Acta Universitatis
Lappeenrantaensis
834

Toni Kuronen

MOVING OBJECT ANALYSIS AND TRAJECTORY PROCESSING WITH APPLICATIONS IN HUMAN-COMPUTER INTERACTION AND CHEMICAL PROCESSES

Toni Kuronen

MOVING OBJECT ANALYSIS AND TRAJECTORY PROCESSING WITH APPLICATIONS IN HUMAN-COMPUTER INTERACTION AND CHEMICAL PROCESSES

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 2310 at Lappeenranta University of Technology, Lappeenranta, Finland on the 13th of December, 2018, at noon.

Acta Universitatis
Lappeenrantaensis 834

Supervisors Professor Lasse Lensu
Adjunct Professor Tuomas Eerola
Professor Heikki Kälviäinen
LUT School of Engineering Science
Lappeenranta University of Technology
Finland

Reviewers Associate Professor Joni Kämäräinen
Department of Signal Processing
Tampere University of Technology
Finland

Professor Pavel Zemčík
Department of Computer Graphics and Multimedia
Brno University of Technology
Czech Republic

Opponents Associate Professor Joni Kämäräinen
Department of Signal Processing
Tampere University of Technology
Finland

Professor Pavel Zemčík
Department of Computer Graphics and Multimedia
Brno University of Technology
Czech Republic

ISBN 978-952-335-314-5
ISBN 978-952-335-315-2 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
LUT Yliopistopaino 2018

Abstract

Toni Kuronen

Moving Object Analysis and Trajectory Processing with Applications in Human-Computer Interaction and Chemical Processes
Lappeenranta, 2018
83 p.

Acta Universitatis Lappeenrantaensis 834
Diss. Lappeenranta University of Technology

ISBN 978-952-335-314-5, ISBN 978-952-335-315-2 (PDF)
ISSN-L 1456-4491, ISSN 1456-4491

In order to better understand processes with moving objects using computer vision, it is important to be able to measure and to analyze how the objects move. The recent advances in imaging technology and in image-based object tracking techniques have made it possible to measure the object movement accurately from video without the need for other sensors. This work focuses on the practical applications of a 3D touch-screen experiment and the moving object analysis of droplets in a chemical mass transfer experiment.

A two-camera framework for tracking finger movements in 3D was developed and evaluated with the study of human-computer interaction. Moreover, trajectory processing and video synchronization were introduced and a 3D trajectory reconstruction technique was proposed. The framework was successfully evaluated in an application where stereoscopic touch-screen usability was studied using stereoscopic stimuli. Finally, a set of hand trajectory features were computed from the trajectory data and it was shown that selected features differ statistically significantly between the targets displayed at different depths.

The image analysis method was proposed for the analysis of moving droplets in a chemical mass transfer experiment enabling the quantification of copper mass transfer. Moreover, the image analysis provided a way to estimate the concentration variation inside the droplet. Furthermore, the method is not limited to a chemical mass transfer experiment with extractants, but it can be used for applications where a detectable color change is present.

This work consisted of: selecting suitable moving object detection and tracking methods for two applications, the post-processing of the trajectories, 3D trajectory reconstruction, and characterizing and visualizing the object data. The applicability of readily available methods for moving object detection and analysis was successfully demonstrated in two application areas. With modifications, both of the used frameworks can be extended for use in other similar applications.

Keywords: object tracking, trajectory processing, trajectory analysis, high-speed video, image analysis, human-computer interaction, 3D reconstruction, liquid-liquid extraction, mass transfer

Acknowledgements

The work presented in this dissertation has been carried out at the Laboratory of Machine Vision and Pattern Recognition at the Department of Computational Processes and Engineering of the Lappeenranta University of Technology, Finland, between 2015 and 2018. I would like to express my deep gratitude to my supervisors, Professors Lasse Lensu and Heikki Kälviäinen, and Adjunct Professor Tuomas Eerola for their guidance, support, comments and cooperation throughout the work. I thank Jukka Häkkinen, Jari Takatalo, for being great co-authors and providing insights about the HCI side of the research. I would like to thank Jussi Tamminen, Esko Lahdenperä and Tuomas Koiranen for the cooperation in several articles and providing information about the chemistry side of the work. I gratefully acknowledge the help of the reviewers Associate Professor Joni Kämäräinen and Professor Pavel Zemčík for their criticism and valuable comments on the work done. I also wish to thank all of my friends and co-workers for their support, patience and help. I would like to thank the Academy of Finland, which funded the computational psychology of experience in human-computer interaction (COPEX) research project (No. 264429) and the analysis of polydispersity in reactive liquid-liquid systems (PORLIS) research project (No. 277189) within which parts of this study were conducted. Finally, I would like to thank my wife, Marika, and our daughter, Nea, for the support, love, and understanding during this work.

Lappeenranta, November 2018

Toni Kuronen

Publication I

Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J., Kälviäinen, H., High-speed hand tracking for studying human-computer interaction, Proceedings of Scandinavian Conference on Image Analysis, 2015, pages 130-141. JUFO 1

Publication II

Lyubanenko, V., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Häkkinen, J., Multi-camera finger tracking and 3D trajectory reconstruction for HCI studies, Proceedings of Advanced Concepts for Intelligent Vision Systems, 2017, pages 63-74. JUFO 1

Publication III

Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Two-camera synchronization and trajectory reconstruction for a touch screen usability experiment, Proceedings of Advanced Concepts for Intelligent Vision Systems, 2018, pages 125-136. JUFO 1

Publication IV

Tamminen, J., Lahdenperä, E., Koiranen, T., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Determination of single droplet sizes, velocities and concentrations with image analysis for reactive extraction of copper, Chemical Engineering Science, 2017, Vol. 167, pages 54-65. JUFO 2

Publication V

Lahdenperä, E., Tamminen, J., Koiranen, T., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Modeling mass transfer during single organic droplet formation and rise, Journal of Chemical Engineering & Process Technology, 2018, Vol. 9. JUFO 1

In this dissertation, these publications are referred to as *Publication I*, *Publication II*, *Publication III*, *Publication IV*, and *Publication V*.

ASMS	scale adaptive mean shift
AtCF	attentional feature-based correlation filter
CT	real-time compressive tracking
CNN	convolutional neural networks
DAT	distractor aware tracker
EKF	extended Kalman filter
FCT	fast compressive tracking
FFT	fast Fourier transform
fps	frames per second
HCI	human-computer interaction
HMI	hydrargyrum medium-arc iodide
HOG	histogram of oriented gradients
HT	Hough-based tracking of non-rigid objects
IVT	incremental learning for robust visual tracking
KCF	high-speed tracking with kernelized correlation filters
KCF2	high-speed tracking with kernelized correlation filters v2
KF	Kalman filter
LED	light emitting diode
LRS	log-Euclidean Riemannian subspace and block-division appearance model tracking
LOESS	local regression
LOWESS	locally weighted scatterplot smoothing
MIL	robust object tracking with online multiple instance learning
MA	moving average
MCMC	Markov chain Monte Carlo
PCA	principal component analysis
RSCM	robust object tracking via sparse collaborative appearance model
S3D	stereoscopic 3D
SCT	structuralist cognitive model for visual tracking

SDC	sparsity-based discriminative classifier
sKCF	scalable kernel correlation filter with sparse feature integration
SGM	sparsity-based generative model
SRPCA	online object tracking with sparse prototypes
STAPLE	sum of template and pixel-wise learners
STAPLE+	improved STAPLE tracker with multiple feature integration
STC	fast visual tracking via dense spatio-temporal context learning
struck	structured output tracking with kernels
SVM	support vector machine
S-G	Savitzky-Golay
TLD	tracking-learning-detection
TV	total variation
TVD	total variation denoising
UKF	unscented Kalman filter
UKS	unscented Kalman smoother
VOT	visual object tracking
yadif	yet another deinterlacing filter

Chapter I

Introduction

Moving object detection and object tracking have been popular topics in the field of computer vision. For example, a survey of various moving object detection and tracking methods was carried out in [46], tracking surveys considering various aspects of tracking have been provided in [66, 75, 11, 121], and the accuracy and robustness of object trackers have been evaluated in visual object tracking (VOT) challenges, such as, VOT2014 [58] and VOT2016 [56]. Object detection can be thought as a basic step for the further analysis of video since all tracking methods require object detection, or manual setting of the object location at the initialization phase. The basic idea of video tracking is to follow one or more objects in an image sequence. On a general level, tracking can be divided into initialization, target representation, motion estimation, localization, model update, and track management phases. In the initialization phase, the tracker is initialized either manually or automatically. The target representation phase compiles the representation model using selected features from an image. The following step is to estimate and localize the target in an image. As the last step, the object representation is updated if needed. Finally, as a result of tracking, a trajectory can be formed and high-level features from the trajectory can be extracted [75].

The applications for object detection and tracking are numerous. Moving object detection and tracking can be used, for example, in human-computer interaction (HCI) [105], augmented reality [50, 77], media production [85], biological research [112], chemistry applications [94, 108], surveillance [46], robotics and unmanned vehicles [75]. In the HCI field, detection and tracking can be used to track and detect the movement of a person when they perform different tasks and the produced trajectories can be later used for movement analysis. For the augmented reality cases, it is possible to detect and track the movement of a moving person and change a virtual scene based on the movement. In the media production field, certain elements of videos or image sequences can be appended, adjusted or discarded based on the results from detection and tracking. For surveillance purposes, detection and tracking can be used to recognize the abnormal moving patterns of people. In the robotics field and with unmanned vehicles, the results of detection and tracking can be used to gather information about the surroundings and the vehicle or

robot can act accordingly [75].

This research focuses on moving object analysis and trajectory post-processing with applications in the field of HCI. The research involves a study of hand movement in a touch-screen experiment, and a study of droplets in a chemical mass transfer experiment in the field of chemistry. Moving object analysis was done for both normal-speed and for high-speed video data. The interest in using high-speed videos derives from the temporal resolution which is better than the one of the videos with a standard frame rate. A higher temporal resolution, i.e., a higher frame rate in the camera means that smaller and faster motions can be captured compared to videos with the common rates of 24 to 60 frames per second (fps). This results in more accurate measurements. Furthermore, increased sharpness and reduced motion blur of the images of fast-moving objects can be achieved with shorter exposure times. The general steps involved in the moving object analysis of this study, from designing the experimental setup to computing real-world features i.e., features in physical units, are shown in Figure 1.1. The dashed line in the figure, from camera calibration to computing real-world features, indicates the usage of camera calibration results to interpret real-world features.

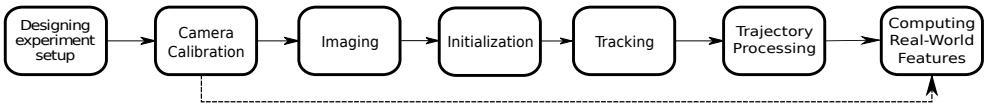


Figure 1.1: A general flow chart of moving object analysis using an imaging setup.

The need for trajectory processing arises from the fact that the object trajectory produced by tracking contains noise. This noise is amplified when certain features are calculated from the trajectory. Thus, in order to obtain appropriate features from the trajectories, post-processing of the tracking data is needed. Moreover, image analysis can be used to find useful information about the tracked objects. For example, the information may consist of color changes or the size of the object. To produce accurate measurements, camera calibration and 3D reconstruction provide a way to acquire the undistorted real-world measurements of the moving object trajectories. With the real-world measurements, it is possible to study the phenomena based on information in physical units such as real velocity and acceleration. Post-processed trajectories can be subjected to further analysis, for example, the categorization of movements based on trajectory features.

1.1 Objectives

In this work, the problem of moving object analysis in two different applications is considered. These applications included a 3D touch-screen experiment and a chemical mass transfer experiment. This work deals primarily with issues of detection, tracking, and trajectory post-processing and analysis. The work consists of four topics forming four separate research questions: object motion detection and tracking from high-speed and normal-speed video material, the post-processing of the tracking data, trajectory understanding and conversion to real-world measurements, i.e., measurements in physical units. The objectives of the research are as follows:

- To evaluate the potential of high-speed and normal-speed imaging to assist in analyzing HCI and monitoring a chemical mass transfer experiment.
- To detect moving objects from high-speed and normal-speed video sequences.
- To devise a way to track moving objects and appearance changes in applications robustly and accurately.
- To study a way to detect and handle errors in the detection and tracking.
- To review methods to make the tracking data more reliable and accurate in case of small fluctuations in the trajectories.
- To select and compute reliable trajectory features such as speed, acceleration, direction, and distance.
- To study other possible features and phenomena that emerge during the work conducted.

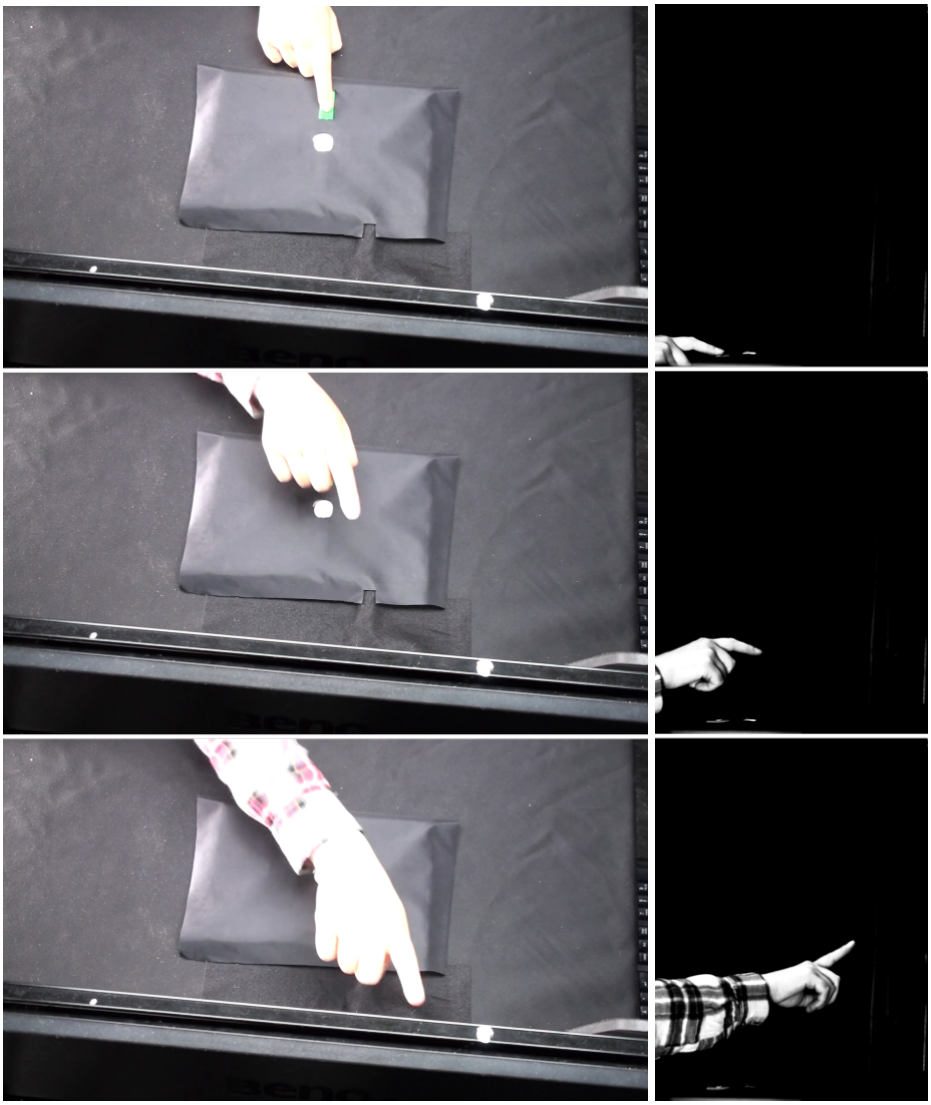
Example images of the 3D touch-screen experiment are shown in Figure 1.2. Figure 1.2a contains images from a normal-speed video with a user's finger moving from a trigger-box button towards the screen. Corresponding high-speed video images of a user performing the pointing action towards the screen are shown in Figure 1.2b.

Example results from the 3D touch-screen experiment are visualized in Figure 1.3. Figure 1.3a and Figure 1.3b show normal-speed and high-speed video frames with trajectories. 3D trajectory features, speed and acceleration, are visualized in Figure 1.3c.

Example images of tracking one droplet frame by frame in the chemical mass transfer experiment are shown in Figure 1.4. The figure shows the low contrast between the foreground, droplet, and background, constant phase liquid, which makes the droplet almost invisible in Figure 1.4a. In order to make the droplets visible in Figure 1.4b, the color channel values were adjusted and the images were made gray-scale. Moreover, the formation of a new droplet at the tip of the needle is visible in the subsequent frames.

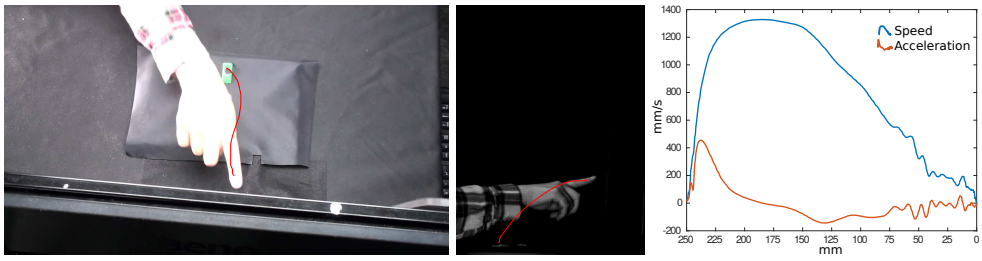
1.2 Contributions and Publications

The main contribution of this work was in designing and implementing two frameworks for collecting and processing data from a 3D touch-screen experiment and from a chemical mass transfer experiment. Designing the frameworks included the evaluation of moving object detection and tracking methods, and the evaluation and selection of optimal trajectory post-processing techniques. Moreover, for the 3D touch-screen experiment, a video and trajectory synchronization and analysis of trajectory features were carried out. As a result, hand movements were tracked with a high success rate and real-world trajectories were constructed. Based on the features calculated from the real-world trajectories, small differences were observed in the trajectories towards targets at different disparities. In the chemical mass transfer experiment, a reliable way to detect and track moving droplets was discovered along with the assumption of an oblate spheroid shape. Moreover, a way to measure chemical changes in the droplets was determined and the accuracy and reliability of the method was found to be on a millimole scale.



(a) **(b)**

Figure 1.2: Example images of the 3D touch-screen experiment: (a) example images of normal-speed video; (b) corresponding example images of high-speed video. The contrast of the high-speed video examples was enhanced for visualization purposes.



(a) **(b)** **(c)**

Figure 1.3: Example trajectory and feature visualization of the 3D touch-screen experiment: (a) example finger trajectory from a normal-speed video; (b) example finger trajectory from a high-speed video; (c) speed (blue line) and acceleration (red line) as a function of a distance from the end point of the trajectory.

This dissertation contains five publications: two journal articles which have been published in international journals, and three conference papers. The publications can be divided into two topic areas. *Publication I*, *Publication II*, and *Publication III* are dedicated to the topic of moving object analysis connected with a 3D touch-screen experiment, while *Publication IV* and *Publication V* addressed the problem of moving object analysis for droplets in a chemical mass transfer experiment.

Publication I introduces a high-speed tracking and finger trajectory filtering evaluation. The author of this dissertation developed the framework, performed the experiments and was the principal author of the article.

Publication II introduces additional tracking from normal-speed hand movement videos and a 3D reconstruction pipeline. *Publication II* is based on the ideas of all the authors of the article. The implementation and the experiments were performed by Lyubanenko with supervision and advice provided by the other authors of the article. The author of this dissertation composed the article, with the help of the other authors, based on the experimental results by Lyubanenko.

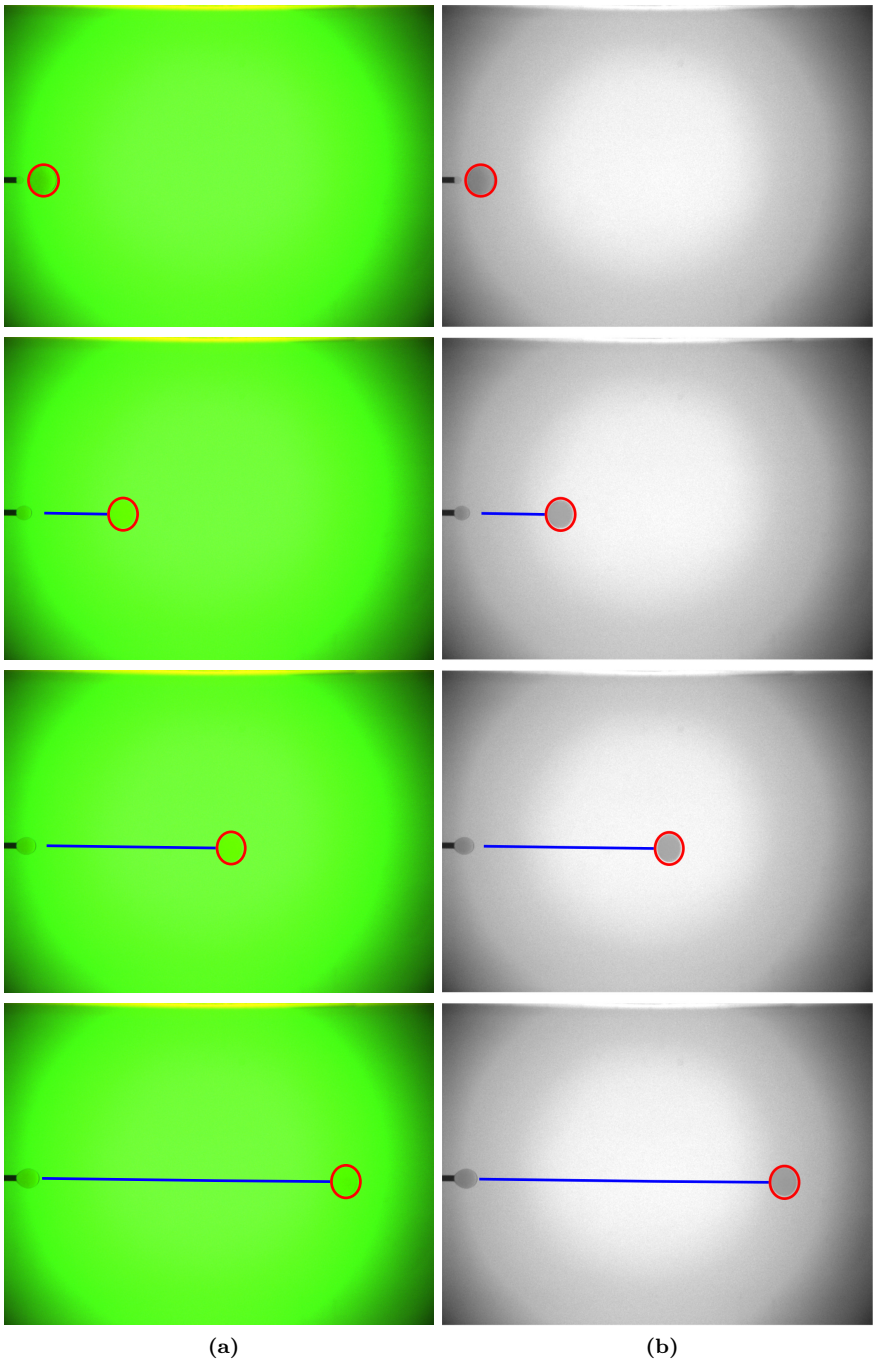
In *Publication III*, a large-scale 3D reconstruction from tracked movements was presented with additional topics covering video synchronization and statistical feature analysis. The implementation and the experiments were performed by the author of this dissertation. Moreover, the author of this dissertation was the principal author of the article.

Publication IV addresses the issue of determining single droplet sizes, velocities and concentration with image analysis. The author of this dissertation designed and prepared the imaging setup, developed the algorithm for the detection and tracking of the droplets and implemented the image analysis pipeline and was a co-author of the article.

Publication V addresses the issue of mass transfer during droplet formation and rise. The publication is based on the work started in *Publication IV*. The author of this dissertation selected the imaging equipment, designed and prepared the imaging setup, developed the algorithm for the detection and tracking of the droplets and implemented the image analysis pipeline, and was a co-author of the article.

1.3 Thesis outline

Chapter 2 contains the main motivation behind the moving object detection and analysis and provides an overview of the methods used. Chapter 3 is the main part of this dissertation. It discusses moving object analysis in a practical application of a 3D touch-screen experiment. In Chapter 4, a practical application for the moving object analysis of droplets in a chemical mass transfer experiment is discussed. Finally, Chapter 5 provides a short conclusion of the dissertation.



(a) (b)

Figure 1.4: A droplet detection and tracking example: (a) sequence of RGB images and (b) gray-scale images based on modified RGB images. The detections are shown with red ellipses and trajectories are indicated with blue lines.

Chapter II

Moving Object Detection, Tracking and Movement Analysis

This chapter covers the main ideas and motivations of the previous work on the moving object detection, tracking, trajectory processing and 3D reconstruction methods used in this work.

2.1 Moving Object Detection and Tracking

Many applications benefit from the detection and tracking of various moving objects [75]. However, it is a challenging task because of the varying appearance of the objects, distortions, occlusions and noise. In the simplest form, a moving object can be detected from a static background by calculating the temporal difference between the video frames. However, this technique, called frame differencing, only works for static backgrounds and fixed settings. Moreover, frame differencing can have problems in detecting all the relevant pixels of a foreground object if the object moves slowly or has a uniform texture. Furthermore, if the target object stops moving, frame differencing fails to detect any changes and loses the target [46]. Detection can be also performed using more sophisticated background subtraction methods [46] or object detectors [88, 89]. However, background subtraction methods are easily distracted by challenges such as sudden illumination changes or dynamic background [9]. Moreover, background subtraction and object detection techniques usually have longer frame processing times than object tracking which is another possible method used to follow moving objects. A wide variety of trackers for different purposes are available. There are methods for tracking rigid objects and non-rigid objects. There are also trackers that can learn different poses for the tracked objects and continue tracking them, even after they momentarily lose the target object [49]. However, trackers need to be initialized with the location of the object and in these cases manual initialization or automatic detection methods need to be used.

2.1.1 Background

The detection and tracking of moving objects can be divided into steps, including: target initialization, target representation, motion estimation, target localization, and model update. The target initialization for tracking is usually given manually, but this stage can

be performed automatically by using object detectors or background subtraction methods, for example. The model or representation describes how the features of the object and the surrounding background are represented. The motion estimation phase attempts to estimate where the target object will be in the next frame. Motion estimation provides information for the target localization phase where the target location is determined, for example, by using normalized cross correlation or maximizing likelihood functions. The model update phase concerns updating the appearance model of the target when needed.

In general, tracking algorithms can be classified into discriminative and generative approaches based on the appearance models used [123]. The generative methods learn the appearance model of the target object and use it to search for the image region with minimum possible reconstruction error. Discriminative methods treat tracking as a binary classification problem, where a decision boundary between the target and the background is sought. The generative approaches can typically deal better with missing data, which helps in the case of occlusions. Moreover, the generative approaches have better generalization performance when the size of the training data is small. However, it has been shown in [61] that the discriminative classifiers outperform the generative approaches if enough training data is available.

A variety of object trackers use online learning that updates the representation of a target over time. Online learning is used to handle variations in appearance that are usually unavoidable especially in long-term tracking. In generative online learning methods, the appearance model for the target is updated in response to the appearance variations [70]. In discriminative methods, a decision boundary between the foreground and the background is updated adaptively in an online manner as the appearance of the target and the background changes [70].

Variations in an object's appearance are challenging for object trackers. The variations include occlusions, changes in the object poses, scene changes and possible sensor noise. Occlusions occur when the view of the tracked object is blocked by another object in the scene. Changes in the pose of an object arise from object rotation, translation or deformation. Scene changes refer to aspects such as changes in illumination or the weather. Moreover, similar objects in the background pose challenges for most trackers [75]. Some of the object appearance challenges are shown in Figure 2.1, which presents the effects of illumination, occlusion, deformation, noise corruption, out-of-plane rotation, and motion blurring for the object appearance [66]. High-speed imaging introduces other issues, for example, the amount of light needed in imaging increases as the exposure time decreases.

2.1.2 Target Initialization

In the evaluation of object tracking, mostly manual initialization is used by annotating the target object with bounding boxes. Moreover, manual initialization can be utilized in cases where the initial location of the target is known, for example, a trigger button in an HCI experiment. In this case, the target can be initialized with a bounding box over the button since the user has to first press that button to begin the usage of the equipment. Automatic initialization can be also performed using object detectors or movement detection. Automatic initialization needs to be used in cases where the initial location of the target is not known, for example, in gesture recognition. However, initialization is problematic in cases where bounding boxes are used because typically up

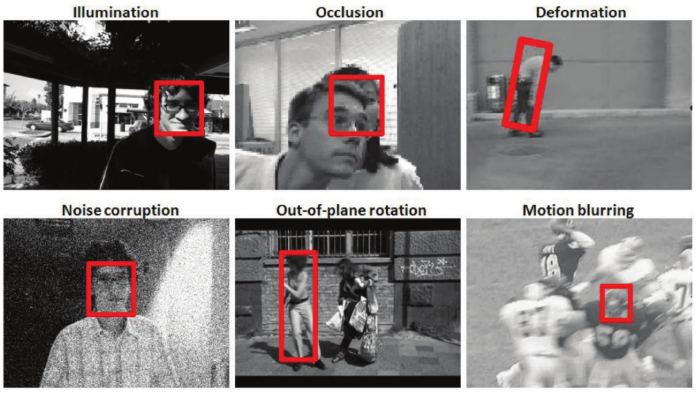


Figure 2.1: Object tracking challenges [66].

to 30% of the bounding box region contains pixels that do not belong to the object [24]. The initialization problem can be addressed by selecting the regions of the bounding box that are highly likely to belong to the object and removing the parts which result in poor performance. Moreover, segmentation techniques can be used to identify the regions that do not belong to the object. Furthermore, optical flow estimation and using areas with good image alignment properties can be used to address the initialization problem [12, 26, 60, 120]. In [24], the authors found an alpha matting method being effective for the VOT2016 [56] dataset. The method predicts an alpha value of pixels based on the pixel belonging to the background or to the foreground, these alpha values are then thresholded using a dynamically changing threshold value based on the proportion of the bounding box belonging to the foreground (object).

Background subtraction is an effective way to initialize tracking for moving objects in relatively static background settings. There are various methods for background subtraction, such as, background subtraction with alpha, statistical methods, and temporal differencing [46]. Heikkilä and Silvén [41] presented a background subtraction with alpha method, where the background B_{t+1} is updated as follows:

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t \tag{2.1}$$

where α is adaption rate, I_t is the current frame and B_t is the previous background. The foreground pixels can be determined by using

$$foreground(x, y) = \begin{cases} 1, & \text{if } |I_t(x, y) - B_t(x, y)| > T \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

where T is a pre-defined threshold value.

Temporal differencing has the same way of determining foreground pixels as the background subtraction with alpha, but the background B_t is replaced with the previous frame I_{t-1}. In the statistical background subtraction methods, such as [84], in each frame the dynamic statistics of pixels that belong to the background are kept and updated. The foreground pixels are detected by comparing the statistics of each pixel with the background model [46].

2.1.3 Target Representation

Many approaches have been used for a target representation in object tracking, among them intensity, color, template, intensity histogram, histogram of oriented gradients (HOG) [20], as well as Haar-like [109], and convolutional neural networks (CNN) [14] features [21, 119]. The intensity model uses intensity values and the color model uses color values of the target area to represent the target. The intensity and color models can be extended to use histograms of the values which allows better handling in the case of small appearance changes. The template model takes an image patch and this is used to represent the target. The target may be presented as a whole or as parts. Part-based representation can help to address the problem of occlusions [75, 98].

Representation using HOG features is used in many current trackers, for example, in [4, 21, 23, 22, 34, 43, 65, 67, 73]. The representation in HOG is based on the idea that the shape of an object can be characterized using edge directions. The idea is to divide the image into small spatial regions, called cells, and to calculate a 1D histogram of the edge orientations for each cell. Finally, the combined histogram entries form a representation. In order to make the method more robust towards intensity changes, it is useful to contrast-normalize the local responses. This can be achieved by accumulating the values of local histograms over larger spatial regions, known as blocks, and using the results to normalize all of the cells in the region [20]. The example results of the HOG feature extraction for 24×24 images of the digit one and digit eight with cell sizes 1×1, 2×2 and 4×4 are shown in Figure 2.2. Moreover, the length of the feature vectors for each cell sizes are shown in the figure. With a cell size of 1×1, the feature vector contains 19044 elements whereas a cell size of 4×4 produces a feature vector with 900 elements.

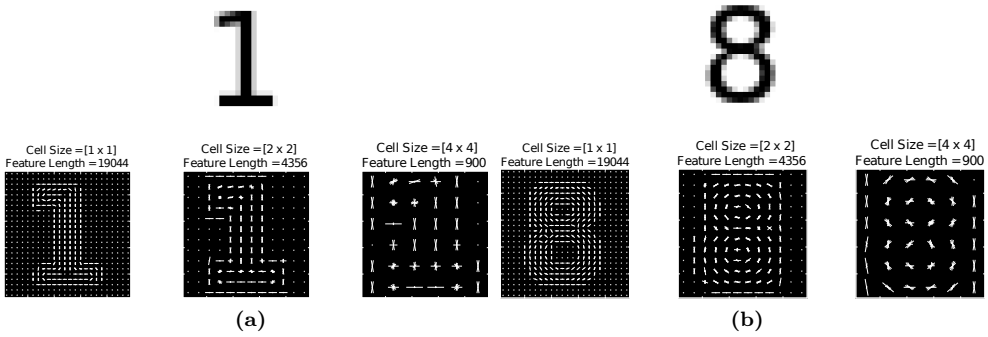


Figure 2.2: An HOG feature plot of: (a) the digit one; (b) the digit eight. The plots include HOG features with cell sizes of 1×1, 2×2 and 4×4.

Haar-like features have been used, for example, in trackers introduced in [1, 37, 123, 124]. The basic idea behind Haar-like features is that the sum of the pixels which lie in the one side of the rectangles are subtracted from the sum of pixels on the other side. The value of a two-rectangle feature is the difference between the sum of the pixels within the two rectangular regions. The rectangular regions have the same size and shape and are horizontally or vertically adjoined. A three-rectangle feature computes

the sum within two outside rectangles subtracted from the sum of a center rectangle. Finally, a four-rectangle feature is computed from the difference between diagonal pairs of rectangles [109]. A Haar-like feature set was extended in [69] by adding 45° rotated features. The extended feature set is shown inside the rectangular region in Figure 2.3 while the original features introduced in [109] are at the top of the figure. The features are grouped according to the dotted lines into edge features, line features and center-surround features.

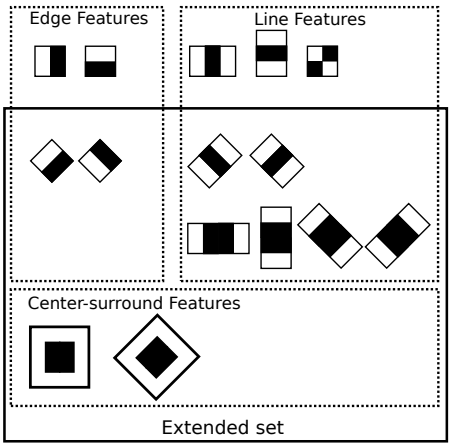
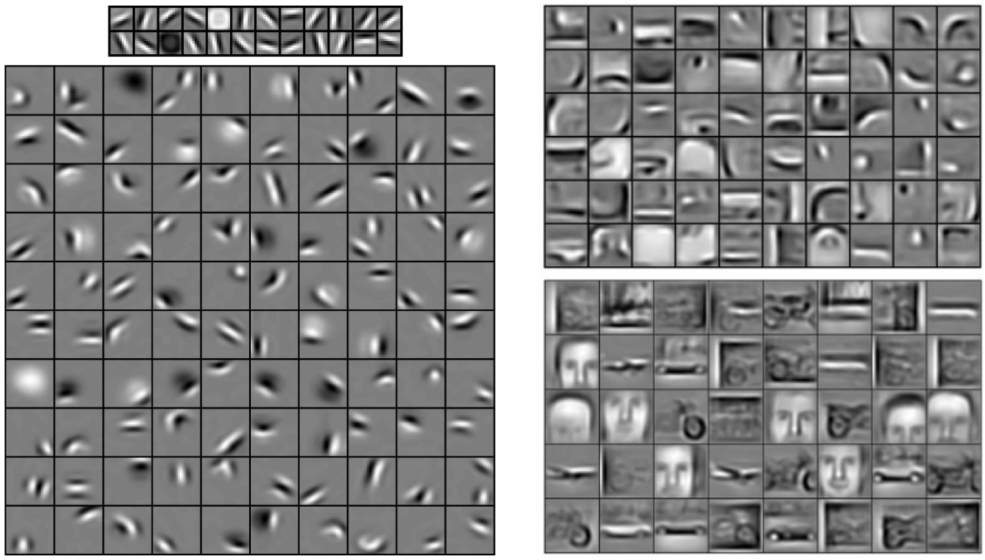


Figure 2.3: Haar-like and center-surround features. The white areas have positive weights and the black areas are negative.

The latest generation of CNN based on the ideas provided in [62] has achieved good results in benchmarks on image recognition and object detection, as well as object tracking, and has significantly raised the interest in these methods [14]. In CNN based methods, a network learns the features that in conventional algorithms are hand-crafted. Visualization of learned network layers are shown in Figure 2.4. From the figure it is possible to see that the first layer contains edge features whereas the third layer features are already recognizable parts of faces, motorbikes, airplanes, and cars. Moreover, the network needs to be pre-trained in order to be effective and this is done via back-propagation. In back-propagation the initialized random weights of the layers are adjusted based on the correctness of the output. However, this process needs a large number of labeled images. To address the training problem, there are pre-trained networks, such as, ImageNet [25] which can be utilized. Furthermore, it was shown in [35] that fine-tuning a pre-trained CNN with target data can further improve the performance of the CNN.

The Hough transform is a voting technique which maps lines from an image to points in Hough space. The technique can be used, for example, to detect lines, circles and ellipses from an image. A generalized Hough transform can be used to define a model shape from boundary points and a reference point. In the procedure, a displacement vector is computed for each boundary point of the model and stored in a table indexed by the gradient orientation. The detection can then be performed by using voting to see which displacement vectors correspond to the stored ones [2].



(a) (b)

Figure 2.4: A CNN layer visualization plot of: (a) the first and second layer learned from natural images; (b) the second and third layer learned from a mixture of faces, cars, airplanes, and motorbikes images [63].

2.1.4 Motion Estimation

Motion estimation tries to estimate the target location in the following frames. Similarly to different object representation methods, there are also various motion estimation methods, including gradient descent, particle filters, Markov chain Monte Carlo (MCMC), local optimum search, and a dense sampling search [75]. Based on the features and a score function that defines the quality of the next state, the gradient method tries to find a local maximum of the score. In tracking, a gradient descent is generally used. In the gradient descent methods, the score function is an error function and the minimum of the error is sought iteratively [75].

In the Monte Carlo approach, the essential idea is to define a domain of possible points and generate random points from a probability distribution over the domain. The random points should be uniformly distributed over the domain. In motion estimation, the MCMC methods are used to approximate the posterior distribution of possible next locations by random sampling in a probabilistic space. The particle filter method is a Monte Carlo technique for the state estimation problem. The idea is to represent the posterior density function from a set of random samples, particles [90]. The trick in the MCMC method is that for a pair of input values, it is possible to compute which one is the better value. This is done by computing how likely the value explains the data, given the prior information. If this randomly generated value is better than the last one, then it is added to the chain with a certain probability determined by how much better it is than the last one.

In the dense sampling methods, a search grid is formed around the previous location of the object and a search window is then moved pixel by pixel over the search grid. Random sampling methods provide a similar approach, but the search grid is formed from random locations around the previous location of the object and these random locations are then searched for locating the target object in the current frame [42].

2.1.5 Target Localization

Target localization usually goes hand in hand with motion estimation. Motion estimation provides samples from which the localization method selects the best possible candidate for the updated target region. Target localization can be carried out by using the gradient descent method, for example, where an error function of the appearance differences is minimized, or cross correlation, maximizing location likelihood function, as well as a discriminative classifier. For the discriminative classifiers, the classifier is learned in the initialization phase and the algorithm attempts to separate the target object from the background. This is achieved by sampling positive samples of the target object and negative samples of the background.

The cross correlation cc for an image f with a template t shifted to (u, v) is calculated as

$$cc(u,v) = \sum_{x,y} f(x,y)t(x-u, y-v). \tag{2.3}$$

The cross correlation is typically evaluated at each point (u, v) for f and the template t, which is shifted by u steps to x direction and by v steps to y direction. However, the cross correlation is easily distracted by image intensity changes so normalized cross correlation is typically used. The effect of image intensity on cross correlation is easily demonstrated with a case of two images with constant gray values, v and $2v$. Regardless of the template, the image with $2v$ is selected as a better match because it gives the higher score.

Normalized cross correlation is a process where the intensities of the template and the search area are normalized and it can be calculated as

$$ncc(u,v) = \frac{\sum_{x,y}(f(x,y) - \bar{f}_{u,v})(t(x-u, y-v) - \bar{t})}{\sqrt{\sum_{x,y}(f(x,y) - \bar{f}_{u,v})^2 \sum_{x,y}(t(x-u, y-v) - \bar{t})^2}} \tag{2.4}$$

where $\bar{t}$ is the mean of the feature and $\bar{f}_{u,v}$ is the mean of $f(x, y)$ in the region under the feature. The score values range from the perfect match of 1 to completely anti-correlated value of -1. However, it should be noted that normalized cross correlation is not invariant to scale, rotation, and perspective distortions [8]. Moreover, the cross correlation between functions $f(t)$ and $g(t)$ is equivalent to the convolution of $f^*(-t)$ and $g(t)$, i.e.,

$$f(t) \star g(t) = f^*(-t) * g(t), \tag{2.5}$$

where $\star$ is the cross-correlation operation, f^* denotes the complex conjugate of f, and $*$ is the convolution operation. Furthermore, Henriques et al. [42] showed that by sampling all the sliding windows, the resulting data matrix can be made circulant, i.e., the first row is a vector u, the second row is u shifted one element to the right, and so

on. The sums, products and inverses of circulant matrices are also circulant which helps in their manipulation. Moreover, circulant matrices encode the convolution of vectors. Since the product $C(u)v$ represents a convolution of vectors u and v, it can be computed in the Fourier domain taking advantage of the convolution theorem which states that an element-wise product in the Fourier domain representation is equivalent to the convolution of two image patches. The fast Fourier transform (FFT) method enables fast tracking with the computational complexity of $\mathcal{O}(n \log n)$. Since the pioneering works conducted in [7, 42], correlation filters have been adapted in many recent trackers [4, 21, 23, 43, 56, 65, 110, 122].

2.1.6 Model Update

The representation of the target can be updated with the combination of a fixed reference and the most recent frame, or the whole representation can be updated with the most recent target. The use of a fixed reference provides a memory for the model and it can help to address the problem of occlusions. There are different strategies used for performing the update, for example, after every frame or after a few frames [21, 23, 58, 57, 75, 98, 11, 119]. However, there have been recent works where the possibility of having no model update at all has been explored with good tracking performance [5, 102].

2.2 Trajectory Processing

This section focuses on the post-processing and analysis of trajectory data obtained by tracking a moving object. Filtering and smoothing methods such as, the moving average (MA), Savitzky-Golay (S-G), and Kalman filter (KF) methods are introduced. Moreover, a short introduction to camera calibration, imaging, multi-view geometry, and 3D reconstruction is provided.

2.2.1 Filtering and Smoothing

The main goal for experiments should be to extract quantifiable information about the measurements obtained from the experiments, but usually these contain noise. The noise can be described as random errors that contaminate the information and it should be suppressed as much as possible without weakening the signal or underlying information [92]. Filters can be used, for example, to remove unwanted noise from the measurements, and remove specific frequencies [82].

2.2.2 Smoothing Trajectory Data

Extracting higher level features for the analysis of a moving object from trajectories, which are sequences of center location points, can be challenging [*Publication I*]. The center locations of an object over time can be useful as such for checking the location of the object at a certain time, but when higher level features such as the velocity or acceleration are calculated from the location data, their values are erroneous. This happens because most trackers operate at pixel level accuracy, and in videos the object movement can be smaller than one pixel per frame or the selected tracking method

may produce small tracking shifts during the tracking process. These issues become problematic when the accurate measurement of velocities and accelerations is needed. Single pixel movement after staying in one region for multiple frames results in erratic acceleration and deceleration values, which are difficult to analyze when evaluated. The tracked movement locations can be thought of as a set of measures from the actual movement trajectory containing a measurement error. Moreover, since the movement itself is typically smooth, sudden movements indicated by the tracking values need to be smoothed. This is where filtering or data smoothing of the tracking values can help. To get better results without filtering, there is a need to adopt sub-pixel tracking, for example, with a marker allowing the tracker to determine the exact tracked location at the sub-pixel level.

Moving Average

An MA filter method operates by averaging a number of points from the input data to calculate the output. The data point to be filtered can be from the start of the averaging sequence, from the end of the sequence, or from the middle of the sequence so that the group of points to be included in the averaging are chosen symmetrically around the output point. Selecting the points symmetrically is common since it does not introduce a relative shift between the input and output signals. Depending on the implementation, the end point(s) of the output signal cannot be smoothed because the span cannot be defined for the end point(s) [99].

Let us assume that a signal x is corrupted by noise ϵ resulting in signal y,

$$y = x + \epsilon, \quad y, x, \epsilon \in \mathbb{R}^N. \tag{2.6}$$

In the MA process, the smoothed value for the ith data point $y_s(i)$ is

$$y_s(i) = \frac{1}{2N+1}(y(i+N) + y(i+N-1) + \ldots + y(i-N)) \tag{2.7}$$

where N is the number of neighboring data points on both sides of $y_s(i)$, and $2N+1$ is the size of the smoothing window, otherwise known as the span [71]. In general, the MA approach works by adding values of a fixed number of points together and dividing the result by the number of points. This approach smooths out peaks from the data. One solution to preserve the peaks during smoothing would be to use a Savitzky-Golay (S-G) filter [92].

Kalman Filtering and Smoothing

The KF method is well researched and highly used in the area of autonomous or assisted navigation. It is an optimal recursive data processing algorithm. The KF method can be thought as a set of mathematical equations that provide recursive means to estimate the state of a process, while minimizing the mean of the squared error [116, 117].

Table 2.1 illustrates a KF process. The first step in the time update stage of the process is to calculate a priori state estimate $\hat{x}_t^-$ and a priori error covariance P_t^- using initial estimates of the state $\hat{x}_0$ and the error covariance P_0. In the table, A denotes a state

Table 2.1: Time and measurement update stages of the Kalman filter.

Time Update (prediction)	Measurement Update (correction)
$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_t + w_t$	$K_t = P_t^- H^\intercal (HP_t^- H^\intercal + R)^-1$
$P_t^- = AP_{t-1}A^\intercal + Q$	$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-)$
	$P_t = (I - K_t H)P_t^-$

transition matrix, B is a control matrix, u_t is a control vector, w_t is zero-mean Gaussian white process noise, and Q is an estimated process noise covariance. After the time update stage, the process moves to the measurement update stage where the calculations of Kalman gain K_t, posterior state estimate $\hat{x}_t$, and posterior error covariance P_t are performed. First, the Kalman gain K_t is calculated using the illustrated equation, where H is an observation matrix, $^\intercal$ denotes transpose, and R is measurement noise covariance. The measurement z_t, which is used in the calculation of the posterior state estimate $\hat{x}_t$, is calculated as

$$z_t = HX_t + v_t, \tag{2.8}$$

where v_t is measurement noise. Finally, the posterior error covariance P_t is calculated with the illustrated equation, where I is an identity matrix [117, 116].

The state transition matrix A, control matrix B, observation matrix H, estimated process noise covariance matrix Q, and estimated measurement noise covariance matrix R are the values which are predefined in the KF equation set. The control vector u_t and the measurement vector z_t are the inputs to the KF calculations. The process model represents the current state at time t from the previous state at $t-1$. Q is the process noise covariance which contributes to the overall uncertainty. When the Q is large, the KF tracks large changes in the data more closely than with a smaller Q. The measurement noise covariance R determines how much measurement information is used. The KF considers the measurements to be inaccurate if R is high: if R is smaller, then the measurements are followed more closely [117, 116].

The time update stage estimates the parameter values based on the current measurements. The KF estimates the parameter values by using the previous and current measurements. The KF smoothing algorithm estimates the parameter values by using the previous, current, and future measurements, that is, all the available data can be used for smoothing [117]. The future measurements can be used because the Kalman smoother proceeds backward in time. This also means that the KF algorithm needs to be run before running the smoother.

The KF can be used for trajectory filtering by using a constant velocity model, for example. For simplicity let us assume a constant velocity model for the trajectory filtering.

The state X_t for an object is defined as

$$X_t = \begin{bmatrix} x_t \\ y_t \\ x'_t \\ y'_t \end{bmatrix} \tag{2.9}$$

where x_t and y_t are the x and y locations of the object at time t, x'_{t-1} and y'_{t-1} are the velocities of the object. The dynamics of the location components of the moving object in 2D can be described as

$$\begin{aligned} x_t &= x_{k-1} + x'_{t-1}T + \frac{1}{2}a_x T^2 \\ y_t &= y_{k-1} + y'_{t-1}T + \frac{1}{2}a_y T^2 \end{aligned} \tag{2.10}$$

where a_x and a_y are the accelerations. The dynamics of the velocity components of the moving object can be described as

$$\begin{aligned} x'_t &= x'_{t-1}T + a_x T \\ y'_t &= y'_{t-1}T + a_y T. \end{aligned} \tag{2.11}$$

From the dynamics equations, the following state transition can be formed

$$\begin{bmatrix} x_t \\ y_t \\ x'_t \\ y'_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & \delta T & 0 \\ 0 & 1 & 0 & \delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ x'_{t-1} \\ y'_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\delta T^2 & 0 \\ 0 & \frac{1}{2}\delta T^2 \\ \delta T & 0 \\ 0 & \delta T \end{bmatrix} \times \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{2.12}$$

which can be written as

$$X_t = AX_{t-1} + Bu_{t-1}, \tag{2.13}$$

where Bu_{t-1} can be seen as the noise component. In the case of trajectory filtering, it is usually an external force causing acceleration for the object. In the case of trajectory filtering, the location of the moving object is the observation. Therefore, the measurement matrix H can be defined as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{2.14}$$

If the process to be estimated is non-linear, extended Kalman filter (EKF) can be used to linearize the process about the current mean and covariance. The EKF method is considered as the de-facto standard in nonlinear state estimation [78]. The EKF method uses first-order terms of the Taylor series expansion of nonlinear functions. However, large errors in filtered values are introduced when the models are highly nonlinear and the local linearity assumption breaks down when the higher order terms become significant. For the EKF method, the three first steps of the process are linearization using the Jacobian matrix then computing the predicted mean, and the predicted covariances. After these three simplified steps, the rest of the Kalman process calculates the Kalman gain and, using measurements, updates the state estimate. The time and measurement update stages of the EKF are illustrated in Table 2.2. Notice that the subscript t is added to the

Table 2.2: Time and measurement update stages of the extended Kalman filter.

Time Update (prediction)	Measurement Update (correction)
$\hat{x}_t^- = f(\hat{x}_{t-1}, u_t, 0)$	$K_t = P_t^- H_t^\intercal (H_t P_t^- H_t^\intercal + V_t R_t V_t^\intercal)^- 1$
$P_t^- = A_t P_{t-1} A_t^\intercal + W_t Q_{t-1} W_t^\intercal$	$\hat{x}_t = \hat{x}_t^- + K_t(z_t - h(\hat{x}_t^-, 0))$
	$P_t = (I - K_t H_t) P_t^-$

Jacobians A, W, H, and V to indicate their recalculation at each time step. A and W are the Jacobian matrices of partial derivatives of f with respect to x and w, respectively. H and V are the Jacobian matrices of partial derivatives of h with respect to x and v, respectively [78, 117].

In the unscented Kalman filter (UKF) method [47], an unscented transformation is used to calculate the statistics of a random variable which undergoes a nonlinear transformation. It is built on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function. In the UKF method, the process starts with a sigma point creation. Sigma points are formed by selecting a minimal set of carefully chosen samples that represent the state distribution. After the sigma points are selected, they are run through the process model and, finally, the transformed mean and covariance are computed. After these steps, the rest of the Kalman process is similar to the last three steps of the EKF algorithm. The UKF approach is highly efficient, has almost the same complexity as the EKF method, and slower only by a constant factor in typical practical applications. The UKF method achieves better linearization than the EKF approach, and it is accurate in the first two terms of the Taylor expansion while the EKF method is accurate only in the first term. In the UKF method, there is no need to calculate the Jacobian matrix, but the state estimation for nonlinear systems in the UKF process is still not optimal [78, 47, 76].

LOESS, LOWESS and Robust Versions

Local regression (LOESS) and locally weighted scatterplot smoothing (LOWESS) are methods estimating the regression surface through a smoothing procedure. The estimation is done by fitting a function inside a sliding window into the variables. The weight function in the LOESS and LOWESS method work in such a way that the points closer to the curve play a larger role in the determination of the smoothed values of the curve. The smoothed values are calculated by fitting a polynomial of nth degree by using weighted least squares with a certain weight w_i at point x_i. Robust versions of the LOESS and LOWESS methods give less weight to the points further away from the curve than the standard versions [16, 17, 45].

As in the MA method, each smoothed value is determined by the neighboring data points defined within the span, and a regression weight function is applied to the points included

within the span. A robust weight function, which makes the process more resistant to outliers, can also be used in addition to the regression weight function [16, 17, 45].

The methods are discriminated by their use of regression model: LOWESS uses a linear, 1st degree polynomial whereas LOESS uses a quadratic, 2nd degree polynomial. This section considers the implementations of the LOWESS and LOESS methods proposed in [16, 45]. If there are the same number of neighboring data points available on each side of the data to be smoothed, the weight function is symmetric, if not, then the function is asymmetric. Thus, unlike in the case of the MA method, the span is constant when using LOESS, LOWESS, or their robust versions. This means that there can be phase changes in the beginning and at the end of the data.

In the LOESS and LOWESS methods, the weight w_i is defined as

$$w_i = (1 - \left| \frac{x - x_i}{d(x)} \right|^3)^3, \tag{2.15}$$

where x is the point of evaluation to be smoothed, x_i are the neighbors of x defined by the span, and $d(x)$ is the distance from x to the most distant neighbor within the span. Outside the span, the weights are set to zero. After the weight calculation, weighted linear least-squares regression is performed.

First, the coefficients b_k that minimize the following equation

$$\sum_{i=1}^{n} w_i(x)(y_i - [\sum_{k=0}^{\lambda} b_k x_i^k])^2, \tag{2.16}$$

need to be found. The parameter λ controls the degree of the polynomial used. In the case of LOWESS, λ is 1 and the in the case of LOESS, λ is 2. When the minimizing coefficients b_k are found, the smoothed value at x is obtained by [71, 45]

$$x_s = \sum_{k=0}^{\lambda} b_k x^k. \tag{2.17}$$

Robust versions of the LOESS and LOWESS methods include calculating

$$r_i = y_i - \sum_{k=0}^{\lambda} b_k x_i^k, \tag{2.18}$$

where r_i is the residual of the ith data point from the preceding local regression. Then, r_i^* is defined as

$$r_i^* = \frac{r_i}{6\mu} \tag{2.19}$$

where μ is the median absolute deviation of the residuals. The median absolute deviation measures how spread-out the residuals are. When r_i is small in comparison to 6μ, the robust weight is close to one. The robust weights w_i^* are then calculated by a bi-square function defined as

$$w_i^* = \begin{cases} (1 - |(r_i^*)^2|)^2 & \text{for } |r_i^*| < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

The robust weights are used to estimate a new set of coefficients b_k^*, which minimize the term

$$\sum_{i=1}^{n} w_i^* w_i(x)(y_i - [\sum_{k=0}^{\lambda} b_k^* x_i^k])^2. \tag{2.21}$$

When the minimizing coefficients b_k^* are found, the smoothed value x_s^* at x is obtained by

$$x_s^* = \sum_{k=0}^{\lambda} b_k^* x^k. \tag{2.22}$$

The robustness steps are repeated until the values of the estimated coefficients converge which typically happens quickly [45]. In [71], the robust weight calculation and smoothing are repeated for a total of five iterations.

Savitzky-Golay

Savitzky-Golay (S-G) smoothing reduces noise while maintaining the shape and height of the peaks in the signal. In particular, the locations, heights, and widths of the peaks in the signal waveform are preserved [93].

The S-G filter fits a polynomial to a set of input samples for each input X_n in a least-squares sense and the value of the polynomial at time n is the filter output. A function $f_K(x)$ describes a polynomial of order K:

$$f_K(i) = \sum_{i=0}^{K} b_i x^i = b_0 x^0 + b_1 x^1 + b_2 x^2 + \ldots + b_K x^K, \tag{2.23}$$

where b_is are the coefficients of the polynomial. If N preceding and M subsequent samples are used as the neighboring samples, then the S-G filter determines the b_i coefficients that minimize the term

$$\sum_{i=-M}^{N} (X_{n-i} - f_K(n-i))^2, \tag{2.24}$$

where the polynomial value at time n is the filter output $\hat{X}_n = f_K(n)$. Thus, when the polynomial describes the data accurately, there is minimal distortion in the result. It has been shown in [92] that the filter can be expressed as a weighted MA filter:

$$\hat{X}_n = \sum_{i=-M}^{N} a_i X_{n-i}, \tag{2.25}$$

where the filtering coefficients a_i are constants for all X_n. The coefficients a_i can be calculated using the available algorithms or by using the available coefficient tables to get the values for various ranges and polynomial degrees [92].

The output from the S-G filter is not shifted when the filtering is applied so the signal has zero phase. The filtering effect of the S-G method is not as destructive as the filtering effect of the MA method, and the loss of signal information is smaller than with the MA approach [82, 92]. For the S-G smoothing to work, there needs to be at least as many

data samples as there are coefficients in the polynomial approximation. The S-G filter response of order $N = 0$ and $N = 1$ is identical to the MA filter response [92]. The degree of smoothing in the S-G method is regulated by the filtering window size and by the degree of the fitted polynomial.

Total Variation Denoising

The total variation denoising (TVD) method was developed to preserve sharp edges in the underlying signal. However, TVD can introduce a staircase effect to the data with gradually changing values. These regions appear because the total variation (TV) regularizer promotes piece-wise-constant behavior. For this reason it is not the best filtering method for piece-wise-smooth signals [95, 19].

The TV for a discrete N-point signal $x(n)$, $1 \leq n \leq N$ is

$$TV(x) = \sum_{n=2}^{N} |x(n) - x(n-1)|. \tag{2.26}$$

Let us assume that a signal x is corrupted by additive white Gaussian noise ϵ resulting in signal y,

$$y = x + \epsilon, \quad y, x, \epsilon \in \mathbb{R}^N. \tag{2.27}$$

The TVD approach estimates x by finding the signal x, minimizing the objective function

$$J(x) = \|y - x\|_2^2 + \lambda \|TV(x)\|_1, \tag{2.28}$$

where the degree of smoothing is controlled by the parameter $\lambda > 0$. Increasing the λ value gives more weight to the term that measures the fluctuation of the signal. Iteration count is another parameter in TVD. It controls how many iterations the process will continue with if the error criterion is not yet met in the algorithm.

2.2.3 3D Trajectory Reconstruction

In many applications it is beneficial to study natural object movement in 3D. However, imaging transforms a three-dimensional world into a two-dimensional representation of it and this results in the loss of depth information. Nevertheless, the lost information can be recovered from images with 3D reconstruction using a multi-view or stereo camera setup [39]. The task of reconstructing a 3D trajectory from multiple 2D trajectories, with at least two different viewpoints, is equivalent to the process of 3D scene reconstruction. The first step, the estimation of image point correspondences, can be interpreted as a problem of pairwise trajectory point alignment. It means that for each 2D trajectory point, the matching point of the complementary trajectory, which corresponds to the same world point, has to be found.

According to [125] and [40], an object point $P = [X, Y, Z]^T$ can be used to acquire the corresponding pinhole camera image point p_n via a perspective projection as follows:

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}. \tag{2.29}$$

In order to make accurate measurements from camera images, the camera parameters need to be known. The intrinsic (internal) parameters are the focal length, principal point and the image sensor format the last of which defines the pixel size in the horizontal and vertical directions. The intrinsic parameters depend only on the camera characteristics. The extrinsic (external) parameters contain rotation and translation which are used to define the location and orientation of the camera related to the world reference frame. The camera calibration parameters can be estimated by known points in the images by using information on the imaging geometry [39].

The pinhole camera image coordinates $[x_n, y_n]^T$ are transformed into distorted image coordinates $[x_d, y_d]^T$ using distortion coefficients k_i as follows:

$$p_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_1 r_n^2 + k_2 r_n^4 + k_5 r_n^6) \begin{bmatrix} x_n \\ y_n \end{bmatrix} + dx \tag{2.30}$$

where r_n^2 is $x_n^2 + y_n^2$ and dx is the tangential distortion vector defined as

$$dx = \begin{bmatrix} 2k_3 x_n y_n + k_4(r_n^2 + 2x_n^2) \\ k_3(r_n^2 + r y_n^2) + 2k_4 x_n y_n \end{bmatrix}. \tag{2.31}$$

The intrinsic camera parameters can be obtained by solving

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} fc_x & 0 & cc_x \\ 0 & fc_y & cc_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{2.32}$$

where $[x_p, y_p]^T$ are the pixel coordinates, fc_x and fc_y represent the focal distance, and cc_x and cc_y represent the principal point.

The geometric relationship between two perspective views of the same 3D scene can be described using epipolar geometry. The main idea in this method is to determine the epipolar lines. These map the corresponding scene points from two images to particular image lines. Finding the epipolar lines makes image point matching less complex as there is then only a line from which the corresponding points need to be found whereas the alternative would be a 2D region [33, 39].

Figure 2.5 illustrates epipolar geometry. Any 3D point M and the camera projection centers C_L and C_R define an epipolar plane. Image points m_L and m_R lie on the epipolar plane since they are located on the lines connecting the corresponding camera projection center, and point M. The epipolar lines l_L and l_R are the intersections of the epipolar plane with the image planes. The line between the camera projection centers, C_L and C_R, forms the baseline which intersects both image planes at a point called an epipole. The epipolar constraint describes an epipolar plane that is fully defined by the camera projection centers and the image point. Thus, when a point m_L is given from the left image, one can determine the epipolar line l_R in the right image on which the corresponding point m_R lies [33, 39].

The fundamental matrix F is the algebraic representation of the described epipolar geometry. It is a unique 3×3 matrix of rank 2 which satisfies

$$m_R^T F m_L = 0. \tag{2.33}$$

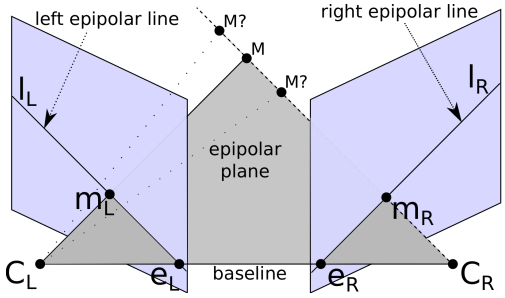


Figure 2.5: Epipolar geometry and the constraint [39].

For any image point m, the corresponding epipolar line may be determined as $l_R = Fm$. Since the epipole lies on an epipolar line, the following holds true

$$e_R^T l_R = e_R^T F m_R = 0. \tag{2.34}$$

The essential matrix E is the specialization of the fundamental matrix in the case of normalized image coordinates. If the calibration matrix K is known, then the normalized coordinates may be obtained as $\hat{m} = K^{-1}m$, defining the equation for the essential matrix

$$\hat{m}^T E \hat{m} = 0. \tag{2.35}$$

Both fundamental and essential matrices allow the reconstruction of the projection matrices of the cameras and to reconstruct the observed scene. The advantage of utilizing the essential matrix is that it shrinks the reconstruction ambiguity from projective to scale. The availability of information about the real-world scene dimensions allows a true Euclidean reconstruction, which includes the determination of the overall scale [39]. The 3D trajectory reconstruction method modified from the scene reconstruction algorithm presented in [39] takes four steps. The steps are explained in Algorithm 2.1.

Algorithm 2.1 Algorithm for 3D trajectory reconstruction [39]

1. Find the corresponding trajectory points from multiple-view trajectories.
2. Compute the essential matrix from the point correspondences.
3. Compute the camera projection matrices from the essential matrix.
4. For each point correspondence, compute the 3D location of a trajectory point.

With eight point correspondences, it is possible to solve the system of linear equations defined by Equation 2.35 directly to obtain an essential matrix. Finding the least-squares solution requires more than eight points. The normalized 8-point algorithm [39] introduces data normalization before finding a linear solution. However, linear methods are not stable for noisy data with outliers. More robust methods can be used in the case of noisy data or when there is a huge number of outliers, such as the MLESAC [104], LMedS [91] or RANSAC [32] methods.

Chapter III

Moving Object Analysis in 3D Touch-Screen Experiment

This chapter contains the main findings of the 3D touch-screen experiment case where a framework for tracking finger movements was constructed for studying HCI. The framework included designing and building the experiment setup, the evaluation of object trackers for the task of hand and finger tracking, the synchronization of the normal-speed and the high-speed videos, 3D reconstruction for tracked trajectories and trajectory feature analysis. The detailed descriptions are provided in *Publication I*, *Publication II* and *Publication III*. This chapter covers the main ideas and motivations of the results presented in the papers.

3.1 Background

In order to develop better touch and gesture user interfaces, it is important to be able to measure how humans move their hands while using a user interface. Earlier research on hand movements in pointing actions has shown that, in addition to the primary movement towards the target, there are corrective sub-movements that are observable from the acceleration or velocity changes during the hand movement [27]. These processing events in goal-directed movements are visualized in Figure 3.1. The illustration shows that a corrective sub-movement is needed when the target is not reached or if it is overshot with the primary sub-movement. The timing and relative locations of these sub-movements belong to the key features of hand movement analysis. According to earlier research, the deceleration part of the first sub-movement starts approximately ten centimeters before the target [27]. Based on this information, the useful features in time and space of the hand movements consist of the point of the maximum velocity, maximal acceleration and deceleration. The points where the deceleration starts as measured from the start-point and/or from the end-point, also provide useful information when defining the smoothness and stability of the movement.

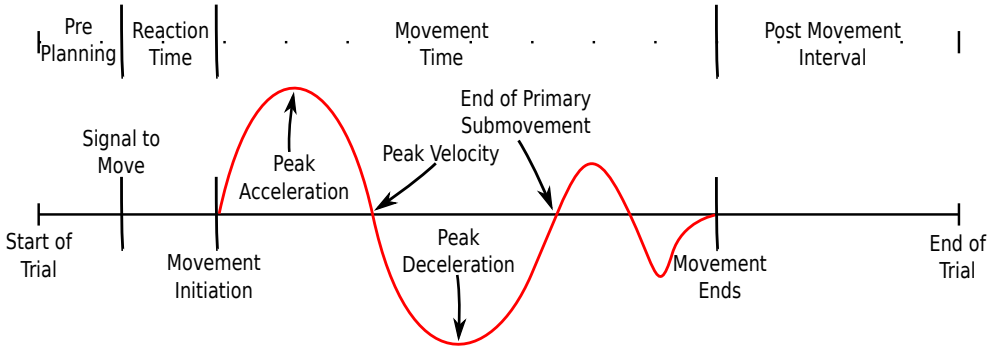


Figure 3.1: Multiple processing events associated with a single goal-directed movement [27].

3.2 Related Work

Advances in gesture interfaces, touch-screens, and augmented and virtual reality have brought new usability concerns that need to be studied in a natural environment and in an unobtrusive way [105]. Several robust approaches for hand tracking exist that can measure hand and finger location with high accuracy using, for example, data gloves with electromechanical, infrared or magnetic sensors [29]. However, such devices affect the natural hand motion and cannot be considered feasible solutions when pursuing natural HCI. Consequently, image-based solutions which provide an unobtrusive way to study and to track human movement and enable natural interaction with the technology have become a subject of research interest. There are commercially available solutions such as the Leap Motion and Microsoft Kinect^{TM}, but they have several limitations. The Leap Motion limits the hand movement to a relatively small area, but it is precise for finger movement measurements [115]. Microsoft Kinect^{TM} on the other hand allows a larger movement area, but it is imprecise for accurate finger movement measurements [52]. However, neither of these devices allows frame rates high enough to capture all the nuances of rapid hand and finger movements.

High-speed cameras have been used for fingertip tracking, for example, in [30] where a fingertip was tracked for the purpose of in-air signature verification using the tracking-learning-detection (TLD) algorithm improved by the authors. They used a 100 fps high-speed camera and achieved promising results with their proposed system. In [103], the authors proposed a fast finger tracking system for an in-air typing interface using a high-speed camera. They used a 120-fps camera also with 60 fps and 30 fps settings and concluded that only a high frame rate over 100 fps was enough for the reliable recognition of rapid typing motions. However, high-speed imaging requires more light than the conventional imaging to allow short exposure times. Gray-scale high-speed imaging is commonly used to keep illumination requirements at a reasonable level, and consequently, the use of hand tracking methods relying specifically on color information becomes impractical.

Recent progress in the domain of HCI has allowed the next generation of user interfaces, combining touch input with stereoscopic 3D (S3D) content visualization. The S3D pro-

vides depth information that helps to distinguish depth and structure from the viewed content as well as enhances the ability to detect camouflaged objects and increases the recognizability of the surface material [55, 106]. Moreover, the stereoscopic presentation enhances the accuracy of visually guided touching and grasping movements [96].

3.3 General Framework

The proposed framework for building a system to measure the hand movements of test subjects in touch-screen experiment tasks is shown in Figure 3.2. The first step of the framework is to design and to build the measurement setup which comprises cameras, illumination, a display, and other interacting devices, and the required hardware for triggering the recording and storing the recordings. The main considerations when designing an HCI measurement setup are that it should not interfere with the usability of the user interface and that it should offer a natural setting for the test subjects. Moreover, in order to accurately record fast phenomena such as reaction times and to robustly track rapid hand movements, high frame rates are needed for the imaging. To produce videos of good quality, the high-speed imaging requires more light compared to imaging at conventional frame rates. This aspect is particularly important in designing the illumination because of the shorter exposure times with high-speed imaging and because bright illumination can disturb the test subjects. Moreover, the illumination should not result in flickering in the recorded videos which can be an issue with common light sources such as fluorescent or tungsten lights. The available flicker-free light sources include light emitting diode (LED) light panels with reliable and constant power sources and hydrargyrum medium-arc iodide (HMI) lamps where the flicker can be avoided by using electronic ballast that operates at high frequencies.

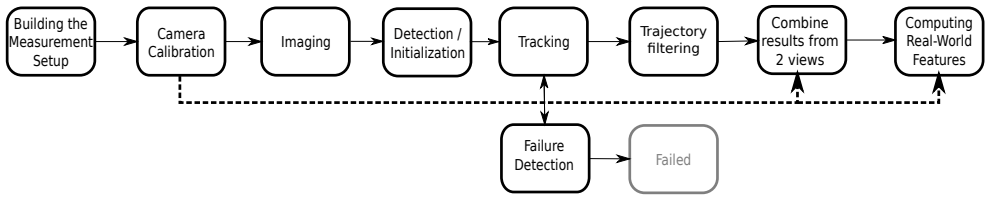


Figure 3.2: An overview of the measurement framework.

The second step in the framework is to calibrate the cameras by determining the intrinsic and extrinsic camera parameters to eventually obtain the mapping to the real-world coordinates from the image point locations. The camera calibration process used in this work is explained in Section 2.2.3. To produce accurate measurements, camera calibration provides a way to acquire the undistorted real-world measurements of the moving object trajectories. This is indicated in Figure 3.2 with the dashed line.

The imaging step should include a synchronized camera startup or a startup marker to obtain a visual cue of the starting point of each action to be recorded. Moreover, it would be beneficial to have an accurate timer visible in all the camera views in order to be able to synchronize the cameras more easily in later parts of the framework.

After imaging an experiment, the object needs to be detected before its movement can be tracked. In a typical controlled HCI study, the hand or finger movement starts from a static trigger box or another predefined location which can be used to initialize the tracking. However, if the initial location is unknown, an object detection module based on computer vision is needed before the tracking module. If the scene background is static, a simple method such as background subtraction can be utilized for the detection. However, in the case of a non-stationary background, simple background detection is insufficient, and the detection can be performed using other detection methods. It should be noted that the initialization of the object location plays an important role in the tracking process since a typical tracking method utilizes the initial location to generate the object model used for tracking.

Tracking is applied in order to follow the location of the detected or otherwise initialized target object while it is moving. In general, the idea of object trackers is to repeatedly estimate the transformation of an object from time step t to $t + 1$, i.e., from one video frame to the next one. In most cases, the transformation is simply the translation of an object. However, there are situations where a more advanced motion model is required that takes into account aspects such as rotation, skew, and scale changes. An extensive comparison of object tracking algorithms for measuring hand movements in the HCI study is presented in *Publication I*. Otherwise, the trackers that perform well in object tracking competitions, such as, [56, 58] should be considered.

A failure detection system is needed in situations where a highly robust tracking system is required or massive datasets are processed. A number of different methods detecting tracking failures can be found in the literature [28, 38, 48, 97, 118]. One approach is to use backtracking and to check whether the backtracked trajectory matches the original tracked trajectory [38]. Other methods include gathering samples of the earlier appearances of the object and comparing them to the currently tracked window using similarity measures [48, 118].

In typical controlled HCI studies, the start and end points of the trajectory are often known. In the touch-screen experiment, for example, the point on the screen that the test subject touches is known, and that can be used to implement a method to detect failures in tracking. When failures are detected, either the tracking can be repeated with another tracking method, or the incorrect trajectory can be excluded from further analysis. For cases where the end location of the trajectory is unknown, a reliable back tracking or a drifting detection method should be applied. In HCI studies, methods such as good features to track [97] and metrics for the performance evaluation of video object segmentation and tracking without the ground-truth [28] that are based on earlier templates of an object work relatively well because the target object is usually a hand or a finger. The hand and finger contain well identifiable features that can be used to detect whether the tracker loses the target. Furthermore, the object detection methods used for the tracker initialization can be applied to the last frames to test whether the end point of the tracked trajectory contains the correct object.

Publication I showed that extracting higher level features from the tracking results can be challenging. Typical object trackers operate at the pixel level and the resulting trajectory often contains noise. Noise causes problems in the determination of derived quantities such as velocity and acceleration, especially in the case of high-speed videos, where movements between frames are very small (often less than a pixel). Consequently, filtering

the trajectories is required. Testing and evaluation of different filtering algorithms for the trajectory smoothing are described in detail in *Publication I*.

Finally, to reconstruct hand trajectories in 3D, tracking results from multiple views obtained from normal and high-speed videos should be synchronized and combined. After the synchronization and finding the corresponding trajectory points from multiple views, 3D trajectory reconstruction can be performed as discussed in Section 2.2.3. The 3D trajectory reconstruction and results are described in detail in *Publication II* and *Publication III*.

3.4 3D Touch-Screen Experiment

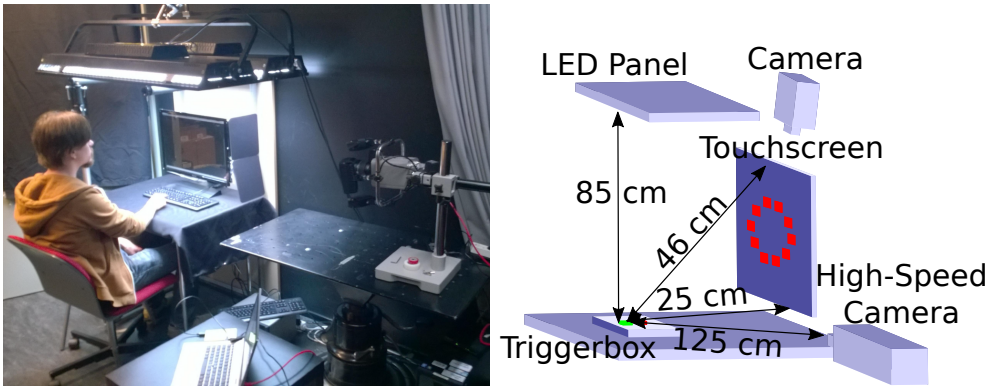
The applicability of the proposed framework is demonstrated in this work with an HCI experiment using a stereoscopic 3D touch-screen setup. In the stereoscopic presentation, the left and right eyes see different stimuli and a depth effect is perceived. However, the depth effect is illusory, which means that the perceived depth is contradicting with the location of the screen. This experiment was built to investigate if different perceived depths affect hand movements and the interaction experience. The data collection, test subject selection, and part of the planning of the experimental setup was carried out at the Institute of Behavioural Sciences of the University of Helsinki, Finland.

3.4.1 Data

In the experiment, test subjects were advised to perform intentional single finger pointing actions from a trigger-box (Cedrus RB-530 response pad) toward a visual stimulus that was at a different parallax than others on the touch-screen (BenQ XL2420Z Display with DigiTouch 24" Touch Overlay). Hand movements were recorded with a Mega Speed MS50K high-speed camera equipped with a Nikon Nikkor AF-S 14-24 mm F2.8G objective lens fixed at a focal length of 14 mm. In addition, a normal-speed Sony HDR-SR12 camera was used. The high-speed camera was positioned on the right side of the test setup, and the distance to the screen was approximately 1.25 meters. The normal-speed camera was mounted above the touch-screen. The lighting was arranged using an overhead LED light panel 85 cm above the table surface. The test subject sat at a distance of approximately 65 cm from the touch-screen and the trigger-box was placed approximately 40 cm away from the test subject. The setup is illustrated in Figure 3.3.

The data for the full 3D touch-screen experiment was collected by using 20 test subjects. There were 10 targets of 100×100 pixels (approx. 28×28 mm) arranged in a circle formation on the screen. The formation of the targets is shown in Figure 3.3. The selected target to be pointed at on the touchscreen was indicated by parallax disparity or by color information. The disparity defines the difference in the target object locations between the images seen by the left and right eyes causing the target object to appear in front or behind the screen. A disparity of 6 pixels means that the object appears clearly in front of the screen and a disparity of -2 pixels means that the object appears slightly behind screen.

Pointing actions were divided into nine blocks. Each of the test subjects did 40 practice pointing actions in order to become acquainted with the test setup. The practice pointing



(a) (b)

Figure 3.3: 3D touch-screen experiment setup: (a) an image of the pilot stage setting; (b) a drawing of the finalized settings with distance measurements.

actions, blocks one and two with 20 pointing actions each, made use of targets with disparities of 6 and -6. After the practice rounds, the test subjects performed four rounds of 40 pointing actions toward disparities $2, -6, 6, -2$. These were blocks three to six. After the main tests, two rounds of 40 pointing actions, blocks seven and eight, where the disparity changed during the pointing action were performed, and 40 control tests (block nine) were also done. These used color information instead of the change in the disparity to indicate the target to be pointed at. In total, the experiments resulted in 6400 pointing actions performed by the test subjects.

All the pointing actions were recorded with the normal-speed camera. The high-speed videos were recorded so that the first ten pointing actions of each block were recorded and after that every third until the end of the block, resulting in a maximum of 20 recorded pointing actions per block. This limitation was needed because of the memory capacity of the high-speed camera.

3.4.2 Comparison of Trackers

There is still lack of consensus about which performance measures should be used in tracker evaluation [11]. Thus, cross-paper tracker comparison can be hard due to unequal performance measures and non-standard video test sets. The lack of generalized tracking evaluation sets has given rise to a number of online object tracking benchmarks [119, 98, 11, 57, 58, 56] for the benefit of tracking algorithm researchers searching for a good online platform with equal measures and easy-to-use tracking evaluation kits.

Selected Methods

The tracker selection for this research included trackers for which source code was available, and which had appeared in the latest tracking benchmark articles or had provided good results in their original papers. Moreover, the novelty value of each particular tracker influenced the choice between them. It should be noted that tracking algorithms

have evolved quite rapidly over the last few years. The aim of the tracker selection for the 3D touch-screen experiments was to find a tracker suitable also for the high-speed videos in gray-scale and real-time usage, and this ruled out some of the trackers. The selected trackers with information about the representation model and motion estimation method used are shown in Table 3.1. The trackers used both in the normal-speed and high-speed video tests are shown with a white background, the trackers used only in the high-speed video tests are indicated with a light cyan background and the trackers used only in the normal-speed video tests are visualized with a light gray background.

The KCF is an improved version of the kernelized correlation filters introduced in [42]. The tracking with KCF [43] is initialized by cropping out a window of a fixed size from the input image (double the target size) at the target location and updating the object model. The object model is based either on pixel intensity values or the HOG features. The original image is weighted by a cosine window so that the pixel values near the boundaries are weighted to zero. This is done to eliminate discontinuities in the Fourier domain representation because the Fourier transform is periodic and does not respect the image boundaries. The new location of the object is calculated by evaluating the classifier response at all possible sliding window shifts and finding the maximum response, i.e., the point where the correlation is the highest. When a new location is found, new parameters for the object model are linearly interpolated with the ones from the previous frame to provide a short memory for the tracker [43]. The KCF2 method introduced in [110] is an extended version of the KCF method with a scale estimation and color-names features [107].

Another improvement to the KCF, sKCF was proposed in [79]. The sKCF method replaces the cosine window with an adjustable Gaussian windowing function to support target size changes in order to produce better background-foreground separation. The new appearance window size is estimated with a forward-backward optical flow strategy. It extracts relevant key-points of the target area at the successive frames and then estimates the scale change by analyzing the pair-wise difference. The STAPLE method [4] extends the KCF approach with color histograms as an additional representation which is robust to deformation, since they do not depend on the spatial structure within the image patch and for scale estimation. In [56], an improved version of the STAPLE method, the STAPLE+ method, was proposed. While the original algorithm extracts HOG features from a gray-scale image, the STAPLE+ method relies on HOG features retrieved from a color probability map, which are expected to better represent the image patch color information.

Another tracker based on the correlation filters, SCT [15], decomposes tracking into two stages: disintegration and integration. In the first stage, multiple cognitive structural units, attentional feature-based correlation filters (AtCFs), are generated. Each unit consists of an attentional weight estimator and KCF. Each AtCF utilizes a unique pair of features, a six-channel average of RGB and LAB color, and a 31-bin HOG, in addition to a kernel of Gaussian, polynomial, or linear types. In the integration step, the object appearance is expressed as a representative combination of AtCFs, which is memorized for future usage. The KCF is extended with a spatial context model between the target and its surrounding background in the STC [122]. The STC first learns the spatial context model between the target and its surrounding background from the target location in the previous frame based on their spatial correlations in a scene by solving a deconvolution

Table 3.1: The selected tracking methods for the experiments. Representation: Binary Pattern (BP), Color Histogram (CH), Color Histogram of Oriented Gradients (CHOG), Color Names (CN), Gaussian Kernel (GK), Histogram of Oriented Gradients (HOG), Principal Component Analysis (PCA), Pixel Intensity (PI), Structured Support Vector Machine (SSVM). Motion estimation: Coarse Sampling (CS), Dense Sampling (DS), Gradient Descent (GD), Gaussian Model (GM), Particle Filter (PF), Random Sampling (RS).

Method	Representation	Motion Estimation
High-speed tracking with kernelized correlation filters (KCF) [43]	HOG, PI	DS
Fast visual tracking via dense spatio-temporal context learning (STC) [122]	PI	DS
Incremental learning for robust visual tracking (IVT) [90]	PCA	PF
Structured output tracking with kernels (struck) [37]	SSVM, Haar, PI	RS
Real-time compressive tracking (CT) [123]	Haar	DS
Fast compressive tracking (FCT) [124]	Haar	CS, DS
Hough-based tracking of non-rigid objects (HT) [36]	Hough	GM
Log-Euclidean Riemannian subspace and block-division appearance model tracking (LRS) [44]	PI	PF
Robust object tracking with online multiple instance learning (MIL) [1]	Haar	DS
Robust object tracking via sparse collaborative appearance model (RSCM) [114]	PI	PF
Online object tracking with sparse prototypes (SR-PCA) [113]	PCA, PI	RS
Tracking-learning-detection (TLD) [49]	BP	RS
High-speed tracking with kernelized correlation filters v2 (KCF2) [110]	HOG, PI, CN	DS
Structuralist cognitive model for visual tracking (SCT) [15]	HOG, CH	DS
Scalable kernel correlation filter with sparse feature integration (sKCF) [79]	HOG	DS
Sum of template and pixel-wise learners (STAPLE) [4]	HOG, CH	DS
Improved STAPLE tracker with multiple feature integration (STAPLE+) [56]	CHOG	DS
Scale adaptive mean shift (ASMS) [111]	CH	GD
Distractor aware tracker (DAT) [86]	CH	DS

problem. In the next step, the spatio-temporal context model is updated by using the learned spatial context model. Finally, the object location is determined by maximizing the confidence map. Moreover, for every Nth frame, the scale is updated. The object location likelihood is defined as the sum of the conditional probability of the spatial relationship between the object location and its spatial context, and a prior probability which models the appearance of the local context [122]. The algorithm can be listed under the hybrid methods because the context includes the target and its neighboring background making it discriminative. Moreover, the context is also of the target and background, giving it the means of a generative approach.

The IVT method [90] is a tracking method that incrementally learns a low-dimensional subspace representation. The appearance model used in the IVT is an eigenbasis which is formed by computing the eigenvectors of the sample covariance matrix. The IVT learns the eigenbases during the object tracking process and can adapt online to changes in pose, view angle, and the illumination of the target. A new incremental principal component analysis (PCA) algorithm that correctly updates the eigenbasis as well as the mean with a forgetting factor, given one or more additional training data, is an extension of the Sequential Karhunen-Loeve algorithm [64]. The IVT uses a particle filter with an affine image warp as a dynamic model for estimating the target object movement. An affine image warp considers the translation, rotation, scale, aspect ratio and skew of the object motion.

Like the IVT approach, the LRS method uses an affine image warp to model the object movement [44]. During tracking, each image region is warped into a normalized rectangular region by using the estimated affine parameters. In order to obtain the optimal object state, the LRS uses an observation model and a dynamic model. The observation model maps the similarity between the learned appearance model and the image region. A dynamic model updates the particle filter and in order to estimate the optimal state, a particle filtering approach is used. The appearance model is incrementally updated with the image region which has the optimal state [44].

The SRPCA online object tracking method, developed by Wang et al. [113], also uses an affine image warp to model the target motion between two consecutive frames. First the algorithm samples candidate states and then uses an observation model to evaluate them. As a result, the algorithm obtains the best candidate and the related occlusion map. Then the target location is updated. Finally, the algorithm updates the observation model fully, partially, or not at all based on the occlusion map of the sample.

The HT method is a tracking-by-detection approach to tracking non-rigid objects [36]. The method uses the generalized Hough-transform [2] as the representation model. The target region is represented as a segmented area instead of a bounding box. This helps when tracking non-rigid objects and is the main difference from most of the tracking methods. A Gaussian motion model and a discriminative classifier are used in order to find the best match for the segmented object region in the consequent frames. Finally, the classifier is updated based on the new segmented target.

The MIL uses a set of Haar-like features that are computed for each image patch within a specific search radius of the target location. A feature contains two to four rectangles and each of them has a real-valued weight. The value of the feature is then a weighted sum of the pixels in all the rectangles. The motion model is simply an area with a radius

s where the object is equally likely to appear in the next frame. The image patches within the area are then classified based on the features using a discriminative classifier. The best matching image patch is used to update the tracker location. The positive and negative sets of image patches are then cropped out around the new tracker location and they are used to update the appearance model of the tracker [1].

The TLD method consist of tracking, learning and detector parts. The learning part is used to initialize the detector in the first frame and to update the detector using a positive-expert and the negative-expert. An explanation of the experts can be found in [49]. In the initialization phase, 10 bounding boxes within a scanning grid which are near the initial bounding box are warped by geometric transformations to generate 20 warped versions resulting in 200 synthetic positive patches. The negative patches are collected outside of the initializing bounding box.

In the detector phase, an image patch is sampled from an image within the object bounding box and then it is resampled to a normalized resolution (15×15 pixels) regardless of the aspect ratio. The image patches are generated on all possible scales and shifts of an initial bounding box with scale steps of 1.2, horizontal steps of 10 percent of the width, and vertical steps of 10 percent of the height. The similarity between two patches is evaluated using normalized cross correlation. The first part of the cascaded classifier is a patch variance process where patches are rejected from further classification if the variance is greater than 50 percent. The second part of the cascaded classifier is an ensemble classifier where each ensemble consists of n base classifiers. In each base classifier, a number of pixel comparisons are made on the patch resulting in posterior values. Based on the posteriors of individual base classifiers, the patch is classified as the object if the average posterior is greater than 50 percent. The third part is the nearest neighbor classifier where the patch is classified as positive or negative. The patches that are classified as positive are the object detector responses [49].

The tracking part of the TLD method is based on the Median-Flow tracker [48] with added failure detection. The tracker works by selecting a set of points within a bounding box. The points are then tracked using the Lucas-Kanade tracker [72] which generates a sparse motion flow. Point predictions are then evaluated using the Forward-Backward error [48] and normalized cross correlation. Half of the most reliable displacements are used to estimate the bounding box movement using the median of the displacements. If the displacement of a point prediction is more than ten pixels away from the median, it is marked as a failed prediction. If a tracking failure is detected, the bounding box is not returned by the tracker.

The integrator phase is used to combine the bounding boxes of the tracker and the detector into a single bounding box output using the candidate window with the highest similarity to the object model. The newly formed bounding box is then selected as the new object. If neither the tracker nor the detector find the object, the TLD method declares that the object is not visible. With this kind of approach TLD can effectively handle short-term occlusions.

In the struck method, a kernelised structured output support vector machine (SVM) is learned online to provide adaptive tracking. A budgeting method is used that prevents the growth of the number of support vectors beyond a set limit that would normally occur during tracking [37]. The struck method extends the Learning to Localise Objects

with Structured Output Regression tracking method [6]. The principal idea of the struck approach is to create positive samples from areas close to the object, still containing the object, and negative samples further away from the object, only containing the background. Using these positive and negative samples, the object in the next frame can be detected by estimating the change in the object location using an SVM classifier to find the most probable transformation. It finds a statistical correlation between the object of interest and its local context based on the probabilities. Furthermore, the struck method uses a confidence map and obtains the best location by maximizing a location likelihood function of the object. When the new location of the object is found, the classifier is updated with the current location and budget maintenance is carried out. The budget maintenance removes support vectors with a minimum impact on the classifier. New samples of the target object and background are taken. Based on the relevance of the new samples, the model is updated. After that, the samples are optimized and the bounding box is returned [37].

The basis for the CT method is a very high-dimensional multi-scale image feature vector of Haar-like features which is multiplied with a very sparse random measurement matrix. The random measurement matrix is computed only once at the beginning of the tracking. In compressed sensing, using a very sparse random measurement matrix enables an efficient classification of the compressed projected features. The appearance model of CT is generative as the object can be represented based on the features extracted in the compressed domain. CT uses a naive Bayes classifier with online updates in the compressed domain to separate the target from the surrounding background which makes the tracker also discriminative [123]. The FCT and CT methods share the same algorithm for most of the parts. The main novelty of FCT is a coarse-to-fine search strategy where the object is first searched within a large search radius with large shifts of the sliding window. After the coarse search, approximate object location is determined and the fine detection is done in a smaller area with small shifts of the sliding window to find the exact location of the target object [124].

The RSCM approach uses a collaborative model that includes a discriminative classifier, sparsity-based discriminative classifier (SDC), based on holistic templates and a generative model, sparsity-based generative model (SGM), using local representations [114]. In the SDC, positive templates within a radius of a few pixels from the initialized target location are selected and normalized to the same size (32×32 pixels). Intensity values are used as features. Object candidates in the current frame are sampled with a particle filter, and for each particle, a confidence value for the SDC is calculated. The confidence value is used in a likelihood function that combines the confidence value based on the holistic templates, SDC, and the similarity measures based on the local patches, SGM [114]. In RSCM, positive templates remain the same during the entire sequence, but negative templates are updated after a few frames. In [114], the update was conducted every fifth frame.

The DAT method [86] uses a color histogram-based Bayes classifier for differentiating the object from the background. The DAT uses an additional distractor-aware model which allows robust detection of distracting objects. The distractor-aware model works by selecting possible object-distractors from the object surroundings and reduces their impact on the combined object model. For localization, DAT uses votes based on the combined object model and distance score based on the Euclidean distance to the previous

object center to penalize larger inter-frame movements.

The ASMS [111] tracker extends the mean-shift tracking algorithm [18] by targeting the problem of a fixed size tracking window. The ASMS encompasses a regularized scale estimation mechanism. The regularization is done with a technique that the authors named the forward-backward consistency check. This uses reverse tracking to check that the object size has not changed mistakenly. In the case of scale inconsistency, the object size is updated as a weighted combination of the previous size, the new estimated size and the initial size. Another improvement is the introduction of a background ratio weighting that uses the target background color information computed over its neighborhood in the first frame.

Data

The dataset used for the tracker performance comparison in *Publication I* contained a set of 11 high-speed videos with a spatial resolution of 800×600 recorded at 500 fps consisting of 11 individual pointing actions. This dataset is referred to as *Dataset I* in the following sections of this work. The videos were between 544 and 1407 frames long, and contained 10798 frames in total. The start-point for all the sequences was the same, but there were ten possible end point areas on the touch-screen where the finger movement should stop. The ground truth was annotated manually for every fifth frame and then interpolated using spline interpolation to obtain the ground truth for every frame.

The dataset used for tracker performance comparison in *Publication II* contained a set of 17 pointing action videos. This dataset is referred to as *Dataset II* in the following sections of this work. *Dataset II* contains 17 test subjects with a varying appearance of hands. These normal-speed videos were recorded using interlaced encoding with 50 fields per second and a resolution of 1440×1080 (4:3). A yet another deinterlacing filter (yadif) [31] was used for deinterlacing with frame-to-frame conversion, producing 25 fps videos, and a field-to-frame conversion producing double frame rate 50 fps videos. The ground truth for the evaluation was done manually by annotating each frame in all the 17 videos. The corresponding high-speed videos were manually synchronized with the normal speed videos.

Experimental Arrangements

The tracking experiments were carried out using the original implementations of the authors of each method, except in the case of the MIL and KCF2 methods. For MIL, the implementation by Luo [74], and for KCF2, the implementation by Vojir [110] are used. The search area parameters of the trackers were tuned for the used datasets if the implementation allowed this. This was done to see whether the trackers could perform faster without losing their target. For the other parameters, the default values proposed by the authors were used. To minimize random factors in tracking, the tracking methods were run ten times for each video, and the results were averaged. In this research, both background subtraction and the fixed location were used to initialize the trackers. The fixed locations were used in the evaluation phase of the tracking algorithms and background subtraction was used for automated finger location determination in the large-scale video processing.

Table 3.2: Tracking results for *Dataset I*: the percentage of correctly tracked frames (TR%) and average center location errors (Err.), and the processing speed (fps). The value ranges of the results are also shown. The best results are shown in bold.

Method	TR%	TR% range	Err.	Err. range	fps	fps range
CT	79.43%	0-100%	18.43	3.5-76	99.97	63-121
FCT	17.14%	0-73%	58.74	16-92	118.73	72-150
HT	97.12%	36-100%	15.29	3.1-226	4.65	4-4.9
IVT	74.50%	15-100%	86.75	2.0-448	63.38	51-70
KCF	**100%**	-	**4.65**	1.4-7.4	979.97	728-1236
LRS	20.51%	2-47%	291.32	76-540	8.79	7.8-9.4
MIL	93.82%	24-100%	11.35	2.8-138	0.55	0.4-0.6
RSCM	86.81%	40-100%	18.84	2.2-126	2.50	2.0-2.8
SRPCA	83.64%	24-100%	72.38	1.7-366	10.52	8.3-12.5
STC	**100%**	-	5.13	2.3-6.9	**1291.03**	1156-1330
struck	**100%**	-	4.72	1.6-6.5	118.62	99-153
TLD	68.48%	16-100%	43.55	4.4-139	16.46	8.8-24

The performances of the selected trackers were compared against the ground-truth. The accuracy of the trackers is measured as the distance from the ground-truth center point to the tracker's center point. This is known as the center location error. The trackers were evaluated using *Dataset I* in *Publication I*. The tracking rate was evaluated as the percentage of frames where the tracking window center location was within 32 pixels from the ground truth center location. The value of 32 pixels for the threshold was experimentally selected - values above this threshold would imply that the tracking had completely lost the target.

In *Publication II*, the tracking evaluation moved to normal-speed video tracking where eleven trackers were evaluated with *Dataset II*. The trackers were evaluated using both backward and forward tracking and at the rates of 25 fps and 50 fps. The threshold for the calculation of the percentage of correctly tracked frames was lowered to 16 pixels. The new value of 16 pixels for the threshold was also experimentally selected and values above the threshold would imply that the tracking had drifted half way of the target.

Results

High-speed video tracking was the first stage of the research and 12 trackers were evaluated with 11 high-speed videos. In *Publication I* with *Dataset I*, the tracking rate of 100% was achieved by three of the trackers: KCF, STC and struck. Table 3.2 shows the results of this evaluation. The KCF method provided the best overall results with the smallest average center location error of 4.65 pixels. Also, struck and STC achieved a high degree of accuracy. The KCF method was also the second fastest to compute, 980 fps, and was selected for the further use during this work.

The fps measure used in the experiments was calculated without including the image loading times in the calculations to get the raw frame processing speed. The highest value was measured for the STC method which showed the best average performance

and for KCF which had the peak performance of over 1200 fps. Both of these achieved processing speeds which were well over the frame rate of the videos. However, it should be noted that due to the different programming environments (MATLAB, C, etc.) and levels of performance optimization, these results should be considered only directive.

Normal-speed video tracking was the second phase of the research and 11 trackers were evaluated with 17 normal-speed videos. All tracking methods were evaluated for normal frame rate (NFR) and double frame rate (DFR) videos. Moreover, in addition to the normal forward tracking (FT), backward tracking (BT) from the end of video to the beginning was evaluated. Thereby, the following four tracker evaluation cases were considered: NFR/FT, DFR/FT, NFR/BT and DFR/BT.

The accuracy of tracking was measured with the center location error (Err.) in respect to the ground truth. The percentage of correctly tracked frames (TR%), i.e. the frames where the distance between the ground truth and the estimated location was below a fixed threshold (16 pixels), was used as the measure of robustness. The results for all the test videos are shown in Table 3.3.

Table 3.3: Tracking results. The best results for Forward and Backward tracking are given in bold.

	Forward tracking				Backward tracking			
	NFR		DFR		NFR		DFR	
Method	Err.	TR%	Err.	TR%	Err.	TR%	Err.	TR%
Staple	141.7	44	102.5	43	129.5	45	14.6	84
Staple+	146.7	43	102.6	40	113.5	43	14.9	82
DAT	100.3	41	52.2	46	41.8	44	33.9	51
KCF	120.3	42	45.5	50	72.4	62	**10.8**	83
KCF2	145.0	49	63.1	61	56.3	75	19.4	**86**
ASMS	146.1	39	109.6	28	52.7	40	53.7	41
SCT	90.6	57	**25.7**	**77**	29.7	83	17.6	81
STC	136.4	39	52.1	51	41.2	75	13.6	83
sKCF	146.4	46	71.1	49	136.4	48	29.3	80
STRUCK	117.3	46	43.4	62	294.6	15	112.6	38
IVT	177.4	43	183.7	31	374.4	7	336.2	10

The best forward tracking results with 50 fps were produced by the SCT method, with 77% correctly tracked frames, and the best backward tracking results were achieved with the KCF2 tracker, with 86% of correctly tracked frames. The KCF2 was selected for the further use in this work because the backward tracking provided better results overall.

In the experiments, a tracking failure detection method was needed in order to be able to reliably process a large number of trajectories in the case of all the normal-speed and high-speed videos collected in the experiments. The implemented failure detection system was based on the fact that the trajectory had to end in a specific area of the projected touch-screen point or in an area near the trigger-box start button. If the correct end point was not reached in the high-speed videos with the default gray-level features used by the KCF tracker, the tracking was repeated with more computationally demanding HOG features. If the tracking failed again, it was considered incorrect and was excluded

Table 3.4: Minimal mean of Location Errors (LE), Velocity Errors (VE), and Acceleration Errors (AE) with different filtering methods. In the parentheses is the filtering window size which gave the best result for the filter. The best results are shown in bold.

Error	Moving Average	LOWESS	LOESS	Savitzky-Golay	TVD	UKS	original
LE	4.6057 (3)	4.6057 (4)	**4.6029** (34)	4.6030 (25)	4.6474	4.6033	4.6099
VE	0.0440 (17)	0.0419 (18)	**0.0415** (34)	0.0428 (31)	0.2052	0.0421	0.2085
AE	0.0137 (83)	**0.0118** (23)	0.0119 (53)	0.0137 (97)	0.3026	0.0125	0.3074

from further analysis. In the normal-speed videos, if the tracking with KCF2 failed to reach the trigger-box button, it was considered incorrect and was excluded from further analysis.

3.4.3 Comparison of Filtering Methods

Raw trajectory data usually contain small spatial location fluctuations that can make the calculation of accurate velocities and accelerations impossible. Consequently, filtering raw trajectory data with an appropriate filtering method is needed. For filtering the trajectory data, eight filtering methods were considered: MA [99], KF [116, 117], EKF [78], UKF [47], LOESS [16], LOWESS [16], S-G [82], and TVD [13].

Data

The trajectories produced by the best performing trackers were used to evaluate the filtering performance of each of the methods. The trajectory data for *Publication I* was produced by the KCF tracker, and the trajectory date for *Publication II* was produced by the SCT and KCF2 trackers. The derivatives of the location data, velocity and acceleration, were used in addition to the location data in *Publication I*. This was done in order to gain an idea of how much filtering was needed to produce appropriate results also for the location derivatives.

Results

Table 3.4 summarizes the trajectory filtering results of *Publication I*. The results were calculated by averaging the results from all the dataset trajectories tracked with the KCF tracker. The window size and method parameters were optimized separately for each filtering method. Filtering with the unscented Kalman smoother (UKS) and TVD are included for comparison. The UKS method was selected to represent the Kalman smoother algorithms since the Extended Kalman Smoother and UKS produced similar results. Velocity and acceleration curves for trajectories obtained using Kalman filtering based methods were computed using the Kalman filtering motion model. For the trajectories obtained using the other filtering methods, velocity and acceleration curves were computed based on Euclidean distances between the trajectory points in consecutive frames.

It was found that the LOESS filtering method achieved the best results for the purposes of this work, providing the lowest errors in velocity, acceleration and location results, with the most constant filtering window sizes, against the ground truth values. Based on the findings, the LOESS filtering method with the window size of 40 frames was selected for the further use during this work. Moreover, during the experiments it was noted that filtering the location data alone was not enough to get appropriate results from the velocity and acceleration data. Therefore the velocity and acceleration data were filtered after they were calculated from the location data. Moreover, the LOESS filtering method was the least sensitive to the window size with optimal filtering results from a window size range of 34 to 53.

Based on the results in *Publication I*, the filtering evaluation in *Publication II* was conducted only for the LOESS method with different window sizes and using the trajectories retrieved from the double frame rate videos. The forward and back-tracking trajectories of the best performing trackers (SCT and KCF2) were used for the filtering evaluation. The efficiency was evaluated against the ground truth using the mean of location error measure. The results are shown in Table 3.5.

Table 3.5: Mean and variance (in parentheses) of the location error for the original and smoothed trajectory data with three filtering window sizes. The best results are shown in bold.

Case	Original	Filtered (window size)		
		5	7	9
DFT/FT [SCT]	4.02 (6.08)	**3.97 (5.10)**	4.40 (6.62)	5.29 (10.78)
DFR/BT [KCF2]	**3.27** (4.48)	3.32 **(4.36)**	3.68 (5.29)	4.30 (8.47)

The filtering with a span size of five frames reduced the variance of the location error measure. For the DFT/FT tracking case with the SCT tracker, filtering also marginally improved the location error.

3.4.4 Video Synchronization and 3D Reconstruction

Publication III introduced the automated video synchronization of the normal-speed and high-speed trajectories to the framework. This was done to successfully reconstruct the trajectories in 3D for each of the videos.

Data

Publication II provided the initial 3D reconstruction results using seven manually aligned corresponding videos from *Dataset II*. In *Publication III*, tracking, video synchronization and 3D reconstruction of the framework were considered with all the available normal- and high-speed videos and trajectories.

Results

Seven manually aligned corresponding normal- and high-speed videos were used to evaluate the 3D reconstruction error in *Publication II*. The average re-projection error of all

the trajectory points used in the initial 3D reconstruction experiment comprised 3.89 pixels for the high-speed and 5.72 pixels for the normal-speed camera videos.

In *Publication III*, all the available normal- and high-speed videos were considered, and therefore manual alignment would have been too laborious and there was a need for automatic alignment. In order to automatically align the normal-speed videos with the high-speed videos, the following procedure was performed. First, the normal-speed videos were divided into blocks, of one to nine, based on the longer breaks in pointing actions caused by the memory capacity of the high-speed camera. A coarse alignment was performed using timestamps accompanied by the high-speed videos and the starting time of the normal-speed videos. The final step of the trajectory synchronization was to find a point of the finger trajectory which could be detected from both videos. The trigger box had a white button that was visible in both views, and the point where the trajectory passed the button was used for the trajectory time event matching. The final alignment was done by searching for the delay which maximized the correlation between the timestamp sequences for normal-speed and high-speed videos. As it can be seen in Figure 3.4, the timestamp correlations for the corresponding events passing the white button are periodical. Therefore, simply finding the minimal timestamp difference would not work. The figure presents an example of synchronizing a block of pointing actions. In Figure 3.4a, all the samples are well correlated within a tight time range. Figure 3.4b presents a challenging situation and based on the figure it is impossible to say which time difference gives a good result. This is mainly caused by the low count of sequences which were tracked correctly in both videos because only correctly tracked videos were included in the video synchronization process.

The normal-speed videos contained 6400 pointing actions out of which 6125 were detected as correct pointing actions without interfering objects or another hand in the view. There were in total 4216 out of 6125, 69%, of good trajectories tracked from the normal-speed videos. Moreover, 2597 high-speed videos were recorded out of which 1999 in total, 77%, were tracked correctly. In total, 1161 of the pointing actions were correctly tracked from the both videos and were synchronized correctly. Since there was no ground truth data for the 3D trajectories, the 3D reconstruction accuracy was assessed by using the re-projection error measure [39].

The average re-projection error for all the trajectory points used in the initial 3D reconstruction experiment comprised 3.89 pixels for the high-speed and 5.72 pixels for the normal-speed camera videos. These values correspond to a one to three millimeter degree of accuracy in real-world units. However, the mean re-projection error for all the trajectory points used from 1161 videos in the 3D reconstruction experiment was 31.2 pixels. This average re-projection error corresponds to approximately ten millimeters in real-world units. The resulting 3D trajectories are visualized in Figure 3.5. Units displayed in the figure are centimeters and the high-speed camera is placed at (0,0,0) location. Moreover, example trajectories from one block of pointing actions are visualized in Figure 3.6. Features calculated from the corresponding trajectories are visualized in Figure 3.7. One line-style in these two figures corresponds to a single pointing action.

It was noticed that the normal-speed videos did not capture the full trajectories of the hand movement. This was caused by the touch interface in front of the monitor blocking the view of the fingertip near the monitor surface. As a consequence, the depth information from the 3D reconstruction was used only to supplement the trajectories obtained

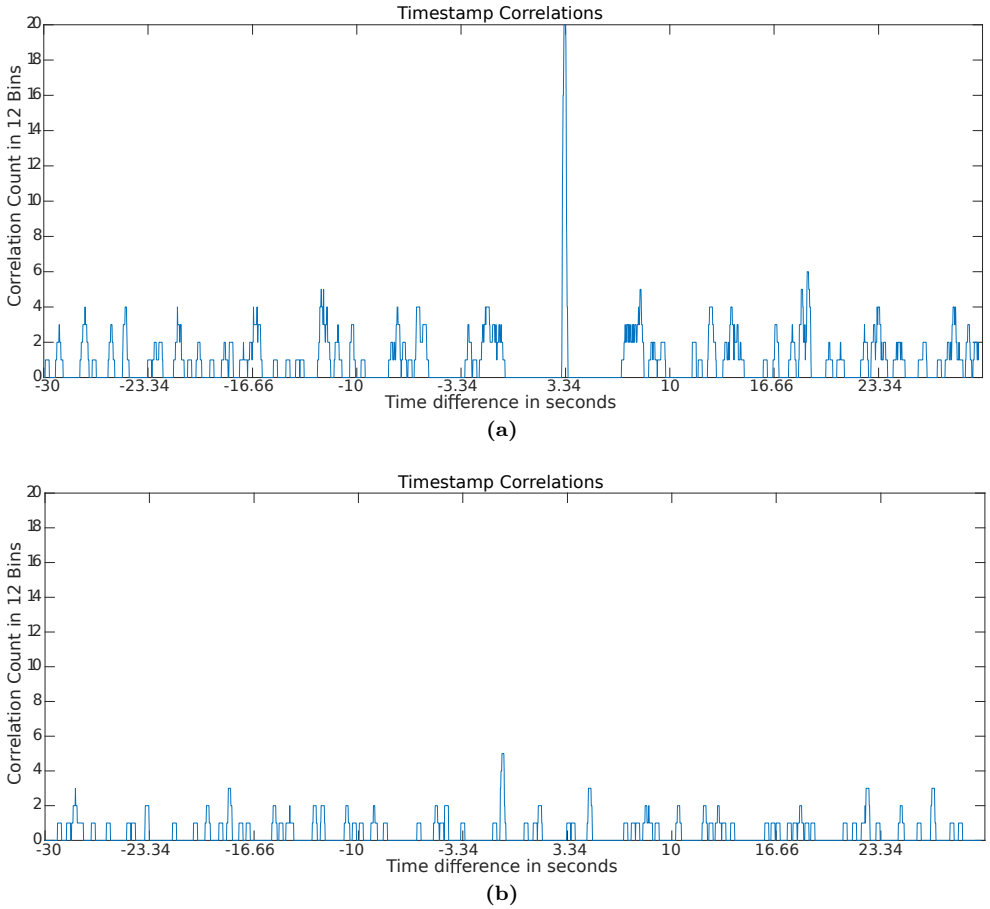


Figure 3.4: Example of synchronizing two blocks of pointing actions. (a) High correlation in one time range (around 3.34 seconds). (b) Low correlations within the whole time range.

from tracking the high-speed videos. The depth information was interpolated, using a fourth order polynomial, to match 500 fps and the missing parts at the end of the trajectories were extrapolated with the last known depth information. This resulted in full 3D trajectories where the error in depth information was approximately ten millimeters and the error in the other two axes was less than five millimeters.

3.4.5 Trajectory Analysis

It is important to develop appropriate metrics for the purpose of analyzing the experience of intentional hand movements in HCI and to gain more understanding about the movement control. According to [27], the features of interest in time and space of the hand movements include the point of the maximum velocity, maximal acceleration and deceleration. The points where the deceleration starts, as measured from the start-point

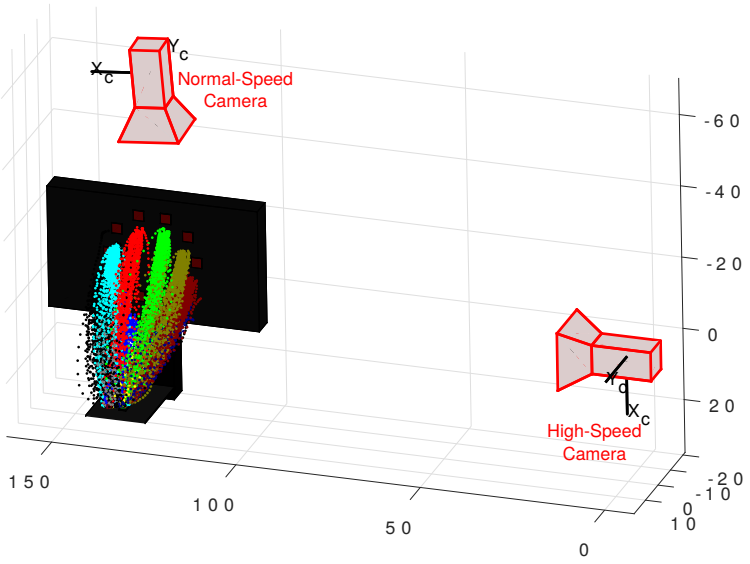


Figure 3.5: Reconstruction results of all the 1172 pointing actions. The different colors indicate trajectories toward different targets.

and/or from the end-point, also provide useful information when defining the smoothness and stability of the movement. The features to be used for trajectory analysis in this work were selected based on this information.

Multiple features of the hand trajectories were measured in the experiments. The features included the acceleration, velocity as well as the movement time and reaction time. The reaction time is the time that it took for the test subject to determine the target after a test image was shown on the screen. The movement time is the time that it took for the test subject to reach the touch-screen surface from the trigger-box.

The used features were calculated based on the tracking data which can provide location, velocity, acceleration and accuracy measures. These can be calculated from tracked 2D trajectories or from reconstructed 3D real-world trajectories. The velocity of a moving object was calculated as the distance it traveled with respect to time. The velocity is the movement of the object between two consecutive frames. To calculate the distance, a Euclidean distance formula is used. The Euclidean distance d between points p and q in a n-dimensional case is calculated as

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}, \tag{3.1}$$

where q_i and p_i are the i-dimension components of the points p and q. The velocity can be negative or positive. The magnitude of velocity is speed, which can only be positive. Velocity v can be calculated as

$$v = \frac{\triangle d}{\triangle t}, \tag{3.2}$$

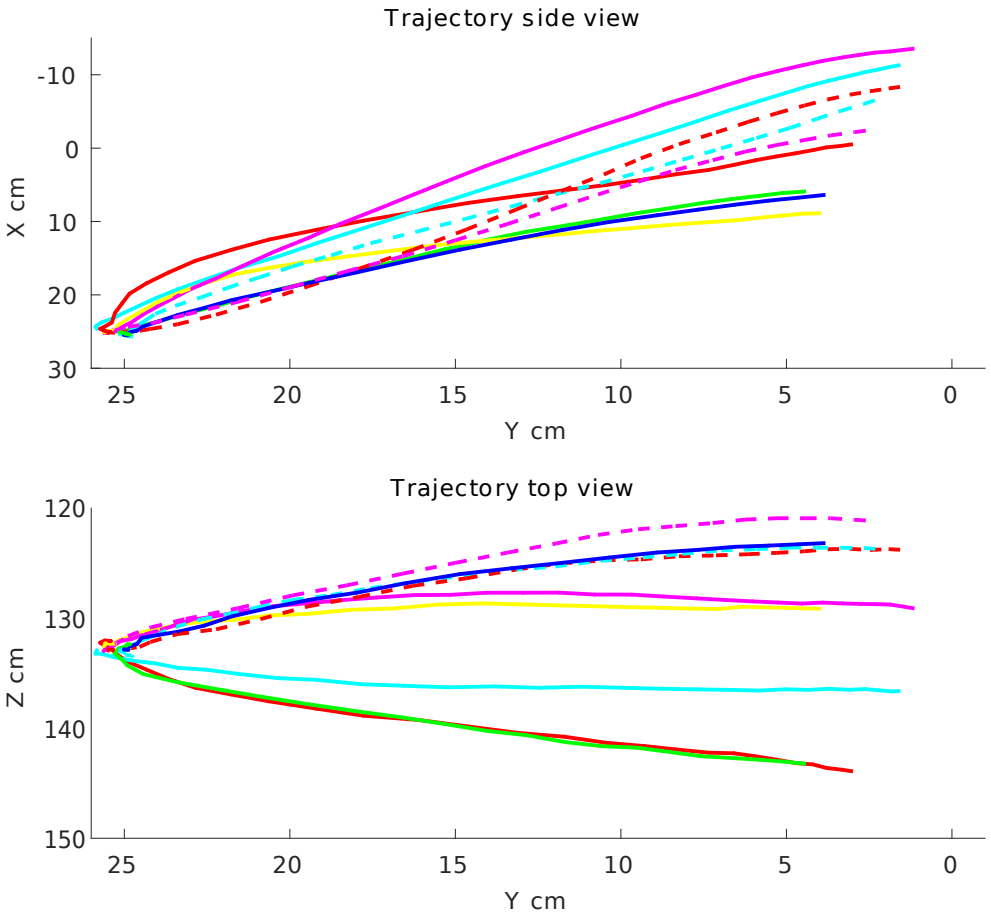


Figure 3.6: Trajectories from one block of pointing actions are visualized from two different viewpoints. The different line-styles indicate different pointing actions.

where $\triangle d$ denotes displacement and $\triangle t$ change in time. The acceleration can be calculated as the first derivative of velocity.

Data

The data for the trajectory analysis was selected from the main test blocks with disparities of 2, -2, 6, and -6. There were 705 3D trajectories in total, 82 with disparity of 2, 196 with -2, 252 with 6, and 175 with a disparity of -6.

Results

Eleven extracted trajectory features of the trajectories of high-speed videos supplemented with depth information were used for the statistical analysis. The analysis was conducted

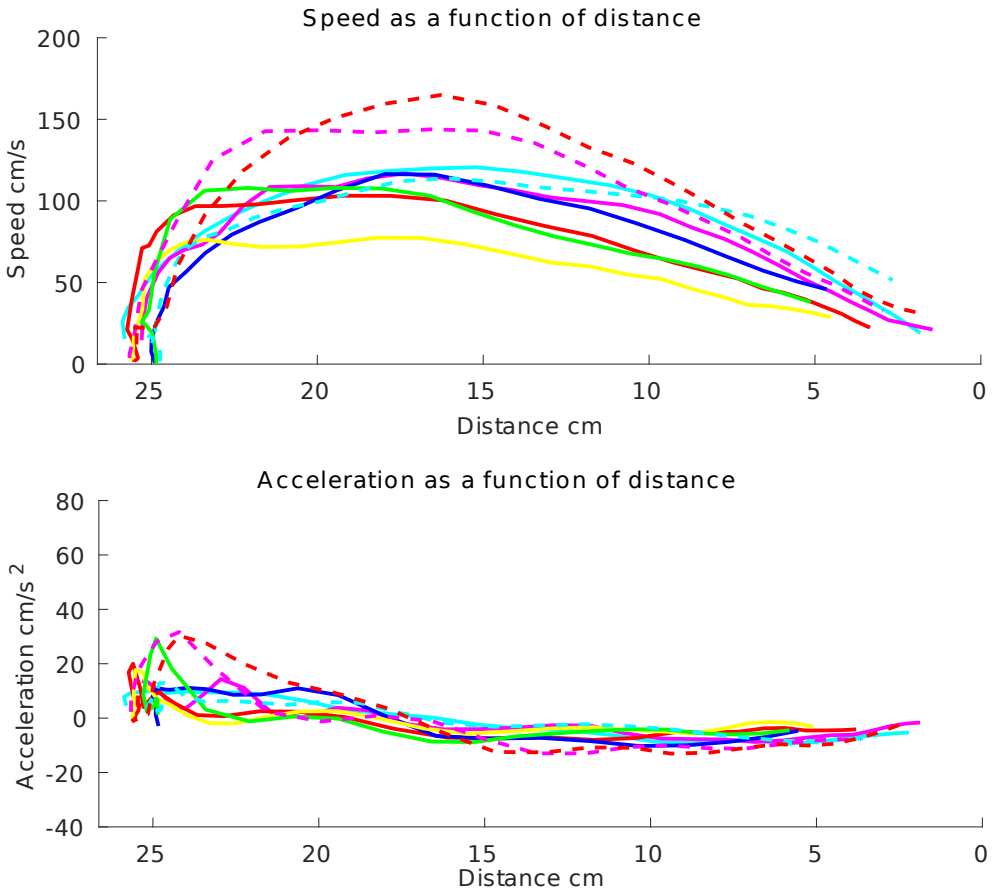


Figure 3.7: Visualization of speed and acceleration features calculated from the trajectories of Figure 3.6. Line-styles correspond to the ones used in Figure 3.6.

using a two-sample t-test, with a 95% confidence interval. The analysis showed that there were features that had a statistically significant difference in their means. The differentiating features were: the mean velocity, maximum velocity, maximum acceleration, mean 2nd sub-movement acceleration and the point where the 2nd sub-movements accelerated after deceleration. However, between two pairs of disparity classes, {2, -2} and {2, -6}, none of the features varied by their means in a statistically significant way. Moreover, the median velocity, maximum 2nd sub-movement velocity, maximum 2nd sub-movement acceleration, mean 2nd sub-movement velocity, the start point of the deceleration and the starting point of acceleration in the 2nd sub-movement features did not provide statistically significant differences between any of the used disparity pairs. The results are shown in Table 3.6.

Table 3.6: Rejection of the null hypothesis for the disparity pairs using the selected features.

Features	**Disparity pairs**						**Total**
	2 -2	2 6	2 -6	-2 6	-2 -6	6 -6	
Mean velocity	0	0	0	**1**	0	**1**	2
Median velocity	0	0	0	0	0	0	0
Maximum velocity	0	**1**	0	0	0	**1**	2
Maximum acceleration	0	0	0	0	0	**1**	1
Maximum 2nd sub-movement velocity	0	0	0	0	0	0	0
Maximum 2nd sub-movement acceleration	0	0	0	0	0	0	0
Mean 2nd sub-movement velocity	0	0	0	0	0	0	0
Mean 2nd sub-movement acceleration	0	**1**	0	**1**	0	0	2
Deceleration start Point	0	0	0	0	0	0	0
2nd sub-movement start point 1	0	0	0	0	0	0	0
2nd sub-movement start point 2	0	0	0	**1**	**1**	0	2
Total	0	2	0	3	1	3	9

3.5 Discussion

A framework for measuring hand movements, in particular pointing actions, in HCI situations using a multi-camera system consisting of a normal-speed and high-speed camera was introduced. The framework was evaluated with a large-scale study of HCI.

A tracking accuracy of 4.65 pixels and 100% correctly tracked frames was achieved in *Publication I* with a KCF tracker. The KCF2 tracker, with 86% of correctly tracked frames and an average accuracy of 19.4 pixels, produced the best results in *Publication II*. Moreover, the full set of normal-speed videos were tracked with a 69% success rate and the full set of high-speed videos with a 77% success rate. The rates indicate that there is still lots of room to improve, but nevertheless the selected methods provided a good solution to track moving objects robustly and accurately enough for the objectives of this work.

The accuracy of ten millimeters for the depth information and the accuracy of five millimeters on other two axes achieved in the 3D reconstruction with the complete framework provides a good starting point for further analysis and improvements, even though the initial test in *Publication II* delivered better results. The larger error that was measured from the full dataset was likely caused by tracking shifts and possible video synchronization issues. Moreover, there were times when the trigger-box and the monitor moved when a user was performing the test. This could have had an impact on the full dataset results as well.

The backward tracking used in the normal-speed videos and the forward tracking used in the high-speed videos caused the trajectories to be the most accurate at different ends

of the movement. This made the inlier search, for the essential matrix calculation, more difficult for the 3D reconstruction task. The issues raised during the synchronization and 3D reconstruction stages could be avoided with a more thorough planning of the setup and more careful selection of the equipment used. Moreover, using trackers which are capable of tracking the target using a rotating bounding box could provide more accurate center locations [10, 68].

The trajectory features that showed statistically different means in the results could potentially be used to differentiate the pointing action performed towards different disparities. However, more features or possibly combinations of features should be used to gain a better understanding of how the movement differs between the different disparity classes. As expected, the smaller disparity changes of 2 and -2 had a minor impact on the hand movements according to the computed features whereas the disparity values of 6 and -6 had a bigger impact. The large positive disparity of 6, where the target appears in front of the screen, seemed to have a more prominent effect on the pointing actions than the others. Overall, the velocity features were better to distinguish the pointing actions toward targets at different disparities than the acceleration features. Furthermore, features that would produce clear difference between the trajectories would be needed to gain a better understanding of the hand movements towards targets at different disparities.

It was learned during the 3D touch-screen research that it would be beneficial to use well fixed test setups to prevent critical parts of the equipment from moving. Moreover, the late decision to include a normal-speed camera in the framework resulted in issues that could have been solved in the setup phase of the project. These issues were mainly the synchronization and positioning of the normal-speed camera in the setup. The issues could have been corrected with a more thorough planning of the experiments. Nevertheless, the framework used achieved good accuracy and a few statistically meaningful features describing the difference of trajectories toward targets at different disparities were found. This would indicate that there were minor differences in the pointing actions towards the targets at different disparities.

Further tuning of the tracker parameters should be conducted, or a new selection of trackers should be considered for high-speed or normal-speed video tracking in order to achieve sub-pixel accuracy and faster operating speeds. Furthermore, specific painted markers could be an unobtrusive way to find the sub-pixel accurate location of the moving object, but this approach is limited to piecewise rigid objects. A failure detection system using, for example, rapid acceleration or velocity changes could be added to the frameworks of both applications to further assist the failure detection systems.

Video synchronization with multiple known locations, which could be used to synchronize the system, should improve the accuracy in the case of different imaging rates. However, these issues could be avoided if the cameras use accurate and synchronized time-stamping systems. Multi-camera setups and better imaging equipment could provide better results for the 3D reconstruction.

Chapter IV

Moving Droplet Analysis in a Chemical Mass Transfer Experiment

This chapter contains the main findings of the chemical mass transfer experiment. The experiment focuses on liquid-liquid extraction using droplets in a glass column. An imaging setup was designed and constructed to view parts of the glass column in order to follow the droplets moving inside the column. This chapter covers the main ideas and motivations behind the results presented in the publications. The detailed descriptions are provided in *Publication IV* and *Publication V*.

4.1 Background

Liquid-liquid extraction is used in a variety of industries, such as the petroleum, food, hydrometallurgy, and chemical industries. It is widely used mainly due to its simplicity and good mass transfer rate [53]. Mass transfer affects the design of liquid-liquid contactor units utilizing reactive extraction. In order to design the units, it is essential to comprehend and quantitatively determine the effect of mass transfer in the process. Mass transfer from one liquid to another involves solute transfer from the bulk to the interface, interfacial reaction, and transport from the interface to the bulk. The mass transfer in solvent extraction depends, for example, on droplet sizes, velocities, and concentrations [*Publication IV*].

In single droplet experiments, a droplet rises or settles in a column filled with liquid. The droplet is collected from the column outlet and its concentration is analyzed. In conventional single droplet experiments, the conditions inside the droplet during its rise and settling phase are not monitored and the mass transfer leads to concentration changes inside the droplet and at the interface during the rise and settling phases. Due to the mass transfer occurring during these phases, it is valuable to be able to monitor droplet velocities and concentration profiles inside the droplet.

The image-based analysis of droplets in chemical mass transfer experiments allows automated real-time concentration measurement for substances which changes color due to the concentration changes, whereas a common concentration analysis would require collecting samples after or during the experiment. By tracking droplets in the column using computer vision, it is also possible to determine their velocities and accelerations and other features automatically which would be laborious to do manually.

The concentration analysis inside a droplet is based on the observations of image intensity changes inside the droplet. The presence of color changes enables the concentration analysis to be made via imaging. The color changes can be observed from a video recording of moving droplets by a digital camera. Concentration analysis by imaging is based on the Lambert-Beer law. The Lambert-Beer law [100] explains the linear relationship between concentration and absorbance. It can be defined as

$$A = \epsilon l c \tag{4.1}$$

where A is the absorbance, ϵ is the molar absorptivity coefficient, l is the absorption path length and c is the analyte concentration. Moreover, transmittance T is defined as

$$T = I/I_0 \tag{4.2}$$

where I is the light intensity after passing through the sample and I_0 is the initial intensity of light before entering the medium. The relation between A and T is then

$$A = \log_{10} T = \log_{10}(I/I_0). \tag{4.3}$$

Furthermore, the Lambert-Beer law can now be formulated as

$$A = \log_{10}(I/I_0) = \epsilon l c \tag{4.4}$$

and finally, to get the concentration c, the following formula can be used:

$$c = \frac{\log_{10}(I/I_0)}{\epsilon l}. \tag{4.5}$$

These equations provide the background for the concentration analysis method used in this study. More detailed explanations of the equations used are given in *Publication IV*.

4.2 Related Work

Liquid-liquid extraction is a method of separating compounds. It is used in a variety of applications in chemical engineering, analytical chemistry and biology. The process of liquid-liquid extraction can be better observed by minimizing the overall extraction process to an interaction of single droplets. In this way, the factors that affect the process can be reduced.

Chemical process evaluation and concentration analysis by using digital cameras and mobile phones have been conducted, for example, in the following laboratory experiments. The corrosion rate of iron in simulated seawater was determined in [80] and crystal violet concentration was analyzed in [54]. In [51], Gatorade instant powder dry mix energy drink, blue food coloring solution and iron chloride hexahydrate concentrations were analyzed with multiple devices including camera phone and a digital camera. Moreover, copper sulfate concentrations were analyzed in [59] using an additional sample box to prevent stray lighting from interfering with the analysis. However, these experiments were performed manually using solutions in stationary cuvettes instead of automatically analyzing moving droplets.

The relationship between the droplet size and extraction efficiency using microdroplets was determined in [87] with a mass transfer coefficient calculation. Moreover, the authors determined that the reaction time until saturation is directly proportional to the droplet radius and that their mass transfer coefficients agreed with the theoretical results. In [81], a photographic method was used to determine droplet sizes. An imaging setup to determine the volumes, spacing and velocities of droplets was used in [108]. The flow pattern and concentration front inside a droplet has been visualized in [94], but the authors did not determine the concentration of the droplets. The concentration profiles near the phase boundary were measured indirectly using a laser induced fluorescence to track tracer concentrations in [3]. However, the direct quantitative determination of the concentration from droplets using image analysis has not been published before.

4.3 Data

The data collection for the chemical mass transfer experiments was carried out by the research group of computational fluid dynamics at the school of engineering science at LUT. The chemical mass transfer experiments were conducted using a glass column measuring 45 mm×45 mm×375 mm. The droplets formed at the flat tip of a needle at the bottom of the column. The setup was illuminated with a 35W LED panel which measured 300 mm×300 mm with a color temperature of 3000 K. The droplets were collected using a small funnel at the top of the column. Two cameras were positioned to view approximately the areas shown in Figure 4.1. An AVT Oscar was positioned so that it captured the top region of the column, while a Canon Legria was positioned to image the middle part of the column.

An AVT Oscar F-510C FireWire 8-bit camera was used for the size and concentration analysis, and a Canon Legria HF R47 was used for the velocity and acceleration measurements. Two cameras were needed because the AVT Oscar provided the manual settings of the camera parameters, but only produces the rate of 7.5 fps, which was not fast enough for the velocity and acceleration measurements. The Canon Legria on the other hand provided a rate of 50 fps, but it did not provide the manual settings of the camera parameters which were needed for the accurate color analysis.

Both cameras were calibrated using a 5 mm checkerboard pattern. During the experiments, a 5 mm grid was in the view of the Canon Legria providing the scale information. The grid was also placed in the view of the AVT Oscar before the actual experiments to set the scale. During the actual experiments, the grid was removed to allow the better visibility of the droplets. Subsequent RGB frames captured by the AVT Oscar are shown in Figure 4.2a. It can be seen from the figure that the green channel is a bit overexposed and the droplets are barely visible. However, as it is visualized in Figure 4.2b, when the RGB values are modified and the images made gray-scale, the droplets are clearly visible in all of the subsequent frames.

The color versus the concentration calibration was done using droplets with known concentrations. Four different concentrations with varying feed rates and two different needle sizes were used. In total, 61 chemical mass transfer experiments were performed and recorded with both cameras resulting in 122 videos analyzed. The proposed setup was used to collect data for *Publication IV* and *Publication V*.

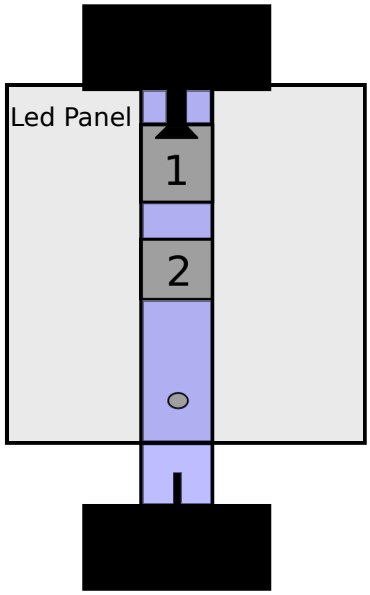
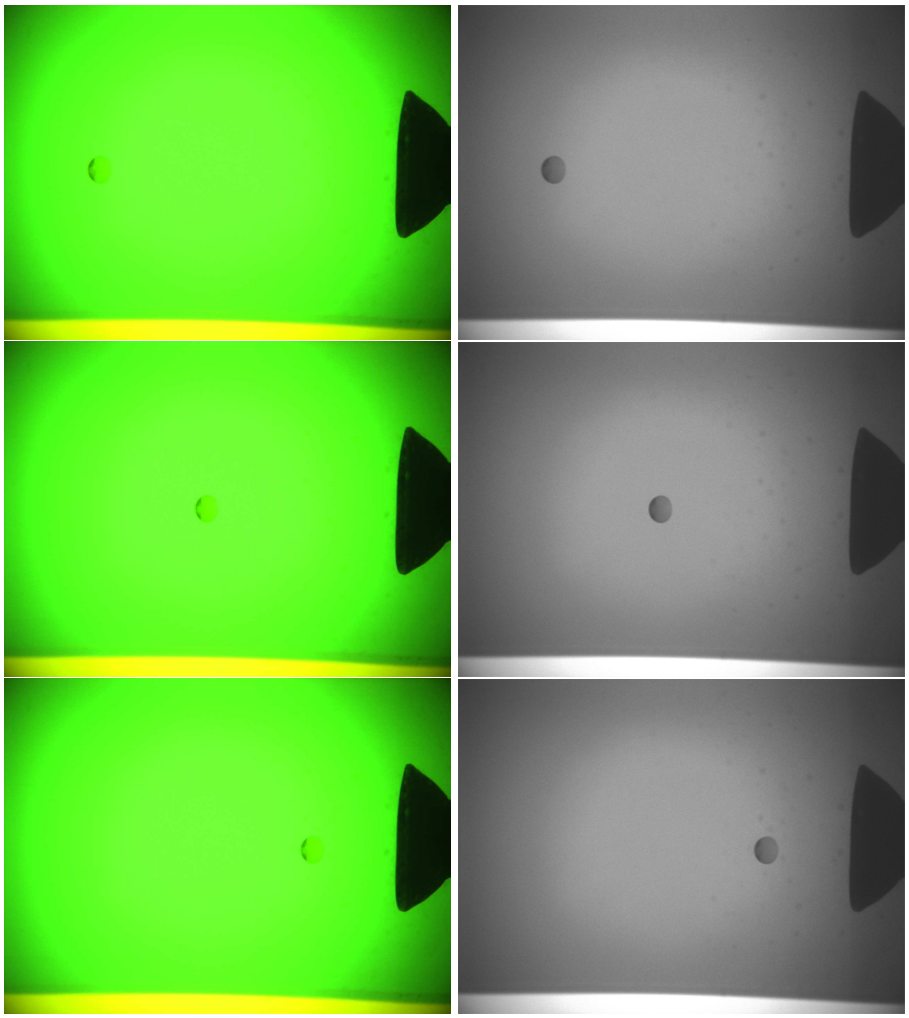


Figure 4.1: The experimental setup for chemical mass transfer experiments. Approximate camera views are marked with rectangles (1. AVT Oscar, 2. Canon Legria).

4.4 Proposed Method

Object trackers for the task of droplet tracking were evaluated with a small set of videos and a selection of trackers. The tested trackers included CT, IVT, LRS, MIL, SRPCA, KCF and RSCM. The trackers were tested with a dataset containing eight droplet videos recorded with the AVT Oscar camera. All the videos contained four frames of a moving droplet. Two of the videos had the most visible droplets used in the experiments and six of the videos had the least visible droplets used in the experiments. These were selected in order to provide the highest available contrast and the lowest available contrast between the droplet and the background. However, it was found that the performance of the trackers was not sufficient. All the tested trackers lost the target after the first frame in all the videos. Consequently, the processing pipeline visualized in Figure 4.3 was adopted. Compared to the general flowchart in Figure 1.1 on page 14, the tracking is done in four steps using tracking by detection. Moreover, the trajectory processing block is only used to calculate the average velocities of droplets after calculating the real-world features.

The resulting images of each step in the pipeline are visualized in Figure 4.4. The pipeline starts with a frame differencing method where the previous frame was subtracted from the current frame. This method provided good initial detection results. It was complemented by using Otsu's method for thresholding [83] and small area removal to remove noise. A morphological closing method was used to fill in the missing parts inside the droplet region. Furthermore, the ellipse fitting method introduced in [101] was used to represent the shape of the droplet. This provided the estimated contours of the droplet in the cases where only half of the droplet or even less was visible in the binary image that was



(a) (b)

Figure 4.2: The example images of one droplet experiment produced by AVT Oscar: (a) subsequent RGB images; (b) corresponding gray-scale images based on modified RGB images.

formed after the frame differencing and the binary operations used. The ellipse fitting method was used based on the assumption that the droplets would assume an oblate spheroid form when moving in the column.

The simplified process flow of the concentration analysis is shown in Figure 4.5. In the figure, the factors that contribute the most to the concentration, absorbance and volume, are visualized. The volume is calculated with the assumption of an oblate spheroid shape using the minor and major axis measurements produced by the ellipse

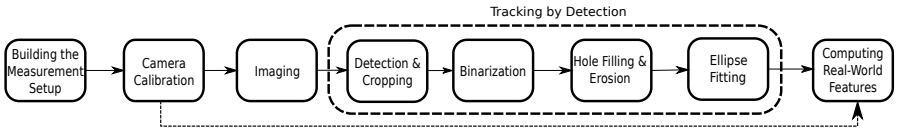


Figure 4.3: Image analysis steps for determining the droplet movement, size, and concentration.

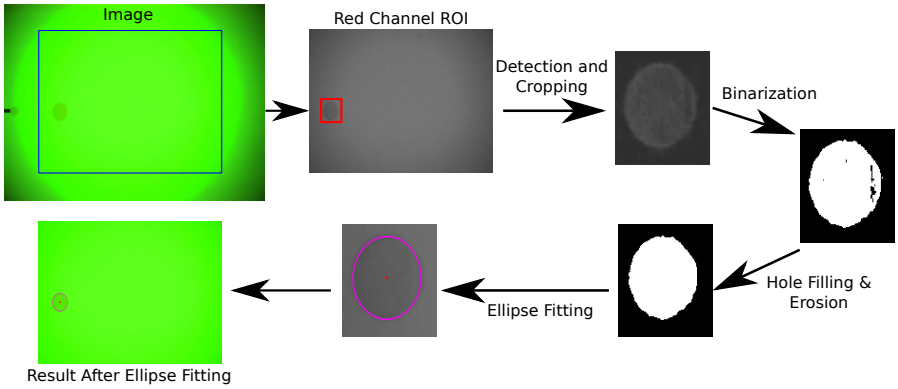


Figure 4.4: Image analysis steps visualized with real images from the process. The contrast of the images in the figure has been enhanced for visualization purposes [*Publication IV*].

fitting procedure. The absorbance A is calculated with Equation 4.3, where I is the light intensity transmitted through the sample and I_0 is the incoming light intensity. The light intensity in the case of an image is the pixel value in the range of 0-255 when the 8-bit camera is used. The concentration analysis is described in more detail in *Publication IV*.

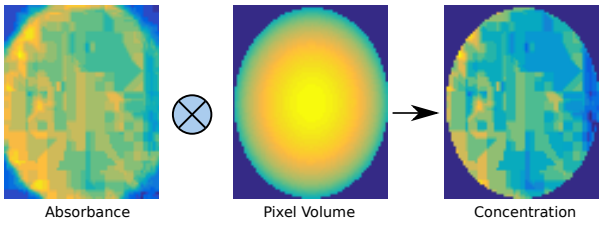


Figure 4.5: The simplified process of the concentration analysis is presented. The figure represents the major contributing factors, absorbance and volume, relevant to the concentration determination [*Publication IV*].

4.5 Results

Based on the comparison with the manual evaluation of the videos used, the processing pipeline managed to detect all the droplets in the chemical mass transfer experiment but not the ones with the smallest concentration, i.e. the droplets with the lowest

contrast. The accuracy of the ellipse fitting method was not directly evaluated, but the minor and major axis values had some fluctuation indicating small errors in the correctness of the fits. However, the accuracy of the ellipse fit was not critical to the concentration calculation because 15% of the outer edge region was cropped out before the concentration analysis. This was done in order to exclude the areas affected mostly by the light scattering.

The results of the concentration analysis were verified against reference samples analyzed using spectrophotometry. Theoretically the smallest detectable concentration change, based on the change of one unit in the red channel value, was approximately 0.15 mmol/L. However, the measured reliability of the method, the width of the measured data distribution at each standard solution concentration, was approximately 2 mmol/L. This translates to approximately 13 unit change in the pixel values.

In *Publication IV*, the velocities calculated for the droplets varied between 112 and 115 mm/s which were within the limits of terminal velocities, 117 and 118 mm/s. The terminal velocity measurements corresponded well with the correlations of contaminated systems. The ratios of the droplet minor axis to the major axis were also measured. The measured values between 0.8 and 0.83 corresponded well with the estimated aspect ratio of 0.83 provided in *Publication IV*. Moreover, this confirms the assumption of the oblate spheroid shape.

In *Publication V*, computational fluid dynamics with a stagnant cap model for a rising droplet simulation was used. The model is based on a concept of two zones, where one is stagnant and the other is circulatory. The obtained experimental average droplet volumes were 26 mm^3 for the 0.8 mm needle and 9.5 mm^3 for the 0.4 mm needle. The corresponding average diameters of the droplets were 3.8 and 2.8 mm. These values obtained from the experiments were slightly larger than the one obtained from the simulation results, which were 22 and 8.3 mm^3 respectively. A computational fluid dynamics model of a non-deforming rising droplet with a rigid interface was used to fit an interfacial reaction kinetic constant. However, the fitted value was much lower than the experimentally determined one. Nevertheless, the mass transfer coefficients calculated from the computational fluid dynamics model and the estimated coefficients based on literature correlations agreed well.

4.6 Discussion

The proposed video-based analysis method was developed for monitoring of single droplets and their reactions in a glass column. The method is based on the observed change of color in the droplets when they react inside the column. The accuracy of the method was evaluated with reference samples which were analyzed using spectrophotometry. The method can be used to measure the concentration of a droplet which is inside the column. This enables direct monitoring of the reactions of individual droplets. The smallest detectable concentration change corresponding to the pixel values of the 8-bit imaging used, was approximately 0.15 mmol/L. The reliability of the method was approximately 2 mmol/L. The results provided promising results for reactions where color changes take place during a reaction. Based on the results, the reactions would be quantifiable using the proposed method. Moreover, the method could be extended to analyze multiple droplets at once.

This method is not specific to the selected setup, but it can be applied to other reactions and different setups where a detectable color change is present. Moreover, the spectral responsiveness of the used sensor can be changed by using filters. Only the red channel of the camera was used in this work, but the method is not limited to a specific color channel. It is possible to use different color channels in the detection of other reactions. Moreover, it is possible to change the illumination. Furthermore, the method is not limited to the visible range, but to the range of the camera optics and sensors available. However, the fluorescence or phosphorescence of the sample changes in the refractive index at high analyte concentrations and light scattering in the sample can cause nonlinearity in the calculations.

The reliability and accuracy of the method could be improved with better hardware. One of these improvements could be, for example, using 10-bit or better imaging equipment. Better sensors, 10- to 16-bit, and specific spectral sensitivity ranges in the imaging sensor would benefit the color-based chemical reaction analysis with the help of properly selected illumination. Moreover, modeling the light refraction and scattering at the edge regions of the droplets would allow the inclusion of the edge region in the concentration analysis. One possibility would be to have an additional camera at a different angle to determine the light scattering and to calculate the optical paths of the edge region. Moreover, a multi-camera setup would allow 3D reconstruction of the droplets. A high-speed camera setup could be used in the concentration analysis in order to get better understanding about the vortices formed inside the droplet. Moreover, formations, coalescence, and breaking of droplets could be analyzed in depth with the high-speed video material. Furthermore, different light sources could be considered for determining the effects of light scattering at the outer edge region of the droplet. Finally, the spectral sensitivities of the camera sensors should be considered, and possibly filters could be added to filter out specific ranges depending on the use case and optical characteristics of the materials used. Moreover, the spectral sensitivities of the sensor and optics define what kind of reactions can be observed. Lastly, the illumination in specific bands of the electromagnetic spectrum could also be used to further improve the system.

Chapter V

Conclusion

In this work, a multi-camera framework for tracking finger movements was proposed for studying HCI. Moreover, a video analysis based framework to analyze droplets in a chemical mass transfer experiment was proposed for chemical reaction studies where suitable color changes are present.

Considering the general framework for moving object analysis introduced in Figure 1.1 on page 14, it can be seen that most of the steps are common for both applications. The experiment setup needed to be designed and built, and camera calibration for appropriate imaging was needed in both applications. The initialization phase used partially approaches of the same kind in both applications. Mainly, this phase was performed by using background subtraction which was used to initialize the tracking of the normal speed videos in the case of 3D touch-screen experiment, and to initialize and to follow the moving droplets in the chemical mass transfer experiment. However, manual initialization was used for the high-speed videos in the case of 3D touch-screen experiment. The tracking was performed using specialized tracking algorithms in the case of 3D touch-screen experiment, but a tracking by detection approach was used for the droplet tracking in the chemical mass transfer experiment. Trajectory processing was carried out in both cases. The trajectories were smoothed in the 3D touch-screen experiment before calculating the real-world features. However, in the chemical mass transfer experiment, the trajectories itself were not filtered or smoothed because it was only necessary to determine the average values of velocities. Computing the real-world features included forming 3D reconstruction in the 3D touch-screen experiment and determining the sizes and the average velocities in the chemical mass transfer experiment. Furthermore, an analysis part could be added to both cases as the gathered real-world features were analyzed in both cases.

The multi-camera framework was evaluated in the application of a 3D touch-screen experiment. In order to select the best tracking methods for the task, a comprehensive evaluation was performed. Even though the tracking rate of 100% was achieved with the KCF tracker for the limited dataset of eleven high-speed videos, the KCF did not manage to track all of the high-speed videos correctly. Moreover, none of the evalu-

ated tracking methods managed to achieve a 100% tracking rate in the evaluation of the dataset of 17 normal-speed videos. Trajectory post-processing and 3D trajectory reconstruction methods were proposed. The trajectory data post-processing enabled more appropriate results for the calculations of the derivatives of the target location, velocity and acceleration. The proposed framework evaluation produced promising results on the applicability of general object trackers for the task of finger tracking in the application of 3D touch-screen experiment. Moreover, the trajectory post-processing methods used provided a way to extract appropriate features that could be used to explain some of the differences in the hand movements toward targets at different disparities.

A droplet analysis framework was evaluated in the application of a concentration analysis of droplets in a chemical mass transfer experiment. Various tracking methods were evaluated, but none of the methods produced good enough results, and a simple background subtraction method was selected to be used. The background subtraction method provided good results for detecting the droplets. The concentration analysis of the droplets is based on the color changes of the droplets. In the experiments, the concentration analysis results were verified against reference samples analyzed with spectrophotometry. Moreover, the estimated droplet velocities were in a good agreement with the measured values.

The 3D touch-screen experiment and concentration analysis of droplets in a chemical mass transfer experiment provided different challenges for movement tracking and analysis. However, the achieved results were insightful and with a few future improvements, such as using better equipment, and more careful experiment planning and execution they could be made to form reliable and accurate systems for the purposes of both application fields.

Both application areas provided different challenges and different approaches were needed in order to obtain useful results for analysis. In the 3D touch-screen experiment, the tracking approach worked relatively well, but in the moving droplets analysis in the chemical mass transfer experiment the tracking methods were not able to track the droplets. However, a simple background subtraction method, frame differencing, provided good results. The failure detection systems implemented for this research provided enough information for the purposes of the applications.

The systems introduced in this work could be extended for other HCI research areas where hand movements are the area of interest and other reaction analysis experiments where the color of reagent changes during a reaction.

[1] Babenko, B., Yang, M.-H., and Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 8 (2011), 1619–1632.

[2] Ballard, D. H. Generalizing the Hough transform to detect arbitrary shapes. In *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann Publishers Inc., 1987, pp. 714–725.

[3] Baumann, K.-H., and Mühlfriedel, K. Mass transfer studies with laser-induced fluorescence across liquid/liquid phase boundaries. *Chemical Engineering & Technology 25*, 7 (2002), 697–700.

[4] Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. H. S. Staple: Complementary learners for real-time tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1401–1409.

[5] Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. S. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)* (2016), pp. 850–865.

[6] Blaschko, M. B., and Lampert, C. H. Learning to localize objects with structured output regression. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2008), pp. 2–15.

[7] Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. Visual object tracking using adaptive correlation filters. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 2544–2550.

[8] Briechle, K., and Hanebeck, U. Template matching using fast normalized cross correlation. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)* (2001), pp. 95–102.

[9] Brutzer, S., Höferlin, B., and Heidemann, G. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 1937–1944.

[10] Böttger, T., Ulrich, M., and Steger, C. Subpixel-precise tracking of rigid objects in real-time. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)* (2017), pp. 54–65.

[11] Čehovin, L., Kristan, M., and Leonardis, A. Is my new tracker really better than yours? In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)* (2014), pp. 540–547.

[12] Čehovin, L., Leonardis, A., and Kristan, M. Robust visual tracking using template anchors. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)* (2016), pp. 1–8.

[13] Chambolle, A. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision 20*, 1-2 (2004), 89–97.

[14] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference (BMVC)* (2014).

[15] Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., and Young Choi, J. Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4321–4330.

[16] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association 74*, 368 (1979), 829–836.

[17] Cleveland, W. S., and Devlin, S. J. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association 83*, 403 (1988), 596–610.

[18] Comaniciu, D., Ramesh, V., and Meer, P. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2000), vol. 2, pp. 142–149.

[19] Condat, L. A direct algorithm for 1-D total variation denoising. *IEEE Signal Processing Letters 20*, 11 (2013), 1054–1057.

[20] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), vol. 1, pp. 886–893.

[21] Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. Eco efficient convolution operators for tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 6931–6939.

[22] Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 8 (2017), 1561–1575.

[23] Danelljan, M., Häger, G., Shahbaz Khan, F., and Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2015), pp. 4310–4318.

[24] De Ath, G., and Everson, R. Visual object tracking: The initialisation problem. In *Proceedings of the Conference on Computer and Robot Vision (CRV)* (2018), p. in press.

[25] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 248–255.

[26] Du, D., Qi, H., Wen, L., Tian, Q., Huang, Q., and Lyu, S. Geometric hypergraph learning for visual tracking. *IEEE transactions on cybernetics 47*, 12 (2017), 4182–4195.

[27] Elliott, D., Hansen, S., Grierson, L. E. M., Lyons, J., Bennett, S. J., and Hayes, S. J. Goal-directed aiming: two components but multiple processes. *Psychological Bulletin 136*, 6 (2010), 1023–44.

[28] Erdem, C. E., Tekalp, A. M., and Sankur, B. Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In *Proceedings of the International Conference on Image Processing (ICIP)* (2001), pp. 69–72.

[29] Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. Vision-based hand pose estimation: a review. *Computer Vision and Image Understanding* (2007), 52 – 73. Special Issue on Vision for Human-Computer Interaction.

[30] Fang, Y., Kang, W., Wu, Q., and Tang, L. A novel video-based system for in-air signature verification. *Computers & Electrical Engineering* (2017), 1 – 14.

[31] FFmpeg. `https://ffmpeg.org/`, 2018. [Online; accessed 01-Jun-2018].

[32] Fischler, M. A., and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[33] Forsyth, D. A., and Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[34] Galoogahi, H. K., Fagg, A., and Lucey, S. Learning background-aware correlation filters for visual tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2017), pp. 1144–1152.

[35] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 580–587.

[36] Godec, M., Roth, P. M., and Bischof, H. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding 117*, 10 (2012), 1245–1256.

[37] Hare, S., Saffari, A., and Torr, P. H. S. Struck: Structured output tracking with kernels. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2011), pp. 263–270.

[38] Hariyono, J., Hoang, V.-D., and Jo, K.-H. Tracking failure detection using time reverse distance error for human tracking. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2015), pp. 611–620.

[39] Hartley, R. I., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[40] Heikkilä, J., and Silvén, O. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (1997), pp. 1106–1112.

[41] Heikkilä, J., and Silvén, O. A real-time system for monitoring of cyclists and pedestrians. In *Proceedings of the Workshop on Visual Surveillance (VS)* (1999), pp. 74–81.

[42] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2012), pp. 702–715.

[43] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 3 (2015), 583–596.

[44] Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S., and Zhang, Z. Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 12 (2012), 2420–2440.

[45] Jacoby, W. G. Loess:: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies 19*, 4 (2000), 577–613.

[46] Joshi, K. A., and Thakore, D. G. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering* (2012), 44–48.

[47] Julier, S. J., and Uhlmann, J. K. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the International Society for Optics and Photonics (SPIE)* (1997), pp. 182–194.

[48] Kalal, Z., Mikolajczyk, K., and Matas, J. Forward-backward error: automatic detection of tracking failures. In *Proceedings of the International Conference on Pattern Recognition (ICPR)* (2010), pp. 2756–2759.

[49] Kalal, Z., Mikolajczyk, K., and Matas, J. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 7 (2012), 1409–1422.

[50] Kato, H., and Billinghurst, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the International Workshop on Augmented Reality (IWAR)* (1999), pp. 85–94.

[51] Kehoe, E., and Penn, R. L. Introducing colorimetric analysis with camera phones and digital cameras: An activity for high school or general chemistry. *Journal of Chemical Education 90*, 9 (2013), 1191–1195.

[52] Khoshelham, K., and Elberink, S. O. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors* (2012), 1437–1454.

[53] Kislik, V. S. Chapter 13 - advances in development of solvents for liquid-liquid extraction. In *Solvent Extraction*. Elsevier, 2012, pp. 451–481.

[54] Knutson, T. R., Knutson, C. M., Mozzetti, A. R., Campos, A. R., Haynes, C. L., and Penn, R. L. A fresh look at the crystal violet lab with handheld camera colorimetry. *Journal of Chemical Education 92*, 10 (2015), 1692–1695.

[55] Kooi, F. L., and Toet, A. Visual comfort of binocular and 3D displays. *Displays 25*, 2 (2004), 99–108.

[56] Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojíř, T., Häger, G., Lukežič, A., Fernández, G., Gupta, A., Petrosino, A., Memarmoghadam, A., Garcia-Martin, A., Solís Montero, A., Vedaldi, A., Robinson, A., Ma, A. J., Varfolomieiev, A., Alatan, A., Erdem, A., Ghanem, B., Liu, B., Han, B., Martinez, B., Chang, C.-M., Xu, C., Sun, C., Kim, D., Chen, D., Du, D., Mishra, D., Yeung, D.-Y., Gundogdu, E., Erdem, E., Khan, F., Porikli, F., Zhao, F., Bunyak, F., Battistone, F., Zhu, G., Roffo, G., Subrahmanyam, G. R. K. S., Bastos, G., Seetharaman, G., Medeiros, H., Li, H., Qi, H., Bischof, H., Possegger, H., Lu, H., Lee, H., Nam, H., Chang, H. J., Drummond, I., Valmadre, J., Jeong, J.-c., Cho, J.-i., Lee, J.-Y., Zhu, J., Feng, J., Gao, J., Choi, J. Y., Xiao, J., Kim, J.-W., Jeong, J., Henriques, J. F., Lang, J., Choi, J., Martinez, J. M., Xing, J., Gao, J., Palaniappan, K., Lebeda, K., Gao, K., Mikolajczyk, K., Qin, L., Wang, L., Wen, L., Bertinetto, L., Rapuru, M. K., Poostchi, M., Maresca, M., Danelljan, M., Mueller, M., Zhang, M., Arens, M., Valstar, M., Tang, M., Baek, M., Khan, M. H., Wang, N., Fan, N., Al-Shakarji, N., Miksik, O., Akin, O., Moallem, P., Senna, P., Torr, P. H. S., Yuen, P. C., Huang, Q., Martin-Nieto, R., Pelapur, R., Bowden, R., Laganière, R., Stolkin, R., Walsh, R., Krah, S. B., Li, S., Zhang, S., Yao, S., Hadfield, S., Melzi, S., Lyu, S., Li, S., Becker, S., Golodetz, S., Kakanuru, S., Choi, S., Hu, T., Mauthner, T., Zhang, T., Pridmore, T., Santopietro, V., Hu, W., Li, W., Hübner, W., Lan, X., Wang, X., Li, X., Li, Y., Demiris, Y., Wang, Y., Qi, Y., Yuan, Z., Cai, Z., Xu, Z., He, Z., and Chi, Z. The visual object tracking VOT2016 challenge results. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)* (2016), pp. 777–823.

[57] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L., Nebehay, G., Fernández, G., Vojíř, T., Gatt, A., khajenezhad, A., Salahledin, A., Soltani-Farani, A., Zarezade, A., Petrosino, A., Milton, A., Bozorgtabar, B., Li, B., Chan, C. S., Heng, C.,

Ward, D., Kearney, D., Monekosso, D., Karaimer, H., Rabiee, H., Zhu, J., Gao, J., Xiao, J., Zhang, J., Xing, J., Huang, K., Lebeda, K., Cao, L., Maresca, M., Lim, M. K., El Helw, M., Felsberg, M., Remagnino, P., Bowden, R., Goecke, R., Stolkin, R., Lim, S., Maher, S., Poullot, S., Wong, S., Satoh, S., Chen, W., Hu, W., Zhang, X., Li, Y., and Niu, Z. The visual object tracking VOT2013 challenge results. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)* (2013), pp. 98–111.

[58] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L., Nebehay, G., Fernández, G., Vojíř, T., Gatt, A., khajenezhad, A., Salahledin, A., Soltani-Farani, A., Zarezade, A., Petrosino, A., Milton, A., Bozorgtabar, B., Li, B., Chan, C. S., Heng, C., Ward, D., Kearney, D., Monekosso, D., Karaimer, H., Rabiee, H., Zhu, J., Gao, J., Xiao, J., Zhang, J., Xing, J., Huang, K., Lebeda, K., Cao, L., Maresca, M., Lim, M. K., El Helw, M., Felsberg, M., Remagnino, P., Bowden, R., Goecke, R., Stolkin, R., Lim, S., Maher, S., Poullot, S., Wong, S., Satoh, S., Chen, W., Hu, W., Zhang, X., Li, Y., and Niu, Z. The visual object tracking VOT2014 challenge results. In *Proceedings of the European Conferance on Computer Vision Workshops (ECCVW)* (2015), pp. 191–217.

[59] Kuntzleman, T. S., and Jacobson, E. C. Teaching beer's law and absorption spectrophotometry with a smart phone: A substantially simplified protocol. *Journal of Chemical Education 93*, 7 (2016), 1249–1252.

[60] Kwon, J., and Lee, K. M. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 1208–1215.

[61] Lasserre, J. A., Bishop, C. M., and Minka, T. P. Principled hybrids of generative and discriminative models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2006), pp. 87–94.

[62] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551.

[63] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the International Conference on Machine Learning (ICML)* (2009), pp. 609–616.

[64] Levey, A., and Lindenbaum, M. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Transactions on Image Processing 9*, 8 (2000), 1371–1374.

[65] Li, F., Tian, C., Zuo, W., Zhang, L., and Yang, M.-H. Learning spatial-temporal regularized correlation filters for visual tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), p. in.

[66] Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., and Hengel, A. V. D. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2013), 58:1–58:48.

[67] Li, Y., and Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)* (2015), pp. 254–265.

[68] Li, Y., Zhu, J., Song, W., Wang, Z., Liu, H., and Hoi, S. C. Robust estimation of similarity transformation for visual object tracking with correlation filters. *arXiv preprint arXiv:1712.05231* (2017).

[69] Lienhart, R., and Maydt, J. An extended set of Haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing (ICIP)* (2002), vol. 1, pp. 900–903.

[70] Liu, Q., Zhao, X., and Hou, Z. Survey of single-target visual tracking methods based on online learning. *Institution of Engineering and Technology (IET) Computer Vision 8*, 5 (2014), 419–428.

[71] Matlab Documentation Filtering and Smoothing Data. `http://www.mathworks.se/help/curvefit/smoothing-data.html`. [Online; accessed 01-Jun-2018].

[72] Lucas, B. D., and Kanade, T. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence* (1981), IJCAI'81, pp. 674–679.

[73] Lukežič, A., Vojíř, T., Čehovin, L., Matas, J., and Kristan, M. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision 126*, 7 (2018), 671–688.

[74] Luo, W. Matlab implementation for multiple instance learning (MIL) tracker. `https://sites.google.com/site/whluoimperial/code-software`, 2018. [Online; accessed 04-Jul-2018].

[75] Maggio, E., and Cavallaro, A. *Video Tracking: Theory and Practice*. Wiley Publishing, 2011.

[76] Merwe, R. V. D., and Wan, E. A. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)* (2000), pp. 153–158.

[77] Mine, M. R., van Baar, J., Grundhofer, A., Rose, D., and Yang, B. Projection-based augmented reality in Disney theme parks. *Computer* (2012), 32–40.

[78] Montemerlo, M., and Thrun, S. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (2003), pp. 1985–1991.

[79] Montero, A. S., Lang, J., and Laganiere, R. Scalable kernel correlation filter with sparse feature integration. In *Proceedings of the International Conference on Computer Vision Workshop (ICCVW)* (2015), IEEE, pp. 587–594.

[80] Moraes, E. P., Confessor, M. R., and Gasparotto, L. H. S. Integrating mobile phones into science teaching to help students develop a procedure to evaluate the corrosion rate of iron in simulated seawater. *Journal of Chemical Education 92*, 10 (2015), 1696–1699.

[81] Oliveira, N., Silva, D. M., Gondim, M., and Mansur, M. B. A study of the drop size distributions and hold-up in short Kühni columns. *Brazilian Journal of Chemical Engineering 25*, 4 (2008), 729–741.

[82] Orfanidis, S. J. *Introduction to Signal Processing.* Prentice Hall, 1996.

[83] Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* (1979), 62–66.

[84] Paragios, N., and Deriche, R. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 3 (2000), 266–280.

[85] Pece, F., and Kautz, J. Bitmap movement detection: HDR for dynamic scenes. In *Proceedings of the Conference on Visual Media Production (CVMP)* (2010), pp. 1–8.

[86] Possegger, H., Mauthner, T., and Bischof, H. In defense of color-based model-free tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 2113–2120.

[87] Q Yu, J., K Chin, L., Chen, Y., J Zhang, G., Q Lo, G., Ayi, T., H Yap, P., L Kwong, D., and Liu, A.-q. Microfluidic droplet-based liquid-liquid extraction for fluorescence-indicated mass transfer. In *Proceedings of the Conference on Micro Total Analysis Systems (μTAS)* (2010), pp. 1079–1081.

[88] Redmon, J., and Farhadi, A. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[89] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017), 1137–1149.

[90] Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision 77*, 1-3 (2008), 125–141.

[91] Rousseeuw, P. J., and Leroy, A. M. *Robust regression and outlier detection.* John Wiley & Sons, Inc., 1987.

[92] Savitzky, A., and Golay, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry 36*, 8 (1964), 1627–1639.

[93] Schafer, R. W. What is a Savitzky-Golay filter? Lecture notes. *IEEE Signal Processing Magazine 28*, 4 (2011), 111–117.

[94] Schulze, K. *Stoffaustausch und Fluiddynamik am bewegten Einzeltropfen unter dem Einfluss von Marangonikonvektion.* PhD thesis, Technische Universität Berlin, 2007.

[95] Selesnick, I. W., and Bayram, I. Total variation filtering. *Lecture Notes* (2010).

[96] Servos, P., Goodale, M. A., and Jakobson, L. S. The role of binocular vision in prehension: a kinematic analysis. *Vision research 32*, 8 (1992), 1513–1521.

[97] Shi, J., and Tomasi, C. Good features to track. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (1994), pp. 593–600.

[98] Smeulder, A., Chu, D., Cucchiara, R., Calderara, S., Deghan, A., and Shah, M. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 7 (2014), 1442–1468.

[99] Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, 1997.

[100] Swinehart, D. The Beer-Lambert law. *Journal of chemical education 39*, 7 (1962), 333.

[101] Szpak, Z. L., Chojnacki, W., and van den Hengel, A. Guaranteed ellipse fitting with the Sampson distance. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2012), pp. 87–100.

[102] Tao, R., Gavves, E., and Smeulders, A. W. M. Siamese instance search for tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1420–1429.

[103] Terajima, K., Komuro, T., and Ishikawa, M. Fast finger tracking system for in-air typing interface. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)* (2009), pp. 3739–3744.

[104] Torr, P. H., and Zisserman, A. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78*, 1 (2000), 138–156.

[105] Valkov, D., Giesler, A., and Hinrichs, K. Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In *Proceedings of the International Conference on Interactive Tabletops and Surfaces (ITS)* (2012), pp. 21–30.

[106] van Beurden, M. H., Van Hoey, G., Hatzakis, H., and Ijsselsteijn, W. A. Stereoscopic displays in medical domains: a review of perception and performance effects. In *Proceedings of the International Society for Optics and Photonics (SPIE)* (2009), pp. 72400A–1.

[107] van de Weijer, J., Schmid, C., Verbeek, J., and Larlus, D. Learning color names for real-world applications. *IEEE Transactions on Image Processing 18*, 7 (2009), 1512–1523.

[108] Vansteene, A., Jasmin, J.-P., Al Amhad, L., Cavadias, S., Cote, G., and Mariet, C. Liquid-liquid extraction in microsystems: establish segmented flows to optimize mass transfer. In *Proceedings of the Le Congrés Francais de Mécanique (CFM)* (2017).

[109] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2001), pp. 511–518.

[110] Vojíř, T. Tracking with kernelized correlation filters. `https://github.com/vojirt/kcf/`, 2018. [Online; accessed 01-May-2018].

[111] Vojíř, T., Noskova, J., and Matas, J. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters 49* (2014), 250–258.

[112] Vostatek, P., Claridge, E., Uusitalo, H., Hauta-Kasari, M., Fält, P., and Lensu, L. Performance comparison of publicly available retinal blood vessel segmentation methods. *Computerized Medical Imaging and Graphics 55* (2017), 2 – 12. Special Issue on Ophthalmic Medical Image Analysis.

[113] Wang, D., Lu, H., and Yang, M.-H. Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing 22*, 1 (2013), 314–325.

[114] Wei, Z., Huchuan, L., and Ming-Hsuan, Y. Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing 23*, 5 (2014), 2356–2368.

[115] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. Analysis of the accuracy and robustness of the leap motion controller. *Sensors 13*, 5 (2013), 6380–6393.

[116] Welch, G., and Bishop, G. An introduction to the Kalman filter. Technical report, Department of Computer Science, University of North Carolina, 1995.

[117] Welch, G., and Bishop, G. An introduction to the Kalman filter: Siggraph 2001 course 8. In *Proceedings of the Conference on Computer Graphics & Interactive Techniques* (2001), pp. 12–17.

[118] Wu, H., Chellappa, R., Sankaranarayanan, A. C., and Zhou, S. K. Robust visual tracking using the time-reversibility constraint. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2007), pp. 1–8.

[119] Wu, Y., Lim, J., and Yang, M.-H. Online object tracking: a benchmark. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 2411–2418.

[120] Xiao, J., Stolkin, R., and Leonardis, A. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4978–4987.

[121] Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Computing Surveys 38*, 4 (2006).

[122] Zhang, K., Zhang, L., Liu, Q., Zhang, D., and Yang, M.-H. Fast visual tracking via dense spatio-temporal context learning. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2014), pp. 127–141.

[123] Zhang, K., Zhang, L., and Yang, M.-H. Real-time compressive tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2012), pp. 864–877.

[124] Zhang, K., Zhang, L., and Yang, M.-H. Fast compressive tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 10 (2014), 2002–2015.

[125] Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the International Conference on Computer Vision (ICCV)* (1999), pp. 666–673.

Publication I

Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J.,
Kälviäinen, H.

High-speed hand tracking for studying human-computer interaction

Reprinted by permission from Springer Nature:
Lecture Notes in Computer Science,
Proceedings of Scandinavian Conference on Image Analysis (SCIA)
Vol. 9127, pp. 130–141, 2015.

High-Speed Hand Tracking for Studying Human-Computer Interaction

Toni Kuronen[1(✉)], Tuomas Eerola[1], Lasse Lensu[1], Jari Takatalo[2], Jukka Häkkinen[2], and Heikki Kälviäinen[1]

[1] Machine Vision and Pattern Recognition Laboratory (MVPR), School of Engineering Science, Lappeenranta University of Technology (LUT), P.O. Box 20, FI-53851 Lappeenranta, Finland
{toni.kuronen,tuomas.eerola,lasse.lensu,heikki.kalviainen}@lut.fi
http://www2.it.lut.fi/mvpr/

[2] Visual Cognition Research Group, Institute of Behavioural Sciences, University of Helsinki, P.O. Box 9, FI-00014 Helsinki, Finland
{jari.takatalo,jukka.hakkinen}@helsinki.fi
http://www.helsinki.fi/psychology/groups/visualcognition/

Abstract. Understanding how a human behaves while performing human-computer interaction tasks is essential in order to develop better user interfaces. In the case of touch and gesture based interfaces, the main interest is in the characterization of hand movements. The recent developments in imaging technology and computing hardware have made it attractive to exploit high-speed imaging for tracking the hand more accurately both in space and time. However, the tracking algorithm development has been focused on optimizing the robustness and computation speed instead of spatial accuracy, making most of them, as such, insufficient for the accurate measurements of hand movements. In this paper, state-of-the-art tracking algorithms are compared based on their suitability for the finger tracking during human-computer interaction task. Furthermore, various trajectory filtering techniques are evaluated to improve the accuracy and to obtain appropriate hand movement measurements. The experimental results showed that Kernelized Correlation Filters and Spatio-Temporal Context Learning tracking were the best tracking methods obtaining reasonable accuracy and high processing speed while Local Regression filtering and Unscented Kalman Smoother were the most suitable filtering techniques.

Keywords: Hand tracking · High-speed video · Hand trajectories · Filtering · Human-computer interaction

1 Introduction

The motivation for this work comes from the human-computer interaction (HCI) research, and the need to accurately record hand and finger movements of test subjects in various HCI tasks. During the recent years, this has become particularly important due to the rapid development of touch display technology

R.R. Paulsen and K.S. Pedersen (Eds.): SCIA 2015, LNCS 9127, pp. 130–141, 2015.
DOI: 10.1007/978-3-319-19665-7_11

and amount of commercially available touchscreens in smartphones, tablets and other table-top and hand-held devices, as well as, the emergence of different gesture based interfaces. Recording the hand movements can be performed by using hand tracking or general object tracking which has been studied since the 1990s and is an active research area also today [4], [13], [15], [24], [25]. Despite the significant effort, however, the problem of hand tracking cannot be considered solved [9]. From a technical perspective, different robust approaches for hand tracking exist, such as data gloves with electro-mechanical or magnetic sensors that can measure the hand and finger location with high accuracy. However, such devices affect the natural hand motion, are expensive, and hence, cannot be considered a good solution when pursuing natural HCI. As a consequence, there is a need for image-based solutions that provide unobtrusive way to study and track human movement and enable natural interaction between technology.

To accurately record fast phenomena such as reaction times and to robustly track rapid hand movements, high frame rates are needed in imaging. To produce videos with good quality, the high-speed imaging requires more light when compared to imaging with conventional frame rates. Therefore, gray-scale high-speed imaging is in common use making the use of hand tracking methods relying specifically on color information unsuitable. This motivates to apply general object trackers for the problem. In [9], various general object trackers were compared for hand tracking with a primary focus on gray-scale high-speed videos. It was found out that by avoiding the most difficult environments and posture changes, the state-of-the-art trackers are capable of reliable hand and finger tracking.

The main problem in using the existing object tracking methods in accurate measurement of hand and finger movements is that they are developed for applications where high (sub-pixel) accuracy is unnecessary. Instead, the research has focused on developing more computationally efficient and robust methods, i.e., losing the target is considered a much more severe problem than a spatial shift of the tracking window. While these are justified choices in most tracking applications, this is not the case in the hand trajectory measurement in high speed videos where small hand movement between the frames and a controlled environment help to maintain higher robustness, but high accuracy is needed. Even small errors in spatial locations can cause high errors when computing the speed and acceleration. Therefore, the existing tracking algorithms are as such insufficient for the accurate measurements of hand movements and further processing of hand trajectories is required.

In this paper, the work started in [9] is continued by further evaluating an extended set of tracking algorithms to find the best methods for accurate hand movement measurements. Moreover, the earlier work is extended by processing tracked hand trajectories with various filtering techniques. The different methods are evaluated using novel annotated data consisting of high-speed gray-scale videos of a human performing HCI tasks using a touch user interface.

Since the trackers specific for hand tracking rely on color information, the focus of this study is on the state-of-the-art general object trackers. Based on

a literature review and preliminary tracking tests, 12 trackers were selected for further study [14]. These methods are summarized in Table 1.

Table 1. Trackers selected for the experiments

Method	Abbreviation	Implementation
Real-time Compressive Tracking [27]	CT	MATLAB+MEX[1]
Fast Compressive Tracking [28]	FCT	MATLAB+MEX[2]
High-Speed Tracking with Kernelized Correlation Filters [8]	KCF	MATLAB+MEX[3]
Hough-based Tracking of Non-Rigid Objects [5]	HT	C++[4]
Incremental Learning for Robust Visual Tracking [19]	IVT	MATLAB+MEX[5]
Robust Object Tracking with Online Multiple Instance Learning [1]	MIL	MATLAB[6]
Tracking Learning Detection [12]	TLD	MATLAB+MEX[7]
Robust Object Tracking via Sparsity-based Collaborative Model [29]	RSCM	MATLAB+MEX[8]
Fast Tracking via Spatio-Temporal Context Learning [26]	STC	MATLAB[9]
Structured Output Tracking with Kernels [6]	struck	C++[10]
Single and Multiple Object Tracking Using Log-Euclidean Riemannian Subspace and Block-Division Appearance Model [10]	LRS	MATLAB+MEX[11]
Online Object Tracking with Sparse Prototypes [21]	SRPCA	MATLAB[12]

Real-time Compressive Tracking (CT) [27] is a tracking-by-detection method that uses a sparse random matrix to project high-dimensional image features to low-dimensional (compressed) features. The basic idea is to acquire positive samples near the current target location and negative samples far away from the target object at each frame, and use these samples to update the classifier. Then, the location for the next frame is predicted by getting samples from around the last known location and choosing the sample that gets the best classification

[1] http://www4.comp.polyu.edu.hk/~cslzhang/CT/CT.htm
[2] http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm
[3] http://www.isr.uc.pt/~henriques/circulant/
[4] http://lrs.icg.tugraz.at/research/houghtrack/
[5] http://www.cs.toronto.edu/~dross/ivt/
[6] http://whluo.net/matlab-code-for-mil-tracker/
[7] http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html
[8] https://github.com/gnebehay/SCM
[9] http://www4.comp.polyu.edu.hk/~cslzhang/STC/STC.htm
[10] http://www.samhare.net/research/struck/code
[11] http://www.iis.ee.ic.ac.uk/~whluo/code.html
[12] http://faculty.ucmerced.edu/mhyang/project/tip13_prototype/TIP12-SP.htm

score. Fast Compressive Tracker (FCT) [28] is an improvement of CT. The speed of the tracker is improved by using a sparse-to-dense search method. First, the object search is done by using a sparse sliding window followed by detection using a dense sliding window for better accuracy.

HoughTrack (HT)[5] is a tracking-by-detection method which is based on the generalized Hough transform. In the method, a Hough-based detector is constantly trained with the current object appearance. Unlike the other selected algorithms, in addition to bounding box tracking, HT outputs also segmented tracking results which is used to limit the amount of background noise supplied to the online learning module.

Incremental learning for robust visual tracking (IVT) [19] learns a low-dimensional subspace representation of the target object and tracks it using a particle filter. Online object tracking with sparse prototypes (SRPCA)[21] is a particle filter based tracking method that utilizes sparse prototypes consisting of PCA basis vectors modeling the object appearance. The main difference to IVT is trivial templates that are applied to handle partial occlusions.

High-Speed Tracking with Kernelized Correlation Filters (KCF) [8] is an improved version of the kernelized correlation filters introduced in [7]. By over-sampling sliding windows, the resulting data matrix can be simplified, the size of the data reduced, and the computation made faster. This can be achieved by taking advantage of Fast Fourier Transform (FFT).

Tracking with online multiple instance learning (MIL) [1] is a tracking-by-detection method that applies the multiple instance learning approach to tracking to account ambiguities in the training data. In the multiple instance learning, positive and negative training examples are presented as sets, and labels are provided for the sets instead of individual instances. By using this approach, updates of the classifier with incorrectly labeled training examples may be avoided and thus, more robust tracking achieved.

Tracking-learning-detection (TLD) [12] is a framework aiming to long-term target tracking by decomposing the task into tracking, learning, and detection sub-tasks. The tracker is tracking the object during the frames whereas the detector localizes all the appearances observed earlier and reinitializes the tracker if required. The final tracker estimate is a combination of the tracker and detector bounding boxes. The third sub-task, learning, tries to estimate the errors of the detector and update it to avoid those in the following frames.

Robust Object Tracking via Sparsity-based Collaborative Model (RSCM) by Zhong et al. [29] contains a sparsity-based discriminative classifier (SDC) and a sparsity-based generative model (SGM). SDC introduces an effective method to compute the confidence value that assigns more weight to the foreground by extracting sparse and determinative features that distinguish the foreground and background better. SGM is a histogram-based method that takes the spatial information of each patch into consideration with an occlusion handling scheme.

Fast Tracking via Spatio-Temporal Context Learning (STC) [26] algorithm works by learning a spatial context model between the target and its surrounding background. The learned model is used to update the spatio-temporal context

model for the following frame. The tracking task is formulated by convolution as a computing task of a confidence map, and the best object location can be estimated by maximizing the confidence map.

The main idea of Structured Output Tracking with Kernels (struck) [6] is to create positive samples from areas containing the object, and negative samples of the background further away from the object. It uses a confidence map and obtains the best location by maximizing a location likelihood function of an object.

Tracker based on Riemannian subspace learning (LRS)[10] is an incrementally learning tracking algorithm that focuses on appearance modeling using a subspace-based approach. The key component in LRS is the log-Euclidean block-division appearance model that aims to adapt to the changes in the objects appearance. In the incremental log-Euclidean Riemannian subspace learning algorithm, covariance matrices of image features are mapped into a vector space with the log-Euclidean Riemannian metric. The log-Euclidean block-division appearance model captures both local and global spatial layout information about the object's appearances. Particle filtering based Bayesian state inference is utilized as the core tracking technique.

2 Trajectory Filtering

In an ideal case, the motion between the frames should be at least one pixel in order to be quantifiable for the trackers. That is not always the case with high-speed videos and can create challenges for the trackers and trajectory analysis. Therefore, filtering of the trajectory data is necessary to obtain accurate velocity and acceleration measurements. Fig. 1 shows an example result of filtering the tracking data.

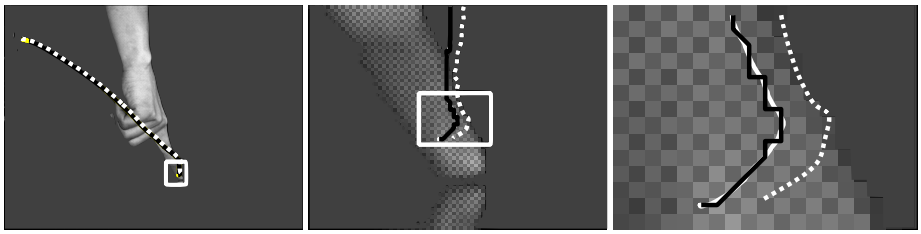


Fig. 1. Raw tracking data (black), the ground truth (dotted white) and filtered tracking data (white)

The following 8 filtering methods were considered in this work: Moving Average (MA) [20], Kalman Filter (KF) [22,23], Extended KF (EKF) [17], Unscented KF (UKF) [11], Local Regression (LOESS) [3], Locally Weighted Scatterplot Smoothing (LOWESS) [3], Savitzky-Golay (S-G) [18], and Total Variation Denoising (TVD) [2] .

MA filter operates by averaging subsets of input data points to produce a sequence of averages. A Kalman filter is an optimal recursive data processing algorithm. EKF is the nonlinear version of the Kalman filter and has been considered as the de-facto standard in nonlinear state estimation. In UKF, unscented transformation is used to calculate the statistics of a random variable which undergoes a nonlinear transformation. It is designed on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function. In KF, the predictor predicts parameter values based on the current measurements. The filter estimates parameter values by using the previous and current measurements. The smoothing algorithm estimates the parameter values by using the previous, current, and future measurements: that is, all available data can be used for filtering [23]. Future measurements can be used because the Kalman smoother proceeds backward in time. This also means that the Kalman filter needs to be run before running the smoother.

LOESS and LOWESS were originally developed to enhance visual information on scatterplots by computing and plotting smoothed points by using locally weighted regression. LOESS and LOWESS are methods to estimate the regression surface through a smoothing procedure. S-G is a smoothing filter, also called the polynomial smoothing or least-squares smoothing filter. S-G smoothing reduces noise while maintaining the shape and height of peaks. Total variation (TV) of a signal measures the changes in the signal between signal values. TVD output is obtained by minimizing a TV-based cost function. It was developed to preserve sharp edges in the underlying signal.

3 Experiments

3.1 Data

Data was collected during a HCI experiment where test subjects were advised to perform intentional single finger pointing actions from trigger-box toward a colored target on a touchscreen. The target on the touchscreen was one of 13 objects which formed a circle on the screen, were of different sizes, and lay on different parallaxes. Hand movements were recorded with a Mega Speed MS50K high-speed camera equipped with Nikon Nikkor AF-S 14-24mm F2.8G objective fixed to a 14mm focal length. The camera was positioned on the right side of the test setup, and the distance to the screen was approximately 1.5 meters. The lighting was arranged using an overhead light panel 85 cm above the table surface and 58 cm in depth. The test subject was sitting at the distance of 65 cm from the touch screen and a trigger-box was placed 40 cm away from it.

Dataset contained 11 high-speed videos with 800×600 resolution recorded at 500 fps. Sample frames from the dataset can be seen in Fig. 2. These images illustrate the different end-points of the trajectories. The start-point for all the sequences was the same. The ground truth was annotated manually. Annotations were done for every 5th frame and then interpolated using spline interpolation to get the ground truths for every frame.

Fig. 2. The sample images are from the dataset used in the experiments. Those were all taken from the end point of respective videos. The ground-truth bounding-box can be seen as a white rectangle in the images.

3.2 Results

The tracking experiments were carried out using the original implementations of the authors except in the case of MIL; for that, the implementation by Luo [16] was used. Search area parameters of the trackers were tuned for the video data used, if it was possible with the implementation. For the other parameters, the default values proposed by the original authors were used. The tracking methods were run 10 times for each video and the results were averaged to minimize random factors in tracking. Table 2 shows the results of the trackers for the dataset. The tracking rate of 100% with threshold of 32 pixels center location error was achieved by three of the trackers, KCF being the best one in overall results with the smallest average center location error of 4.65. Also, struck, and STC achieved high accuracy. Length of the videos in total was 10798 frames and individual videos were between 544 and 1407 frames long.

Table 2. Tracking results for Dataset: percentage of correctly tracked frames (TR%) and average center location errors (Err.), and the processing speed (fps). Also, the range of the values from the results are shown. The best results are shown in bold.

Method	TR%	TR% range	Err.	Err. range	fps	fps range
CT	79.43%	0-100%	18.43	3.5-76	99.97	63-121
FCT	17.14%	0-73%	58.74	16-92	118.73	72-150
HT	97.12%	36-100%	15.29	3.1-226	4.65	4-4.9
IVT	74.50%	15-100%	86.75	2.0-448	63.38	51-70
KCF	**100%**	-	**4.65**	1.4-7.4	979.97	728-1236
LRS	20.51%	2-47%	291.32	76-540	8.79	7.8-9.4
MIL	93.82%	24-100%	11.35	2.8-138	0.55	0.4-0.6
RSCM	86.81%	40-100%	18.84	2.2-126	2.50	2.0-2.8
SRPCA	83.64%	24-100%	72.38	1.7-366	10.52	8.3-12.5
STC	**100%**	-	5.13	2.3-6.9	**1291.03**	1156-1330
struck	**100%**	-	4.72	1.6-6.5	118.62	99-153
TLD	68.48%	16-100%	43.55	4.4-139	16.46	8.8-24

When working with high-speed videos, the importance of processing speed is emphasized. The experiments were carried out using a desktop computer with an Intel i5-4570 CPU and 8 GB of memory. The fps measure used in the experiments was calculated without including the image loading times in the calculations to get the raw frame processing speed. The highest fps was measured for STC which showed the best average performance and for KCF which had the peak performance of over 1200 fps. Both achieved processing speeds well over the frame rate of the videos. However, it should be noted that due to the different programming environments (MATLAB, C, etc.) and levels of performance optimization, these results should be considered merely suggestive.

KCF was selected for the further study since it correctly tracked all the frames, had one of the smallest average center location error, was able to process the high-speed videos in real-time. Moreover, earlier tracking experiments [14] have shown that KCF is more robust than STC on diverse video content.

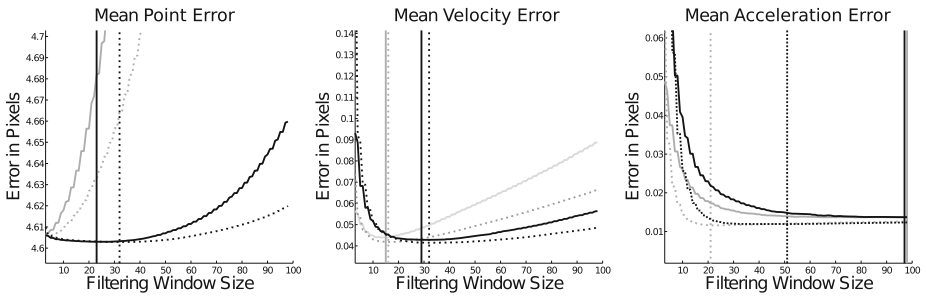
Table 3 summarizes the trajectory filtering results. The results were calculated by averaging the results from all dataset trajectories tracked with KCF tracker. The window size and method parameters were optimized separately for each filtering method. Filtering with Unscented Kalman Smoother (UKS) and TVD are included for comparison. UKS was selected to represent Kalman smoother algorithms since Extended Kalman Smoother and UKS produced similar results. Velocity and acceleration curves for trajectories obtained using Kalman filtering were computed using the Kalman filtering motion model. For the trajectories obtained using other filtering methods, velocity and acceleration curves were computed based on Euclidean distances between trajectory points in consecutive frames.

Table 3. Minimal mean and standard deviations of Position Errors (PE), Velocity Errors (VE), and Acceleration Errors (AE) with different filtering methods. In parentheses is the filtering window size which gave the best result for the filter. The best results are shown in bold.

Error	Moving Average	LOWESS	LOESS	Savitzky-Golay	TVD	UKS	unfiltered
Mean PE	4.6057 (3)	4.6057 (4)	**4.6029** (34)	4.6030 (25)	4.6474	4.6033	4.6099
Mean VE	0.0440 (17)	0.0419 (18)	**0.0415** (34)	0.0428 (31)	0.2052	0.0421	0.2085
Mean AE	0.0137 (83)	**0.0118** (23)	0.0119 (53)	0.0137 (97)	0.3026	0.0125	0.3074
std PE	1.3496 (5)	1.3495 (8)	**1.3459** (38)	1.3461 (29)	1.3860	1.3468	1.3917
std VE	0.0544 (15)	0.0522 (18)	**0.517** (34)	0.0532 (27)	0.2599	0.0526	0.2641
std AE	0.0210 (95)	0.0185 (23)	**0.0185** (39)	0.0206 (85)	0.4599	0.0190	0.4652

From the results shown in Fig. 3, it is obvious that different window sizes were optimal for each derivative of the position. The velocity and acceleration curves needed larger window sizes to get better results than the position. LOESS filtering was the least sensitive to window size with optimal filtering results from the window size range of 34 to 53. The problem with a large window size is that

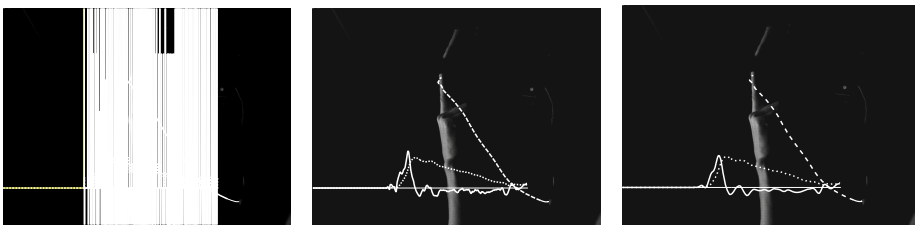
the estimated position starts to drift off from the true position which is very clear in case of moving average and LOWESS filtering.



(a) (b) (c)

Fig. 3. Filtering the effect of the window size on the means of (a) point error; (b) velocity error and (c) acceleration errors. The location of minimum error for each of the methods is indicated with the vertical line. Moving average is shown in grey, LOWESS in dotted grey, LOESS in dotted black, and Savitzky-Golay in black.

An example of how filtering affects the tracking data is shown in Fig. 4. In Fig. 4(a) no filtering is applied to tracking data before calculating velocity and acceleration values. Fig. 4(b) shows the result when position data after tracking is filtered with LOESS filtering with a span of 40 frames, and the velocity and acceleration values are calculated from that filtered data. In Fig. 4(c), also the velocity data is filtered after position data filtering with the same LOESS filtering method. From these results, it is clearly visible that filtering is needed to achieve appropriate velocity and acceleration curves from the tracked hand movement data.



(a) (b) (c)

Fig. 4. Tracking data and velocity and acceleration curves computed from it using: (a) Raw data; (b) Position data filtered with LOESS (span of 40); (c) Position and velocity data filtered with LOESS (span of 40). Trajectory is shown in dashed, velocity in dotted, and acceleration in continuous.

4 Conclusion

In this paper, hand tracking in high-speed videos during HCI tasks, and post-processing of the tracked hand trajectories were studied. The results showed that objects in high-speed video feeds with almost black background can be tracked in real-time with two of the tested trackers. For this research, this meant reaching speeds of over 970 (KCF) and over 1290 (STC) fps on average for the test video sequences which were recorded at 500 frames per second. Thus, the trackers satisfied real-time needs. Even though the performance evaluation for the trackers in this setup did not include the image-loading times, 2.3 milliseconds on average per image with MATLAB, the results are still impressive.

Filtering helps to find smooth acceleration curves to allow us see clearly where the moments of maximum and minimal acceleration are. With appropriate filtering, the velocity and acceleration features of the trajectories got closer to the ground truth. Two filtering methods, LOESS and UKS, produced the most consistent results for all the tests. Selecting one method as the winner raised the question, which one is simpler to use, and that happened to be LOESS. To conclude, with filtering and smoothing the hand-tracking data, it is possible to get to the underlying characteristics of the real movement sequence.

Smoothing the trajectories produced by the trackers gave good results for the derivatives of the position, but sub-pixel accuracy for video sequences which require high precision could be alternative way. By having more accurate positions of the object, one would not need to smooth the trajectories and more accurate results also for the velocities and accelerations of the moving object would be generated. The videos used in this work did not have large scale changes, but adapting to the scale changes on sub-pixel level could help to make the tracking process even more accurate. Also, ground-truth annotation process proved to be a hard undertaking. Clearly visible and accurate marker in test subject's finger would have helped the ground-truth annotation process.

The results provide observations about the suitability of tracking methods for high-speed hand tracking and about how filtering can be applied to produce more appropriate velocity and acceleration curves calculated from the tracking data.

Acknowledgments. The research was carried out in the COPEX project (No. 264429) funded by the Academy of Finland.

References

1. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8), 1619–1632 (2011)
2. Chambolle, A.: An Algorithm for Total Variation Minimization and Applications. Journal of Mathematical Imaging and Vision **20**(1–2), 89–97 (2004)
3. Cleveland, W.S.: Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association **74**(368), 829–836 (1979)

4. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108**(1–2), 52–73 (2007). special Issue on Vision for Human-Computer Interaction
5. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. Computer Vision and Image Understanding **117**(10), 1245–1256 (2012)
6. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)
7. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012)
8. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 583–596 (2015)
9. Hiltunen, V., Eerola, T., Lensu, L., Kälviäinen, H.: Comparison of general object trackers for hand tracking in high-speed videos. In: International Conference on Pattern Recognition (ICPR), pp. 2215–2220 (2014)
10. Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S., Zhang, Z.: Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(12), 2420–2440 (2012)
11. Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: Proceedings of The International Society for Optics and Photonics (SPIE) AeroSense: International Symposium on Aerospace/Defense Sensing, Simulations and Controls (1997)
12. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7), 1409–1422 (2012)
13. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., et al.: The visual object tracking VOT2014 challenge results. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014 Workshops. LNCS, vol. 8926, pp. 191–217. Springer, Heidelberg (2015)
14. Kuronen, T.: Post-Processing and Analysis of Tracked Hand Trajectories. Master's thesis, Lappeenranta University of Technology (2014)
15. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A Survey of Appearance Models in Visual Object Tracking. ACM Transactions on Intelligent Systems and Technology (TIST) **4**(4), 58:1–58:48 (2013)
16. Luo, W.: Matlab code for Multiple Instance Learning (MIL) Tracker. http://whluo.net/matlab-code-for-mil-tracker/. (accessed: August, 2013)
17. Montemerlo, M., Thrun, S.: Simultaneous localization and mapping with unknown data association using FastSLAM. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 2, pp. 1985–1991 (2003)
18. Orfanidis, S.J.: Introduction to Signal Processing. Prentice Hall international editions, Prentice Hall (1996–2009)
19. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental Learning for Robust Visual Tracking. International Journal of Computer Vision **77**(1–3), 125–141 (2008)
20. Smith, S.W.: The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing (1997)
21. Wang, D., Lu, H., Yang, M.H.: Online Object Tracking With Sparse Prototypes. IEEE Transactions on Image Processing **22**(1), 314–325 (2013)

22. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Tech. rep. Department of Computer Science, University of North Carolina (1995)
23. Welch, G., Bishop, G.: An Introduction to the kalman filter: SIGGRAPH 2001 course 8. In: Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques, pp. 12–17 (2001)
24. Wu, Y., Lim, J., Yang, M.H.: Object Tracking Benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence (2015)
25. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. ACM Computing Surveys **38**(4) (2006)
26. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 127–141. Springer, Heidelberg (2014)
27. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
28. Zhang, K., Zhang, L., Yang, M.H.: Fast Compressive Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(10), 2002–2015 (2014)
29. Zhong, W., Lu, H., Yang, M.H.: Robust Object Tracking via Sparse Collaborative Appearance Model. IEEE Transactions on Image Processing **23**(5), 2356–2368 (2014)

Publication II

Lyubanenko, V., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Häkkinen, J.

Multi-camera finger tracking and 3D trajectory reconstruction for HCI studies

Reprinted by permission from Springer Nature:
Lecture Notes in Computer Science
Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS)
Vol. 10617, pp. 63–74, 2017.

Multi-camera Finger Tracking and 3D Trajectory Reconstruction for HCI Studies

Vadim Lyubanenko[1,2], Toni Kuronen[1(✉)], Tuomas Eerola[1], Lasse Lensu[1], Heikki Kälviäinen[1], and Jukka Häkkinen[3]

[1] School of Engineering Science, Machine Vision and Pattern Recognition Laboratory, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland
{vadim.lyubanenko,toni.kuronen,tuomas.eerola,lasse.lensu,heikki.kalviainen}@lut.fi

[2] Institute of Mathematics, Mechanics and Computer Science, Laboratory of Artificial Intelligence and Robotics, Southern Federal University, Rostov-on-Don, Russian Federation

[3] Institute of Behavioural Sciences, University of Helsinki, P.O. Box 9, 00014 Helsinki, Finland
jukka.hakkinen@lut.fi

Abstract. Three-dimensional human-computer interaction has the potential to form the next generation of user interfaces and to replace the current 2D touch displays. To study and to develop such user interfaces, it is essential to be able to measure how a human behaves while interacting with them. In practice, this can be achieved by accurately measuring hand movements in 3D by using a camera-based system and computer vision. In this work, a framework for multi-camera finger movement measurements in 3D is proposed. This includes comprehensive evaluation of state-of-the-art object trackers to select the most appropriate one to track fast gestures such as pointing actions. Moreover, the needed trajectory post-processing and 3D trajectory reconstruction methods are proposed. The developed framework was successfully evaluated in the application where 3D touch screen usability is studied with 3D stimuli. The most sustainable performance was achieved by the Structuralist Cognitive model for visual Tracking tracker complemented with the LOESS smoothing.

Keywords: Human-computer interaction · Object tracking · Finger tracking · Multi-view tracking · 3D reconstruction

1 Introduction

The motivation for this work comes from the human-computer interaction (HCI) research, and the need to accurately record hand and finger movements of test subjects in various HCI tasks. Recent progress in the domain of HCI has allowed to form the next generation of user interfaces, combining touch input with stereoscopic 3D (S3D) content visualization.

J. Blanc-Talon et al. (Eds.): ACIVS 2017, LNCS 10617, pp. 63–74, 2017.
https://doi.org/10.1007/978-3-319-70353-4_6

Stereoscopically rendered views provide additional depth information that makes depth and structure judgements easier, enhances the ability to detect camouflaged objects as well as increases the ability to recognize the surface material [3,17]. Furhermore, stereoscopic presentation enhances the accuracy of visually guided touching and grasping movements [26]. Although touch input has already proved its utility and indispensability for various HCI applications, interacting with stereoscopically rendered content is still a challenging task. Usually the touch recognition surface is placed onto another plane than the displayed content, which being stereoscopically rendered floats freely in front of or behind the monitor. It has been shown that touching an intangible surface (i.e., touching the void) leads to confusion and a significant number of overshooting errors [4].

Advances in gesture interfaces, touch screens, and augmented and virtual realities have brought new usability concerns that need to be studied in a natural environment and in an unobtrusive way [28]. Several robust approaches for hand tracking exist that can measure the hand and finger location with high accuracy, for example, data gloves with electromechanical, infrared or magnetic sensors [10]. However, such devices affect the natural hand motion and cannot be considered feasible solutions when pursuing natural HCI.

Image-based solutions provide an unobtrusive way to study and to track human movement and enable natural interaction with the technology. Commercially available solutions such as Leap Motion[1] and Microsoft Kinect[2] limit the hand movement to a relatively small area, do not allow frame rates high enough to capture all the nuances of rapid hand movements, and are imprecise for accurate finger movement measurements.

State-of-the art object tracking techniques allow automatic estimation of motion trajectories from videos. The main problem with using existing object tracking methods for accurate measurement of hand and finger movements is that they are developed for applications where high (sub-pixel) accuracy is unnecessary. While this is a justified choice for most tracking scenarios, this is not the case in hand trajectory measurement where high spatial accuracy is the main concern. Even small errors in spatial locations can lead to large fluctuations in the speed and acceleration calculated from the location data. Smoothing raw trajectory data with an appropriate filtering method provides a solution for small irregularities in the trajectory data without compromising the tracking results [20].

Another challenge for the finger tracking in HCI studies is the fact that the pointing actions are typically fast which causes large shifts in object locations between frames. While this problem can be solved by using high-speed cameras [15,20], a setup consisting multiple high-speed cameras required for recording 3D trajectories is both expensive and difficult to build. Therefore, it is important to invest on selecting of the suitable tracker that can handle this issue also on normal speed videos.

[1] Leap motion: https://www.leapmotion.com/product.
[2] Microsoft Kinect: http://www.xbox.com/en-US/kinect.

In this paper, a multi-camera framework for measuring hand movements in HCI studies is presented. The framework is developed for the measurement setup consisting of a high-speed camera and normal-speed camera with different viewing angles. The comparison of object trackers for high-speed videos has been provided in an earlier study [20]. In this paper, this is complemented with a tracker comparison on normal-speed videos to find methods that can handle the large shifts in object locations between frames. Moreover, necessary trajectory post-processing, failure detection, and 3D trajectory reconstruction methods are proposed to produce accurate 3D measurements. The framework is generic in nature, and here its functionality is demonstrated in an application where 3D touch screen usability is studied with 3D stimuli.

2 Experiment Setup

The framework was developed for a HCI experiment that uses a S3D touch screen setup. During the trials, test subjects were asked to perform a clear pointing action towards the observed 3D stimuli. Stereoscopic presentation of stimuli was done with the NVIDIA 3D Vision kit. The touch screen was placed at a distance of 0.65 m in front of the person. The trigger-box, which button pressing denoted the beginning of a single pointing action, was set up 0.25 m away from the screen. The process was recorded with two cameras: a Mega Speed MS50K high-speed camera equipped with the Nikon 50 mm F1.4D objective, and a normal-speed Sony HDR-SR12 camera. The high-speed camera was installed on the right side of the touch screen with an approximately 1.25 m gap in-between, while the normal-speed camera was mounted on the top (see Fig. 1). Example frames from both cameras are presented in Fig. 2.

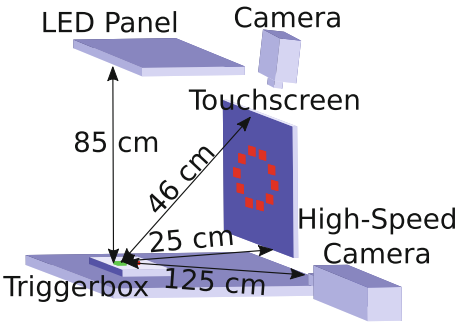
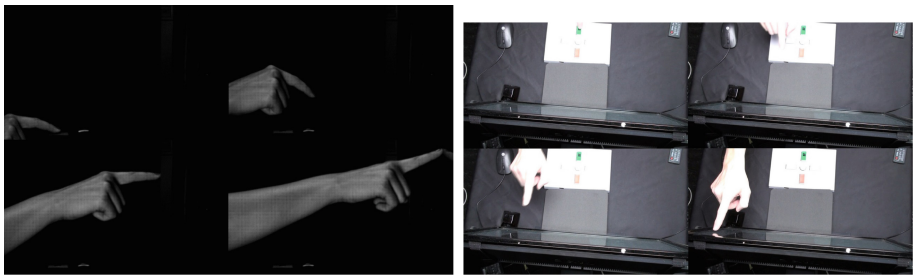


Fig. 1. 3-D touch display experiment.

Similar to earlier pointing action research, e.g., [8], the experiment focused on studying intentional pointing actions. The stimuli were generated by a stereoscopic display with the touch screen to evaluate the effect of different parallaxes, i.e., perceived depth. This arrangement enables study of (potential) conflict between visually perceived and touch-based sensations of depth.



(a) (b)

Fig. 2. Example video frames of volunteer interaction with the 3D touch screen display captured with the high-speed camera (a) and normal-speed camera (b).

3 Finger Tracking

The object trackers were selected to the evaluation based on the following criteria: (1) a high ranking in the Visual Object Tracking (VOT2016) challenge [18], (2) real-time performance on 25 fps videos, and (3) publicly available implementation. The selected trackers are listed in Table 1.

Table 1. Trackers selected for the experiments.

Method	Abbrev.	Year
Sum of template and pixel-wise LEarners [2]	Staple	2016
An improved staple tracker with multiple feature integration [18]	Staple+	2016
Distractor aware tracker [24]	DAT	2015
Scale adaptive mean shift [31]	ASMS	2014
Kernelized correlation filter tracker [14]	KCF	2014
Structuralist cognitive model for visual tracking [5]	SCT	2016
Scalable kernel correlation filter with sparse feature integration [21]	sKCF	2015
Structured output tracking with kernels [11]	STRUCK	2016
Incremental learning for robust visual tracking [25]	IVT	2008
Spatio-temporal context tracker [33]	STC	2014

Correlation filters have shown excellent performance for visual object tracking [14]. Trackers based on these filters are highly sensitive to target appearance deformation because of using a rigid template, but they can be complemented

with a target representation that is insensitive to shape variation. Sum of Template and Pixel-wise LEarners (Staple) [2] utilizes color histograms as an additional representation robust to deformation, since they do not depend on the spatial structure within the image patch. In [18], an improved version of the Staple tracker (Staple+) was proposed. While the original algorithm extracts HOG features from a gray-scale image, Staple+ relies on HOG features retrieved from color probability map, which are expected to better represent the image patch color information.

Distractor Aware Tracker (DAT) [24] relies on a discriminative object surrounding model employing a color histogram for differentiating the object from the background. To suppress the risk of drifting DAT proposes an additional distractor-aware model which allows to robustly detect possible distracting objects whenever they appear within the field-of-view.

The Scale Adaptive Mean Shift (ASMS) [31] tracker enhances the mean-shift tracking algorithm [7] by targeting the problem of a fixed size tracking window. ASMS encompasses a novel scale estimation mechanism based on the mean-shift procedure for the Hellinger distance [23]. Moreover, the authors present a technique to validate the estimated output, called the Backward scale consistency check. It uses reverse tracking to check that the object size has not changed mistakenly. Another improvement is the introduction of a background ratio weighting (BRW), which uses the target background color information computed over its neighborhood in the first frame.

High-Speed Tracking with Kernelized Correlation Filters (KCF) [14] is an improved version of the kernelized correlation filters introduced in [13]. The correlation filters produce a correlation peak for a target object in a scene and low response for background. In [30] the original algorithm is extended by a scale estimation (7 different scale steps) and by color-names features [29] (denoted as KCF2). As an improvement to KCF, Scalable Kernel Correlation Filter with Sparse Feature Integration (sKCF) was proposed in [21]. sKCF replaces the cosine window with an adjustable Gaussian windowing function to support target size changes and, hence, produce better back- and foreground separation. The new appearance window size is estimated with a forwards-backwards optical flow strategy. It extracts relevant keypoints of the target area on the successive frames and then estimates the scale change by analyzing the pair-wise difference.

The Structuralist Cognitive model for visual Tracking (SCT) [5] tracker decomposes tracking into two stages: disintegration and integration. In the first stage, multiple cognitive structural units, attentional feature-based correlation filters (AtCFs), are generated. Each unit consists of an attentional weight estimator and KCF. Each AtCF utilizes a unique pair of a feature (color, HOG, etc.) and a kernel (linear, Gaussian, etc.) type. In the integration step, the object appearance is expressed as a representative combination of AtCFs, which is memorized for future usage.

The main idea of Structured Output Tracking with Kernels (struck) [11] is to create positive samples from areas containing the object and negative samples from the background further away from the object. It uses a confidence map

and obtains the best position by maximizing a location likelihood function of an object.

Incremental Learning for Robust Visual Tracking (IVT) [25] learns a low-dimensional subspace representation of the target object and tracks it using a particle filter.

The Fast Tracking via Spatio-Temporal Context Learning (STC) [33] algorithm is also based on correlation filters, but it adds a spatial context model between the target and its surrounding background. The learned model is used to update the spatio-temporal context model for the following frame.

4 Post-processing and 3D Trajectory Reconstruction

The trajectory data retrieved as the result of tracking usually presents an ordered list of object location coordinate points in an image plane. These measures may contain movement noise or completely incorrect position estimation (if the tracker lost the target) since none of the currently available visual trackers achieve an irreproachable accuracy. Moreover, most visual trackers estimate the object location only with a pixel precision, and therefore the obtained trajectory presents a broken line instead of a desired smooth curve. As noted in [19], the rough-edged transforms between the trajectory points noticeably affect the precision of succeeding calculations. These negative effects can be eliminated by an introduction of trajectory smoothing and tracking failure detection methods into the processing flow.

For the trajectory smoothing Local Regression (LOESS) [6] was selected based on the comparison performed in [20]. A commonly used approach to detect failures is to calculate sum-of-square differences (SSD) [22] or utilize other similar measures [9] between the consequent object area patches. This measure allows to detect various occlusions or target rapid leaps, but it does not recognize gradual trajectory drifting. Comparison between the current object appearance and affine warps of the initial patch can be utilized for drift detection [27], but this method is only applicable for rigid objects and uniform environment. In [32] and [16] the strategy of forward and backward in time analysis of the video sequence was proposed. First the backtracking is used for an estimation of the reverse trajectory from the current timestamp to an earlier moment. Then the divergences between these two trajectories are measured.

To obtain 3D trajectory the 2D trajectories estimated using different-view calibrated cameras need to be combined. The task of computing a 3D trajectory from multiple 2D trajectories is essentially equivalent to the process of 3D scene reconstruction. For this purpose, we utilize the well-known method [12] described in Algorithm 1.

5 Experiments

The experiments were performed on the laptop running Windows 10 x 64 equipped with Intel Core i5-6200U processor and 6 GB memory.

Algorithm 1. The algorithm for three-dimensional scene reconstruction [12]

1. Find point correspondences from the two trajectories.
2. Compute the essential matrix from point correspondences.
3. Compute camera projection matrices from the essential matrix.
4. For each point correspondence, compute the position of a scene point using triangulation.

5.1 Dataset

17 pointing action videos were selected to study the performance of the trackers. The set contains multiple test subjects with a varying appearance of fingers (e.g. with and without glossy nail polish). The videos were recorded using interlaced encoding with 50 field rate and 1440 × 1080 (4:3) resolution. For deinterlacing Yet another deinterlacing filter (yadif) [1] was utilized with both frame-to-frame conversion producing 25 fps videos and field-to-frame conversion producing double frame rate (50 fps) videos. To obtain the ground truth for the evaluation, the finger was manually annotated to each frame in all 17 videos.

The corresponding high-speed videos were recorded at 500 fps and 800 × 600 resolution. The finger tracking and trajectory smoothing for high-speed videos were performed as proposed in [20]. High-speed videos were manually aligned with the normal speed videos using timestamp information.

5.2 Tracking

Each tracking method was evaluated with both normal and double frame rate videos. Moreover, in addition to the forward tracking also backward tracking from the end of video was considered. Finger tracking in high-speed videos was not considered in this work. Thereby, the following tracker evaluation cases were considered: (1) normal frame rate (25 fps), forward tracking (NFR/FT), (2) double frame rate (50 fps), forward tracking (DFR/FT), (3) normal frame rate, backward tracking (NFR/BT), and (4) double frame rate, backward tracking (DFR/BT).

The accuracy of tracking was measured with the center location error (CLE) with respect the ground truth. The percentage of correctly tracked frames (PCF), i.e., the frames where the distance between the ground truth and the estimated position was below a fixed threshold ($\tau = 16$ pixels) was used as a measure of robustness. The results over all test videos are shown in Table 2. The performance visualization was done via precision plots (Fig. 3).

The best result in forward tracking across the both frame rate cases was achieved with the SCT tracker, which was able to track the target through 77% of frames with CLE of 25.7 pixels. STRUCK was the second with 62% of successfully tracked frames, followed by KCF2 with 61%. None of the trackers handled all the 17 sequences in the dataset. In DFR/FT, SCT tracked the target till the end only in 11 sequences, KCF2 in 7, while the others successfully tracked 3 sequences or less. In backward tracking, KCF2 achieved the first place with

Table 2. Tracking results. The best results for Forward and Backward tracking are given in bold.

	Forward tracking				Backward tracking			
	NFR		DFR		NFR		DFR	
Method	CLE	PCF	CLE	PCF	CLE	PCF	CLE	PCF
Staple	141.7	44	102.5	43	129.5	45	14.6	84
Staple+	146.7	43	102.6	40	113.5	43	14.9	82
DAT	100.3	41	52.2	46	41.8	44	33.9	51
KCF	120.3	42	45.5	50	72.4	62	**10.8**	83
KCF2	145.0	49	63.1	61	56.3	75	19.4	**86**
ASMS	146.1	39	109.6	28	52.7	40	53.7	41
SCT	90.6	57	**25.7**	**77**	29.7	83	17.6	81
STC	136.4	39	52.1	51	41.2	75	13.6	83
sKCF	146.4	46	71.1	49	136.4	48	29.3	80
STRUCK	117.3	46	43.4	62	294.6	15	112.6	38
IVT	177.4	43	183.7	31	374.4	7	336.2	10

86% and CLE of 19.4, while KCF took the first place in CLE measure with 10.8 and shared the third place in PCF with 83%. In DFR/BT, KCF2 tracked the target till the end in 13 videos, Staple and sKCF - 12, while Staple+, SCT, and STC tracked 11 records each. In some of the videos, the target appearance was altered due to light reflection, and only STRUCK successfully adapted to these changes. Some videos were distorted with motion blur, while in few videos the test subject bent his/her forefingers during the experiment making them partially occluded with respect to the camera view.

5.3 Smoothing

The efficiency of the trajectory smoothing with the LOESS algorithm was evaluated using the trajectories retrieved from the double frame rate videos. The SCT and KCF2 tracker outputs were selected for forward and back-tracking, respectively. The efficiency was evaluated against the ground truth using the CLE measure. The results are shown in Table 3.

Table 3. Mean and variance (in parentheses) of center location error (CLE) for raw and smoothed (with various span size in frames) trajectory data. The best results are shown in bold.

Case	Raw	Smoothed (with various span size)		
		5	7	9
DFT/FT [SCT]	4.02 (6.08)	**3.97 (5.10)**	4.40 (6.62)	5.29 (10.78)
DFR/BT [KCF2]	**3.27** (4.48)	3.32 **(4.36)**	3.68 (5.29)	4.30 (8.47)

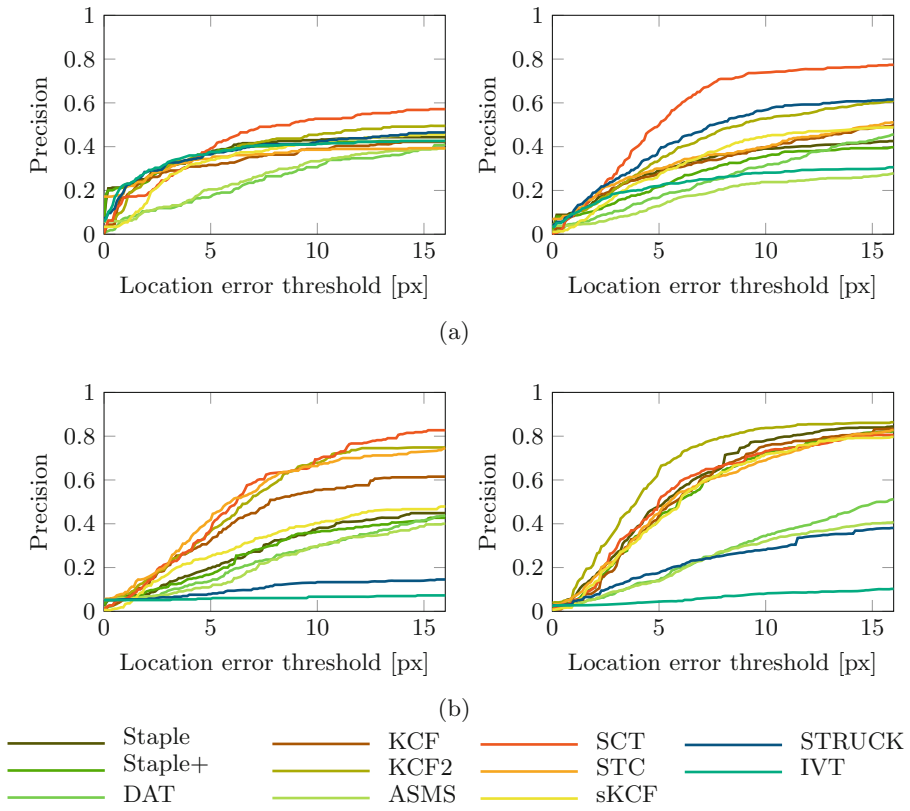


Fig. 3. Forward tracking (a) and backward tracking (b) precision plots. NFR is shown on the left side and DFR on the right side.

The LOESS smoothing with the span size of 5 frames systematically decreased the variance of the CLE measure. For the DFT/FT tracking case, smoothing also fractionally shrank the deviation between the estimated and the ground truth position.

5.4 Reconstructing 3D Trajectories

To demonstrate the 3D reconstruction of trajectories, seven videos were used. Camera calibration was done with a standard checkerboard calibration target with a pattern consisting of 26.5 mm patches. A set of captured calibration images was used to compute the intrinsic camera parameters and distortion coefficients. As a result, seven finger trajectory points corrected for lens distortion from both normal and high-speed videos were used to form the basis of 3D-reconstruction. The reconstruction scale ambiguity was eliminated by a fixed setup with known distances between the high-speed camera (HS), normal speed camera (NL), and the monitor. The reconstruction results are visualized in Fig. 4.

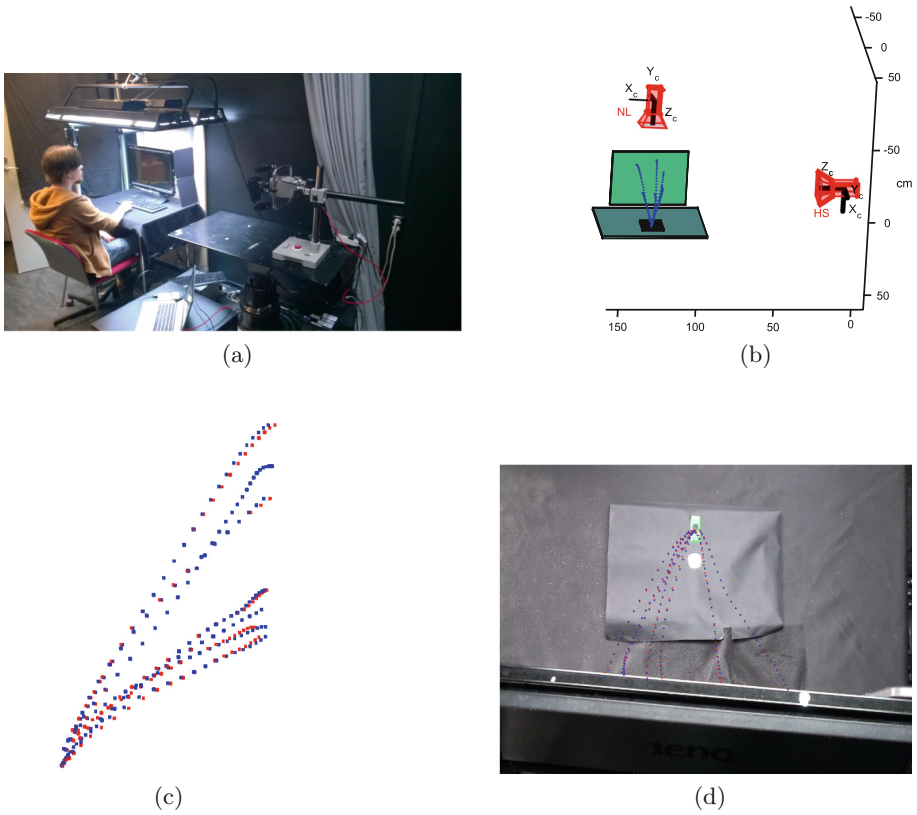


Fig. 4. (a) Experiment setup, (b) 3D reconstruction result from one viewpoint, and 3D point reprojection to (c) the high-speed camera image plane and (d) the normal camera image plane. The trajectories include seven captured pointing actions displayed as blue dotted curves.

With the absence of ground truth for the 3D trajectories, the 3D reconstruction accuracy was assessed with the calculation of the reprojection error measure [12]. The average reprojection error over all the trajectory points used in the 3D reconstruction experiment comprised 3.89 pixels for the high-speed and 5.72 pixels for the normal camera videos respectively (see Fig. 4), which corresponds to a 1–3 mm accuracy in real world units.

6 Conclusion

In this work, a multi-camera framework to track finger movements in 3D was proposed for studying human-computer interaction. To select the best tracking method for the task, a comprehensive method evaluation was performed. SCT outperformed the other methods. However, none of the studied algorithms

achieved the ideal performance since they failed under various conditions. Moreover, trajectory post-processing and 3D trajectory reconstruction methods were proposed. Trajectory data processing was shown to smoothen the produced trajectories, enabling more accurate calculations of the derivatives of the target position. The proposed framework was successfully evaluated in the application where stereoscopic touch screen usability was studied with stereoscopic stimuli.

References

1. FFmpeg (2017). https://ffmpeg.org/. Accessed 01 May 2017
2. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
3. van Beurden, M.H., Van Hoey, G., Hatzakis, H., Ijsselsteijn, W.A.: Stereoscopic displays in medical domains: a review of perception and performance effects. In: IS and T/SPIE Electronic Imaging, p. 72400A. International Society for Optics and Photonics (2009)
4. Chan, L.W., Kao, H.S., Chen, M.Y., Lee, M.S., Hsu, J., Hung, Y.P.: Touching the void: direct-touch interaction for intangible displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2625–2634. ACM (2010)
5. Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., Young Choi, J.: Visual tracking using attention-modulated disintegration and integration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4321–4330 (2016)
6. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**(403), 596–610 (1988)
7. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142–149. IEEE (2000)
8. Elliott, D., Hansen, S., Grierson, L.E.M., Lyons, J., Bennett, S.J., Hayes, S.J.: Goal-directed aiming: two components but multiple processes. Psychol. Bull. **136**(6), 1023–1044 (2010)
9. Erdem, C.E., Sankur, B., Tekalp, A.M.: Performance measures for video object segmentation and tracking. IEEE Trans. Image Process. **13**(7), 937–951 (2004)
10. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: a review. Comput. Vis. Image Underst. **108**(12), 52–73 (2007). Special issue on vision for human-computer interaction
11. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
14. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)

15. Hiltunen, V., Eerola, T., Lensu, L., Kälviäinen, H.: Comparison of general object trackers for hand tracking in high-speed videos. In: International Conference on Pattern Recognition, pp. 2215–2220 (2014)
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: automatic detection of tracking failures. In: International Conference on Pattern Recognition, pp. 2756–2759. IEEE (2010)
17. Kooi, F.L., Toet, A.: Visual comfort of binocular and 3D displays. Displays **25**(2), 99–108 (2004)
18. Kristan, M., et al.: The visual object tracking VOT2016 challenge results. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 777–823. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_54
19. Kuronen, T.: Post-processing and analysis of tracked hand trajectories. Master's thesis, Lappeenranta University of Technology (2014)
20. Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J., Kälviäinen, H.: High-speed hand tracking for studying human-computer interaction. In: Paulsen, R.R., Pedersen, K.S. (eds.) SCIA 2015. LNCS, vol. 9127, pp. 130–141. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19665-7_11
21. Montero, A.S., Lang, J., Laganiere, R.: Scalable kernel correlation filter with sparse feature integration. In: Proceedings of the IEEE Conference on Computer Vision Workshops, pp. 587–594. IEEE (2015)
22. Nickels, K., Hutchinson, S.: Estimating uncertainty in SSD-based feature tracking. Image Vis. Comput. **20**(1), 47–58 (2002)
23. Nikulin, M.S.: Hellinger distance. Encyclopedia of Mathematics, vol. 151. Springer (2001)
24. Possegger, H., Mauthner, T., Bischof, H.: In defense of color-based model-free tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2113–2120 (2015)
25. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1), 125–141 (2008)
26. Servos, P., Goodale, M.A., Jakobson, L.S.: The role of binocular vision in prehension: a kinematic analysis. Vis. Res. **32**(8), 1513–1521 (1992)
27. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
28. Valkov, D., Giesler, A., Hinrichs, K.: Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In: Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces, ITS 2012, pp. 21–30. ACM, New York, NY, USA (2012)
29. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Trans. Image Process. **18**(7), 1512–1523 (2009)
30. Vojir, T.: Tracking with kernelized correlation filters (2017). https://github.com/vojirt/kcf/. Accessed 01 May 2017
31. Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking. Pattern Recogn. Lett. **49**, 250–258 (2014)
32. Wu, H., Sankaranarayanan, A.C., Chellappa, R.: In situ evaluation of tracking algorithms using time reversed chains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
33. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 127–141. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_9

Publication III

Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H.
Two-camera synchronization and trajectory reconstruction for a touch screen usability experiment

Reprinted by permission from Springer Nature:
Lecture Notes in Computer Science
Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS)
Vol. 11182, pp. 125–136, 2018.

Two-Camera Synchronization and Trajectory Reconstruction for a Touch Screen Usability Experiment

Toni Kuronen(✉), Tuomas Eerola, Lasse Lensu, and Heikki Kälviäinen

Machine Vision and Pattern Recognition Laboratory (MVPR), Department of Computational and Process Engineering, School of Engineering Science, Lappeenranta University of Technology, P.O.Box 20, 53851 Lappeenranta, Finland
{toni.kuronen,tuomas.eerola,lasse.lensu,heikki.kalviainen}@lut.fi

Abstract. This paper considers the usability of stereoscopic 3D touch displays. For this purpose extensive subjective experiments were carried out and the hand movements of test subjects were recorded using a two-camera setup consisting of a high-speed camera and a standard RGB video camera with different viewing angles. This produced a large amount of video data that is very laborious to analyze manually which motivates the development of automated methods. In this paper, we propose a method for automatic video synchronization for the two cameras to enable 3D trajectory reconstruction. This together with proper finger tracking and trajectory processing techniques form a fully automated measurement framework for hand movements. We evaluated the proposed method with a large amount of hand movement videos and demonstrated its accuracy on 3D trajectory reconstruction. Finally, we computed a set of hand trajectory features from the data and show that certain features, such as the mean and maximum velocity differ statistically significantly between different target object disparity categories. With small modifications, the framework can be utilized in other similar HCI studies.

Keywords: Human-computer interaction · Multi-view tracking
3D reconstruction · Stereoscopic touch screen · Image processing
Image analysis

1 Introduction

Advances in gesture interfaces, touch screens, and augmented and virtual realities have brought new usability concerns that need to be studied in a natural environment and in an unobtrusive way [15]. In this work we focus on the next generation of user interfaces, combining touch input with stereoscopic 3D (S3D) content visualization. Stereoscopically rendered views provide additional depth information that makes depth and structure judgments easier, enhances the ability to detect camouflaged objects as well as increases the ability to recognize

J. Blanc-Talon et al. (Eds.): ACIVS 2018, LNCS 11182, pp. 125–136, 2018.
https://doi.org/10.1007/978-3-030-01449-0_11

the surface material [2,9]. Furthermore, the stereoscopic presentation enhances the accuracy of visually guided touching and grasping movements [14]. Although touch input has already proved its utility and indispensability for various human-computer interaction (HCI) applications interacting with stereoscopically rendered contents is still a challenging task. Usually the touch recognition surface is placed on another plane than the displayed content which being stereoscopically rendered floats freely in front of or behind the monitor. It has been shown that touching an intangible surface, i.e., touching the void leads to confusion and a significant number of overshooting errors [3].

In order to study the usability of the stereoscopic 3D touch screen, it is important to be able to accurately record hand and finger movements of test subjects in 3D. Several robust approaches to hand tracking exist that can measure the hand and finger location with high accuracy, for example, data gloves with electromechanical, infrared, or magnetic sensors [6]. However, such devices affect the natural hand motion and cannot be considered feasible solutions when pursuing natural HCI. Image-based solutions provide an unobtrusive way to study and to track human movement and enable natural interaction with the technology. Commercially available solutions such as Leap Motion[1] and Microsoft Kinect[2] limit the hand movement to a relatively small area, do not allow frame rates high enough to capture all the nuances of rapid hand movements, and are imprecise for accurate finger movement measurements.

This study continues the work done in [11] and [12] where a camera-based hand movement measurement framework for HCI studies was proposed. In order to analyze automatically a large amount of video data, we complement the framework by proposing a video synchronization procedure for a setup consisting of a high-speed camera and a normal-speed camera with different viewing angles. The high-speed camera produces accurate information on hand movements in 2D while the additional normal-speed camera provides the possibility to measure the movements in 3D. The framework is further evaluated with a large scale HCI experiment where the usability of a 3D touch screen is studied with 3D stimuli. Finally, a set of hand trajectory features is computed from the data and they are compared with the different 3D stimuli, i.e., with target objects with different parallaxes.

2 Experiment Setup

The framework was developed for a HCI experiment that uses a S3D touch screen setup. During the trials, test subjects were asked to perform an intentional pointing action towards the observed 3D stimulus. The stereoscopic presentation of the stimuli were done with the NVIDIA 3D Vision kit. The touch screen was placed at a distance of 0.65 m in front of the person. The trigger box contained a button to be pressed to denote the beginning of a single pointing action and was set up 0.25 m away from the screen. The process was recorded

[1] Leap motion: https://www.leapmotion.com/product.
[2] Microsoft Kinect: http://www.xbox.com/en-US/kinect.

with two cameras: (i) a Mega Speed MS50K high-speed camera equipped with the Nikon 50 mm F1.4D objective and (ii) a normal-speed Sony HDR-SR12 camera. The high-speed camera was installed on the right side of the touch screen with an approximately 1.25 m gap in-between, while the normal-speed camera was mounted on the top (see Fig. 1). Example frames from both cameras are presented in Fig. 2. The high-speed camera was operated by the trigger resulting a separate video file for each pointing action. The normal-speed recorded the whole session for each test subject into one video file. This resulted in the need of camera synchronization and re-calibration.

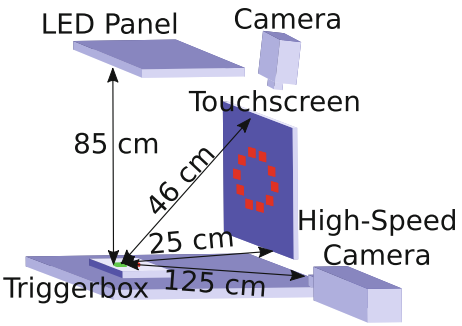
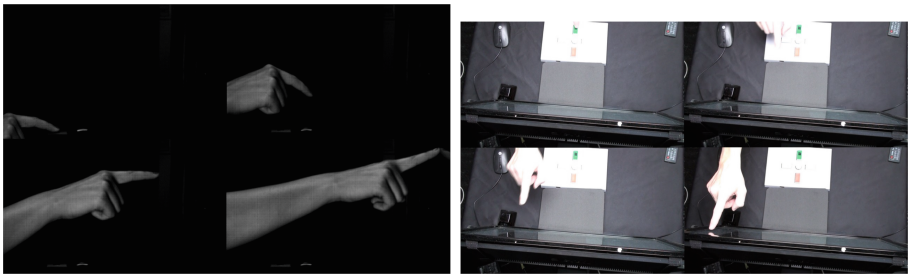


Fig. 1. 3-D touch display experiment.

Similar to earlier pointing action research, e.g., [5], the experiment focused on studying intentional pointing actions. The stimuli were generated by a stereoscopic display with the touch screen to evaluate the effect of different parallaxes, i.e., perceived depth. This arrangement enables study of (potential) conflict between visually perceived and touch-based sensations of depth.



(a) (b)

Fig. 2. Example video frames of volunteer interaction with the 3D touch screen display captured with the high-speed camera (a) and normal-speed camera (b).

2.1 Dataset

For the data collection, the pointing action tests were performed by 20 subjects. The pointing actions were divided into nine test blocks based on the interruptions in the high-speed imaging due to limited camera memory. The main test block contained 40 pointing actions per each parallax disparity. Disparity defines the difference in the target object locations between the images seen by the left and right eyes causing the target object to appear in front or behind the screen. Four disparities were considered: (1) 6 pixels causing the object to appear clearly in front of the screen, (2) 2 pixels causing the object to appear slightly in front of the screen, (3) -6 pixels causing the object to appear clearly behind screen, and (4) -2 pixels causing the object to appear slightly behind screen. Blocks 1 and 2 with disparities 6 and -6 were meant for the user to get acquainted with the setup. Blocks 3–6 were the main testing blocks with disparities 6, -6, 2, and -2. In blocks 7 and 8, the disparity was changed in the middle of the pointing action. Finally, block 9 was a control test with color information used as a target for the pointing actions.

The high-speed videos were recorded at 500 fps and 800×600 resolution. The normal-speed videos were recorded using interlaced encoding with 50 field rate and 1440×1080 (4:3) resolution. For deinterlacing the normal-speed videos the yet another deinterlacing filter (yadif) [1] was utilized with field-to-frame conversion producing double frame rate (50 fps) videos. In total, 2597 pointing actions were recorded with the both cameras.

3 Hand Tracking and Video Synchronization

The hand movements in the high-speed videos were tracked using [8] as proposed in [11]. The tracking window was initialized by a manually placed initial bounding box on the trigger box button image. The normal-speed videos were processed with motion detection near the monitor area. The motion detection was performed using background subtraction (frame differencing). There were few incorrect detections which were filtered out based on the known location of each touch. The detected motions were used to obtain the location of the finger tip which was further used to initialize the tracking window for the normal-speed videos. The tracking was performed using [16] as introduced in [12].

In order to automatically align the normal-speed videos with the high-speed videos, the ratio of framerates and delay are needed. The ratio of framerates of the cameras is known, so only the delay needs to be estimated from data. To do this, first, a coarse alignment was performed using timestamps accompanied with the high-speed videos and the starting time of the normal-speed videos. This made it possible to identify the blocks, i.e. the video sequence containing one block could be cut from the normal-speed videos and the high-speed videos corresponding the same pointing actions can be recognized. Nine blocks were identified based on the longer breaks in pointing actions caused by the limited high-speed camera memory, transfer of the memory contents to the computer and clearing the camera memory. The accurate alignment, i.e. the estimation

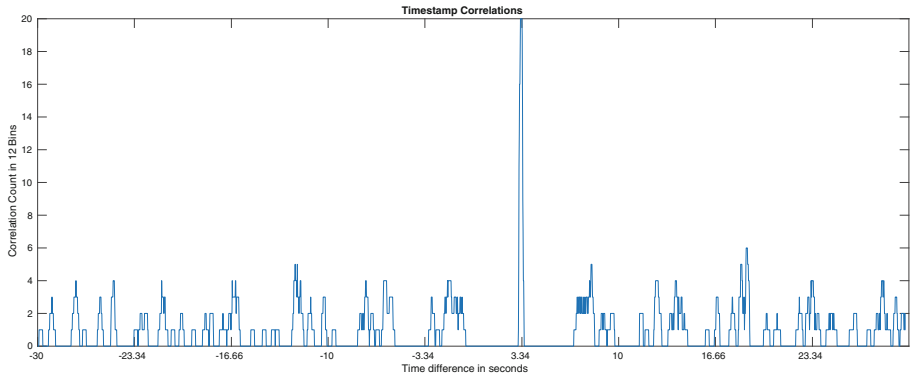
of the delay was done separately for each block. To do this points of the finger trajectories which can be detected from both videos need to be found. The trigger box had a white button that was visible on the both views, and the point where the trajectory passed the button was used to find a timestamp from both videos. Each block contains several point actions and therefore several timestamps were obtained.

The final alignment was done by searching the delay which maximizes correlation between the timestamp sequences for normal-speed and high-speed videos. As it can be seen from Fig. 3, the timestamp correlations of the corresponding events of passing the white button are periodical so that simply finding the minimal timestamp difference would not work in this case. The event matching was done by binning the trajectory event matches of the timestamp differences. One bin was the length of a single frame (0.02 s) and the maximum for the timestamp correlation was found by summing up 12 frames and finding the largest bin. 12 frames bin size were used because it was the minimal bin size that produced full bin of 20 corresponding time differences of the events for some of the blocks. Figure 3 presents examples of synchronizing blocks of pointing actions. In Fig. 3a all the samples are well correlated within a tight time range (around 3.34 s timestamp difference), but in Fig. 3b it is impossible to say which time difference gives a good result. This is mainly due to the low count of sequences which were tracked correctly in the both videos.

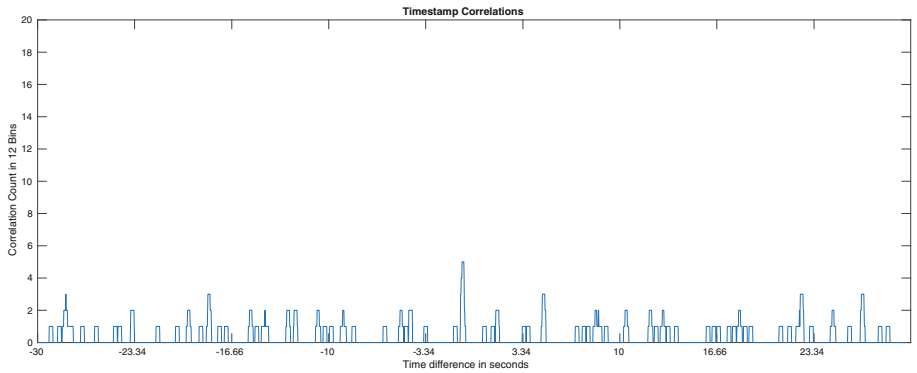
4 Post-processing and 3D Trajectory Reconstruction

The trajectory data retrieved as the result of tracking usually presents an ordered list of object location coordinate points in an image plane. None of the currently available visual trackers achieve immaculate accuracy, and thus, the measures may contain movement noise or completely incorrect position estimation if the tracker lost the target. Moreover, most visual trackers estimate the object location only with a pixel precision and, therefore, the obtained trajectory presents a broken line instead of a desired smooth curve. As noted in [10], the rough-edged transforms between the trajectory points noticeably affect the precision of subsequent calculations. These negative effects can be eliminated by introducing trajectory smoothing and tracking failure detection methods into the processing flow. For the trajectory smoothing, Local Regression (LOESS) [4] was used based on the comparison performed in [11].

To obtain a 3D trajectory, the 2D trajectories estimated using calibrated cameras with a different viewpoint need to be combined. The task of computing a 3D trajectory from multiple 2D trajectories is essentially equivalent to the process of 3D scene reconstruction. For this purpose, we utilized the well-known method [7] sketched in Algorithm 1. The essential matrix is computed with the M-estimator sample consensus (MSAC) algorithm. Finding the best suitable essential matrix was done by minimizing the back-projection errors. The evaluation was performed with confidence levels varying from 90% to 99%, and different Sampson distance [13] thresholds from 5 to 35 pixels.



(a)



(b)

Fig. 3. Example of synchronizing one block of pointing actions from high correlation in one time range (a) and from low correlations within the whole time range (b).

Algorithm 1. The algorithm for three-dimensional scene reconstruction [7]

1. Find point correspondences from the two trajectories.
2. Compute the essential matrix from point correspondences.
3. Compute camera projection matrices from the essential matrix.
4. For each point correspondence, compute the position of a scene point using triangulation.

It was observed that the normal-speed camera did not capture the full trajectories of the hand movement because the touch interface in front of the monitor blocked the view of the finger tip near the monitor. Since the full trajectory was not captured the depth information of the reconstructed 3D trajectories was used to generate the 3D trajectories out of the high-speed video trajectories. The depth information was interpolated by fitting a fourth degree polynomial to all the available sample points, i.e., 3D reconstructed trajectory points which

were available from the normal-speed trajectory. The fourth degree polynomial was selected experimentally by examining the fit error and the behavior of the trajectory past its end. The missing parts of the full trajectory depth information at the end of the trajectories were extrapolated by using the last known depth information.

5 Results

5.1 Measuring 3D Trajectories

The success rate of the finger tracking was measured by the proportion of trajectories which reached the predefined end points. For the high-speed videos, the end points were the touch target areas reprojected onto the image plane, and for the normal-speed videos, the defined end point was the triggerbox button. 77% of point actions were tracked correctly from the high-speed videos and 69% from the normal-speed videos. In total, 1161 (58%) of the pointing actions were correctly tracked from the both videos and were synchronized correctly.

To demonstrate the 3D reconstruction of trajectories, 19 pointing actions with 1127 corresponding tracked points were used. Trajectories used for the 3D reconstruction needed to be limited. Otherwise the inlier point selection for the reconstruction task would have been biased to the triggerbox location due to the slow movement speed in the start of the trajectory. This would have resulted in inaccurate reconstruction, and thus the limited set of pointing actions was used for the reconstruction.

The camera calibration was done with a standard checkerboard calibration target with a pattern consisting of 26.5 mm patches. A set of captured calibration images was used to compute the intrinsic camera parameters and distortion coefficients. 1127 finger trajectory points corrected for the lens distortion from the both normal and high-speed videos were used to form the basis of 3D-reconstruction (302 inlier points in the reconstruction). The reconstruction scale ambiguity was eliminated by a fixed setup with known distances between the high-speed camera (HS), normal speed camera (NL), and the monitor. The reconstruction results of all of the 1161 pointing actions are visualized in Fig. 4. The colors in the figure are just for the visualization purpose.

Since there was no ground truth for the 3D trajectories, the 3D reconstruction accuracy was assessed by using the re-projection error measure [7]. The mean re-projection error over all the trajectory points used from 1161 videos in the 3D reconstruction experiment was 31.2 pixels This corresponds to approximately 10 mm in the real world units. The 3D reconstruction results of all of the 1161 pointing actions are visualized in Fig. 4 with the X- (Fig. 4b), Y- (Fig. 4c) and Z-views (Fig. 4d). The test setup is visible in Fig. 4a.

5.2 3D Trajectory Analysis

Pointing actions from the initial 3D reconstruction results with velocity and acceleration curves from one block are visualized in Fig. 5. From the figure it can

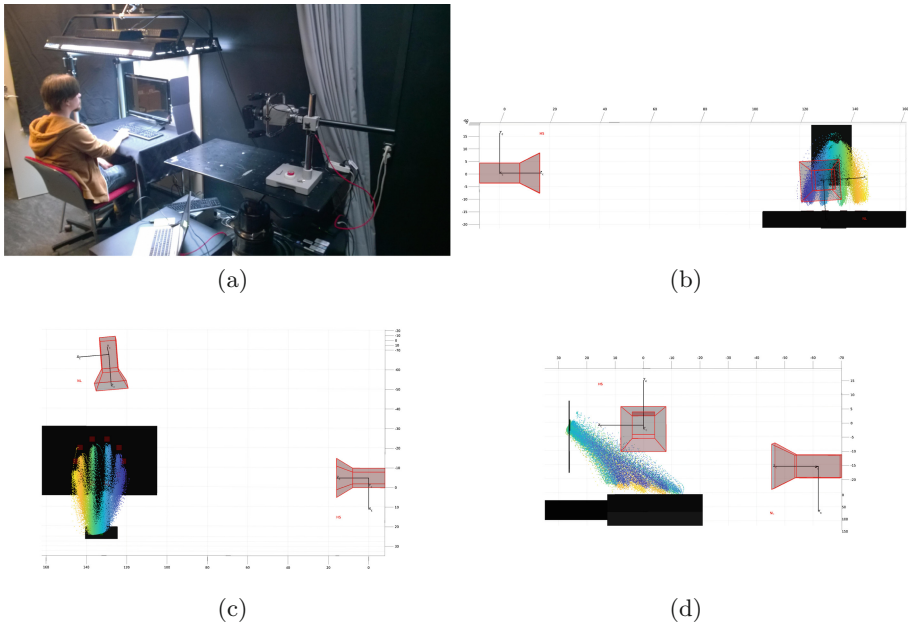


Fig. 4. Visualization of reconstruction results: (a) Setup of the experiments; (b) 3D reconstruction from x-view, and (c) from y-view and (d) from z-view. The trajectories include 1161 captured pointing actions displayed as colored dotted curves. (Color figure online)

be observed that the normal-speed camera did not capture the full trajectories of the hand movement. For the trajectory feature analysis, 3D trajectories reconstructed from the high-speed videos with the extrapolated depth-information were used as described in Sect. 4.

Eleven features were computed from the obtained trajectories: mean velocity, median velocity, maximum velocity, maximum 2nd submovement velocity, maximum 2nd submovement acceleration, mean 2nd submovement velocity, mean 2nd submovement acceleration, the point of the deceleration start, the point where the 2nd submovement acceleration starts for the first time, and the point where the 2nd submovement acceleration starts for the second time. Submovement intervals of the trajectories were detected similarly to [5]. The primary submovement started with the initial acceleration and ended when the acceleration went from negative values to positive values. This was the starting point of the secondary (2nd) submovement of intentional pointing actions where minor adjustments to the trajectory were made and the movement was fixed to the final target position.

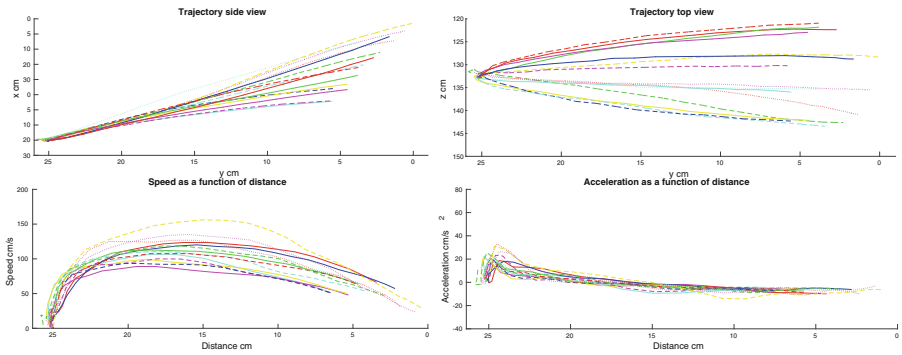


Fig. 5. Trajectory side and top views with velocity and acceleration features extracted from one block of pointing actions. One line style represents one pointing action.

A two sample T-test level was used to analyze the trajectory features. The result h from the test is 1 if the test rejects the null hypothesis at the 5% significance level, and 0 otherwise. It returns a test decision for the null hypothesis that the data in disparity pairs comes from independent random samples from normal distributions with equal means and unequal and unknown variances. Another two sample T-test was used to test the effect of equal, but unknown variances which it did not change the results. These results are visible in Table 1. Using the mean velocities over the whole trajectory in case of the disparity categories -2 and 6, and as well as with 6 and -6 shows that the mean velocities of the disparity categories -2 and 6, and 6 and -6 have unequal means. The null hypothesis is also rejected when using the maximum velocity over the whole trajectory with the disparity categories 2 and 6, as well as with 6 and -6, and with the maximum acceleration over the whole trajectory with the disparity categories 6 and -6. Moreover, using the mean acceleration of the 2nd submovement with the disparity categories 2 and 6, and as well as with -2 and 6 also rejects the null hypothesis.

The 2nd acceleration point of the 2nd submovement showed that the means of the feature in the disparity pairs -2 and 6, and as well as with -2 and -6 are statistically different. Moreover, after the tests with the disparity categories 2 and -2 or 2 and -6, neither could reject the null hypotheses with any of the used features. In Fig. 6 four example cases are shown where statistically significant differences were found in the means of the calculated features with different disparity pairs are shown. It can be observed that the differences are small, but still statistically meaningful.

6 Discussion

Because of the backtracking used for the normal-speed videos, the trajectories are most accurate near the monitor and least accurate near the trigger button. Moreover, it is the opposite situation for the high-speed trajectories which are initialized at the trigger button, meaning that they are most accurate at the trigger button and least accurate near the monitor. Moreover, the fact that the normal-speed videos are most accurate near the monitor and the high-speed videos are most accurate near the trigger button makes the inlier search more difficult for the 3D reconstruction task. The accuracy could be further improved with better tracking results and more careful planning of the test setup.

As expected, the smaller disparity changes 2 and −2 had only minor impact to the hand movements according to the computed features whereas the disparity values 6 and −6 had more significant impact to the movements. Moreover, the large positive disparity 6 (the target object in front of the screen) seemed to have a more prominent effect on the pointing actions than the others. It is shown in Table 1 that there are more features showing the trajectories to be statistically different when using the disparity 6 as one member of the disparity pair than any other disparity category. Furthermore, the velocity features seem to be better than the acceleration features to distinguish the pointing actions towards different disparity values.

Table 1. Rejection of the null hypothesis for the disparity pairs using the selected features.

Features	Disparity pairs						Total
	2 −2	2 6	2 −6	−2 6	−2 −6	6 −6	
Mean velocity	0	0	0	1	0	1	**2**
Median velocity	0	0	0	0	0	0	**0**
Maximum velocity	0	1	0	0	0	1	**2**
Maximum acceleration	0	0	0	0	0	1	**1**
Maximum 2nd submovement velocity	0	0	0	0	0	0	**0**
Maximum 2nd submovement acceleration	0	0	0	0	0	0	**0**
Mean 2nd submovement velocity	0	0	0	0	0	0	**0**
Mean 2nd submovement acceleration	0	1	0	1	0	0	**2**
Deceleration start point	0	0	0	0	0	0	**0**
2nd submovement start 1st	0	0	0	0	0	0	**0**
2nd submovement start 2nd	0	0	0	1	1	0	**2**
Total	**0**	**2**	**0**	**3**	**1**	**3**	**9**

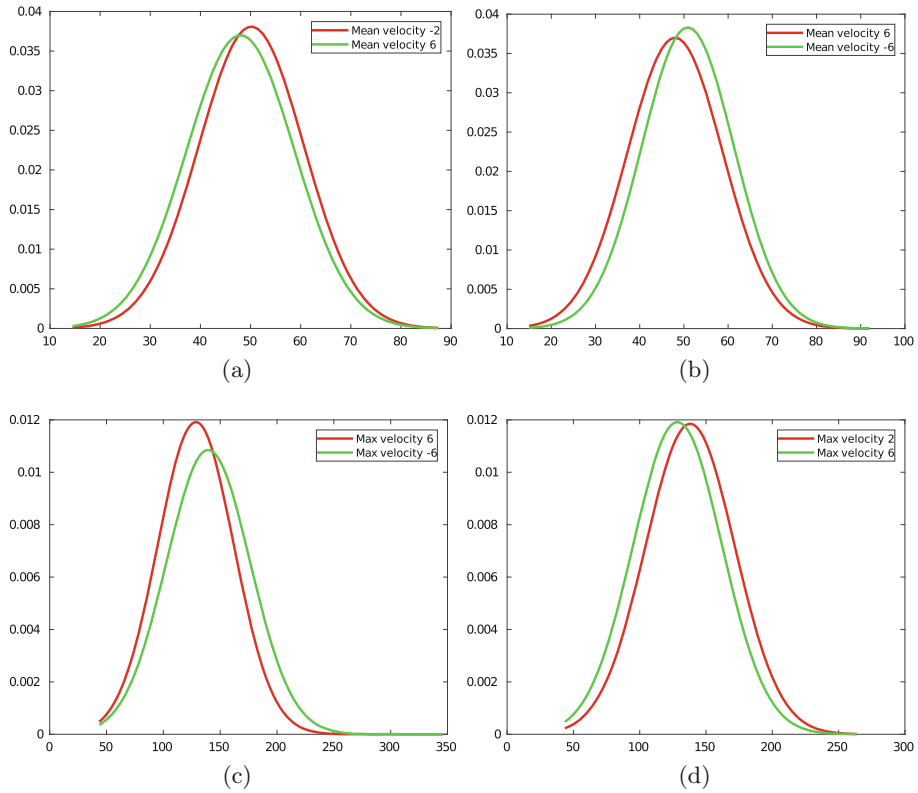


Fig. 6. Normal distribution plot examples where there was statistically significant differences in means of the data. (a) Mean velocity with disparities −2 and 6, (b) Mean velocity with disparities 6 and −6, (c) Maximum velocity with disparities 6 and −6, and (d) Maximum velocity with disparities 2 and 6.

7 Conclusion

In this work, a two-camera framework for tracking finger movements in 3D was evaluated with a large-scale study of human-computer interaction. Moreover, trajectory and video synchronizing processes were introduced and 3D trajectory reconstruction was proposed. The proposed framework was successfully evaluated in the application where stereoscopic touch screen usability was studied with the stereoscopic stimuli.

Overall, the depth information gathered from 3D reconstruction task resulted in full high-speed trajectories with good depth estimation data from the interpolated results of the 3D reconstruction task. Some feature correlation with different parallaxes were already detected, but deeper analysis of the effects of different parallaxes on the trajectories is planned for the future research.

Acknowledgments. The authors would like thank Dr. Jukka Häkkinen and Dr. Jari Takatalo from Institute of Behavioural Sciences from University of Helsinki for producing the data for the research. Their valuable and constructive contributions during the previous steps of the research are very much appreciated.

References

1. FFmpeg (2017). https://ffmpeg.org/. Accessed 1 May 2017
2. van Beurden, M.H., Van Hoey, G., Hatzakis, H., Ijsselsteijn, W.A.: Stereoscopic displays in medical domains: a review of perception and performance effects. In: IS&T/SPIE Electronic Imaging, p. 72400A. International Society for Optics and Photonics (2009)
3. Chan, L.W., Kao, H.S., Chen, M.Y., Lee, M.S., Hsu, J., Hung, Y.P.: Touching the void: direct-touch interaction for intangible displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2625–2634. ACM (2010)
4. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**(403), 596–610 (1988)
5. Elliott, D., Hansen, S., Grierson, L.E.M., Lyons, J., Bennett, S.J., Hayes, S.J.: Goal-directed aiming: two components but multiple processes. Psychol. Bull. **136**(6), 1023–1044 (2010)
6. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: a review. Comput. Vis. Image Underst. **108**(12), 52–73 (2007). Special Issue on Vision for Human-Computer Interaction
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2004)
8. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
9. Kooi, F.L., Toet, A.: Visual comfort of binocular and 3D displays. Displays **25**(2), 99–108 (2004)
10. Kuronen, T.: Post-processing and analysis of tracked hand trajectories. Master's thesis, Lappeenranta University of Technology (2014)
11. Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J., Kälviäinen, H.: High-speed hand tracking for studying human-computer interaction. In: Paulsen, R.R., Pedersen, K.S. (eds.) SCIA 2015. LNCS, vol. 9127, pp. 130–141. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19665-7_11
12. Lyubanenko, V., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H., Häkkinen, J.: Multi-camera finger tracking and 3D trajectory reconstruction for HCI studies. In: Blanc-Talon, J., Penne, R., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2017. LNCS, vol. 10617, pp. 63–74. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70353-4_6
13. Sampson, P.: Fitting conic sections to "very scattered" data: an iterative refinement of the Bookstein algorithm. Comput. Graph. Image Process. **18**(1), 97–108 (1982)
14. Servos, P., Goodale, M.A., Jakobson, L.S.: The role of binocular vision in prehension: a kinematic analysis. Vis. Res. **32**(8), 1513–1521 (1992)
15. Valkov, D., Giesler, A., Hinrichs, K.: Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In: Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces, ITS 2012, pp. 21–30. ACM, New York (2012)
16. Vojir, T.: Tracking with kernelized correlation filters (2017). https://github.com/vojirt/kcf/. Accessed 1 May 2017

Publication IV

Tamminen, J., Lahdenperä, E., Koiranen, T., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H.

Determination of single droplet sizes, velocities and concentrations with image analysis for reactive extraction of copper

Chemical Engineering Science
Vol. 167, pp 54–65, 2017.

Chemical Engineering Science 167 (2017) 54–65

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Determination of single droplet sizes, velocities and concentrations with image analysis for reactive extraction of copper

Jussi Tamminen*, Esko Lahdenperä, Tuomas Koiranen, Toni Kuronen, Tuomas Eerola, Lasse Lensu, Heikki Kälviäinen

School of Engineering Science, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland

HIGHLIGHTS

- Single organic droplet copper extraction was measured using funnel and imaging.
- Droplet inner concentrations were determined from analysis of video frames.
- Results include concentration profiles and average concentrations in droplet.
- Funnel measurement leads to higher mass transfer due to indirect measurement.
- Dynamic droplet analysis is enabled using velocity, shape and concentration.

ARTICLE INFO

Article history:
Received 3 January 2017
Received in revised form 4 March 2017
Accepted 21 March 2017
Available online 23 March 2017

Keywords:
Copper extraction
Single droplet extraction
Image analysis
Video analysis
Photometric concentration analysis

ABSTRACT

The proposed image analysis method allows the measurement of organic phase droplet sizes, velocities, and copper concentrations in single droplet column copper extraction using hydroxyoxime complexation. The method uses image acquisition sequences from video, detection of moving droplets, binarization of background subtracted images, and noise reduction from images. The image analysis method enabled characterizing the shape of droplets, by determining the droplet minor and major axis lengths. The method can detect droplet concentration directly inside the column wherever the droplet is visible. Image based method was validated against reference samples which were analyzed using spectrophotometry. The traditional concentration measurement using the spectrophotometric analysis of column outlet sample collection was performed for comparison purposes. The direct image analysis showed smaller variation in mass transfer results because of longer and non-uniform residence times when using sample collection. However, separately collected sample analysis together with the image analysis enables determination of the copper mass transfer during all the three steps of column experiment. Image analysis can also be used to reveal concentration profiles inside the droplet. This method is not limited to extractants, but it can be applied to systems where a suitable color change is present depending on camera sensor technology.

1. Introduction

Mass transfer is an important phenomenon affecting the design of liquid-liquid contactor units utilizing reactive extraction. To properly design the units, it is important to understand and quantitatively evaluate the effect of different mass-transfer phenomena in the whole process. These phenomena are solute transfer from bulk to the interface, interfacial reaction, and transport from interface to bulk. The mass transfer in solvent extraction depends on, among other variables, droplet sizes, velocities, and concentrations. When the size, velocity, and inner concentration of a single droplet are determined, mass transfer into the droplet is defined.

The presence of suitable reagents enhances mass transfer between the continuous and droplet phases in the reactive extraction. Especially in industrial hydrometallurgical processes, metal extraction with complex forming extractants is in common use. Also substantial application areas of reactive extraction can be found among environmental, petrochemical, chemical, and biochemical applications (Bart and Stevens, 2004).

To experimentally investigate combined interfacial kinetics and mass transfer, different experimental methods are available (Hanna and Noble, 1985). Among these methods, single droplet measurements are widely applied in mass transfer experiments

* Corresponding author.
E-mail address: Jussi.V.Tamminen@lut.fi (J. Tamminen).

http://dx.doi.org/10.1016/j.ces.2017.03.048
0009-2509/

Nomenclature

A	absorbance, [−]
A, B, C, D, E, F	parameters of quadratic formula
c	concentration, [mol/L, mmol/L]
d	diameter [mm]
E	droplet aspect ratio [−]
g	earth gravitational acceleration [9.81 ms^{-2}]
I	light intensity, [−]
L	optical path length, [mm]
l	length or distance, [mm]
n	amount of copper, [mol, mmol]
p	pixel value, [−]
u	droplet velocity, [mm/s]
V	volume, [mL]
$\dot{V}$	droplet phase feed flow rate, [mL/min]
X	conversion ($X = 1 - c/c_0$), [−]

Greek alphabet

$\Delta\rho$	density difference ($\rho_c - \rho_d$), [kg/m^3]
Δc	concentration difference, [mmol/L]
ρ	density, [kg/m^3]
γ	interfacial tension, [mN/m]
ε	molar absorptivity, [L/(mmol mm)]
μ	dynamic viscosity [Pa s]

Subscripts, indices

0	initial value
aq	aqueous phase
bg	background
BOT	column bottom part
c	continuous phase
ch	chord length
cr	critical value
d	droplet or droplet phase
e	equivalent
i, j	pixel location indices
major, minor	major and minor axis of a droplet image
org	organic phase
p	value for pixel
Rise	rise
Sample	analysis from sample
t	terminal velocity
TOP	column top part

Dimensionless numbers

Eo	Eötvös number, $\mathrm{Eo} = g\Delta\rho d^2/\gamma$
Mo	Morton number, $\mathrm{Mo} = g\mu_c^4\Delta\rho/(\rho_c^2\gamma^3)$

of liquid-liquid systems to determine the mass transfer coefficients, interfacial kinetics and extraction efficiencies (for example, Whewell et al., 1975; Henschke and Pfennig, 1999; Kumar and Hartland, 1999; Biswas et al., 1996, 1997; Wegener et al., 2009).

In single droplet systems, a droplet is rising or settling in an ambient continuous liquid. Droplets are collected from a funnel at the column outlet and concentrations are analyzed. Reaction kinetics and mass transfer rates can be determined from this data.

Already in the 1950s, Licht and Conway (1950) and Licht and Pansing (1953) verified that the mass transfer in single droplet extraction is divided into three stages: mass transfer during droplet formation, mass transfer in free rising/settling, and mass transfer during droplet coalescence. Experimental arrangements should be made so that contribution of each phenomena to the extraction process can be determined. It is commonly agreed, that the contribution of droplet formation time to the mass transfer can be substantial, and the related error should be taken into account in the formation of mass transfer correlations (Wegener et al., 2014; Liang and Slater, 1990; Licht and Conway, 1950; Licht and Pansing, 1953). By contrast, the effect of droplet coalescence in the column outlet collector is assumed to be negligible, which has not been clearly shown.

Traditional single droplet experiments do not provide any information on the conditions inside the droplet during its rise. For example, mass transfer leads to concentration changes inside the droplet and at the interface. These changes can generate interfacial tension gradients which in turn lead to the Marangoni convection (Wegener et al., 2009, 2014). The effect of Marangoni convection cannot be directly observed in pure concentration measurements. Because of this, it would be beneficial also to be able to follow droplet velocities and concentration profiles within the droplet, at the interface and in the near vicinity of the droplet in the ambient phase. Flow pattern and concentration front visualization inside a droplet using decolorization with pH indicator have been made by Schulze (2007) and Pawelski et al. (2005) but the concentration profiles have not been measured. Decolorization, however, has been used to reveal Marangoni convection. Mörters and Bart (2000) and Baumann and Mühlfriedel (2002) have determined indirectly concentration profiles near the phase boundary using a laser induced fluorescence to track tracer concentrations. Baumann and Mühlfriedel measured time-dependent average tracer concentration profiles on the flat interface between two immiscible liquids. Mörters and Bart (2000), using D2EHPA system, determined time-dependent tracer concentration profiles inside a droplet to investigate diffusion inside and outside droplets in reactive extraction. Measured tracer concentration profiles were used as a basis to determine organic complex diffusion coefficient. The determination the effect of continuous phase flow on the droplet internal circulation was not successful due to experimental arrangements. In further studies by Mörters and Bart (2003), the measured concentration profiles were a basis for a Stefan-Maxwell based diffusion model for the mass transfer. Diffusion model was applied to zinc-D2EHPA single droplet experiments but the model was not able to describe experimental results satisfactorily and this is probably due to convective effect not included in the model.

In this work, the problems in single droplet extraction experiments are approached with a direct nonintrusive measurement system where the droplet velocity, droplet diameter, and concentration inside the droplet are determined by using digital imaging and subsequent image analysis. In this research, copper extraction from the aqueous solution to the organic solvent using Acorga M5640 extractant is analyzed. The interfacial reaction,

$$Cu^{2+}(aq) + 2HA(org) \rightleftarrows CuA_2(org) + H^+(aq), \quad (1)$$

where the reactant HA (Acorga M5640) exchanges Cu-ions from the aqueous phase and the Cu-complex CuA_2 can be followed directly and visually due to color change.

In experiments, concentrations, droplet velocities, and diameters are determined as averages from several droplets to minimize the effect of experimental variability. This direct droplet concentration analysis allows exact determination of mass transfer rates during the three stages in the single droplet experiment. In tradi-

tional single droplet experiments, droplet rise time have to be controlled by adjusting the difference between the droplet feed and collection locations in order to determine, for example, the mass transfer during the droplet formation period by extrapolation. Also the assumption of the negligible effect of a coalesced droplet phase residence time in the funnel on the total mass transfer can be tested. The method provides also droplets inner concentration profiles and this reveals the inside circulation. The direct concentration analysis can be combined with numerical models and this leads to deeper understanding of mass transfer in reactive extraction and provides a basis of better equipment design. Other geometries than droplets, such as planar interfaces, can be measured if optical path length is determined. Other applications than extraction where suitable color changes exist, can be tracked with this method. The details of experimental procedure are presented in Section 2, where the experimental set-up, experiments in the column, and the analysis methods are shown in Sections 2.1–2.5. Results from experiments and discussion are combined as Section 3. The calibration, droplet size, concentration and velocity results are discussed in Sections 3.1–3.3. To our knowledge the direct spatial quantitative concentration measurement from droplets has not been published previously.

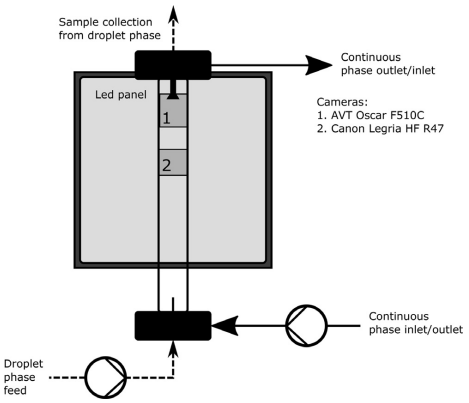


Fig. 1. The experimental set-up for column experiments with rising single organic droplets. Approximate field-of-view positions of the cameras for column top measurements are marked with rectangles.

2. Experiments

2.1. Preparation of feed solutions

The extractant (Acorga M5640 by Cytec Solvay Group, Lot n:o P3GBA524A) was contacted twice with 0.1 mol/L sulfuric acid and once with 0.1 mol/L ammonium sulfate solution prior its use. Equal volumes of organic and aqueous solutions were used. This was made in order to pre-equilibrate extractant and remove remaining soluble impurities from extractant. Finally, extractant was diluted with Exxsol D80 (by Exxon) to 10 vol% and 20 vol% solutions. The extractant active component concentration was measured by titration (Mettler Toledo T50 automatic titrator).

The equilibrated organic copper complex standard solutions were prepared by mixing 30 min the feed solutions with different copper concentrations, and 10 or 20 vol% Acorga solutions ($V_{aq}/V_{org} = 1:1$). The copper content of the aqueous phase was analyzed with a spectrophotometer (Agilent 8543), and organic copper concentrations were calculated from mass balance. Organic solutions were used as standards for both the spectrophotometric analysis and the image analysis.

The copper sulfate solutions were prepared by dissolving copper sulfate ($CuSO_4{\cdot}5H_2O$, Merck, Pro analysis) into water. The pH of solution was adjusted to 3.1 with concentrated sulfuric acid.

2.2. Experimental set-up

The single organic droplet extraction experiments were made in a glass column (45 mm × 45 mm × 375 mm) filled with continuous aqueous phase. The droplets were formed at the flat tip of a 0.8 mm steel needle (nominal inner diameter 0.51 mm) at the bottom of the column using a high precision syringe pump. The droplet and column were back-illuminated with a led panel (300 mm × 300 mm, 35 W, color temperature 3000 K). The measured droplet sizes are introduced in Fig. 10. The droplets were collected at the other end of the column using a small funnel as shown in Fig. 1.

Two milliliter samples from the rising droplets were collected. Copper concentrations of the organic phase samples were directly analyzed with the spectrophotometer (Agilent 8543). The analysis was made using the absorption at 600 nm for the organic solution and at 811 nm for the aqueous solution samples. The wavelengths were selected based on the sample measurements with a spectrophotometer.

The droplet velocities and sizes were determined by analyzing the videos recorded with a Canon Legria HF R47 camera using a framerate of 50 frames per second. The concentrations inside the droplets and droplet sizes were determined from videos recorded with a AVT Oscar F-510C camera using a framerate of 7.5 frames per second. This second camera with a smaller field-of-view was used to capture more detailed images of the droplets and for more accurate representation of color information.

2.3. Column experiments

The column was filled with the continuous phase. The droplet phase was continuously pumped through the needle into the column. Flow rates of 0.1–1.0 mL/min were used in experiments. The average droplet detachment rate increased from about 4 droplet/min, to 35 droplet/min as a function of the feed flow rate. At first, the droplet phase flow rate was set to be at the minimum (0.1 mL/min) so that the droplet formation was slow enough for having a single droplet in the column. Droplets were formed until enough droplets were collected for the analysis (approximately 2 mL). Samples were collected manually using syringe in two or three batches in order to minimize sample residence time in the funnel. The average sample residence times decreased from 420 s to 60 s when flow rate increased from 0.1 to 1.0 mL/min. The funnel samples contained both phases and the phases were separated just after sampling. The samples were later analyzed with the spectrophotometer. Experiments were then repeated using higher feed rates. It is acknowledged here, that it is possible to optimize sampling procedure. However, sample residence time in the funnel cannot be removed completely as droplet coalescence is not an instantaneous process. Samples include also mass transfer over the phase interface at column outlet.

The concentrations of droplets were separately recorded at the bottom of the column (just after the droplet was detached from the tip of the needle), and at the top of column (just before the droplet entered the column outlet funnel). The results from the bottom of the column revealed the mass transfer into the droplet during its formation, whereas the results from the top of the column included the mass transfer into the droplet during its formation and rise.

The effect of rise on the mass transfer into the droplet could be determined by comparing the measured concentrations in the column bottom and top.

2.4. Image analysis

The first step in the image-based droplet analysis was to acquire image sequences using the setup described in Section 2.2. The AVT Oscar F-510C FireWire camera was used to obtain the image sequences (stored as uncompressed image files) for the concentration analysis and droplet size measurements. The Canon Legria HF R47 camera was used to obtain videos for the droplet size, velocity, and acceleration measurements. The second step was to detect moving droplets in the videos. Since the background (the column with the continuous phase) was static except for the moving droplets, the detection problem was solved using a background subtraction method. The background subtraction was performed by subtracting the previous frame in the image sequence from the input frame. This way the regions, where the subsequent frames differ due to the movement of droplets, were separated from the background (Cheung and Kamath, 2005).

The background subtracted image was then binarized by selecting an appropriate threshold value for the red color channel. The red channel was selected based on preliminary experiments where it was found to provide the most robust information for the task. The threshold value was automatically selected by using Otsu's method (Otsu, 1979) that assumes that the image contains foreground (object) and background pixels, and calculates an optimal threshold so that the intra-class variance is minimal.

The resulting binary image typically contains noise. To eliminate the noise, connected components (connected regions of the foreground pixels) were determined and the components smaller than 10% of the predefined minimum droplet area were removed. The resulting binary images were further processed with morphological erosion using a small structuring element to remove small irregularities in the droplet edge regions. The structuring element is a binary area where black pixels (0 pixels) are excluded, and white pixels (1 pixels) are included in the morphological computation. After this step, the binary images contained at least fragments of the droplet edges.

Depending on the concentration, some of the detected regions contained holes. Moreover, with some concentrations the droplet edges were only partly visible.

To obtain full contours of the droplets, fitting an ellipse to the binary image data was used. The ellipse fitting was selected based on an assumption that the shape of droplets is oblate spheroid which causes the droplets to have an elliptical shape in the images. The ellipse fitting provides a good estimate for the true contours of the droplets and produces more reliable droplet shape parameters than using only binary descriptors of the droplet region (Szpak et al., 2012). The limits for the ellipse parameters were used to control the droplets, which were processed further and the ones which were discarded if they did not meet the limits. The full image analysis pipeline is presented in Fig. 2. The image at the top left corner shows the region of interest (ROI) (the blue rectangle) used for processing. ROI was manually selected after visual inspection of the videos to contain all the droplets moving with minimum processing area.

The basic image processing pipeline from the beginning to the ellipse fitting step was the same for the both imaging sources. After this, different feature extraction operations were performed for the Canon Legria HF R47 videos (size, velocity, and acceleration measurements) and the AVT Oscar videos (size and concentration analysis). Velocity and acceleration of the droplets were not determined from AVT Oscar videos since the lower frame rate, only 3 or 4 captured images from one moving droplet, did not allow accurate measurements. Features extracted from the Canon Legria HF R47 videos included the minor and major axis lengths, orientation, velocity, and acceleration of the droplet. The ellipse minor and major axis lengths are

$$d_{minor}|d_{major} = \left[\frac{2(AE^2 - BDE + CD^2) - 4ACF + FB^2}{(4AC - B^2)(A + C) \pm \sqrt{(A - C)^2 + B^2}}\right]^{\frac{1}{2}}. \quad (2)$$

where | denotes the logical or (disjunction) operator (the greater of the two values given by the equation is the major axis) and A, B, C, D, E, and F are the parameters of the general quadratic curve,

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (3)$$

(Weisstein, 2016). These parameters were obtained using the ellipse fitting. The velocity was calculated from the movement of the droplet center point between the consecutive frames and the acceleration as the difference in velocities between the consecutive frames.

In order to convert the pixel values of the features into the real world values (millimeters), the cameras were geometrically calibrated. To calibrate the camera, a sequence of images with a 5 mm grid was captured to provide the points of reference. The calibration was performed using the methods presented in (Heikkila and Silvén, 1997; Zhang, 2000). The process included detecting the grid from the image sequence and calculating the pairwise distances of grid lines. The collected information was used to form corrected, distortion free images. The pixel size in mm was obtained by dividing the real world grid size (5 mm) with the average distance for adjacent grid lines in images.

The concentration calculations were performed using the absorbance in the red color channel as explained in Section 2.5. The droplet geometry was taken into account by making chord length calculations assuming that the shape of droplets was oblate spheroid. This geometry was calculated from the parameters of the fitted ellipse. 15% of the droplet image area, the outer edge region of the droplet, was excluded from the concentration analysis since the light scattering at the droplet phase boundary caused problems when calculating the concentrations. Simplified process flow of the concentration analysis is shown in Fig. 3. Image analysis steps were implemented using MATLAB (2016a). The concentration analysis is described in more detail in Section 2.5.

2.5. Concentration analysis

The concentration analysis inside a droplet is based on the observation of image intensity change inside the sample. The concentration analysis is possible with imaging because of color change, which takes place when copper is complexed with hydroxyoxime. This color change is observed from the video recording of moving droplets by a digital camera.

The spectra recorded with the spectrophotometer revealed that free hydroxyoxime and its copper complex absorb light at different wavelengths (Fig. 4). The highest wavelength at which the free extractant absorbs light is approximately 400 nm whereas the copper complex absorbs also at longer wavelengths. The complex has a wide absorption peak with the absorption maximum at approximately 680 nm. The concentration analysis of organic phase samples with the UV/VIS-spectrophotometer (Agilent 8543) was made at the wavelength of 600 nm. The wavelength of 811 nm was used for the aqueous copper sulfate solution concentration analysis.

The concentration analysis with both the spectrophotometer and the cameras are based on the Lambert-Beer law. The Lambert-Beer or Bouguer-Beer law describes how, in a transmission measurement, the light intensity decreases as a function of the sample concentration and light path length. Each recorded

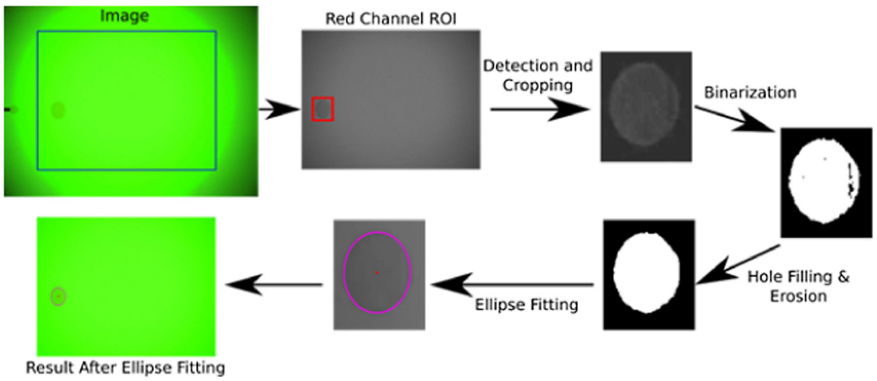


Fig. 2. Image analysis steps for determining the droplet movement, size and concentration. Contrast of the color images on the top row and the difference image has been enhanced for visualization purposes.

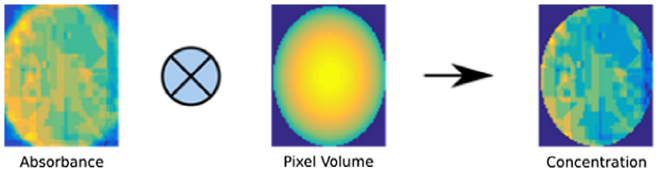


Fig. 3. Simplified process flow of the concentration analysis. The 'logical and'-symbol is used to represent the major contributing factors (absorbance and volume) to the concentration determination in simplified form.

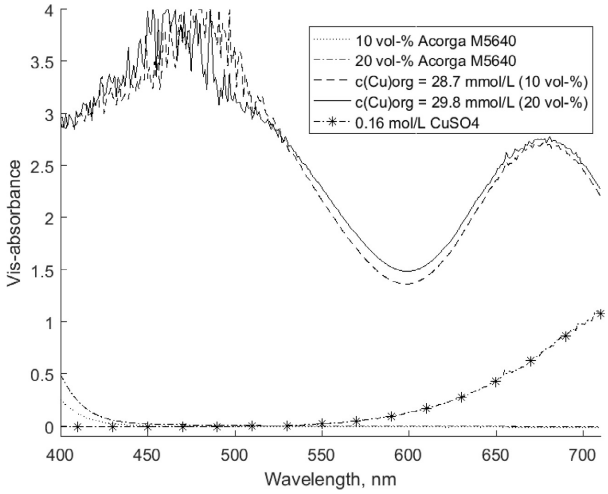


Fig. 4. The basis for detecting the copper complex. The red channel of camera is most sensitive for wavelengths from 550 nm to 700 nm. The visible-range spectra of organic Acorga M5640 solutions dissolved into Exxsol D80 and aqueous copper sulfate solution.

pixel of the droplet image represents a part of droplet. These droplet parts have a volume, here called as the pixel volume. The concentration in the pixel volume may not be uniform along the droplet chord (perpendicular to the image plane) as droplets were imaged only from a single direction. Uniform concentration was assumed for each pixel volume. The Lambert-Beer law and its derivation are presented and discussed by Berberan-Santos (1990), Goldstein and Day (1954) and Liebhafsky and Pfeiffer (1953) and Denney and Sinclair (1987). The Lambert-Beer equation is presented as

$$A = \varepsilon L c \tag{4}$$

where ε is molar absorptivity, L the optical path length, c is concentration and A is the absorbance:

$$A = \log_{10}(I_0/I) \tag{5}$$

that is, the logarithmic ratio of light intensity transmitted through the sample (I) and incoming light intensity (I_0).

The camera used for the concentration analysis (AVT Oscar F-510C) is a RGB camera with 8-bit channels. The approximate range, where red channel of the camera detects light, is 550–700 nm (Fig. 4). The red channel was found to be specific only for the copper complex of extractant, and not for the uncomplexed extractant. It is also noted that copper sulfate absorbs light at the wavelength range of the red channel. The red channel background is assumed to be constant, as the continuous phase volume is much higher compared to the droplet volume and thus the amount of copper and the copper concentration in the continuous phase remains practically constant during an experiment.

The camera records incoming light intensity and encodes the intensities as pixel values in the range of 0–255 (8 bits). The pixel values are assumed to be linearly dependent on the incoming light intensity within the usable dynamic range of the camera. Analogous to Eq. (5), absorbance is determined by:

$$A = \log_{10}(p_0/p), \tag{6}$$

where the incoming light intensity is represented by the pixel value p_0 and the light transmitted through the sample with the pixel value p (compare to Eq. (5)).

The camera records a droplet as it moves through the column. The spatial uniformity of the light source is even, and the pixels of the camera are assumed to be identical. Both droplet and background are visible simultaneously in recorded images, and thus the situation is similar to the double-beam spectrophotometer where the light beam is divided into two beams, and one travels through the sample and another through the solvent (i.e., background) (Mann et al., 1974). As the background of the image, i.e., the continuous phase, absorbs light also in the red channel used for the concentration analysis, the background is corrected by subtracting the background absorbance (A_{bg}) from the absorbance of the droplet image (A_d):

$$A = A_d - A_{bg}. \tag{7}$$

The droplet concentration can be calculated from absorbances, when droplet geometry is taken into account. The optical path length, i.e., the distance which light travels in a sample effects absorbance as it can be seen from the Lambert-Beer law (Eq. (4)). The chord lengths of the droplet at each pixel position were calculated by assuming that the droplet was oblate spheroid and that the axis normal to the image plane is equal to the measured major axis length.

The effect of light scattering was determined by calculating molar absorptivities (ε) for each pixel of single droplet image using the Lambert-Beer law (Eq. (4)). The absorptivities are shown as a function of calculated chord lengths in Fig. 5b. The edge of a droplet is darker than the rest of the droplet, which is due to light refraction and scattering. The apparent absorptivity increases towards the droplet edge and it includes both the effect of light absorption due to the copper complex and the effect of scattering. While there are correlations for the light intensity changes due to scattering (Gumprecht and Sliepcevich, 1953), the use of apparent absorptivity was chosen for this work, as the present camera detects a wide range of wavelengths. The applied interpolation function for the apparent absorptivity was

$$\varepsilon = 3.786 \times 10^{-3}/\ln(l_{ch} + 1) + 0.8977 \times 10^{-3}, \tag{8}$$

where l_{ch} is the chord length. The function parameters were fitted to the droplet data and the function was used in the calculation of concentration profiles and the average concentrations of the droplets.

When the optical path length of the Lambert-Beer law (L in Eq. (4)) is set equal to the calculated chord length and the absorptivities are calculated with Eq. (8), the concentrations at the positions of the pixels (c_p) can be calculated from the image data based absorbances and the Lambert-Beer law (Eq. (4)):

$$c_p = A/\varepsilon l_{ch} \tag{9}$$

The calculated chord lengths (l_{ch}) together with the image scale, i.e., the distance per pixel (l_p), are used when the representative volume of each pixel is calculated:

$$V_p = l_p^2 l_{ch}, \tag{10}$$

where l_p is the distance per pixel. The amount of copper (n_p) at the position of each pixel, can be calculated from the volume and the measured concentration of the pixel:

$$n_p = c_p V_p \tag{11}$$

When the sum of the values of n_p of the whole droplet image is determined, the amount of copper transferred into the droplet can be calculated with

$$n = \sum_{i=0}^{d_{major}} \sum_{j=0}^{d_{minor}} n_p(i,j) \tag{12}$$

where i and j are indices of pixel location along the droplet image axes.

The average concentration of copper transferred into the droplet is

$$c = n/V_d, \tag{13}$$

where V_d is the droplet volume. The organic phase copper concentration is presented as a conversion of extractant, as organic feed solution includes the extractant and no copper.

The conversion (X) of extractant (HA) is calculated (Levenspiel, 1999) by:

$$X_{HA} = 1 - c_{HA}/c_{HA,0} \tag{14}$$

The extractant concentration is calculated using the extractant mass balance:

$$c_{HA} = c_{HA,0} - 2c \tag{15}$$

Copper reacts with two extractant molecules to form the complex, as also it can be seen from the reaction (Eq. (1)).

The calibration standards (see Fig. 5a) for the single droplet experiments were imaged as droplets in the column. The apparent molar absorptivity was calculated for each pixel of a one droplet image (cCu_{org} = 25.4 mmol/L). The interpolation function (continuous line) and ε obtained from droplet center data (dotted line) in Fig. 5b are also shown. The continuous line represents the interpolation function of apparent molar absorptivity and the dotted line denotes the value of ε obtained from the fit. The scattering is at minimum at droplet center, as can be seen from Fig. 5b.

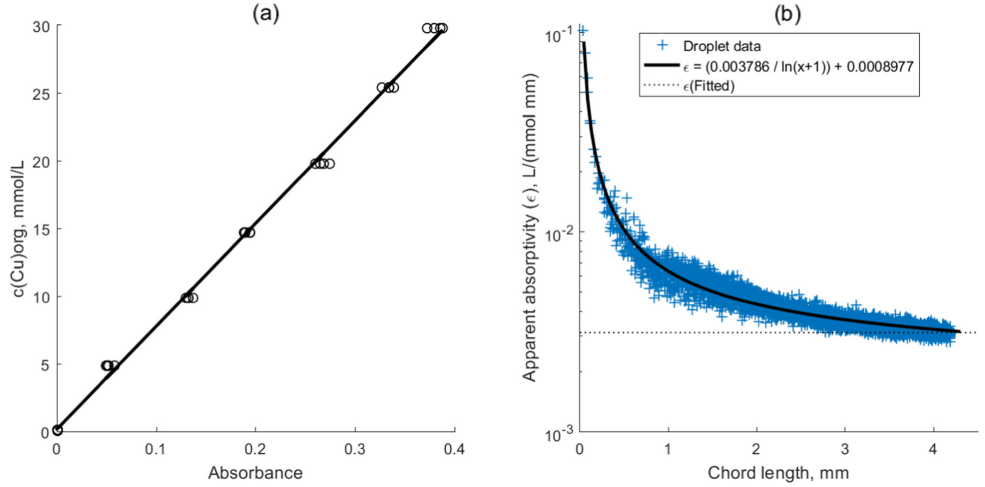


Fig. 5. The calibration of copper complex concentration for organic phase droplet image analysis with the AVT Oscar F-510C camera. (a) The fitted calibration line for droplet center data and (b) the effect of light scattering due to droplet interface curvature on molar absorptivity.

The calibration standard solutions of 20 vol% Acorga M5640 containing different copper concentrations were measured from the 0.8 mm needle tip using the aqueous copper sulfate solution (0.16 mol/L) as the continuous phase. The calibration line (Fig. 5a) was computed based on the center region of the droplets where the effect of light scattering is minimized. The copper complex standard solutions of 10 vol% Acorga M5640 in 0.06 M $CuSO_4$ continuous phase were measured for the comparison and verification purposes. The organic standard solutions were pumped ($\dot{V} = 1.5$ mL/min) through the needle into the column filled with the copper sulfate solution. The copper concentration in organic standard solution was varied between 0 and 30 mmol/L. The standard solution analysis with the spectrophotometer revealed that calibration lines of the both extractant concentrations were essentially the same.

The similar observations can be seen in Fig. 6, where samples with known copper concentrations were analyzed. The concentration in the droplets was analyzed using the same method and the calibration line as in the actual column experiments. The results at two different extractant concentrations are overlapping, indicating that also in the image analysis the calibration lines are practically the same.

Each droplet was detected several times from the subsequent video frames. The median value of these separate detections was used as the representative value for the droplet. The subsequent droplets are here considered as repetitive measurements. The χ^2 goodness of fit test for the normality was performed. The χ^2 test was passed in the case of elliptical droplet major and minor axis lengths, velocity, and concentration, that is, the data is normally distributed. Standard deviation of each dataset was chosen as the basis for quantifying variation in the measurement data. The measure for the error in this work is three times dataset standard deviation. The errors are approximately 1% for the droplet minor and major axis lengths, approximately 3% for the droplet velocity and approximately 22% for the droplet concentration in mmol/L (Fig. 7). The measured data is shown as frequency histograms and the calculated normal distributions as lines. The parameters of the normal distribution were estimated based on the data. The data analysis was performed using MATLAB (2016a).

The theoretical detection limit for the proposed method is estimated by assuming that copper complex concentration in the droplet is uniform. The smallest possible detectable concentration change corresponds to pixel value p change from 255 to 254. Then the absorbance is 1.7×10^{-3}. The concentration can be calculated from this using the calibration line (Fig. 5). Thus, the smallest detectable concentration is approximately 0.15 mmol/L. The method accuracy and reproducibility are presented in Fig. 6. The method reproducibility (i.e., the distribution width of the measured data at each standard solution concentration) is approximately 2 mmol/L. The accuracy of method is defined as the difference between the average of the measured data and the nominal standard solution value. The accuracy of method may be possible to improve since there are some differences between the measured and nominal concentrations (the line in Fig. 6). The method standard line is linear in the concentration range 0–30 mmol/L (Fig. 5a).

The instrument bandwidth affects recorded absorbance values (Denney and Sinclair, 1987). In case of the spectrophotometer, the spectra was recorded with interval of 1 nm. The absorbance peak of complex is at least 150 nm wide (see Fig. 4) and the ratio of instrument and sample spectral bandwidths are below 0.01 in case of the spectrophotometer. The bandwidth of the red channel of the camera is close to 150 nm. The ratio of the instrument and sample spectral bandwidths is thus approximately 1.

3. Results and discussion

3.1. Analysis method

The camera calibration line forms a straight line (see Fig. 5a). Thus the Lambert-Beer law is obeyed also in determining the concentration from the droplet image data. The molar absorptivity coefficients obtained for the 10 vol% and 20 vol% Acorga standard

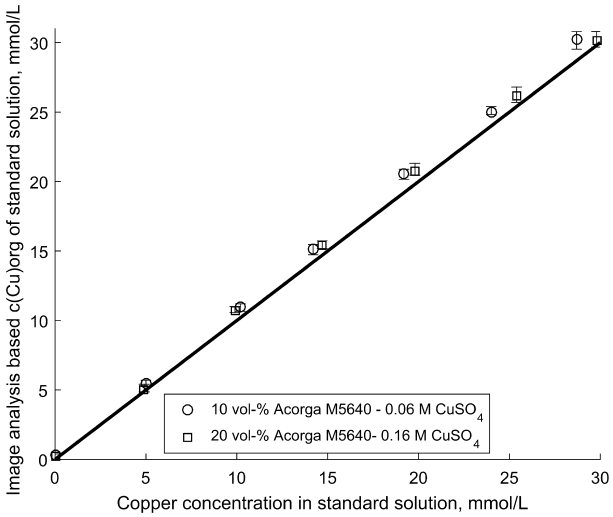


Fig. 6. Comparison of known standard solution concentrations and concentrations from image analysis. Mean of measured data (symbols) and variation of data (error bars) are shown. Nominal values (line) are shown for comparison purposes.

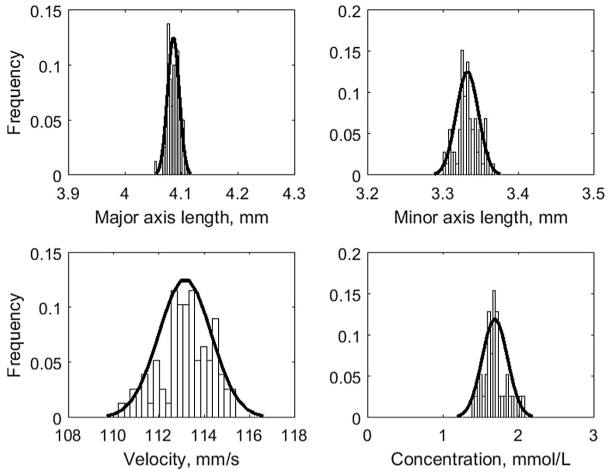


Fig. 7. The example distributions of measured droplet sizes, velocities, and concentrations. The histograms represent the measured data distributions and the lines denote normal distributions whose parameters are estimated from the data. The width of normal distribution is ±3 standard deviations. The average and standard deviation were estimated from data.

solutions using the spectrophotometer at the wavelength of 677 nm, are 9.62×10^{-3} and 9.43×10^{-3} L/(mmol mm), respectively (L = 10 mm). The coefficients are nearly identical. For that reason, the only species present, which absorbs light at 677 nm, is the copper complex. The corresponding estimated absorptivities at the droplet center for the camera (AVT Oscar F-510C) were $3.4 \times 10^{-3} \pm 0.2 \times 10^{-3}$ L/(mmol mm) for the 10 vol% Acorga solutions and $3.2 \times 10^{-3} \pm 0.2 \times 10^{-3}$ L/(mmol mm) for the 20 vol% Acorga solutions. The optical path length (L) was taken to be equal to the droplet major axis length in the calculations. The measured

major axis lengths were 4.2 ± 0.1 mm in the case of 10 vol% Acorga standards and 4.1 ± 0.2 mm in the case of 20 vol% Acorga standard solutions (see also Fig. 10).

The observed absorptivities of the 10 and 20 vol% standard solutions are almost identical also in the case of droplet image analysis. The complex absorbs light at the wavelength range of the red channel. The background was constant in each case and was subtracted from the images. The observed absorptivities were lower for the camera due to the wider spectral bandwidth. The similarity of absorptivities implies that the mass transfer during droplet formation was indeed minimal.

The effect of scattering due to droplet interface curvature was taken into account by calculating molar absorptivities for each pixel of standard solution droplet (Fig. 5b). The apparent molar absorptivity includes both the effect of copper complex light absorption and the effect of light scattering. The apparent absorptivity was assumed to depend only on the chord length, but the variation in the droplet data is notable. The interpolation function was fitted to the droplet data and it was used in the calculation of droplet copper concentrations. The interpolation function gives apparent absorptivity of 3.2×10^{-3} L/(mmol mm) at the droplet center (L = 4.1 mm), which is identical to the absorptivity obtained by fitting to all different concentrations. The difference between the continuous and dotted line in Fig. 5b indicates the effect of scattering. Scattering is minimal at the droplet center and increases towards the droplet edge.

It is acknowledged that copper can be transferred into the standard solution during droplet formation in the calibration experiment. The average formation time of droplets was estimated to be approximately 1.3 s. Due to the short formation time, the effect of mass transfer during the droplet formation with the standard solution is estimated to be approximately 0.3 mmol/L. This is within the variation in the analyzed concentrations of standard solutions. The estimation was made by extrapolating measured data from column bottom experiments, which contain mass transfer from droplet formation (see Fig. 8 and Table 1). The fitted extrapolation power function was used to calculate concentration at flow rate of 1.5 mL/min.

3.2. Concentration analysis

The present method determines organic phase copper complex concentration in the droplet. The results are presented as an extractant conversion and they are shown in Fig. 8. The conversion based on sample analysis (c_{Sample}) taken from the column outlet funnel are clearly higher than based on the droplets imaged at the top (c_{TOP}) and bottom of the column (c_{BOT}) (see Table 1). Results from bottom of column show mass transfer during droplet formation, while results from column top include also mass transfer during droplet rise. The difference between column top and bottom analyses, reveal mass transfer during droplet rise (see also Fig. 8b). Samples include mass transfer during droplet formation, rise and column outlet funnel.

The high concentrations in the analyzed samples are due to long residence times at the column outlet. The average sample residence times at column outlet varied from 60 s to 420 s. It is acknowledged, that the sample concentrations are high and that sampling can be optimized. However, the purpose of this work is to present direct method for determining droplet inner concentrations and sample concentrations were determined for comparison purposes. Also noteworthy is the variation between the samples at high feed flow rates (over 0.3 mL/min) compared to the variation from direct droplet analysis.

The imaging of a droplet at the column top was made just prior it enters the column outlet funnel and imaging at the column bot-

Table 1
Measured average droplet concentrations at column top and bottom (c_{TOP} and c_{BOT}), concentration change during rise (Δc_{Rise}) and collected sample direct analysis (c_{Sample}).

$\dot{V}$ mL/min	c_{TOP} mmol/L	c_{BOT} mmol/L	Δc_{Rise} mmol/L	c_{Sample} mmol/L
0.1	5.3	5.3	0.0	29.9
0.3	2.4	1.7	0.6	21.5
0.5	2.1	1.0	1.1	24.5
0.7	2.1	0.7	1.4	28.5
1	2.3	n.d.	2.3	24.9

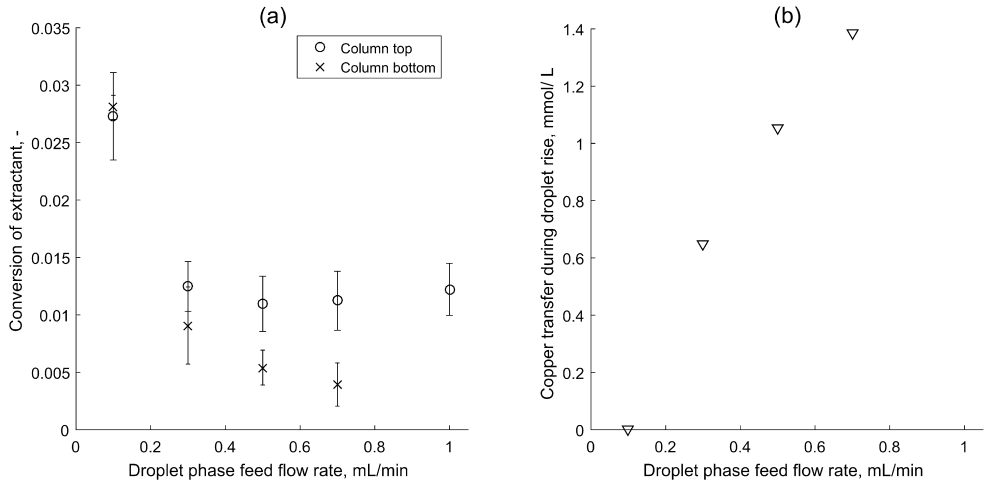


Fig. 8. Copper complex concentration change in column experiments. (a) Conversion of extractant to copper complex. (b) Copper complex concentration change during droplet rise.

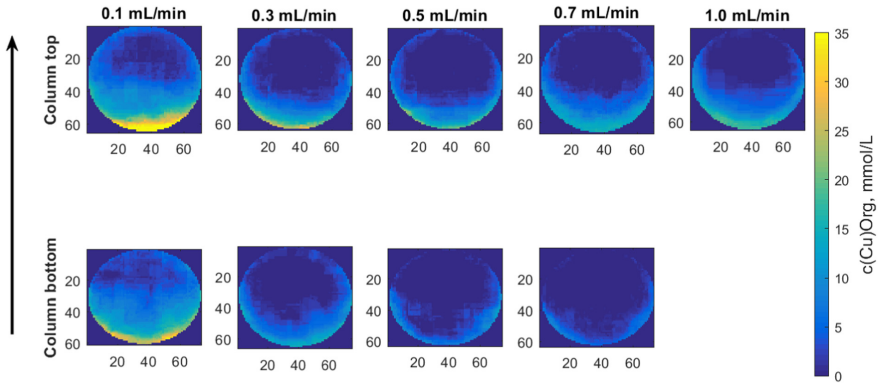


Fig. 9. Examples of determined copper organic phase concentration profiles inside droplets. Separate measurements were made for moving droplets at the column top (top row) and column bottom (bottom row). The arrow indicates the direction of droplet movement. Axis units are pixels.

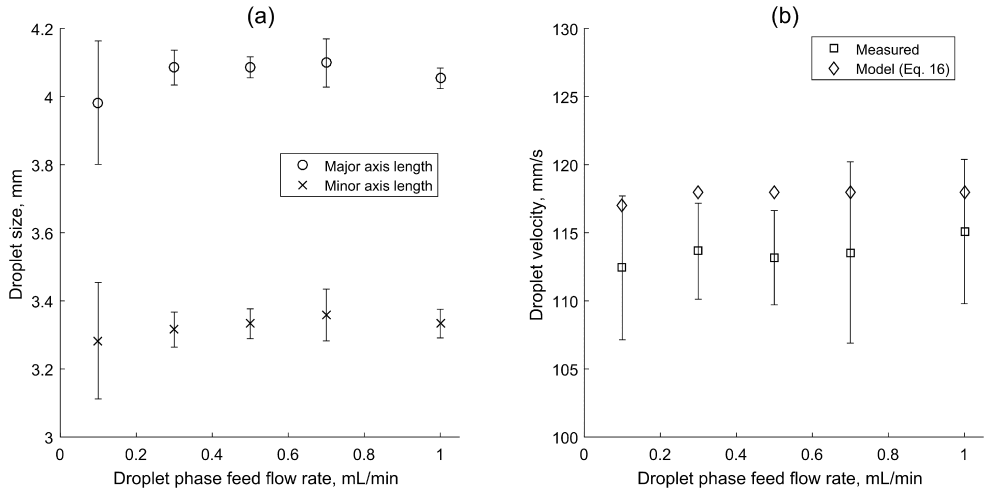


Fig. 10. The droplet sizes and velocities measured at the top of the column with organic feed flow rates 0.1–1.0 mL/min. Data is from Canon Legria HF R47. (a) The measured droplet minor and major axis lengths, (b) The measured and estimated terminal velocities of droplets.

tom was made just after a droplet detached from the needle tip. These two positions were measured in separate experiments. Comparison of the column bottom and column top measurements reveals that the majority of copper is transferred during the droplet formation. When the flow rate increases, copper transfer during the droplet formation decreases rapidly as the droplet formation time decreases. The droplet formation time is approximately 17 s at 0.1 mL/min and it decreases below 2 s when the flow rate is 1.0 mL/min.

The copper mass transfer during the rise increases as the droplet feed flow rate increases (see Fig. 8b). The droplet sizes and the measured velocities of droplets are approximately constant (see Figs. 9 and 10) and the rise time does not vary significantly and cannot explain the difference because the rise length remains constant. Possible explanation is that, the concentration difference between droplet and continuous phase increases as a function of flow rate, due to decreasing mass transfer during droplet formation (see Fig. 8 and Table 1). This leads to observed increased mass transfer during droplet rise.

For determining droplet concentration profiles, droplets were imaged at the bottom and the top of the column, and the concentrations were calculated from the recorded video images (see Fig. 9). The shown droplet images represent examples of droplets at the top and bottom positions of the camera. The highest copper concentrations are found near the droplet interface, which is due to interfacial complexation of copper and limited interaction time for

mass transfer. The concentrations at the droplet bottom tend to be higher than at the top. The explanation for this is not clear, but a possible reason is the density difference between the extractant and copper complex because the loaded extractant is heavier. Another possible reason is the circulatory flows at both sides of the droplet phase boundary. One possible explanation for the observed concentration distribution inside the droplets is the uneven distribution of surfactants. According to Slater (1995), surfactants have a tendency to concentrate in the droplet lower part and to stop internal circulation. In this stagnant cap model the droplet upper part, having lower surfactant concentration, has a mobile interface leading to circulation and reducing concentration differences.

The effect of decreasing the formation time can be observed in Fig. 9 as the concentrations decrease from left to right. The effect is especially clear in the case of observations from the column bottom. The effect of mass transfer during the droplet rise can be seen by comparing images taken at the top and the bottom of the column with the same flow rate. The images recorded at the top of the column show higher concentrations. Comparison of the images is straightforward, as the droplet sizes are approximately equal.

3.3. Droplet sizes and velocities

Fig. 10 shows measured droplet sizes and measured and estimated droplet velocities. Terminal velocities were calculated by using Eq. (16). Data was obtained by image analysis of videos recorded by Canon Legria HF R47.

The droplet sizes are similar in all experiments (Figs. 9 and 10), which is due to identical solution properties and the same needle gauge used in the droplet formation. The increase in feed flow rate of the droplet phase from 0.1 to 1.0 mL/min had no significant effect in the drop sizes.

Droplet velocities were determined at the top of the column and like droplet sizes, velocities are also close to each other. They varied between 112 and 115 mm/s (Fig. 10b). To validate the velocity measurement, correlation by Grace et al. (1976) was used. This correlation is for systems where, for example, surfactants are present as contaminants and correlations for pure systems are not usable. Their correlation for droplet terminal velocity is

$$u_t = \mu_c/(\rho_c d_e)\mathrm{Mo}^{-0.149}(J-0.857) \tag{16}$$

with

$$J = 0.94H^{0.757}, \quad 2 < H \leqslant 59.3 \tag{17}$$

and

$$J = 3.42H^{0.441}, \quad H > 59.3 \tag{18}$$

and H defined as

$$H = 4/3\mathrm{EoMo}^{-0.149}(\mu_c/\mu_w)^{-0.14}, \ \mu_w = 0.9 \text{ mPa s} \tag{19}$$

where μ_c and ρ_c are continuous phase viscosity and density, d_e volume equivalent sphere diameter, Mo is Morton number and Eo is Eötvös number. The property values for the 20 vol% Acorga and 0.16 mol/L $CuSO_4$-solution pair at room temperature were $\rho_c = 1025$ kg/m^3, $\rho_d = 834$ kg/m^3 and $\gamma = 22$ mN/m. Copper sulfate solution density was interpolated from literature values (Lobo, 1981) and interfacial tension was measured using the drop weight method (Adamson, 1990). Using average droplet major and minor axis lengths $\bar{d}_{major} = 4.0$ mm or 4.1 mm and $\bar{d}_{minor} = 3.3$ mm (see Fig 10a), volume equivalent sphere diameters d_e are 3.76 and 3.81 mm. With given physical property values Mo and Eo numbers are $0.242 \cdot 10^{-9}$ and 1.36 and terminal velocities u_t using Eqs. (16)–(19) are 117 and 118 mm/s, which agree well with the measured droplet velocities.

Based on the measurements, the droplet aspect ratio $E = d_{minor}/d_{major}$ is between 0.80 and 0.83. Grace et al. (1976) have presented a correlation for the droplet aspect ratio

$$E = 1/(1 + 0.163\mathrm{Eo}^{0.757}), \quad \mathrm{Mo} < 10^{-6}, \quad \mathrm{Eo} < 40 \tag{20}$$

Using Eo = 1.36, the estimated aspect ratio is 0.83 which corresponds well with the observed aspect ratios. To determine the droplet shape region, diagram presented by Grace et al. (1976) is used. With droplet Eötvös number 1.34 and Morton number $0.242 \cdot 10^{-9}$ the shape region is ellipsoidal which is confirmed from droplet observations.

In addition to surfactants, the droplet size and form have an effect on the terminal velocities. The droplet size defines the mass transfer area, and the droplet velocity affects internal circulation in the droplet and the diffusion layer thickness. The changes in velocity lead to differences in the diffusion layer thickness and wake outside the droplet. The properties of the diffusion layer and circulation both inside and outside the droplet, have an effect on the mass transfer during extraction. The velocity of the droplet affects the rise time of the droplet. Both the droplet size and velocity are measured in order to differentiate between the mass transfer during droplet formation and rise.

4. Conclusions

The presented method was developed for direct monitoring of single organic phase droplets and their reactive copper extraction in a glass column. The developed method is based on the observation of droplet color change, which is due to the formation of strongly light absorbing copper–hydroxyoxime complex. It was verified against reference samples which were analyzed using spectrophotometry. The method can be used to detect droplet concentration directly inside the column from any position where the droplet is visible. This enables monitoring of mass transfer into a single rising droplet in the column, which can be done in separate experiments or in a single one using several cameras.

In this paper the problematic concentration analysis using the traditional sample collection has been performed for comparison purposes, and the advantages of the direct image analysis based method has been presented. The method enabled the determination of individual effect of droplet formation, droplet rise and sample collection on copper mass transfer. The effect of sample collection is most notable due to the long residence time of the droplet phase at the column outlet. The droplet formation has a notable effect on copper mass transfer at low feed flow rates, due to long droplet formation times. The image analysis method also enabled characterizing the shape of droplets, by determining the droplet minor and major axis lengths. There was a good agreement with measured and estimated droplet velocities.

The developed method can be improved in several ways. For example, when a more sensitive camera with a wide dynamic range and improved conversion to digital intensity values is used, a wider range of concentrations can be measured and with better signal-to-noise ratio. Calculation of optical paths and modeling the light scattering at the outer edge region of the droplet would possibly allow inclusion of the edge region into the concentration analysis. One possible way could be to add additional camera(s) at different angle(s) to determine the light scattering and optical properties of the edge region. Different light sources could also be considered, when determining the effects of the light scattering at the outer edge region of the droplet. While the present work is designed for copper solvent extraction, the same method can be used in any metal-extractant pair, where the color change in extraction is large enough for the imaging. This method is by no means limited to metal extraction substances, but it can be applied

also to other systems, where a suitable color change is present. It is also possible to limit the observation bandwidth by installing special filters onto the camera optics. The camera red channel was used here, but present method is not limited to specific color channel. Depending on application, it may be possible to use different channels in detection of different reactive species. On the other hand, it is also possible to change the spectral characteristics of the illumination used and even use laser illumination. The method is not limited to the visible light, assuming that the camera sensor is sensitive to and the optics is suitable for other wavelengths, such as ultraviolet light or infra-red.

Acknowledgements

The authors gratefully acknowledge the Academy of Finland (project "Analysis of polydispersity in reactive liquid-liquid systems") for financial support.

References

Adamson, A.W., 1990. Physical Chemistry of Surfaces. Wiley, New York, USA.
Bart, H.-J., Stevens, G.W., 2004. Reactive solvent extraction. In Ion exchange and solvent extraction. A Series of Advances 17, 37–83. CRC Press.
Baumann, K.-H., Mühlfriedel, K., 2002. Mass transfer studies with laser-induced fluorescence across liquid/liquid phase boundaries. Chem. Eng. Technol. 25 (7), 697–700.
Berberan-Santos, M., 1990. Beer's law revisited. J. Chem. Educ. 67 (9), 757–759.
Biswas, R.K., Hanif, M.A., Bari, M.A., 1996. Kinetics of forward extraction of manganese(II) from acidic chloride medium by D2EHPA in kerosene using the single drop technique. Hydrometallurgy 42 (3), 399–409.
Biswas, R.K., Habib, M.A., Bari, M.F., 1997. Kinetics of backward extraction ofMn(II) from Mn-D2EHP complex in kerosene to hydrochloric acid medium using single droptechnique. Hydrometallurgy 46 (3), 349–362.
Cheung, S.-C.S., Kamath, C., 2005. Robust background subtraction with foreground validation for urban traffic video. Eurasip J. Appl. Signal Process. 2005 (14), 2330–2340.
Denney, R.C., Sinclair, C., 1987. Visible and Ultraviolet Spectroscopy. John Wiley & Sons, Chichester.
Goldstein, J.H., Day Jr., R.A., 1954. A kinetic interpretation of the Bouguer-Beer law. J. Chem. Educ. 31 (8), 417–418.
Grace, J.R., Wairegi, T., Nguyen, T.H., 1976. Shapes and velocities of single drops and bubbles moving freely through immiscible liquids. Trans. Inst. Chem. Eng. 54 (3), 167–173.
Gumprecht, R.O., Sliepcevich, C.M., 1953. Scattering of light by large spherical particles. J. Phys. Chem. 57 (1), 90–95.
Hanna, G.J., Noble, R.D., 1985. Measurement of liquid-liquid interfacial kinetics. Chem. Rev. 85 (6), 583–598.
Heikkila, J., Silvén, O., 1997. Four-step camera calibration procedure with implicit image correction. IEEE International Conference on Computer Vision and Pattern Recognition.
Henschke, M., Pfennig, A., 1999. Mass-transfer enhancement in single-drop extraction experiments. AIChE J. 45 (10), 2079–2086.
Kumar, A., Hartland, S., 1999. Correlations for prediction of mass transfer coefficients in single drop systems and liquid-liquid extraction columns. Chem. Eng. Res. Des. 77 (5), 372–384.
Levenspiel, O., 1999. Chemical Reaction Engineering, vol. 3. John Wiley & Sons, New York, USA.
Liang, T.-B., Slater, M.J., 1990. Liquid-liquid extraction drop formation: mass transfer and the influence of surfactant. Chem. Eng. Sci. 45 (1), 97–105.
Licht, W., Conway, J.B., 1950. Mechanism of solute transfer in spray towers. Ind. Eng. Chem. 42 (6), 1151–1157.
Licht, W., Pansing, W.F., 1953. Solute transfer from single drops in liquid-liquid extraction. Ind. Eng. Chem. 45 (9), 1885–1896.
Liebhafsky, H.A., Pfeiffer, H.G., 1953. Beer's law in analytical chemistry. J. Chem. Educ. 30 (9), 450–452.
Lobo, V.M.M., 1981. Electrolyte Solutions: Literature Data on Thermodynamic and Transport Properties, vol. 1. University of Coimbra, Coimbra, Portugal.
Mann, C.K., Vickers, T.J., Gulick, W.M., 1974. Instrumental Analysis. Harper & Row, USA.
MATLAB Release 2016a. The MathWorks, Inc., Natick, Massachusetts, United States.
Mörters, M., Bart, H.-J., 2000. Fluorescence-indicated mass transfer in reactive extraction. Chem. Eng. Technol. 23 (4), 353–359.
Mörters, M., Bart, H.-J., 2003. Mass transfer into droplets undergoing reactive extraction. Chem. Eng. Process. 42 (10), 801–809.
Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9 (1), 62–66.
Pawelski, A., Paschedag, A.R., Kraume, M., 2005. Beschreibung des Stofftransports am Einzeltropfen in Anwesenheit einer schnellen chemischen Reaktion mittels CFD-Simulation. Chem.-Ing.-Tech. 77 (7), 874–880.
Schulze, K., 2007. Stoffaustausch und Fluiddynamik am bewegten Einzeltropfen unter dem Einfluss von Marangonikonvektion PhD Thesis. Technische Universität Berlin.
Slater, M.J., 1995. A combined model of mass transfer coefficients for contaminated drop liquid-liquid systems. The Can. J. Chem. Eng. 73 (4), 462–469.
Szpak, Z.L., Chojnacki, W., van den Hengel, A., 2012. Guaranteed ellipse fitting with the Sampson distance. Paper presented at Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy.
Wegener, M., Paschedag, A.R., Kraume, M., 2009. Mass transfer enhancement through Marangoni instabilities during single drop formation. Int. J. Heat Mass Transf. 52 (11), 2673–2677.
Wegener, M., Paul, N., Kraume, M., 2014. Fluid dynamics and mass transfer at single droplets in liquid/liquid systems. Int. J. Heat Mass Transf. 71, 475–495.
Weisstein, E.W., Ellipse. From MathWorld-A Wolfram Web Resource. <http://mathworld.wolfram.com/Ellipse.html> (16. Dec. 2016)
Whewell, R.J., Hughes, M.A., Hanson, C., 1975. The kinetics of the solvent extraction of copper(II) with LIX reagents–I single drop experiments. J. Inorg. Nucl. Chem. 37 (11), 2303–2307.
Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22 (11), 1330–1334.

Publication V

Lahdenperä, E., Tamminen, J., Koiranen, T., Kuronen, T., Eerola, T., Lensu, L., Kälviäinen, H.,

Modeling mass transfer during single organic droplet formation and rise

Journal of Chemical Engineering & Process Technology
Vol. 9(2):378, doi: 10.4172/2157-7048.1000378, 2018

Journal of
Chemical Engineering & Process Technology

Lahdenperä et al., J Chem Eng Process Technol 2018, 9:2
DOI: 10.4172/2157-7048.1000378

Research Article Open Access

Modeling Mass Transfer During Single Organic Droplet Formation and Rise

Esko Lahdenperä*, Jussi Tamminen, Tuomas Koiranen, Toni Kuronen, Tuomas Eerola, Lasse Lensu and Heikki Kälviäinen

School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland

Abstract

Copper reactive extraction from ambient aqueous solution to organic droplets using single droplet experiments was performed. Extractant was Agorca M5640 hydroxyoxime in Exxsol D80. An image analysis based method was used to determine droplet concentration directly after droplet formation and rise. Mass transfer during formation is correlated using literature. Level Set interface tracking method was used to find formation hydrodynamics and as a result the assumption of non-circular velocity field could be validated. This was also supported by the circulation criteria based on needle Reynolds number. A model to estimate extraction rate as function of droplet Fourier number was based on a literature correlation and it was found that a model where the interface effect was described using interface mobility parameter was able to predict satisfactorily mass transfer. For a rising droplet stagnant cap model was used. Stagnant cap volumes were estimated from droplet images. A CFD model of a non-deforming rising droplet with rigid interface was used to fit interfacial reaction kinetic constant. Fitted value was much lower than experimentally determined by high a shear reactor. Mass transfer coefficients calculated from CFD model and estimated using literature correlations agreed well. By applying a two-film model it was shown that major part of the resistance is located at the interface between the phases.

Keywords: Mass transfer; Copper extraction; Liquid-liquid extraction; Mathematical model; CFD

Introduction

Single droplet experiment is a common method to determine mass transfer rates between the feeds in liquid-liquid extraction (LLE). During a single droplet experiment three stages are identified: (1) droplet formation, detachment and acceleration, (2) droplet rise/fall and (3) droplet coalescence [1]. In traditional experiments a droplet concentration are measured before formation and after coalescence and this provides an overall mass transfer rates for all stages. By using suitable experimental arrangements the effect of coalescence can be minimized but the droplet formation stage is often substantial [2-5] and by using several different droplet rise times the effect of formation is found by extrapolation of results into zero rise time.

This indirect method to determine the amount mass transfer during formation has some drawbacks. Application of the method requires several experimental points in order to get statistically relevant results. In this work this problem is avoided by using an image analysis based direct measurement method developed by Tamminen et al. [6] to measure concentrations directly from the column. This also simplifies experimental setup as there are no strict limitations for the coalescence stage arrangements.

Mass transfer correlations during droplet formation correlate the amount of extracted Δn or the extraction ratio E as a function of Fourier number $FO_d=4D_m t_F/d_e^2$ [7-10]. Diameter d_e is volume equivalent sphere diameter after droplet formation. Those correlations assume that mass transfer inside the droplet is purely diffusion based. To take into account the intensifying effect of internal circulation and the hindering effect of surfactants, Liang and Slater [4] formulated overall effective diffusivity $D_{F,eff}=k_{H,F}(D_m+D_{E,F})$ where $D_{F,E}$ is time dependent diffusivity due to circulation. The empirical parameter $k_{H,F}$ takes into account the effect of surfactants. Depending on the interface properties, $k_{H,F}$ varies between 0 and 1. Liang and Slater [4] also propose a criterion based on the needle Reynolds number, whether there is circulation during droplet formation.

Kumar and Hartland [11] have published a collection of mass transfer correlations for a rising droplet. For the continuous phase correlations are expressed in form $Sh_c=f(Re, Sc_c, K)$ where K is viscosity ratio between dispersed and continuous phase. Droplet side correlations are based on Newman [12] model, which assumes no circulation. The intensifying effect of droplet internal circulation is taken into account by (1) using the effective diffusion coefficient D_{eff} which is D_m multiplied with a constant [12,13], (2) using eddy diffusivity D_E [14] or (3) combining eddy and molecular diffusivities into an effective diffusivity D_{eff} [15-17]. To take into account the effect of surfactants, Slater [18] applied the stagnant cap model where a droplet is divided into a circulating and stagnant regions. Effect of surfactants on interface mobility is implemented by using a similar experimental parameter $k_{H,R}$ as was used in droplet formation.

DNS (=Direct Numerical Simulation) method to solve transport equations can, in principle, provide parameters for the constitutive equations, like mass transfer coefficients. During droplet rise, if a droplet is smaller than the critical diameter then it maintains sphericity and it can be modelled as a sphere with constant shape and diameter. This approach has been used by Piarah et al. [19], Wegener et al. [20], Jeon et al. [21] and Pawelski et al. [22]. When a droplet is deformed due to diameter being larger than critical diameter, interface tracking offers a method to model the combined effect of hydrodynamics and interface evolution on mass transfer. A rising droplet interface tracking has been applied by, for example, Deshpande and Zimmerman [23],

***Corresponding author:** Esko Lahdenperä, School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland, Tel: +35850553425; E-mail: Esko.Lahdenpera@lut.fi

Received March 27, 2018; **Accepted** April 02, 2018; **Published** April 20, 2018

Citation: Lahdenperä E, Tamminen J, Koiranen T, Kuronen T, Eerola T, et al. (2018) Modeling Mass Transfer During Single Organic Droplet Formation and Rise. J Chem Eng Process Technol 9: 378. doi: 10.4172/2157-7048.1000378

Yang and Mao [24], Kenig et al. [25] and Wang et al. [26] who all have used Level Set (LS) method. For the droplet formation, LS method is used by Lu et al. and Soleymani et al. [27,28]. Soleymani et al. however, model only droplet hydrodynamics during formation and rise and they do not calculate mass transfer.

When the classical two-film theory is used to describe mass transfer between two feeds LLE, it is assumed that two films having finite thickness is formed between the phases. The interface itself is assumed to have infinitely small width thus providing negligible mass transfer resistance [29]. However, the assumption of negligible interface thickness and resistance is questionable [30]. Hu et al. [31] modelled mass transfer in LLE using molecular simulation and according to their result, as a contrary to assumption used in two-film theory, the effect of the interface on the mass transfer cannot be neglected. Hu et al. also claim that the surfactants restructure the interface and the mass transfer mechanism is modified. An additional component to be considered is an interface reaction which in two-film theory is assumed to be a surface phenomenon and proceeding with the kinetics of its own. But according to Hu et al. the interface structure is complex and it can be assumed that the empirical parameters for reaction kinetic models determined, for example, in high shear mixers, are not usable because of missing interface effects.

In this study the focus is on mass transfer from continuous phase to organic droplet. Models of copper reactive extraction from aqueous phase into an organic droplet during droplet formation and droplet rise were formulated. Droplets were formed using 0.4 and 0.8 mm needles with several different feed rates. Droplet concentrations, diameters and velocities after the formation and in the end of the rise were measured with an image analysis method. The method is documented in the previous work by Tamminen et al. [6]. An empirical model of mass transfer in droplet formation is based on a methodology proposed by Walia and Vir [9] enhanced by Liang and Slater [4]. Results are compared with Popovich [7] model. The droplet formation hydrodynamics is simulated with CFD using LS method to provide support for the selected velocity profile during droplet formation.

Copper transport from aqueous continuous phase to the organic droplet during droplet rise was modelled with CFD using a stationary spherical droplet with non-deforming interface. The ambient aqueous phase is moving with the measured droplet terminal velocity. Stagnant cap model was implemented by dividing the droplet interface into two domains. Velocity boundary condition was used to adapt corresponding interface mobility to reflect stagnant cap properties. By assuming a fully rigid interface (k_{HR}=0) the reaction kinetic coefficient and extraction rate were estimated. Fractional mass transfer resistances were calculated from the film model.

Materials and Methods

Experiments

Experiments were made using same feeds, setup and method as in previous work by Tamminen et al. [6]. Results of previous work were extended here by using a smaller diameter needle (0.4 mm) in drop formation.

The droplet formation flow rates were 0.041, 0.21, and 0.29 mL/min in case of 0.4 mm needle and 0.10, 0.30, 0.50, 0.70 mL/min, when 0.8 mm needle was used. Needle Reynolds number Re_N was between 1-10. Droplet formation times were not explicitly measured. They were calculated based on feed rate and measured droplet volume.

Contact angles were measured from sessile droplets with 20 vol-% Acorga M5640 in Exxsol D80 (by Exxon). The aqueous phase was 0.16 M copper sulfate solution. The size of droplet and the height (h) of a sessile droplet from needle tip to droplet apex were measured. Contact angle is calculated from $\tan(\theta/2) = h/r_d$ [32]. Droplet image axes measured in x- and y-directions confirmed spherical droplet assumption. Estimated contact angle θ is 120° Measured physical properties of 20 vol-% Agorca solution are shown in Tables 1 and 2.

Details of feed solution preparation, experimental arrangement, droplet imaging and analysis method are documented [6]. Local droplet concentrations were measured at the end of the droplet rise and just after the droplet detachment.

Mass transfer in droplet formation

Popovich [7] has presented a model to describe the total mass transfer during droplet formation:

$$\Delta n = a_1(c^*_d - c_{d,o})\sqrt{\pi D_m t_F d_e^2} = a_1(c^*_d - c_d)\frac{de}{2}\sqrt{\pi Fo_d d^2_e} \quad (1)$$

Where c^*_d is equilibrium concentration, $c_{d,O}$ is initial concentration and t_E is the formation time of a spherical droplet having volume equivalent diameter d_e. The model is based on assumption that the mass transfer process is diffusion controlled so interfacial instabilities and internal circulation is not taken into account. The constant a_1 varies between 0.857 and 3.43 [5].

In the model by Liang and Slater [4] the extraction ratio E is calculated with the model developed by Walia and Vir [8,9]

$$E = E_d - (7/8)E^2_d + (49/72)E^3_d - 0.476E^4_d \quad (2)$$

Liang and Slater [4] define the term E_d as a function of modified Fourier number Fo'_d

$$E_d \frac{36}{\sqrt{21\pi}}\frac{1}{2}\sqrt{Fo'_d}\left(1+\frac{1}{2}\sqrt{\pi Fo'_d}\right) \quad (3)$$

Modified Fourier number Fo'_d uses overall effective diffusivity $D_{F,eff}$ instead of molecular diffusivity D_m The overall effective diffusivity is defined as:

$$D_{F,eff} = k_{H,F}(D_m + D_{F,E}) \quad (4)$$

Where $k_{H,F}$ describes the effect of surfactant and has values between 0 to 1, and $D_{F,E}$ is pseudo-eddy diffusivity to take into account the effect of droplet internal circulation. Liang and Slater [4] propose a method to judge if system is (1) diffusion controlled Re_N<10, or (2) circulation enhanced diffusion: 10<Re_N<34, or (3) circulation controlled: 10<Re_N>34. In a system with pure diffusion control, the diffusivity D_m is to be used. In cases 2 and 3 the enhancing effect of circulation on the mass transfer is taken into account using $D_{F,E}$.

Liang and Slater [4] considered only interfacial effects to be included in the constant $k_{H,F}$ but also other effects like the resistance generated by interfacial reaction can also taken into account [33,34].

Mass transfer during droplet rise

When a droplet is rising the mass transfer is affected by diffusion and internal circulation and also the outside convection as well. Depending on the interface mobility, the interface can be rigid or mobile and this has an effect on droplet internal circulation strength.

Correlations provided by the literature are mostly for systems without contaminations and surfactants [11]. Slater [18] has formulated

a model to take into account decrease of mass transfer rates due to surface effects and defines a correction factor $K_{H,R}$:

$$k_{H,R} = \overline{U}_i / U_t \quad (5)$$

Where $\overline{U}_i$ is the average interfacial velocity and is U_t droplet terminal velocity.

The stagnant cap model for a rising droplet by Slater [18] is applied. The model is based on the concept of two zones, where one zone is stagnant and another zone is circulatory (Figure 1). Surfactants and other contaminants act against circulation and as a result the interface becomes more rigid [18].

A droplet is divided into two zones having relative sizes f_v and 1-f_v, where f_v is size of the stagnant zone. An overall effective diffusivity is calculated based on f_v, molecular diffusivity D_m and eddy diffusivity $D_{R,E}$.

$$D_{R,eff} = f_v D_m + (1-f_v)(D_m + D_{F,E}) \quad (6)$$

Eddy diffusivity describes the effect of internal circulation on mass transfer and is calculated using Handlos and Baron Method [14]:

$$D_{R,E} = \frac{k_{H,R} U_t d_e}{2048(1+\mu_d/\mu_c)} \quad (7)$$

It can be assumed that in this system the interface is rigid so $k_{H,R}=0$. Droplet phase mass transfer coefficient is calculated using the model by Newman [12]. The overall effective diffusivity defined in eq. (7) is used here:

$$k_{CuA2} = -\frac{d_e}{6t_R} \ln\left(\frac{6}{\pi^2}\right) \sum_{z=1}^{\infty} \frac{1}{Z^2} \exp\left(-\frac{4z^2\pi^2 D_{R,eff} t_R}{d_e^2}\right) \quad (8)$$

This equation is valid for a case where the main mass transfer resistance is on the droplet side [18].

Stagnant zone size was estimated from droplet concentration distribution images. An example of concentration profiles in a rising droplet is shown in Figure 2. The average overall mass transfer coefficient K_d during droplet rise is:

$$K_d = -\ln\left(\frac{c_d^* - c_d}{c_d^* - c_{d,0}}\right) \frac{d_e}{6t_R} \quad (9)$$

Where t_R is droplet rise time and $c_{d,0}$ is droplet concentration after formation. This equation can be used as well to calculate mass transfer coefficient during formation. The time to use, then, is the droplet formation time t_F and $c_{d,o}$ is feed concentration which in most cases is zero [35].

CFD model for droplet formation

Droplet coalescence simulations were performed with Comsol Multiphysics v.5.2 [36] using LS method for two-phase laminar flows. A 2-d axisymmetric geometry was used. Geometry and boundary conditions are shown in Figures 3a and 3b. Two needle diameters were used: 0.8 mm o.d/0.51 mm i.d and 0.41 mm o.d and inner diameter 0.21 mm. Calculation domain dimensions were 3.8 mm width and height 10.2 mm. Same domain size was used for both needles.

A hemisphere having diameter of needle inlet was formed before calculations (Figure 3b). Initially both phase velocities and pressures were set to 0. The Comsol LS solver performs a steady state calculation at time t=0 to get consistent initial state for the transient calculation.

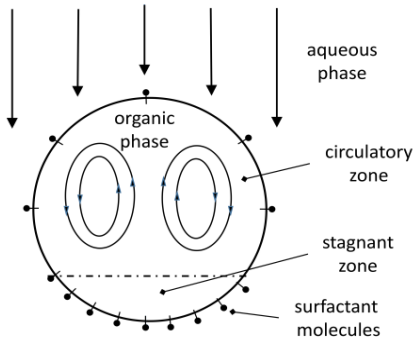


Figure 1: Description of the stagnant cap model of a rising droplet. Arrows describe the flow relative to a droplet.

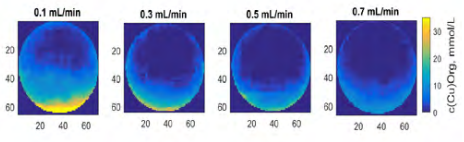


Figure 2: An example of measured concentration profiles of a rising droplet with different feed rates. Droplets are moving upwards (compare with Figure 1). The complex CuA_2 concentrations are higher in the droplet lower part thus supporting the stagnant cap assumption.

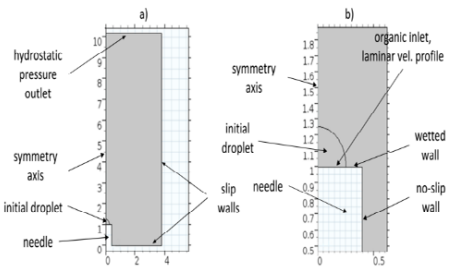


Figure 3: Droplet formation and CFD boundary conditions. Units on the geometry axis are in mm. Calculation domain meshed using triangular elements and the domain divided into: initial droplet, around the initial droplet and remaining part outside these two zones. Inside the initial droplet mesh element limits (max/min) are 25.6 and 0.0765 µm. In the mesh control domain and remaining area outside the limits are 0.574 and 49.7 µm. Total amount of mesh elements: 7800. a) Geometry and certain boundary conditions b) Close up figure containing details near the needle.

The default LS solver was used. Two user controllable LS parameters, reinitialization parameter and interface thickness parameter were adjusted to reach convergence. Reinitialization parameter was set to 0.1 ms^{-1} and interface thickness was half of the maximum element size.

CFD model for droplet rise

Mass transfer between the continuous phase and droplet was calculated with a CFD-model using Comsol Multiphysics v.5.2 [36]. The surface velocity correction factor $k_{H,R}$ was determined with this model to take into account the drag increasing effect of surface active agents, local mass transfer coefficients K_{cu} and k_{CuA2} of Cu transfer in continuous phase and in droplet (eqs. 12b, 12c) and the overall mass transfer coefficient K_D (eq. 12a).

The system is modeled as a stationary spherical droplet and the continuous aqueous phase is moving with a measured terminal velocity. Both phases are separate calculation domains. The following assumptions are used: 2-d axisymmetric geometry, the spherical droplet, laminar velocity fields, steady and constant droplet volume.

Droplet was divided into two domains: stagnant cap zone having volume fraction of f_v and circu lating zone with 1-f_v volume fraction.

The rectangular continuous phase domain width and height were set to 3 and 10 times the droplet diameter, respectively. Unstructured triangular mesh was used for the droplet and continuous phase domains. The fine grid resolution along the interface was created by specifying the amount of cells at the interface. Mesh sensitivity was tested by refining the grid near the interface. Variation of mass transfer coefficient was used as criteria for grid independence. Calculation was performed in two stages. In the first stage, laminar Navier-Stokes equation was solved to provide flow field. Boundary conditions for the continuous phase were: uniform velocity U_t at inlet, which is the experimentally determined droplet terminal velocity; sliding wall at calculation domain vertical sides; uniform velocity U_t at outlet. At droplet side $p=0$ as a pressure constraint; sliding wall with tangential velocity set to average interface velocity $\overline{U_i} = k_{H,R}U_t$ at droplet interphase; full rigid interface is simulated by setting $K_{H,R}$ to 0.

Boundary conditions for the droplet phase were: moving wall with radial velocity u_d=u_c and axial velocity w_d=w_c at droplet and continuous phase interface pressure is equal to pressure on the continuous side as pressure constraint. Species diffusivities presented in Table 2.

Boundary condition between organic and droplet phases is set by the interfacial reaction rate. The extraction of copper with a hydroxyoxime HA at the phase interface [37-39] is:

$$Cu^{2+}(aq)+2HA(org) \rightleftharpoons 2H^{+}(aq)+CuA2(org) \quad (10)$$

and the reaction rate equation is:

$$R_A(t) = k_{R,A}\frac{[Cu^{2+}(i)][HA(i)]^2}{[H^{+}(i)]} + \frac{k_{R,A}}{K_E}[CuA_2(i)][H^{+}(i)] \quad (11)$$

Where R_A is area based reaction rate, $k_{R,A}$ kinetic constant, K_E equilibrium constant and $[Cu^{2+}(i)]$, $[HA(i)]$, $[H^{+}(i)]$ and $[CuA2(i)]$ are concentrations at the interface. Tamminen et al. [38] determined kinetic constant in a high shear reactor: $k^0_{R,A}$=1.4 × 10^{-3} dm^4 /(mol s). Equilibrium constant k_E is 8 at hydroxyoxime concentration 20vol-%=0.38 M [39]. In this work due to very mild hydrodynamic conditions, the expected kinetic constant will be lower than determined in the high shear experiments.

Overall mass transfer coefficient, K_d and local mass transfer coefficients on droplet and continuous phase side kcuA2 and k_{cu} are determined from:

$$K_d = \frac{V_d([CuA_2](t_R)-[CuA_2]_F)}{A_d\int_0^{t_R}\{[CuA_2]^{*}-[CuA_2](t)\}\,dt} \quad (12a)$$

$$k_{cu} = \frac{V_d([CuA_2](t_R)-[CuA_2]_F)}{A_d\int_0^{t_R}\left\{[Cu^{2+}]-\overline{[Cu^{2+}(i)]}(t)\right\}dt} \quad (12b)$$

$$k_d = k_{cu,A_2} = \frac{V_d([CuA_2](t_R)-[CuA_2]_F)}{A_d\int_0^{t_R}\left\{\overline{[CuA2(i)]}(t)-[CuA_2]\right\}dt} \quad (12c)$$

$\overline{[CuA_2(i)]}$ and $\overline{[Cu^{2+}(i)]}$ are average concentrations at the phase interface.

Results and Discussion

The estimation and application of mass transfer models are described in Figure 4.

Droplet formation

In Table 3 is shown droplet experiment results. Droplet volumes were calculated using formula:

$$V_d=(\pi/6)d^2_{Maj}d_{Min} \quad (13)$$

Average droplet volumes were 29 mm^3 for the 0.8 mm needle and 10 mm^3 for the 0.4 mm needle and corresponding average volume equivalent diameters d_e were 3.8 and 2.7 mm. Droplet average formation times t_F were estimated by dividing the average droplet volume with a feed rate: for the 0.8 mm needle 16, 5, 3 and 2 seconds and for the 0.4 mm needle 21, 4 and 3 seconds.

In Figure 5 is plotted the cumulative mass transfer Δn against the Fourier number and the and the cumulative transfer model by Popovich (eq. 1). The estimated value of the coefficient a_1 is 0.35. This is smaller than the minimum 0.857 noted by Wegener et al. [5] which confirms that in addition to diffusion there are other phenomena

X_{HA}	C_{HA}	ρ	μ	γ	Θ
vol-%	M	kg/m³	mPas	mN/m	degrees
20	0.38	834	3	21.6	120

Table 1: The measured physical properties of Acorga M5640 solutions dissolved in Exxsol D80. Measurements at room temperature (22-24°C). Interfacial tension measured in 0.16 M $CuSO_4$ solution. pH of copper sulphate solution was adjusted to 3.1 with concentrated sulphuric acid.

Species	Cu^{2+}	H^{+}	HA	CuA_2
$D_m/10^{-9}$ m²/s	0.72	9.4	0.46	0.3

Table 2: Species molecular diffusivities in 25°C, Cu^{2+} and H^{+} diffusivities are from Haynes [33], CuA_2 diffusivity and hydroxyoxime HA were estimated by Wilke-Chang method [34,35].

d_N	Q	d_{Maj}	D_{Min}	V_d	$c_{F,m}$	Re_N
mm	mL/min	mm	mm	mm³	Mmol⁻¹	
0.8[a]	0.1	3.9	3.3	26	5.3	1.2
	0.3	3.9	3.4	27	1.7	3.4
	0.5	3.9	3.3	26	1	5.7
	0.7	3.8	3.2	24	0.8	8
0.4[b]	0.04	3.1	2.7	14	6.1	1.1
	0.21	3	2.6	12	1.8	5.7
	0.29	3.3	20.8	16	1.3	8.0

Table 3: Measured droplet dimensions and concentrations with different feed rates Q after droplet formation. Symbols: d_N, needle diameter, d_{Maj}, d_{Min}, droplet major and minor axes lengths, V_d, droplet volume $c_{F,m}$, measured droplet concentration, Re_N, Reynolds number in needle outlet.

(interfacial reaction combined with effect of surfactants) and Popovich model does not follow experimental points.

In Figure 6 is plotted extraction ratio E versus square root of droplet modified Fourier number Fo'_d using Walia and Vir model (eq. 2) with the overall effective diffusivity $D_{F,eff}$ using eq. (4). The fitted values of the surface mobility correction factor $k_{H,F}$ are 0.070 (R^2=0.76) for the 3.8 mm and 0.071 (R^2=0.90) for the 2.7 mm droplets and they are practically equal. The needle Reynolds numbers (Table 3) are less than 10 and according to the criteria proposed by Liang and Slater [4], droplet internal mass transfer is diffusion controlled. Therefore the eddy diffusivity was set to zero and the molecular diffusivity was used in calculation of the overall effective diffusivity. Compared to Popovich model Walia-Vir model combined with effective diffusivity model by Liang and Slater [4] is able to better describe mass transfer during droplet formation.

LS simulation and non-circular assumption

The non-circulatory assumption and the effect of contact angle on the formation hydrodynamics was examined by simulation of droplet formation with three different contact angles using LS method for the studied chemical system. Angles in simulations were 1, 120 and 179 degrees. Based on the simulation results shown in Figure 7 the velocity streamline profile is non-circulatory with the measured θ=120°. When the contact angle approaches 0°, the velocity profile becomes circulatory. Similar non-circulatory droplet formation hydrodynamics was recognized by Lu et al. [27]. Simulated and experimental formation times are shown in Table 4. For both needles experimental formation times $t_{F,m}$ mare somewhat larger compared to the simulated times $T_{F,LS}$. Experimental formation time was determined by dividing droplet feed rate with average droplet volume. The experimental error in formation time determination is 10%, for the 0.8 mm needle formation time is between 1.5 to 1.7 seconds and for the 0.4 mm needle between 1.2 to 1.6 seconds. Interfacial tension is affected by copper extraction [40,41]. When the interfacial tension is 25.5 mNm^{-1} the simulated formation times equals the experimental value for the 0.8 mm needle.

Droplet rise

Based on concentration measurements during a droplet rise (Figure 2) it was recognized that reacted copper complex has a tendency stay at the droplet bottom zone. This supports the assumption of stagnant cap model. Terminal velocity measurements Tamminen et al. [6] gave substantially smaller values than determined from correlations for pure systems which is due to the presence of surfactants i.e., extractant, reducing the interface mobility. Low value of the measured terminal velocity (113 mm/s) corresponds well with the correlation given by Grace et al. [40] for contaminated systems. Based on this it is assumed that the interface is rigid thus signaling a very low value of interface mobility parameter $k_{H,R}$.

Two coefficients are to be determined experimentally when stagnant cap model is combined with effective diffusivity: stagnant volume fraction f_v and contamination coefficient $k_{H,R}$. In this work f_v is determined with image analysis. $k_{H,R}$ was set to zero based on the rigid interface assumption.

Volume fractions of stagnant cap were estimated by measuring droplet major and minor axis lengths and height of stagnant cap from the droplet image. Cap boundary was visually recognized and can also be seen in concentration profiles (Figure 2). Determination was repeated at least 15 times in order to estimate the variation. Fractions are shown in Table 5.

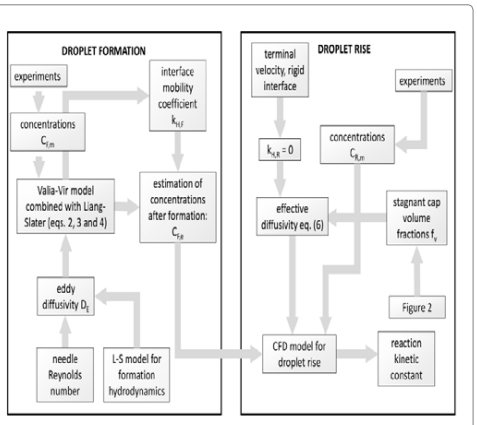


Figure 4: Flow diagram describing the calculation and application mass transfer models.

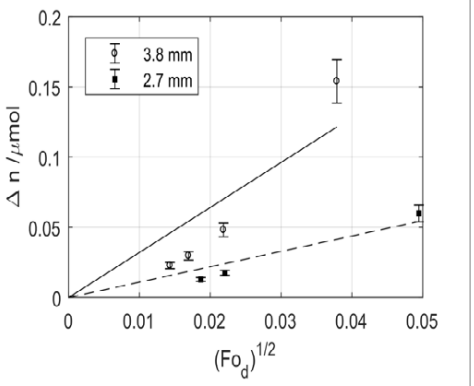


Figure 5: Measured and estimated cumulative mass transfer Δn against the Fourier number and the cumulative transfer model by Popovich (eq. 1): 3.8 mm, 2.7 mm. Symbols refer to experimental values. The fitted value of coefficient a_1=0.35.

Mass transfer simulation using CFD was performed for 0.8 mm needle. It was assumed that $k_{H,R}$ does not depend on the feed rate Q. Droplet concentration $C_{F,e}$ after formation was estimated using the model presented in the section *Mass transfer in droplet formation*. Droplet terminal velocity U_t was set to experimental value 113 mm/s. Droplet diameter was 3.8 mm and rise time 2.3 s. Droplet acceleration to terminal speed was neglected because of a very short acceleration time.

Value for the kinetic constant $k_{R,A}$ was found by fitting with the CFD model. The sum of squared difference between the estimated and measured concentrations was minimized:

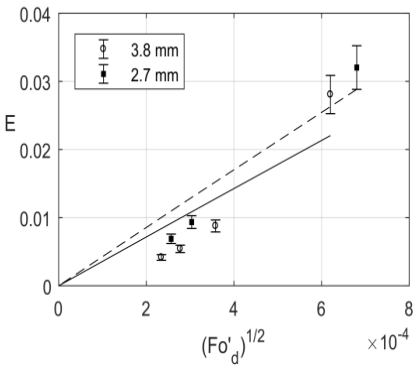


Figure 6: Measured and estimated extraction ratio E vs. square root of droplet modified Fourier number Fo'_d (eqs 2 and 4): 3.8 mm, 2.7 mm. Symbols refer to experimental values. Correction factor $k_{H,F}$ due to surface mobility is approximately 0.07 for both droplet sizes.

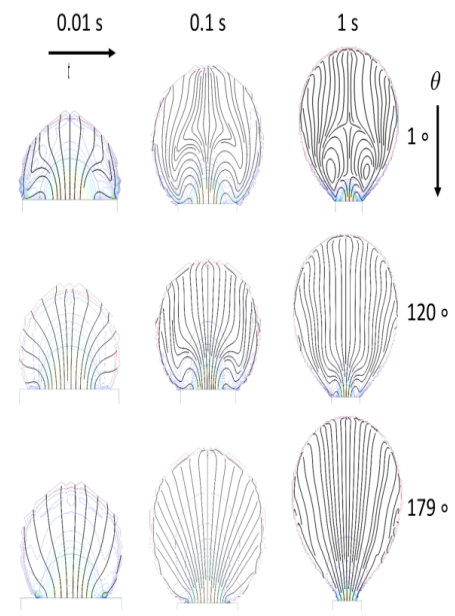


Figure 7: Simulated velocity profiles during droplet formation with different contact angles θ and different times. Feed rate Q is 1 ml/min and needle diameter 0.8 mm.

$$SSQ = \min\left(\sum_{j=1}^{4}\left(c_{R,\mathrm{m,j}} - c_{R,e,j}\right)^2\right) \quad (14)$$

$C_{R,m,j}$ is measured concentration in *j*:th feed flow and $C_{R,e,j}$ is the estimated concentration calculated with the CFD model.

Fitted value of $k_{R,A}$ is 0.13×10^{-6} dm⁴/(mol s). This is substantially smaller than reported by Tamminen et al. [38]. It is obviously due to different hydrodynamic conditions and smaller droplet size compared to this work. Estimated concentration after formation, and measured and estimated droplet concentrations after rise are shown in Table 5.

CFD model was used to calculate mass transfer coefficients using eqs. 12a-c. the most dense mesh (Table 6) was used. Mass transfer coefficients are K_d=0.7 × 10^{-6} m/s, k_{CuA2}=13 × 10^{-6} m/s and k_{Cu}=34 × 10^{-6} m/s. Average K_d estimated from experiments is 2 × 10^{-6} m/s. k_{CuA2} calculated from Newman model (eq. 8) with eddy diffusivity $D_{R,E}$=0 (rigid interface) is 13 × 10^{-6} m/s which is same magnitude compared to value provided by the CFD model. Cu mass transfer coefficient k_{Cu} in continuous phase is estimated with a correlation proposed by Clift et al. [1].

$$Sh_c = 1 + Re^{0.48} Sc_c^{1/3} \quad (15)$$

With U_t=113 mm/s, d_e=3.8 mm, ρ_c=1020 kg/m³, μ_c=1.1 mPas and D_{cu}=0.72 × 10^{-9} m²/s, Re is 388, Sc_c=1498, Sh_c=146 and $k_{Cu}=Sh_c D_{Cu}/d_e$=28 × 10^{-6} m/s. This is slightly smaller than the value 34 × 10^{-6} calculated from CFD model.

Fractional mass transfer resistances for continuous phase (m/k_{Cu})/($1/K_d$) is 2% and for droplet ($1/k_{CuA2}$) / ($1/K_d$) is 5%. Copper distribution ratio *m* between aqueous and droplet phase is 1.2. The reaction fractional resistance is 100%-7%=93% which is a substantial proportion of the total resistance. Similar results were also reported by Ferreira et al. [34].

Solution sensitivity for the selected mesh was tested by simulating using six meshes and calculating mass transfer coefficients (Table 6). Mass transfer coefficients are nearly constant when more than 1 × 10^6 mesh elements were used. Newman model assumes that mass transfer resistance is totally on the droplet side. Continuous side resistance is 2.5 times smaller than the droplet side but the reaction provides the

d_n	Q	$t_{F,LS}$	$t_{F,m}$	V_m	$V(t_{F,LS})$	$V_{F,LS}$
mm	mL/min	S	s	mm³	mm³	mm³
0.8	1	1.3	1.7	26	22	19.4
0.4	0.41	1.1	1.4	9.5	8.3	7.8

Table 4: Comparison of calculated and experimental formation times and volumes for both needles with maximum feed rate Q. $t_{F,LS}$, droplet formation time in simulation $t_{F,m}$ experimental droplet formation time, V_m, experimental droplet volume, $V(t_{F,LS})=t_{F,LS}$ × Q droplet volume at simulated formation time, $V_{F,LS}$ LS-method based droplet volume at the simulated droplet formation time $t_{F,LS}$.

Q	f_v	$c_{F,e}$	$c_{R,m}$	$c_{R,e}$
mL/min		mmol/dm³	mmol/dm³	mmol/dm³
0.1	0.39	4.1	5.1	4.6
0.3	0.25	2.3	2.5	2.8
0.5	0.23	1.8	2.1	2.3
0.7	0.25	1.5	2.1	2

Table 5: Stagnant cap fractions and estimated initial and measured and estimated final droplet copper complex concentrations. $K_{H,R}$=0 (=Rigid interface), estimated $k_{R,A}$=0.13 × 10^{-6} dm⁴/(mol s). Symbols: Q feed rate, f_v, stagnant cap fraction, $c_{F,e}$ estimated concentration after formation, eqs. (2 and 3), $c_{R,m}$, measured average concentration after the rise, $c_{R,e}$, estimated average droplet concentration after the rise.

N_{elem}	K_d	k_{CuA2}	k_{cu}
	10^{-6} m/s	10^{-6} m/s	10^{-6} m/s
475000	0.6	15	50
670000	0.71	15	45
1060000	0.71	14	41
1300000	0.71	14	39
1580000	0.71	14	36
2240000	0.7	13	34

Table 6: Mass transfer coefficients using the CFD model with six meshes. Feed rate is 0.1 ml/min.

most of the resistance so validity to use only Newman model is not justified.

Conclusion

Based on simulation of a rising droplet using stagnant cap assumption and estimating an interfacial reaction kinetic constant it was found that value of measured kinetic constant with high shear reactors is much larger than estimated by simulation. This supports the hypothesis that in this case the interface provides a substantial mass transfer resistance which decreases the overall mass transfer coefficient. The effect is assumed to be due to the structure of the interface which is affected by the hydrodynamic conditions and in this case due to surfactants which probably is been adsorbed into the interface thus making it rigid and on the other hand slowing the reaction reactant's and product's mobility to and away the interface. For the formation stage as well it was found that an empirical model combined with effective diffusivity using interface mobility parameter to describe effect of surfactants is able to predict mass transfer in droplet formation better than a model assuming no explicit interfacial effects.

Acknowledgements

Authors gratefully acknowledge financial support from the Academy of Finland (Project "Analysis of polydispersity in reactive liquid-liquid systems").

Notations

a_i	Coefficient in eq (1)
c	Concentration, M
D	Diffusion coefficient, m^2/s
d	Diameter, m
d_e	Volume equivalent sphere diameter,
E	Extraction ratio
E_d	Extraction ratio assuming constant concentration,
f_v	Fraction of droplet stagnant zone
g	Gravitational acceleration, m/s^2
h	Height of droplet, mm
i	Species label
K	Overall mass transfer coefficient, m/s
K_E	Equilibrium constant in extraction reaction rate equation
k	Local mass transfer coefficient, m/s
k_E	Empirical coefficient in pseudo-eddy diffusivity equation
k_H	Interface mobility parameter
$k_{R,A}$	Reaction kinetic constant, $dm^4/(mol\ s)$
$K^o_{R,A}$	Experimental kinetic constant determined in high shear reactor, $dm^4/(mol\ s)$
m	Partition coefficient, distribution ratio between the phases
n	Molar amount, mol
N_{elem}	Number of elements in CFD mesh
p	Pressure, pa
Q	Feed rate, ml/min
R_A	Reaction rate, $mol/(dm^2\ s)$
R^2	Coefficient of the determination for a fitted model
r	Radius, mm
t	Time, s
u	Radial velocity in cfd model, m/s
U	Velocity, m/s
V	Droplet volume, ml, mm^3
w	Axial velocity in cfd model, m/s
x	Volume fraction
z	Summing index
SSQ	Sum of squared differences

Greek alphabet

a	Species stoichiometric coefficient
Δ	Difference
γ	Interfacial tension, n/m
k	Viscosity ratio, μ_d/μ_c
μ	Viscosity, pa s
ρ	density, kg/m^3
Θ	Contact angle, degrees

Subscripts

-	Average
c	Continuous phase
d	Droplet
e	Estimated
E	Eddy diffusivity
Eff	Effective diffusivity
*	Equilibrium
F,c	Formation in cfd model
F	Formation
i	Interface, species index
LS	Levelset
Maj	Droplet major axis
Min	Droplet minor axis
m	Molecular, measured
N	Needle
R	Rise
0	Initial time, maximum value of reaction kinetic constant
t	Terminal velocity

Abbreviations

(aq)	Aqueous phase
CFD	Computational fluid dynamics
Cu^{2+}	Copper ion
CuA2	Copper complex
DNS	Direct numerical simulation
H^+	Proton
HA	Hydroxyoxime
(i)	Interface
i.d, o.d	Needle inner and outer diameters
LS	Levelset
LLE	Liquid-liquid extraction
(org)	Organic phase
SDS	Sodium dodecyl sulphate
VOF	Volume-of-fluid

Dimensionless numbers

Fo_d	Droplet fourier number, $4D_m t_r/d^2e$
Fo'_d	Modified droplet fourier number, $4D_{F,eff}t_r/d^2_e$
Maj	Droplet major axis
Re	Droplet reynolds number, $Ud_e\rho_c/\mu c$
Re_N	Needle Reynolds number, $\overline{U}_n\rho_d d_n / \mu_d$
Sh_C	Continuous phase sherwood number, $k_c d_e/D_m$
SC_c	Continuous phase schmidt number, $\mu_c/(D_m\rho_c)$

References

1. Clift R, Grace JR, Weber ME (1978) Bubbles, Droplets and particles, Dover Publication, USA.
2. Licht W, Conway JB (1950) Mechanism of Solute Transfer in Spray Towers, Industrial & Engineering Chemistry 42: 1151-1157.
3. Licht W, Pansing WF (1953) Solute Transfer from Single Drops in Liquid-Liquid Extraction. Industrial and Engineering Chemistry 45: 1885-1896.
4. Liang TB, Slater MJ (1990) Liquid-liquid extraction drop formation: mass transfer and the influence of surfactant. Chemical Engineering Science 45: 97-105.
5. Wegener M, Paul N, Kraume M (2014) Fluid dynamics and mass transfer at single droplets in liquid/liquid systems. International Journal of Heat and Mass Transfer 71: 475-495.
6. Tamminen J, Lahdenperä E, Koiranen T, Kuronen T, Eerola T, et al. (2017) Determination of single droplet sizes, velocities and concentrations with image analysis for reactive extraction of copper. Chemical Engineering Science 167: 54-65.
7. Popovich AT, Jervis RE, Trass V (1964) Mass transfer during single drop formation. Chemical Engineering Science 19: 357-365.
8. Walia DS, Vir D (1976) Extraction from single forming drops. The Chemical Engineering Journal 12: 133-141.
9. Walia DS, Vir D (1976) Interphase mass transfer during drop or bubble formation, Chemical Engineering Science 31: 525-533.
10. Heideger WJ, Wright MW (1986) Liquid extraction during drop formation: Effect of formation time. AIChE Journal 32: 1372-376.
11. Kumar A, Hartland S (1999) Correlations for Prediction of Mass Transfer Coefficients in Single Drop Systems and Liquid-Liquid Extraction Columns. Chemical Engineering Research and Design 77: 372-384.
12. Newman AB (1931) The Drying of Porous Solid: Diffusions and Surface Emission Equations.Trans. Am. Inst. Chem. Eng. 27: 203-220.
13. Kronig R, Brink JC (1951) On the theory of extraction from falling droplets. Applied Scientific Research 2: 142-154.
14. Handlos AE, Baron T (1957) Mass and heat transfer from drops in liquid-liquid extraction. AIChE Journal 3: 127-136.
15. Henschke M, Pfennig A (1999) Mass-transfer enhancement in single-drop extraction experiments. AIChE Journal 45: 2079-2086.
16. Henschke M, Pfennig A (2002) Influence of sieve trays on the mass transfer of single drops. AIChE Journal 48: 227-234.
17. Altunok MY, Kalem M, Pfennig A (2012) Investigation of mass transfer on single droplets for the reactive extraction of zinc with D2EHPA. AIChE Journal 58: 1346-1355.
18. Slater MJ (1995) A combined model of mass transfer coefficients for contaminated drop liquid-liquid systems. The Canadian Journal of Chemical Engineering 73: 462-469.
19. Piarah WH, Paschedag AR, Kraume M (2001) Numerical simulation of mass transfer between a single drop and an ambient flow. AIChE Journal 47: 1701-1704.
20. Wegener M, Paschedag AR, Kraume M (2009) Mass transfer enhancement through Marangoni instabilities during single drop formation. International Journal of Heat and Mass Transfer 52: 2673-2677.
21. Jeon SJ, Pawelski A, Kraume M, Hong WH (2011) Mass transfer enhancement by the alkaline hydrolysis of ethyl acetate in a single droplet system. Journal of Industrial and Engineering Chemistry 17: 782-787.
22. Pawelski A, Jeon SJ, Hong WH, Paschedag AR, Kraume M (2013) Interaction of a homogeneous chemical reaction and mass transfer in a single moving droplet. Chemical Engineering Science 104: 260-268.
23. Deshpande KB, Zimmerman WB (2006) Simulation of interfacial mass transfer by droplet dynamics using the level set method. Chemical Engineering Science 61: 6486-6498.
24. Yang C, Mao ZS (2005) Numerical simulation of interphase mass transfer with the level set approach. Chemical Engineering Science 60: 2643-2660.
25. Kenig EY, Ganguli AA, Atmakidis T, Chasanis P (2011) A novel method to capture mass transfer phenomena at free fluid-fluid interfaces. Chemical Engineering and Processing: Process Intensification 50: 68-76.
26. Wang Z, Lu P, Wang Y, Yang C, Mao Z (2013) Experimental investigation and numerical simulation of Marangoni effect induced by mass transfer during drop formation. AIChE Journal 59: 4424-4439.
27. Lu P, Wang Z, Yang C, Mao ZS (2010) Experimental investigation and numerical simulation of mass transfer during drop formation. Chemical Engineering Science 65: 5517-5526.
28. Soleymani A, Laari A, Turunen I (2008) Simulation of drop formation in a single hole in solvent extraction using the volume-of-fluid method, Chemical Engineering Research and Design. ECCE 686: 731-738.
29. Lewis WK, Whitman WG (1924) Principles of gas absorption. Industrial and Engineering Chemistry 16: 1215-1220.
30. Krishna R, Taylor R (1986) Multicomponent Mass Transfer: Theory and Applications. In Handbook of Heat and Mass Transfer. Cheremisinoff NP (ed.), Volume 2: Mass transfer and reactor design. pp: 259-432.
31. Hu Y, Liu Z, Yuan X, Zhang X (2017) Molecular mechanism for liquid-liquid extraction: Two-film theory revisited. AIChE Journal 63: 2464-2470.
32. Adamson AW (1990) Physical Chemistry of Surfaces 5th Ed,Wiley, New York, USA, pp: 88.
33. Haynes WM (2017) CRC Handbook of Chemistry and Physics, 97th Edition CRC Press.
34. Ferreira AE, Agarwal S, Machado RM, Gameiro ML, Santos S, et al. (2010) Extraction of copper from acidic leach solution with Acorga M5640 using a pulsed sieve plate column. Hydrometallurgy 104: 66-75.
35. Reid CR, Prausnitz JM, Poling BE (1987) The properties of gases and liquids (4th Ed.) McGraw-Hill, USA.
36. COMSOL (2017) COMSOL Multiphysics V5.2.

Citation: Lahdenperä E, Tamminen J, Koiranen T, Kuronen T, Eerola T, et al. (2018) Modeling Mass Transfer During Single Organic Droplet Formation and Rise. J Chem Eng Process Technol 9: 378. doi: 10.4172/2157-7048.1000378

37. Szymanowski J (2000) Kinetics and interfacial phenomena. Solvent Extraction and Ion Exchange 18: 729-751.

38. Tamminen J, Sainio T, Paatero E (2013) Intensification of metal extraction with high-shear mixing. Chemical Engineering and Processing: Process Intensification 73: 119-128.

39. Vasilyev F, Virolainen S, Sainio T (2017) Modeling the phase equilibrium in liquid-liquid extraction of copper over a wide range of copper and hydroxyoxime extractant concentrations. Chemical Engineering Science 17: 88-99.

40. Inoue K, Tsunomachi H, Maruuchi T (1986) Interfacial adsorption equilibria of a hydroxyoxime and its metal chelates. Journal of Chemical Engineering of Japan 19: 131-133.

41. Grace JR, Wairegi T, Nguyen TH (1976) Shapes and Velocities of Single Drops and Bubbles Moving Freely through Immiscible Liquids. Trans. Inst. Chem. Eng 54: 167-173.

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

796. JANHUNEN, SARI. Determinants of the local acceptability of wind power in Finland. 2018. Diss.

797. TEPLOV, ROMAN. A holistic approach to measuring open innovation: contribution to theory development. 2018. Diss.

798. ALBATS, EKATERINA. Facilitating university-industry collaboration with a multi-level stakeholder perspective. 2018. Diss.

799. TURA, NINA. Value creation for sustainability-oriented innovations: challenges and supporting methods. 2018. Diss.

800. TALIKKA, MARJA. Recognizing required changes to higher education engineering programs' information literacy education as a consequence of research problems becoming more complex. 2018. Diss.

801. MATTSSON, ALEKSI. Design of customer-end converter systems for low voltage DC distribution from a life cycle cost perspective. 2018. Diss.

802. JÄRVI, HENNA. Customer engagement, a friend or a foe? Investigating the relationship between customer engagement and value co-destruction. 2018. Diss.

803. DABROWSKA, JUSTYNA. Organizing for open innovation: adding the human element. 2018. Diss.

804. TIAINEN, JONNA. Losses in low-Reynolds-number centrifugal compressors. 2018. Diss.

805. GYASI, EMMANUEL AFRANE. On adaptive intelligent welding: Technique feasibility in weld quality assurance for advanced steels. 2018. Diss.

806. PROSKURINA, SVETLANA. International trade in biomass for energy production: The local and global context. 2018. Diss.

807. DABIRI, MOHAMMAD. The low-cycle fatigue of S960 MC direct-quenched high-strength steel. 2018. Diss.

808. KOSKELA, VIRPI. Tapping experiences of presence to connect people and organizational creativity. 2018. Diss.

809. HERALA, ANTTI. Benefits from Open Data: barriers to supply and demand of Open Data in private organizations. 2018. Diss.

810. KÄYHKÖ, JORMA. Erityisen tuen toimintaprosessien nykytila ja kehittäminen suomalaisessa oppisopimuskoulutuksessa. 2018. Diss.

811. HAJIKHANI, ARASH. Understanding and leveraging the social network services in innovation ecosystems. 2018. Diss.

812. SKRIKO, TUOMAS. Dependence of manufacturing parameters on the performance quality of welded joints made of direct quenched ultra-high-strength steel. 2018. Diss.

813. KARTTUNEN, ELINA. Management of technological resource dependencies in interorganizational networks. 2018. Diss.

814. CHILD, MICHAEL. Transition towards long-term sustainability of the Finnish energy system. 2018. Diss.

815. NUTAKOR, CHARLES. An experimental and theoretical investigation of power losses in planetary gearboxes. 2018. Diss.

816. KONSTI-LAAKSO, SUVI. Co-creation, brokering and innovation networks: A model for innovating with users. 2018. Diss.

817. HURSKAINEN, VESA-VILLE. Dynamic analysis of flexible multibody systems using finite elements based on the absolute nodal coordinate formulation. 2018. Diss.

818. VASILYEV, FEDOR. Model-based design and optimisation of hydrometallurgical liquid-liquid extraction processes. 2018. Diss.

819. DEMESA, ABAYNEH. Towards sustainable production of value-added chemicals and materials from lignocellulosic biomass: carboxylic acids and cellulose nanocrystals. 2018. Diss.

820. SIKANEN, EERIK. Dynamic analysis of rotating systems including contact and thermal-induced effects. 2018. Diss.

821. LIND, LOTTA. Identifying working capital models in value chains: Towards a generic framework. 2018. Diss.

822. IMMONEN, KIRSI. Ligno-cellulose fibre poly(lactic acid) interfaces in biocomposites. 2018. Diss.

823. YLÄ-KUJALA, ANTTI. Inter-organizational mediums: current state and underlying potential. 2018. Diss.

824. ZAFARI, SAHAR. Segmentation of partially overlapping convex objects in silhouette images. 2018. Diss.

825. MÄLKKI, HELENA. Identifying needs and ways to integrate sustainability into energy degree programmes. 2018. Diss.

826. JUNTUNEN, RAIMO. LCL filter designs for paraller-connected grid inverters. 2018. Diss.

827. RANAEI, SAMIRA. Quantitative approaches for detecting emerging technologies. 2018. Diss.

828. METSO, LASSE. Information-based industrial maintenance - an ecosystem perspective. 2018. Diss.

829. SAREN, ANDREY. Twin boundary dynamics in magnetic shape memory alloy Ni-Mn-Ga five-layered modulated martensite. 2018. Diss.

830. BELONOGOVA, NADEZDA. Active residential customer in a flexible energy system - a methodology to determine the customer behaviour in a multi-objective environment. 2018. Diss.

831. KALLIOLA, SIMO. Modified chitosan nanoparticles at liquid-liquid interface for applications in oil-spill treatment. 2018. Diss.

832. GEYDT, PAVEL. Atomic Force Microscopy of electrical, mechanical and piezo properties of nanowires. 2018. Diss.

833. KARELL, VILLE. Essays on stock market anomalies. 2018. Diss.

Acta Universitatis
Lappeenrantaensis
834

LUT University

ISBN 978-952-335-314-5
ISBN 978-952-335-315-2 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491
Lappeenranta 2018