

Semester Thesis

Navigation of an Intraocular Camera with Visual Feedback

Moritz Graule

Janis Edelmann

Adviser

Prof. Dr. Bradley J. Nelson

Institute of Robotics and Intelligent Systems

Swiss Federal Institute of Technology Zurich (ETH)

2014-02

Acknowledgements

I would like to express my gratitude to my advisor Janis Edelmann for introducing me to the topics of computer vision and visual servoing. The task was highly interesting and I hope to get a deeper insight into visual servoing with future work. I enjoyed the fruitful discussions and profited from his valuable inputs throughout all stages of the project. I am especially thankful for the lessons on how to write clean code in C++.

Furthermore, I would like to offer my thanks to Prof. Dr. Bradley Nelson for providing me with the opportunity to participate in cutting-edge research in his institute.

Abstract

A magnetically controlled miniature camera probe for intraocular applications is currently under development at the Institute of Robotics and Intelligent Systems (ETH Zurich). The presented work shows how motion feedback for precise navigation can be obtained from the vision of the camera probe by extracting and tracking retinal vessels in real-time. A vessel detection method tailored specifically to the videos from the camera probe was developed. The method is based on a low-pass luminosity correction and binary thresholding and outperforms previously presented approaches in terms of computational efficiency. The Iterative Closest Point algorithm is applied to provide the motion between subsequent frames. Different variants of the Iterative Closest Point algorithm were implemented and expanded to achieve high robustness and fast convergence. An efficient and accurate algorithm was realized, allowing for real-time processing. Real-time application of feature extraction and Iterative Closest Point algorithm was demonstrated by mapping a retinal vessel network from a video. The combination of the two methods will enable closed-loop navigation of the camera probe.

Zusammenfassung

Am Institut für Robotik und Intelligente Systeme (ETH Zürich) wird derzeit eine magnetisch gesteuerte Miniatur-Kamerasonde für Anwendungen im Auginnern entwickelt. Die vorliegende Arbeit zeigt auf, wie ein Feedback der Bewegung der Kamerasonde durch das Extrahieren und Tracken von Gefässen auf der Netzhaut erhalten werden kann. Eine Methode zur Gefässerkennung wurde speziell auf die Videos der Sonde zugeschnitten. Die Methode zur Gefässerkennung basiert auf einer Tiefpass-Korrektur der Lichtverhältnisse und einem binären Schwellenwert zur Klassifizierung der Pixel. Bezüglich Recheneffizienz übertrifft die Methode bisher präsentierte Ansätze. Der "Iterative Closest Point" Algorithmus wird angewendet, um die Bewegung zwischen verschiedenen Bildern zu berechnen. Verschiedene Varianten des Algorithmus wurden implementiert und erweitert, um eine hohe Robustheit und schnelle Konvergenz zu erreichen. Die Identifikation eines effizienten und genauen Algorithmus ermöglicht Bildverarbeitung in Echtzeit. Anwendung der Gesamtlösung in Echtzeit wurde durch die Abbildung eines zusammenhängenden Gefässnetzwerkes von einem Video demonstriert. Die Kombination der beiden Methoden ermöglicht eine präzise Navigation der Kamerasonde in einem geschlossenen Regelkreis.

Contents

Abstract	iii
Zusammenfassung	iv
List of Tables	vii
List of Figures	viii
Notation	viii
1 Introduction	1
1.1 The EyeCam Probe	2
1.2 Project aim	3
2 Related Work	4
2.1 Eye-in-hand Visual Servoing	4
2.2 Extraction and Tracking of Intraocular Features	6
2.2.1 Vessel Detection	7
2.2.2 Iterative Closest Point Algorithm	7
3 Implementation and Evaluation	10
3.1 Preprocessing	11
3.1.1 Noise Reduction	11
3.1.2 Contrast Enhancement and Luminosity Normalization	11
3.2 Feature Extraction	15
3.2.1 Post-processing	16
3.3 Variations of the Iterative Closest Point Algorithm	17
3.3.1 Standard Algorithm	17
3.3.2 Variant 1: Feature Skeletonization	18
3.3.3 Variant 2: Removal of Multiple Links	19
3.3.4 Variant 3: Weighted Pairs	19
3.3.5 Variant 4: Initial Perturbation	20
3.3.6 Variant 5: Overshoot in Translation	21
3.4 Comparison of the ICP Variants	22

3.5 Retinal Mapping	23
4 Summary and Contributions	27
4.1 Outlook	27
References	29
A Closed-form Solution to Planar Registration	35
B Performance of the ICP variants	38

List of Tables

1 Performance of the ICP variants 38

List of Figures

1	The EyeCam	2
2	Closed-loop motion control	6
3	Comparison of fundus photograph and EyeCam image	8
4	Signal-processing cascade	10
5	Superresolution by image averaging	11
6	Channel selection	12
7	Contrast enhancement and illumination normalization	14
8	Feature extraction methods	16
9	Post-processing	17
10	Feature skeletonization	19
11	Removal of multiple links	20
12	Faulty error estimate of the ICP variants	21
13	Convergence of different ICP variants	23
14	Characteristics of different ICP variants	24
15	Approaches to retinal mapping	25
16	Retinal maps	26

Notation

ICP	Iterative closest point
IRIS	Institute of Robotics and Intelligent Systems, ETH Zurich
SIFT	Scale-invariant feature transform
SURF	Speeded up robust features

1 Introduction

Eye surgery involves the handling of tissue with precision in the micrometer range. At this scale, precision of hand-held tools is strongly effected by human tremor, resulting in particularly demanding procedures requiring highly skilled professionals [1]. Patients with complications during and after the operations often require further treatment to avoid gradual loss of sight (e.g. retinal detachment during or after vitrectomy [2]), causing significant additional healthcare costs (e.g. posterior capsule tear at cataract surgery [3]). New technologies to improve procedures would thus be invaluable.

Medical robots have improved surgery in many areas, including neurology [4], orthopaedics [5] and urology [6]. Many advantages such as quicker healing time, reduced pain and lower risk of infection stem from the fact that medical robots enable surgery in a minimally invasive manner. The higher precision obtained with robotic aid can further increase success rate and improve surgical outcome [7]. Several robotic systems have been proposed and tested with the aim of transferring these advances to eye surgery, including teleoperated systems [8, 9, 10] and tools [11], stabilized hand-held tools [12, 13] and microrobots [14, 15]. The teleoperation and guided surgery systems generally consist of tools similar to conventional ones. In teleoperation systems, these tools are actuated with high precision upon a remote control input by the surgeon. Stabilized hand-held tools account for the error in human tool guidance, allowing for a higher precision. In both cases, tools are rigid, resulting in a limited range of action within the eye. A less classical approach to tackle this problem are untethered magnetic intraocular microrobots. Wireless intraocular microrobots are barely limited in their mobility. Relying on the force and torque exerted on them within an electromagnetic field, they can be navigated within the eye with small restrictions [15]. As force and torque on a magnet within an electromagnetic field are volume-dependent, such microrobots have significant limitations in terms of force generation and are thus not applicable for all tasks in eye surgery. They are further unable to inject or remove large amounts of liquids into or from the eye.

In this work the focus lies on magnetically controlled intraocular devices with flexible tethering that have the potential to combine the advantages of the above-mentioned approaches while overcoming some of their shortcomings, as they can be navigated electromagnetically with high precision to reach almost any position

within the eye. At the same time, their connection to the outside allows the exertion of relatively high forces, increase safety and enable injection of fluids (e.g. markers, drugs). As a representative example of such tethered devices, the Institute of Robotics and Intelligent Systems (IRIS, ETH Zurich) is developing a magnetically controlled intraocular camera probe (further referred to as the EyeCam).

1.1 The EyeCam Probe

The EyeCam probe consists of a miniature CMOS camera equipped with three cubes of permanent magnetic material as shown in Figure 1. A magnetic field generated by solenoids placed around the patients head (similar to the OctoMag set-up [16]) can be used to apply a magnetic torque to the camera probe. The camera is connected to the exterior of the eye with a flexible tether for power supply and data transmission.

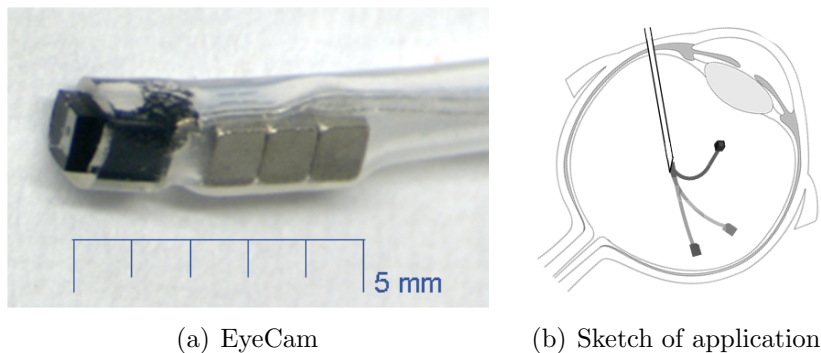


Figure 1: (a) The EyeCam is a miniature camera probe equipped with permanent magnets for wireless navigation within the eye. (b) The flexible tether of the EyeCam would allow it to reach most positions within the eye, making it a promising device for intraocular surgery.

The EyeCam's ability to move around within the eye with little limitations enables a variety of potential applications. The presented work aims at providing means of imaging areas of the eye that can hardly be observed by traditional ophthalmic microscopes. Other tasks under investigation include using the camera probe to provide an image from other surgical robots or tools to improve their precision (e.g. depth control). It could further be equipped with tools (e.g. a needle) to probe the retina, perform remotely controlled surgery with minimal

invasion or localized drug delivery.

The currently implemented strategy to navigate the EyeCam is model-based feed-forward control. A mechanical model of the tether is used to estimate the torque \mathbf{T} required to achieve the desired camera position. The OctoMag then generates the electromagnetic field \mathbf{H} required to exert said torque onto the camera probe according to Equation 1.1 with μ_0 indicating the vacuum permeability, v the volume and \mathbf{M} the magnetization of the magnets.

$$\mathbf{T} = \mu_0 v \mathbf{M} \times \mathbf{H} . \quad (1.1)$$

The mechanical model of the tether is inaccurate, not accounting for certain non-linear effects in bending (e.g. twisting of the tether). The current open-loop control can thus not achieve the precision required for intraocular surgery.

1.2 Project aim

The goal of the subsequent work is to develop software tools that allow for exact control of the EyeCam, as high precision is essential in ophthalmic surgery. This task requires real-time motion feedback to correct errors caused by the inexact mechanical model or external disturbances. Motion feedback would enable exact closed-loop motion control via a magnetic field generated by solenoids. The project aims at extracting the required information solely relying on the vision of the EyeCam, without further observation of the camera probe. An additional target is to allow for mapping of the retina in decent quality, preferably exceeding the quality of single frames.

2 Related Work

2.1 Eye-in-hand Visual Servoing

The task of controlling the motion of a camera based on its view is referred to as eye-in-hand visual servoing. This section shortly summarizes the most important aspects behind visual servoing following the description in [17].

A 3D point with coordinates $\mathbf{X} = (X, Y, Z)$ projects onto the image plane as a 2D point with coordinates $\mathbf{x} = (x, y)$, which compute as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix} = \begin{pmatrix} (u-c_u)/f\alpha \\ (v-c_v)/f \end{pmatrix} . \quad (2.1)$$

Here, (u, v) indicate the coordinates of the image point in pixel units, (c_u, c_v) the coordinates of the focal point, f the focal length and α the ratio of pixel dimensions. With v_c denoting the linear velocity of the origin of the camera frame and ω_c its angular velocity, the velocity of the 3D point can be obtained as

$$\dot{\mathbf{X}} = -v_c - \omega_c \times \mathbf{X} . \quad (2.2)$$

Combining Equations 2.1 and 2.2, one can relate the velocity of a point in image coordinates to the camera velocity as

$$\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \mathbf{v}_c , \quad (2.3)$$

with $\mathbf{v}_c = (v_c, \omega_c)$ and

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \end{bmatrix} . \quad (2.4)$$

$\mathbf{L}_{\mathbf{x}}$ is generally referred to as the interaction matrix. Equation 2.3 can be expanded to a set of n feature points (with coordinates x_n) by stacking the respective interaction matrices $\mathbf{L}_{\mathbf{x}_n}$ to

$$\mathbf{L}_{\mathbf{x}} = \begin{pmatrix} \mathbf{L}_{\mathbf{x}_1} \\ \mathbf{L}_{\mathbf{x}_2} \\ \vdots \\ \mathbf{L}_{\mathbf{x}_n} \end{pmatrix} . \quad (2.5)$$

Since the depth Z_n is generally not available, \mathbf{L}_x must be approximated with $\hat{\mathbf{L}}_x$. Details can be found in [17].

For precise navigation, one can define a set of desired feature coordinates \mathbf{s}^* and an error $\mathbf{e}(t)$ as the difference between the desired features and the actual features $\mathbf{s}(t)$. Differing only by an additive constant, $\mathbf{s}(t)$ and $\mathbf{e}(t)$ have the same time derivative. The estimated interaction matrix with respect to $\mathbf{s}(t)$ can thus be used to relate the time derivative of the error to the camera velocity as

$$\dot{\mathbf{e}} = \hat{\mathbf{L}}_s \mathbf{v}_c . \quad (2.6)$$

This allows the definition of the control law in Equation 2.7 to achieve an exponential decoupled decay in the error ($\dot{\mathbf{e}} = -\lambda \mathbf{e}$) between a desired and the momentary set of features. The control law relies on the Moore-Penrose pseudo-inverse of $\hat{\mathbf{L}}_s$, denoted as $\hat{\mathbf{L}}_s^+$, ensuring a minimal $\|\mathbf{v}_c\|$.

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^+ \mathbf{e} \quad (2.7)$$

Two different approaches for precise control of the EyeCam are feasible, both requiring the extraction and tracking of intraocular features. In the first approach, the operator monitors the video stream from the camera probe and navigates the camera by selecting an area of interest on the stream. This area is then brought into the center of the camera view without further input of the user. This approach makes use of the control law defined in Equation 2.7. The software is required to compute the feature position \mathbf{s}^* associated with the area of interest being in the center of the frame. Equation 2.7 is then used to steer the camera probe to the desired position.

In the second approach, the operator defines the desired camera speed \mathbf{v}_c^* via a joystick or a similar haptic device. This is followed by computing the magnetic field to achieve the desired motion based on the mechanical model. Applying this field to the EyeCam results in an actual camera motion $\mathbf{v}_{c,a}$, which can differ from the desired camera motion due to inadequacies in the model. Following the theory above, the actual camera motion $\mathbf{v}_{c,a}$ is related to the observed feature motion $\dot{\mathbf{s}}_a$ over the pseudo-inverse of the estimated interaction matrix ($\hat{\mathbf{L}}_{s,a}$)⁻¹ as

$$\mathbf{v}_{c,a} = (\hat{\mathbf{L}}_{s,a})^{-1} \dot{\mathbf{s}}_a . \quad (2.8)$$

The actual speed of the EyeCam obtained from tracking the motion of features within the camera vision can then be employed in a feed-back control loop to correct the errors obtained with open-loop control as shown in Figure 2.

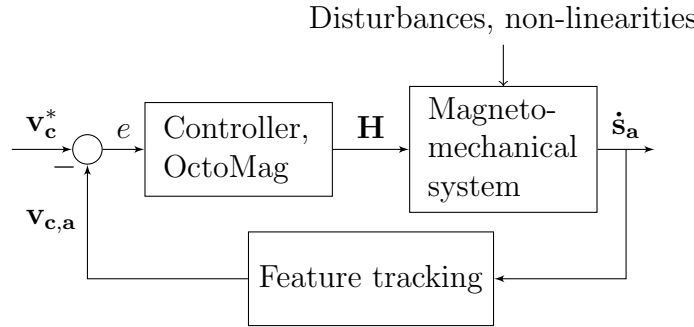


Figure 2: Closed-loop motion control based on a feedback on camera motion obtained by tracking the features within camera view.

2.2 Extraction and Tracking of Intraocular Features

A variety of approaches to calculate the transformation between retinal images have been proposed. These generally match distinctive local features such as key points (e.g. corners) [18, 19] or unique structural elements (e.g. vessel crossings, optic disc, fovea) [20, 21, 22]. The tracking of key points is a common technique widely applied to object tracking and motion detection in computer vision. This usually relies on image descriptors like SIFT (scale-invariant feature transform, [23]) or SURF (speeded up robust features, [24]) which take the surrounding pixels of the selected key points into account for an increased distinguishability.

The high-magnification images obtained with the EyeCam typically contain very few or no landmarks such as vessel crossings. Algorithms further tend not to find a sufficient number of unique key points, because individual points on the vessels are very similar and the retina otherwise lacks a detectable texture. A recent approach addresses these problems by first assigning all points belonging to the vessel network in each frame to a point cloud and then applying the Iterative Closest Point (ICP) algorithm to obtain the transformation between these clouds [25]. The following sections give a brief overview on vessel detection and the ICP algorithm.

2.2.1 Vessel Detection

Many algorithms for retinal vessel detection have been proposed. A complete review is beyond the scope of this report and only a short overview is given instead to highlight the general challenges and identify potentially helpful approaches.

Fundus photographs as shown in Figure 3(a) are images of the interior surface of the eye. They are generally taken at high resolution and low magnification such that they contain the retina, macula, and optic disc. They are a standard tool in diagnosing and monitoring diseases. The driving application in the development of vessel detection algorithms is the automated screening for pathologies using these photographs [26], where an accurate detection is more important than computational efficiency. Thus, many algorithms aim at completely detecting vessel networks at all vessel widths and are not optimized towards high speed [27, 28, 29, 30]. However, a few faster algorithms were reported [31, 32, 25, 33, 34, 35], but most of them still struggle with real-time detection [34, 25]. The computationally most efficient approaches rely on an exploratory algorithm, but they were reported to have difficulties in handling sudden or fast motion and are incapable of coping with zoom between images [25]. The images taken with the EyeCam (high magnification, low resolution) further differ significantly from fundus photographs, as can be seen in Figure 3 and are characterized by a higher noise than the input processed by other methods [25]. Thus, none of the presented algorithms for vessel detection is directly applicable to videos from the EyeCam system, which requires an algorithm able of real-time vessel detection that can deal with significant motion, zoom and noise; accuracy is secondary. Accordingly, the development of a new extraction method was targeted that detects vessels at high speed but limited quality.

2.2.2 Iterative Closest Point Algorithm

Finding the transformation between images with distinct features is straightforward. The transformation between two point clouds can be directly computed if the points contained can be allocated correctly between the two clouds. However, extracting vessels or their borders as binary images without additional information results in ambiguous features. It is impossible to distinguish between the different pixels making up such features. Accordingly, the features from two different frames are point clouds with unknown correspondences. Obtaining the

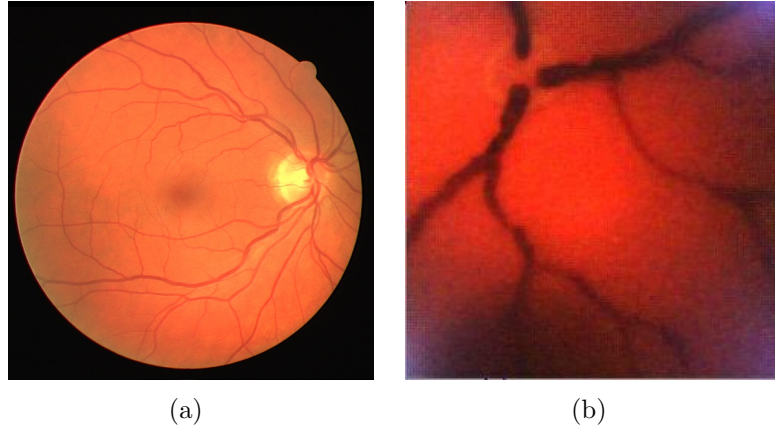


Figure 3: Comparison of a fundus photograph (a) from the DRIVE database [26] and an EyeCam image (b) taken in an artificial eye model. A significant difference in magnification, contrast, illumination and resolution can be observed.

motion between binary vessel networks thus has to rely on approximating the correspondences between these point clouds. This task can be addressed with the ICP algorithm, which iteratively estimates correspondences according to a nearest neighbour criterion. The different stages of the algorithm are briefly outlined below.

Given two sets of points A and B with unknown correspondences, the ICP algorithm aims at aligning these sets by iterating the following steps [41, 42]:

1. Partner each point in A with the closest point in B
2. Use these correspondences to calculate the transformation (rotation, translation and scale) that minimizes the error between the paired points
3. Apply the transformation to A
4. Exit iteration, if a certain determination requirement is met

Many variations of the ICP algorithm have been proposed to address the principal challenges computational speed, stability and robustness in presence of outliers, noise and occlusions. A crucial difference between variants lies in the error metric used (point-to-point versus point-to-plane). While the point-to-plane variant was found to generally outperform the point-to-point variant [43], it has been suggested that the point-to-point error metric is beneficial when working

with two-dimensional scenes [41]. An extensive overview on different approaches is given in [41].

In general, the error-minimizing transformation in the second step can only be solved approximatively; several methods for this task have been presented [44, 45, 46, 47]. In specific situations, a closed form solution to the minimization problem exists. This is the case for the registration of planar scenes using a point-to-point euclidean distance as error metric. The optimal transformation between two sets in the xy-plane A and B of n points then computes as

$$t = \overline{r_B} - sR_\theta(\overline{r_A}) , \quad (2.9)$$

$$s = \left(\frac{\sum_{i=1}^n w_i \| r_{B,i}^C \|^2}{\sum_{i=1}^n w_i \| r_{A,i}^C \|^2} \right)^{\frac{1}{2}} , \quad (2.10)$$

$$\sin \theta = \frac{S}{\sqrt{S^2 + C^2}} \quad \text{and} \quad \cos \theta = \frac{C}{\sqrt{S^2 + C^2}} \quad (2.11)$$

with

$$C = \sum_{i=1}^n w_i r_{B,i}^C \cdot r_{A,i}^C \quad \text{and} \quad S = \sum_{i=1}^n w_i (r_{B,i}^C \times r_{A,i}^C) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} . \quad (2.12)$$

Here, s denotes the scale factor, θ the angle of rotation, t the translation, w_i the weight of the i -th pair, and $R_\theta(\overline{r_A})$ denominates $\overline{r_A}$ rotated by θ . $\overline{r_A}$ and $\overline{r_B}$ refer to the coordinates of the centroids of the two sets and $r_{A,i}^C$ (and $r_{B,i}^C$) the coordinates of the i -th point in A (and B) measured from the centroids. The derivation is outlined in Appendix A (following the solution from [47]).

3 Implementation and Evaluation

As previously mentioned, vision-based feedback navigation of the EyeCam requires the motion between frames to be known. The presented approach relies on pre-processing to enhance image quality and contrast to facilitate the ensuing steps. Pre-processing is then followed by subroutines for feature extraction and feature tracking as shown in Figure 4.

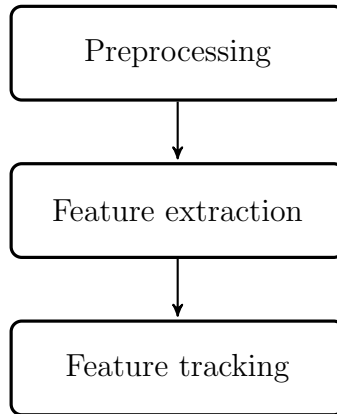


Figure 4: The signal-processing cascade of the implemented method to obtain the motion between frames taken with the EyeCam.

Due to the lack of methods that fit the requirements for feature extraction from EyeCam videos, various signal processing cascades were implemented and compared. The prime evaluation criterion was low computational effort at sufficient quality and repeatability. The selection of the most important steps in pre-processing and feature extraction is presented in Sections 3.1 and 3.2.

The subroutine for feature tracking must find the transformation between the features extracted from two different frames. The promising feature extraction methods all return the features as sets of indistinguishable points. Thus, tracking has to rely on the ICP algorithm as introduced in Section 2.2.2. As a variety of approaches to improve stability and convergence speed of the ICP have been proposed, different variations were implemented and compared to identify the variant that best suits the task of tracking retinal vessels in real-time (Sections 3.3 and 3.4).

All methods were implemented in C/C++ and rely on the OpenCV library for image processing (version 2.4.6, [48]). The tests were done using an eye phantom.

Application within a real eye might require adjustment of the parameters chosen below.

3.1 Preprocessing

3.1.1 Noise Reduction

The magnetic field generated to navigate the EyeCam causes increased noise. Gaussian blur and superresolution by image averaging were applied to reduce high-frequency noise. Four images were combined in the superresolution step, leading to a significant improvement as seen in Figure 5. These are standard methods in image processing and are not discussed here.

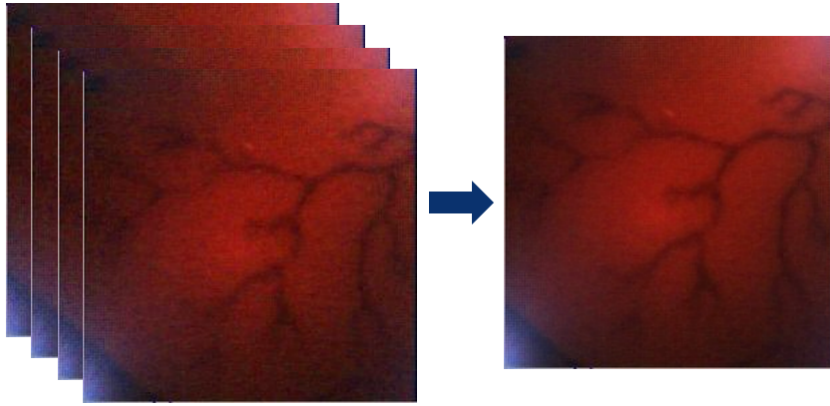


Figure 5: Averaging the four images on the left results in the image on the right, which is of higher quality. This is commonly referred to as superresolution by image averaging.

3.1.2 Contrast Enhancement and Luminosity Normalization

Algorithms both developed for real-time detection[25] and high accuracy [26, 36, 37] select the green channel of the images for a high contrast between blood vessels and background. Thus, images from the EyeCam were split into red, blue and green channel to investigate the contrast level in the different channels. As demonstrated in Figure 6, the highest contrast was found in the red channel which was consequentially chosen as the basis for all following operations. The high contrast observed in the red channel may result from the model eye. The channel filtering may thus have to be adapted for real-world applications.

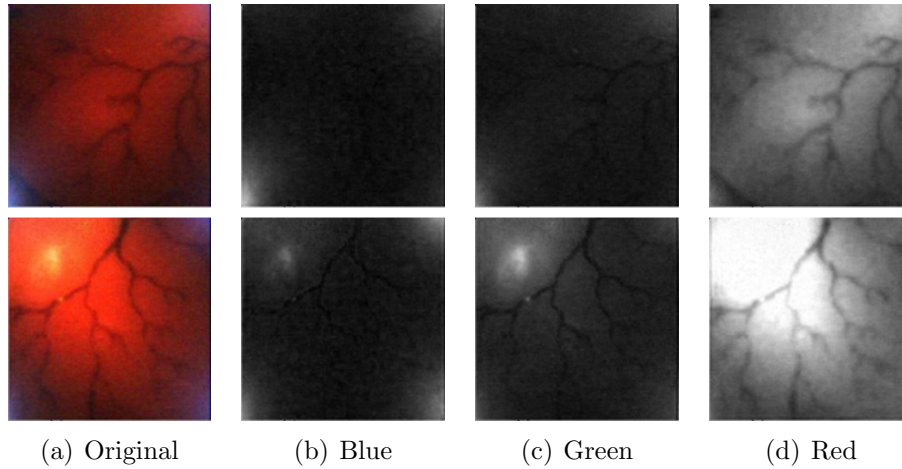


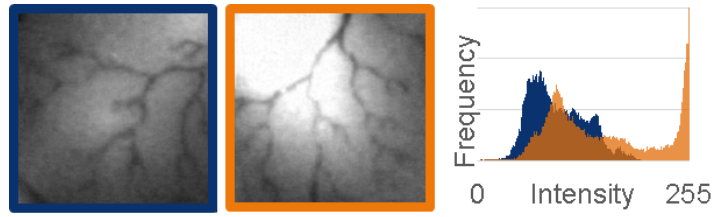
Figure 6: Two frames of an EyeCam video split into their RGB channels.

Algorithms for vessel detection not only benefit from a high contrast but also from consistent illumination conditions in the input images. This facilitates the definition of a unique feature extraction method for all frames. Namely, bringing histograms of differently illuminated frames into alignment in a single operation would be highly beneficial. Suggested normalization methods include a correction with a mixture of Gaussians, which was shown to result in slightly better results than standard low-pass correction [38]. Another method relies on computing the drift in illumination and contrast based on the background pixels and subtracting this from the initial photograph for luminosity normalization [39].

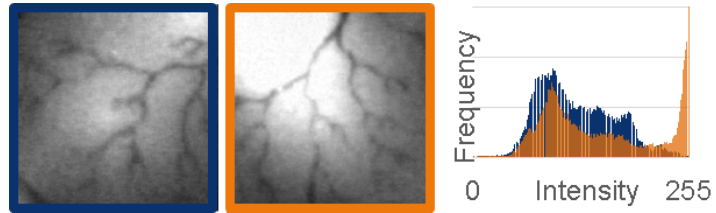
To keep the computational effort limited, three slightly simpler procedures than the suggested ones were investigated: histogram equalization, (linear) contrast stretching, and a spatial low-pass correction. In histogram equalization, ranges of highly populated intensities in a histogram are stretched out. In linear contrast stretching, the highest (and lowest) intensity in an image are set to the maximum and minimum possible intensity values and the other intensities are mapped linearly between these values. As can be seen in Figure 7, these methods increase contrast by some extent but do not have the desired affect of reducing the strong variations between frames.

The implemented low-pass correction disposes of low frequency luminosity drift by applying a strong Gaussian blur to filter out high-frequency intensity changes. The blurred image is then subtracted from the original image for normalization. The effect of the low-pass correction is demonstrated with two ex-

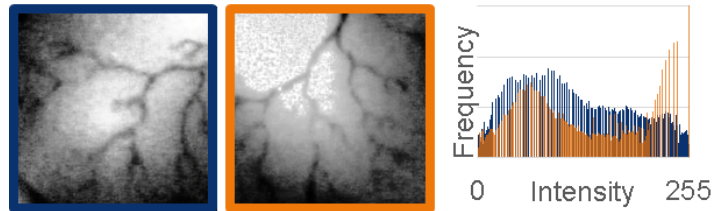
emplary images below. As can be seen in Figure 7, low-pass correction removes the illumination differences between frames well and brings the histograms in alignment. Low-pass correction was thus chosen as the pre-processing step for illumination normalization.



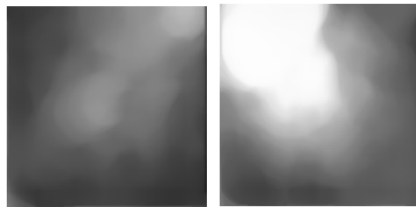
(a) Red channel



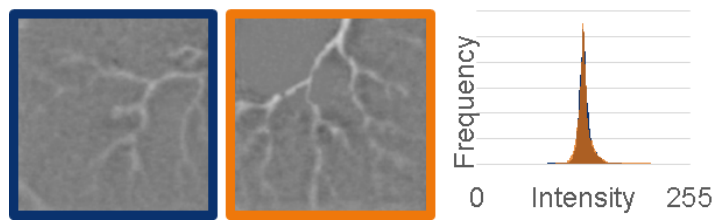
(b) Linear contrast stretching



(c) Histogram equalization



(d) Estimated background



(e) Low-pass correction

Figure 7: The different methods of contrast enhancement and illumination normalization applied to two images for comparison and their effect on intensity distribution.

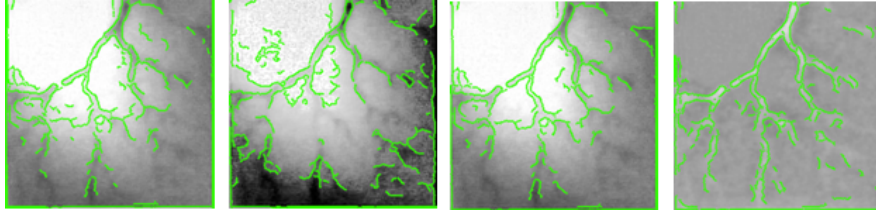
3.2 Feature Extraction

Three common methods of feature extraction were compared in combination with the preprocessing steps described above, namely the Canny edge detector [40], corner detection and binary thresholding.

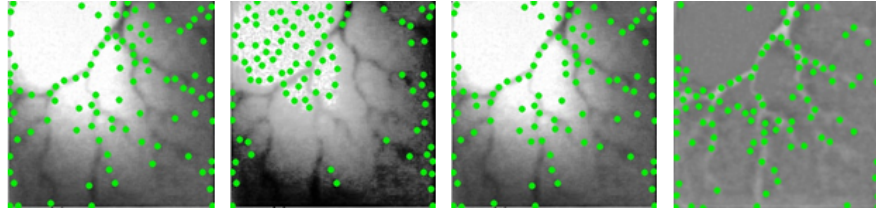
The vessel borders can be characterized by a significant change in intensity. The Canny edge detector (or optimal detector) as presented in [40] is a very common method to find such features (so-called edges). The method uses the intensity gradient of an image to identify its edges. It does so relying on two thresholds T_{up} and T_{low} . A pixel is classified as an edge pixel if its intensity gradient is either above T_{up} or if its intensity gradient lies between the two thresholds and the pixel is next to a pixel considered to be on an edge. The recommended T_{up}/T_{low} ratio is between two and three [40]. The detector was applied to several test frames from an EyeCam video and the threshold were varied to determine which values allow for an optimal detection of vessel edges.

As mentioned in Chapter 2.2, the extraction (and tracking) of corners or key points is a common technique widely applied to object tracking and motion detection in computer vision. For comparison with the other extraction methods, the 100 strongest key points according to [49] were extracted.

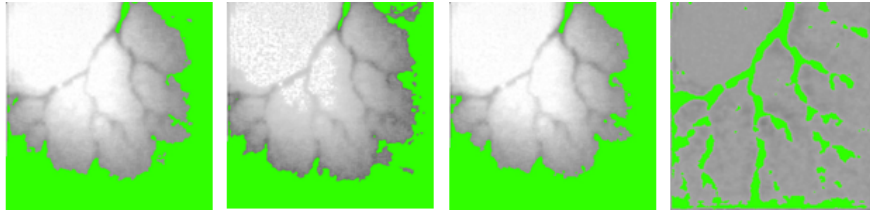
Figure 8 shows the results of the extraction methods applied to an exemplary image in combination with all pre-processing methods mentioned above. The parameters were tuned for optimal detection for the image in Figure 8. These parameters were then applied for feature extraction on a video from the EyeCam. All three detection methods showed the best results following low-pass correction. The Canny edge detector and binary thresholding detected the vessels in comparable quality, but applied to a whole video from the EyeCam, binary thresholding performed slightly better. The latter was chosen for feature extraction, as it is also computationally less expensive.



(a) Canny edge detector (pre-processing steps from left: red channel, histogram equalization, contrast stretching, low-pass correction)



(b) Corner detection (pre-processing steps from left: red channel, histogram equalization, contrast stretching, low-pass correction)



(c) Binary thresholding (pre-processing steps from left: red channel, histogram equalization, contrast stretching, low-pass correction)

Figure 8: The different methods of feature extraction applied to an exemplary frame of the EyeCam video with parameters tuned for optimal detection. The extraction methods are combined with the pre-processing methods discussed above.

3.2.1 Post-processing

A post-processing step was implemented to further reduce false positives in vessel detection based on binary thresholding. This routine uses the continuity of vessels in a neighbour criterion: only pixels that are in close proximity to a minimum number of detected vessel pixels are considered to be on a vessel. The resulting effect is demonstrated in Figure 9. As the borders of the images are particularly noisy with many false positives in the vessel detection, pixels close to the border were not considered as vessel points.



Figure 9: Removing vessel pixels that are not neighbours to a sufficient number of other vessel pixels results in a less noisy feature extraction.

3.3 Variations of the Iterative Closest Point Algorithm

3.3.1 Standard Algorithm

The standard ICP algorithm takes the binary images (I_1, I_2) containing the vessel pixels of two subsequent frames as input. A point cloud C_1 (implemented as a linked list) is assigned to the vessel pixels in I_1 and projected into the second image I_2 . The cloud C_1 is copied to a temporary cloud $C_{1,t}$.

All variants assign correspondences based on a nearest neighbour search. For every point p_i in $C_{1,t}$, I_2 is searched for vessel pixels in a square around p_i . The square size is gradually increased, until a vessel pixel is found in I_2 . The detected vessel pixel is then assigned to a point cloud C_2 and linked to the point p_i . If several vessel points in I_2 are found, the one with the shortest euclidean distance to p_i is chosen. If two or more points have the same distance, one of them is chosen randomly to avoid bias. The ICP is intended to process consecutive frames and a continuous camera motion without jumps is expected, resulting in small transformations. The maximum search distance is thus limited to R_{max} (variable parameter). This reduces the computational cost of the search and the number of point pairs with a wrong correspondence, thus increasing the robustness of the algorithm.

Next, each point pair is assigned the weight $w_i = 1$. The transformation (translation t , scale factor s , rotation angle θ) to align the clouds with the previously assigned correspondences is then computed as introduced in Section 2.2.2:

$$t = \overline{r_B} - sR_\theta(\overline{r_A}) \quad (3.1)$$

$$s = \left(\frac{\sum_{i=1}^n w_i \|r_{B,i}^C\|^2}{\sum_{i=1}^n w_i \|r_{A,i}^C\|^2} \right)^{\frac{1}{2}} \quad (3.2)$$

$$\sin \theta = \frac{S}{\sqrt{S^2 + C^2}} \quad , \quad \cos \theta = \frac{C}{\sqrt{S^2 + C^2}} \quad (3.3)$$

$$C = \sum_{i=1}^n w_i r_{B,i}^C \cdot r_{A,i}^C \quad , \quad S = \sum_{i=1}^n w_i (r_{B,i}^C \times r_{A,i}^C) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.4)$$

The resulting transformation is applied to the point cloud $C_{1,t}$. The order of points in $C_{1,t}$ is not changed. The transformed point cloud is then again partnered with a corresponding cloud from image I_2 as described above. These steps are iterated until the computed transformation is less than predefined values for translation, rotation and scale. At the end of the iteration, the clouds $C_{1,t}$ and C_2 are aligned. Since the order of points in $C_{1,t}$ remains unchanged over all iteration steps, it is known that the k -th point in $C_{1,t}$ corresponds to the k -th point in C_1 . These correspondences are used to obtain the total transformation between C_1 and $C_{1,t}$, which, in the case of correct alignment between $C_{1,t}$ and C_2 , is equal to the transformation between C_1 and C_2 .

3.3.2 Variant 1: Feature Skeletonization

In a first attempt, the ICP algorithm was implemented to directly process the images following low-pass correction and binary thresholding. However, this was too slow for real-time processing and frequently resulted in a translation error of about the vessel width. The low speed was explained with the high number of points per image (5854 – 6696 for four randomly chosen frames) on the vessel network. The error was caused by the pairing of points on opposite borders such that vessels were aligned side-by-side rather than being congruent with one another.

Both problems were addressed by skeletonizing the features, i.e. using only the vessel centrelines in feature tracking. Thus, the average number of points in the four frames could be reduced to 920 – 1120 or by a factor of 5.7 – 7.3. The computational cost of the linear nearest neighbour search as used in the ICP algorithm is $O(2n)$ for n points. Accordingly, the skeletonization made the nearest neighbour search approximately twelve times faster, while maintaining all structural information. As the centreline width is only one pixel, the resulting

error in transformation becomes marginal.



Figure 10: Skeletonization of the features to reduce the alignment error and improve the speed of the ICP.

3.3.3 Variant 2: Removal of Multiple Links

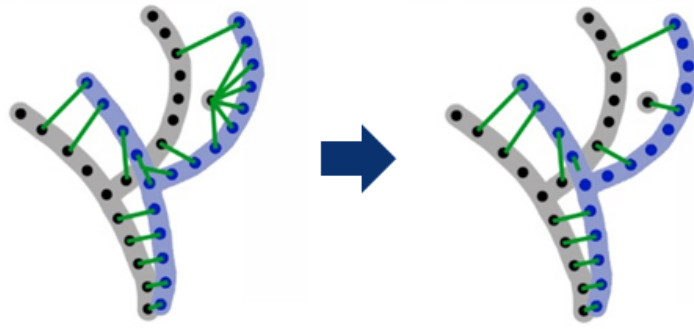
The convergence was found to be inaccurate on selected test pairs. The inaccuracy could partly be explained by the influence of noise points and outliers only contained in one of the two clouds. Such outliers are potentially linked to multiple points in the other cloud, introducing many incorrect correspondences (Figure 11(a), left). Another error source is a bias from points on mesh boundaries in the (common) case that two vessel structures do not overlap completely as shown in Figure 11(b).

Both problems were addressed by changing the implementation of Variant 1 in such a way that if a point in C_2 is linked to multiple points in C_1 , all links except the shortest one are removed as depicted in Figure 11.

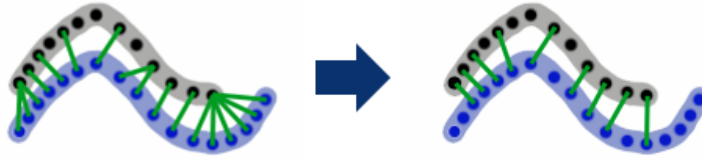
3.3.4 Variant 3: Weighted Pairs

Applying Variant 2 to certain pairs still resulted in an incorrect convergence. Analysing the iterations step by step showed that the error is caused by a number of wrongly partnered point pairs. As transformations are expected to be small (reasoned above), point pairs with a large distance are more likely to describe wrong correspondences. To reduce the influence of such pairs, weights penalizing large distances were introduced as

$$w_i = 1 - d_i/d_{max} , \quad (3.5)$$



(a) Noise and outliers



(b) Incomplete overlapping

Figure 11: In the case where two or more points in cloud C_1 (blue) are linked to the same point in the cloud C_2 (gray), only the shortest link is considered in calculating the transformation. This reduces the error from outliers and the bias from incomplete overlapping of clouds.

where d_{max} denotes the maximum distance over all pairs and d_i the distance of the i -th pair. Other algorithms that reject pairs with a large distance have been suggested [50, 51, 52]. They include ignoring the worst 10% of pairs [50] or pairs with a distance exceeding 2.5 times the standard deviation of distances [51]. The weighing function above was chosen instead as it does not exhibit the discontinuity of the other methods.

3.3.5 Variant 4: Initial Perturbation

The ICP's estimation of the remaining error does not always correspond to the error associated to the ground truth. The ground truth can be obtained by manually ensuring alignment of the starting point cloud C_s with the targeted cloud. At any time during the algorithm's iteration, the ground truth error for a certain

cloud C_{temp} can then be computed as the sum of the point-to-point distances between C_{temp} and the correctly aligned point cloud. The difference between the ICP's error estimate and the ground truth error is demonstrated in Figure 12, where the ground truth error is plotted versus the number of iterations for a specifically selected pair of start and end frame. The ground truth error first decreases, followed by a slight increase before the algorithm is stopped. The example shows how the algorithm converged to a local minimum in error instead of converging towards the global minimum corresponding to the correct transformation. Variant 4 tackles this problem by applying four initial perturbations to the first image I_1 to obtain five images $I_{1,1}$ - $I_{1,5}$. The ICP is then applied to all these images and I_2 , leading to five transformations. Each final transformation is applied to the starting cloud and the sum of squared distances to the cloud C_2 is computed. The transformation resulting in the smallest remaining error is then returned by the method. This increases the probability of finding the global minimum in error, i.e. finding the correct transformation.

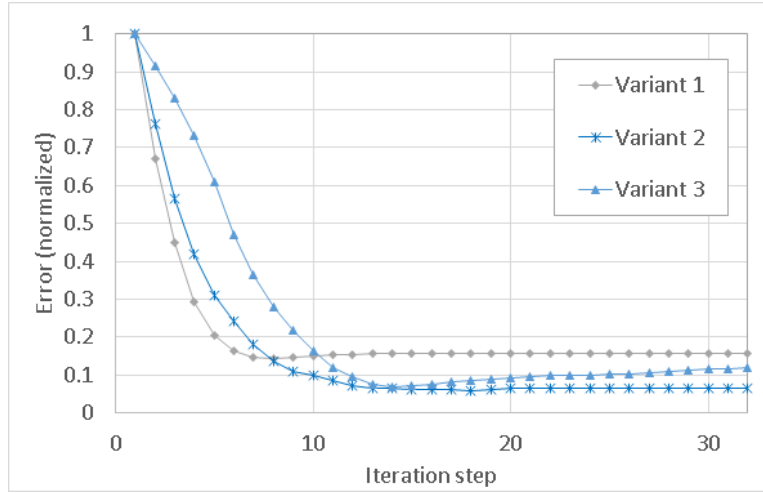


Figure 12: The algorithm's estimate of the error does not necessarily correspond to the ground truth error. Thus, in this example the ground truth error can be seen to increase slightly as the algorithms converge.

3.3.6 Variant 5: Overshoot in Translation

Variations to enhance the speed of previously presented methods would be beneficial as they would facilitate real-time applications. Extrapolating the transforma-

tions over several steps to assure consistence between the the different transformations was reported to improve the convergence rate [53], but time restrictions only allowed for implementation of a more naive approach. This approach is based on the observation that the translations calculated each iteration tend to be less error prone than scale (and to some extend, rotation). A method was implemented that calculates the transformation in each iteration as presented in Variant 3, but then extrapolates the calculated translation by an overshoot of 10% or 50%. The so-obtained transformation is applied to the cloud from the first image. Other steps in the iteration remain unchanged.

3.4 Comparison of the ICP Variants

To evaluate the performance of the different ICP variants, they were applied to 25 randomly selected image pairs (start and end frame). For each iteration step, the error of the current transformation was computed as the sum of the point-to-point-distances with respect to the correct transformation. The results for all pairs were normalized to the initial error. Figure 13 shows the average normalized errors in each iteration step for the Variants 1,2,3 and 5.

For all methods shown in Figure 13, the error decays exponentially. An exponential decay function

$$E(S) = E_0 \exp(-S/\tau) + E_{final} \quad (3.6)$$

can be used to describe the error E in terms of the number of steps S . Fitting the data to such curves allows to extract the exponential time constant τ for the Variants 1-3 and 5. The results of the fit (including the coefficient of determination) are listed in Appendix B and depicted in Figure 13.

The average running time was computed for all variants. The results are shown in Figure 14 with the average final errors for all methods and exponential decay constants. A trade-off between speed of convergence and the robustness of calculations is evident. High speed is relevant for the intended real-time application, but errors in the transformations have a large impact on navigation and mapping and they might render the whole control-loop unstable. A more precise algorithm was thus prioritized. As Variant 3 shows a reasonably fast convergence and a small average final error, it was chosen for further implementations. Vari-

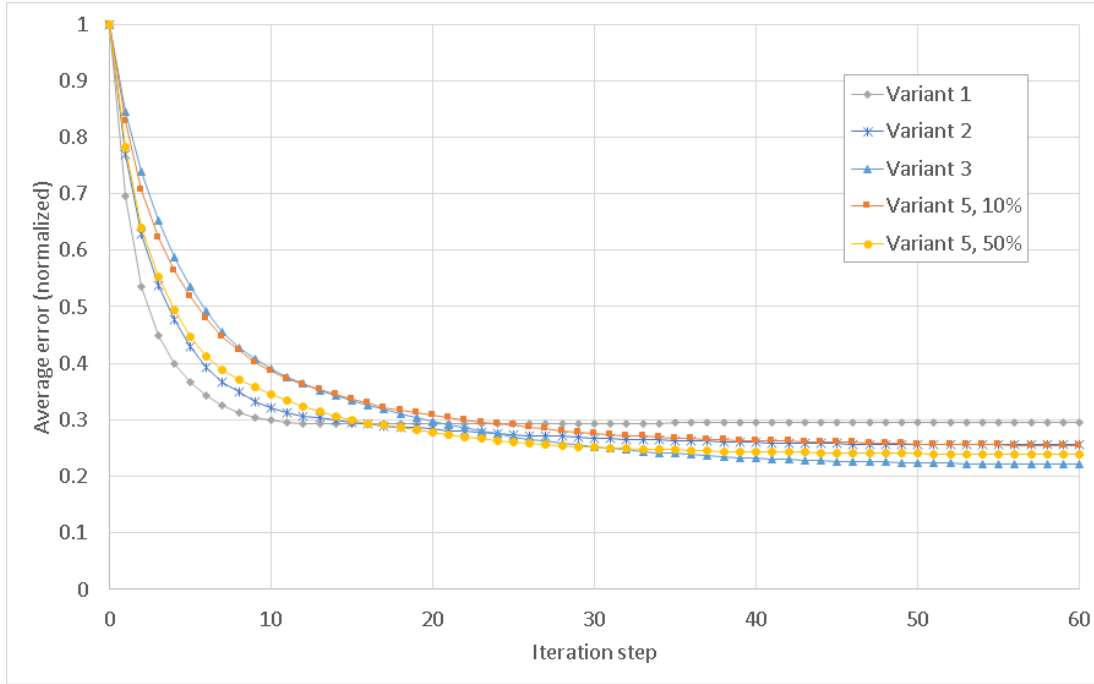


Figure 13: The average error from 25 test pairs (normalized to initial error for every pair) plotted versus the number of iteration steps for the ICP Variants 1-3 and 5. The average error decays exponentially for all variants.

ant 4, exhibiting an even lower average final error, was too slow for real-time applications.

3.5 Retinal Mapping

Combining the single high-magnification frames to a map containing the complete retina (or a big part of it) would be useful both in screening the inner of the eye for pathologies and in navigating the EyeCam. In control, a retinal map would allow the operator to select an area where the camera should be steered to for a detailed inspection or surgical procedures.

Retinal mapping requires a method that returns the transformation between the start of the mapping procedure and the current frame. This transformation gives the relative position of the current frame with respect to the other frames already contained in the retinal map. This can be used to expand the map with the structure contained in the current frame.

Two approaches were implemented for retinal mapping. The first method

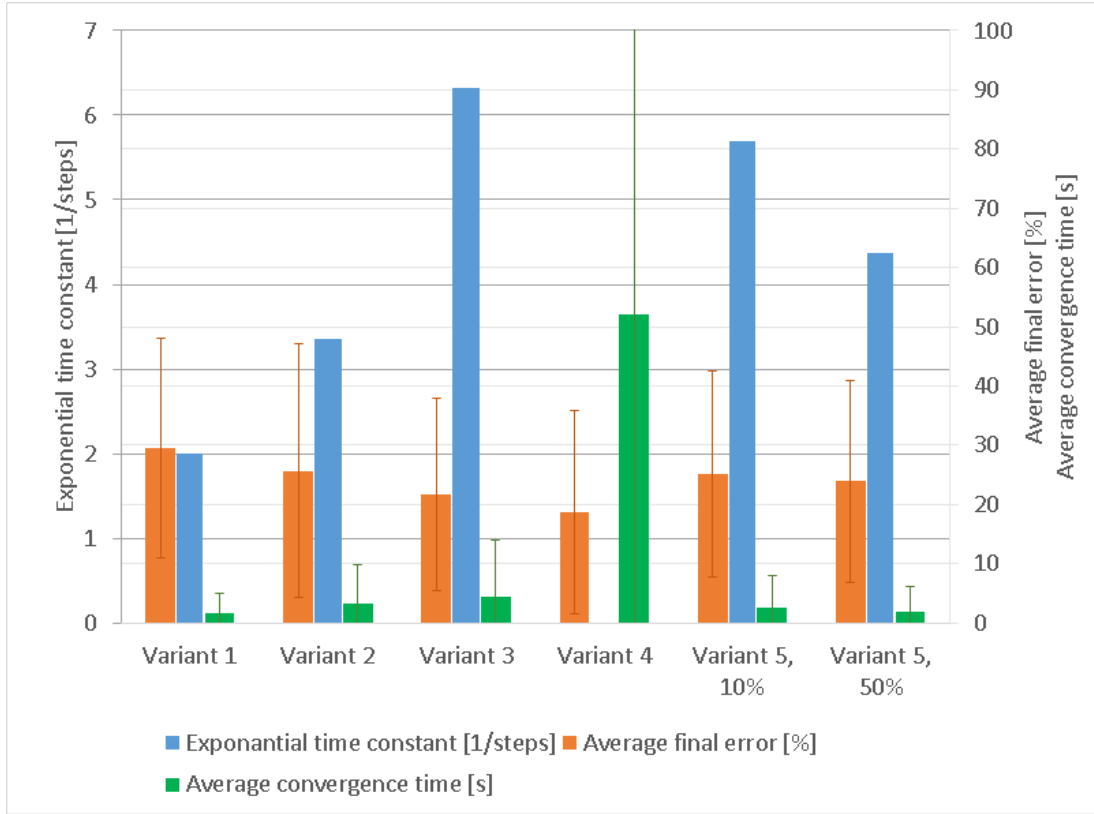


Figure 14: Comparison of the exponential time constant τ , the average final error and the average time to reach convergence for all ICP variants that were implemented. An increase in speed was generally coupled to a decrease in accuracy (and vice versa).

relies on the superposition of the transformations between subsequent frames to obtain the total transformation from the start of mapping to the current frame.

The second approach begins with drawing the centreline of the very first frame (start of mapping) into an auxiliary map M_a . For each following frame, the ICP is applied to align the current frame to M_a . Afterwards, the centreline of the current frame is also drawn onto M_a . In the second approach, the ICP directly returns the total transformation between the first frame and the current frame. This transformation can then be used to update the retinal map with the original frame (in color). Figure 15 shows two images obtained with the different approaches in mapping. For demonstration purposes, only the centrelines of the frames are used to update the maps instead of the complete color images.

This facilitates the differentiation between correctly and incorrectly calculated transformations.

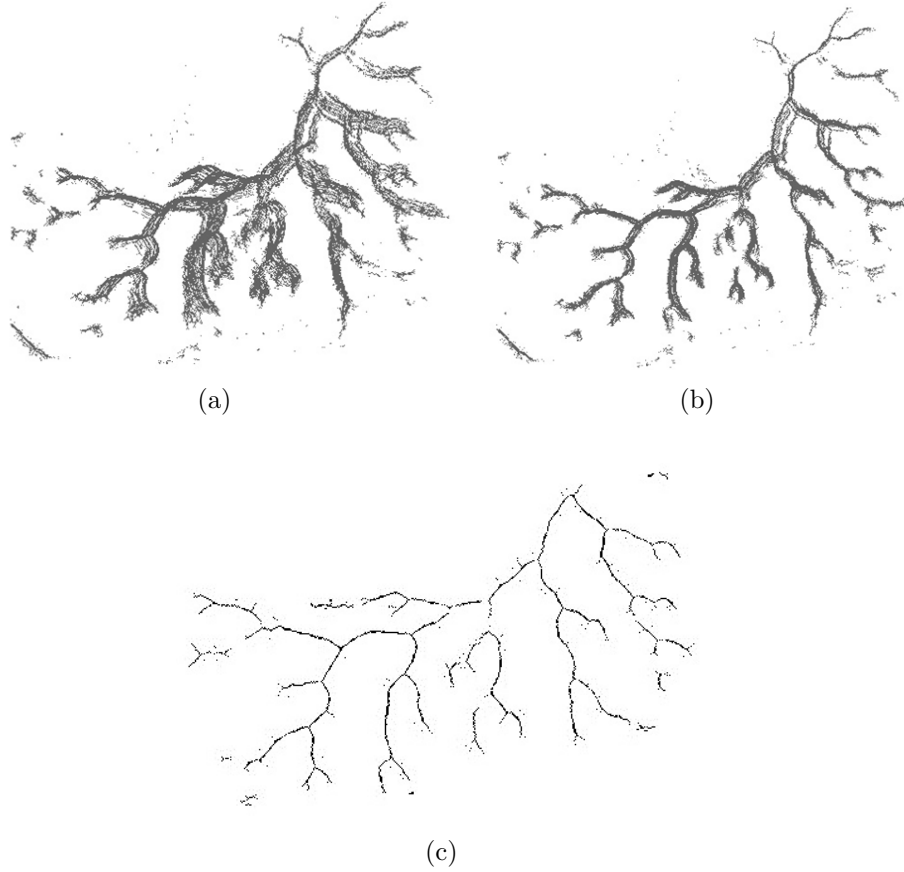


Figure 15: The mapping of the centrelines using the two different approaches superimposition (a) and auxiliary map (b) to get the transformation between the start of mapping and the current frame. A further improvement to the method using the auxiliary map was achieved by drawing the complete vessel networks into this map and skeletonizing it before aligning the current frame to it (c).

The first approach relying on the superimposition of transformations between the frames is inexact due to error propagation and the error increases with every iteration. The approach running the ICP between the auxiliary map and the current frame shows higher precision and was thus selected for the final implementation. It was further improved such that the complete vessel structure instead of only the centreline is transformed and drawn onto M_a . The skeletonization of M_a then provides a much cleaner input for the ICP, leading to a further improvement as shown in Figure 15(c).

With the transformations between the starting frame and every following frame known, the structure in the current frame can be transferred into the retinal map. This is achieved by either simply drawing the current frame on top of the map at the correct position, or by computing the average intensity between all overlapping frames. The resulting maps for a demonstration video are shown in Figure 16. It is evident that the approach averaging intensities leads to a better quality due to superresolution effects.

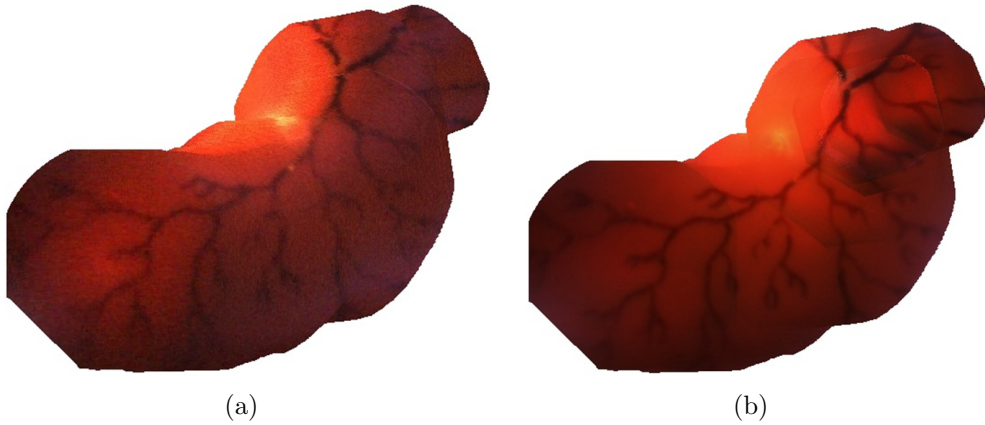


Figure 16: Two retinal maps obtained by (a) overwriting the intensities every time the global map is updated and by (b) averaging the intensity values of the different frames.

Implementation and testing of vessel detection, ICP and retinal mapping in real-time was successful in a stand-alone application. However, integration of the procedures with the Daedalus environment used in the current control of the EyeCam is not stable; the system crashes frequently. This is expected to be caused by memory leakage of parallel threads. The stability issues on the Daedalus system are currently being addressed and real-time navigation experiments will follow.

4 Summary and Contributions

A novel method to extract vessel networks from intraocular images obtained with the EyeCam was implemented and evaluated. A low-pass correction to normalize illumination changes within the different frames of a video was shown to outperform other standard contrast enhancement methods such as contrast stretching and histogram equalization. The low-pass correction was combined with binary thresholding to achieve fast feature extraction. The simplicity of the method allows for real-time detection of retinal vessels, overcoming restrictions faced by other approaches. The method is tailored to the videos from the EyeCam and produces a reliable output even from frames with high noise and luminosity drift.

Multiple variations of the ICP were successfully implemented to align vessel structures extracted from different frames and find the transformation between them. The performance of the ICP variations was analysed to identify the best suited approach for receiving visual motion feedback to navigate the EyeCam. Both the vessel detection and the ICP can process a video from the EyeCam in real-time and thus enable eye-in-hand visual servoing of the camera probe. This is an essential step towards realizing applications of the EyeCam in eye surgery. The algorithms were demonstrated by mapping the retina from an EyeCam video in real-time. Unfortunately, the implementation on the Daedalus system still faces stability issues, delaying navigation experiments. The instabilities are likely caused by memory leaks and faulty interaction between the multiple threads running in parallel.

4.1 Outlook

Many additional variations to the ICP are possible. These include accelerating changes to the baseline algorithm such as reducing the number of points taken into account for the registration or applying a good approximation of the transformation before running the ICP (e.g. Kalman filtering).

It would further be of high interest to combine different ICP variants to achieve both fast convergence and high robustness. This may be possible by applying the steps of the standard ICP (Variant 1) over the first iterations and then switching to an approach penalizing pairs with a large distance that removes multiple links

(as in Variant 3). This could combine the advantages of both methods, resulting in a fast decay in error in the beginning and an accurate convergence towards the later stages of the algorithm.

Finally, the ICP could be expanded to handle shear by splitting the features into different sub-features that are tracked independently, which would allow to extract information on the distance of the camera probe to the different features. Combining this with projecting the retinal map on a 3D sphere would provide a valuable control environment in intraocular surgical procedures.

References

- [1] S. Singhy and C. Riviere, “Physiological tremor amplitude during retinal microsurgery,” in *Bioengineering Conference, 2002. Proceedings of the IEEE 28th Annual Northeast*, pp. 171–172, IEEE, 2002.
- [2] V. Mester and F. Kuhn, “Internal limiting membrane removal in the management of full-thickness macular holes,” *American journal of ophthalmology*, vol. 129, no. 6, pp. 769–777, 2000.
- [3] D. Qatarneh, R. G. Mathew, S. Palmer, C. Bunce, and S. Tuft, “The economic cost of posterior capsule tear at cataract surgery,” *British Journal of Ophthalmology*, vol. 96, no. 1, pp. 114–117, 2012.
- [4] N. Nathoo, M. C. Çavusoglu, M. A. Vogelbaum, and G. H. Barnett, “In touch with robotics: neurosurgery for the future,” *Neurosurgery*, vol. 56, no. 3, pp. 421–433, 2005.
- [5] J. Lang, S. Mannava, A. Floyd, M. Goddard, B. Smith, A. Mofidi, T. Seyler, and R. Jinnah, “Robotic systems in orthopaedic surgery,” *Journal of Bone & Joint Surgery, British Volume*, vol. 93, no. 10, pp. 1296–1299, 2011.
- [6] I. Singh, “Robotics in urological surgery: review of current status and maneuverability, and comparison of robot-assisted and traditional laparoscopy,” *Computer Aided Surgery*, vol. 16, no. 1, pp. 38–45, 2011.
- [7] E. P. Estey, “Robotic prostatectomy: The new standard of care or a marketing success?,” *Canadian Urological Association Journal*, vol. 3, no. 6, p. 488, 2009.
- [8] T. Ueta, Y. Yamaguchi, Y. Shirakawa, T. Nakano, R. Ideta, Y. Noda, A. Morita, R. Mochizuki, N. Sugita, M. Mitsuishi, *et al.*, “Robot-assisted vitreoretinal surgery: development of a prototype and feasibility studies in an animal model,” *Ophthalmology*, vol. 116, no. 8, pp. 1538–1543, 2009.
- [9] H. Meenink and I. Thijs, *Vitreo-retinal eye surgery robot: sustainable precision*. PhD thesis, PhD thesis, TU Eindhoven, Eindhoven, The Netherlands, ISBN: 978-90-386-2800-4, 2011.

- [10] D. H. Bourla, J. P. Hubschman, M. Culjat, A. Tsirbas, A. Gupta, and S. D. Schwartz, “Feasibility study of intraocular robotic surgery with the da vinci surgical system,” *Retina*, vol. 28, no. 1, pp. 154–158, 2008.
- [11] M. Nasser, M. Eder, S. Nair, E. Dean, M. Maier, D. Zapp, C. Lohmann, and A. Knoll, “The introduction of a new robot for assistance in ophthalmic surgery,” in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp. 5682–5685, IEEE, 2013.
- [12] A. Uneri, M. A. Balicki, J. Handa, P. Gehlbach, R. H. Taylor, and I. Iordachita, “New steady-hand eye robot with micro-force sensing for vitreoretinal surgery,” in *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, pp. 814–819, IEEE, 2010.
- [13] R. A. MacLachlan, B. C. Becker, J. Cuevas Tabares, G. W. Podnar, L. A. Lobes, and C. N. Riviere, “Micron: an actively stabilized handheld tool for microsurgery,” *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 195–212, 2012.
- [14] C. Bergeles, B. E. Kratochvil, and B. J. Nelson, “Visually servoing magnetic intraocular microdevices,” *Robotics, IEEE Transactions on*, vol. 28, no. 4, pp. 798–809, 2012.
- [15] F. Ullrich, C. Bergeles, J. Pokki, O. Ergeneman, S. Erni, G. Chatzipirpiridis, S. Pané, C. Framme, and B. J. Nelson, “Mobility experiments with microrobots for minimally invasive intraocular surgery,” *Investigative ophthalmology & visual science*, vol. 54, no. 4, pp. 2853–2863, 2013.
- [16] M. P. Kummer, J. J. Abbott, B. E. Kratochvil, R. Borer, A. Sengul, and B. J. Nelson, “Octomag: An electromagnetic system for 5-dof wireless micromanipulation,” *Robotics, IEEE Transactions on*, vol. 26, no. 6, pp. 1006–1017, 2010.
- [17] F. Chaumette and S. Hutchinson, “Visual servo control. i. basic approaches,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

- [18] B. C. Becker, S. Yang, R. A. MacLachlan, and C. N. Riviere, "Towards vision-based control of a handheld micromanipulator for retinal cannulation in an eyeball phantom," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pp. 44–49, IEEE, 2012.
- [19] J. Chen, R. T. Smith, J. Tian, and A. F. Laine, "A novel registration method for retinal images based on local features," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 2242–2245, IEEE, 2008.
- [20] J. Chen, J. Tian, N. Lee, J. Zheng, R. Smith, and A. F. Laine, "A partial intensity invariant feature descriptor for multimodal retinal image registration," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 7, pp. 1707–1718, 2010.
- [21] A. Can, C. V. Stewart, B. Roysam, and H. L. Tanenbaum, "A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 3, pp. 347–364, 2002.
- [22] D. E. Becker, A. Can, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Image processing algorithms for retinal montage synthesis, mapping, and real-time location determination," *Biomedical Engineering, IEEE Transactions on*, vol. 45, no. 1, pp. 105–118, 1998.
- [23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [24] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision—ECCV 2006*, pp. 404–417, Springer, 2006.
- [25] B. C. Becker and C. N. Riviere, "Real-time retinal vessel mapping and localization for intraocular surgery," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 5360–5365, IEEE, 2013.
- [26] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 4, pp. 501–509, 2004.

-
- [27] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. D. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *Medical Imaging 2004*, pp. 648–656, International Society for Optics and Photonics, 2004.
 - [28] B. Al-Diri, A. Hunter, and D. Steel, "An active contour model for segmenting and measuring retinal vessels," *Medical Imaging, IEEE Transactions on*, vol. 28, no. 9, pp. 1488–1497, 2009.
 - [29] X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 1, pp. 131–137, 2003.
 - [30] J. V. Soares, J. J. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 9, pp. 1214–1222, 2006.
 - [31] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 3, no. 2, pp. 125–138, 1999.
 - [32] P. Bankhead, C. N. Scholfield, J. G. McGeown, and T. M. Curtis, "Fast retinal vessel detection and measurement using wavelets and edge location refinement," *PloS one*, vol. 7, no. 3, p. e32435, 2012.
 - [33] M. A. Amin and H. Yan, "High speed detection of retinal blood vessels in fundus image using phase congruency," *Soft Computing*, vol. 15, no. 6, pp. 1217–1230, 2011.
 - [34] Y. Wang and S. C. Lee, "A fast method for automated detection of blood vessels in retinal images," in *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, vol. 2, pp. 1700–1704, IEEE, 1997.

- [35] M. Sofka and C. V. Stewart, “Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures,” *Medical Imaging, IEEE Transactions on*, vol. 25, no. 12, pp. 1531–1546, 2006.
- [36] A. M. Mendonca and A. Campilho, “Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction,” *Medical Imaging, IEEE Transactions on*, vol. 25, no. 9, pp. 1200–1213, 2006.
- [37] E. Ricci and R. Perfetti, “Retinal blood vessel segmentation using line operators and support vector classification,” *Medical Imaging, IEEE Transactions on*, vol. 26, no. 10, pp. 1357–1365, 2007.
- [38] A. Bhalerao, S. Anand, and P. Saravanan, “Retinal fundus image contrast normalization using mixture of gaussians,” in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pp. 647–650, IEEE, 2008.
- [39] M. Foracchia, E. Grisan, and A. Ruggeri, “Luminosity and contrast normalization in retinal images,” *Medical Image Analysis*, vol. 9, no. 3, pp. 179–190, 2005.
- [40] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
- [41] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.
- [42] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” *International journal of computer vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [43] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets,” *Autonomous Robots*, pp. 1–16, 2013.
- [44] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 5, pp. 698–700, 1987.

- [45] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: image understanding*, vol. 54, no. 3, pp. 358–367, 1991.
- [46] C. Yang and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [47] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.
- [48] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.
- [49] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600, IEEE, 1994.
- [50] K. Pulli, "Multiview registration for large data sets," in *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pp. 160–168, IEEE, 1999.
- [51] T. Masuda, K. Sakaue, and N. Yokoya, "Registration and integration of multiple range images for 3-d model construction," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, pp. 879–883, IEEE, 1996.
- [52] C. Dorai, G. Wang, A. K. Jain, and C. Mercer, "Registration and integration of multiple object views for 3d model construction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 83–89, 1998.
- [53] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.

A Closed-form Solution to Planar Registration

Let A and B be sets of n points each. The coordinates of the points in set A (and B respectively) will be referred to as $r_{A,i}$ (and $r_{B,i}$) with $i \in \{1, 2, \dots, n\}$. The aim is to find the scale factor s , the rotation $R_\theta(r)$ and the translation t that aligns the two sets such that

$$r_{B,i} = sR_\theta(r_{A,i}) + t \quad \forall i. \quad (\text{A.1})$$

In the general case of non-perfect correspondence between the points in the two sets, Equation A.1 cannot be fulfilled for all i . Instead, the sum of the squared residual errors

$$\sum_{i=1}^n \|e_i\|^2 = \sum_{i=1}^n \|r_{B,i} - sR_\theta(r_{A,i}) - t\|^2 \quad (\text{A.2})$$

has to be minimized. If we introduce new coordinates

$$r_{A,i}^C = r_{A,i} - \bar{r}_A \text{ and } r_{B,i}^C = r_{B,i} - \bar{r}_B \quad (\text{A.3})$$

referring to the centroids defined as

$$\bar{r}_A = \frac{1}{n} \sum_{i=1}^n r_{A,i} \text{ and } \bar{r}_B = \frac{1}{n} \sum_{i=1}^n r_{B,i}, \quad (\text{A.4})$$

Equation A.2 becomes

$$\sum_{i=1}^n \|r_{B,i}^C - sR(r_{A,i}^C) - t^C\|^2 \quad (\text{A.5})$$

with $t^C = t - \bar{r}_B + sR_\theta(\bar{r}_A)$. Using that the sum of the centroid coordinates always equals zero, we can rewrite Equation A.5 as

$$\sum_{i=1}^n \|r_{B,i}^C - sR_\theta(r_{A,i}^C)\|^2 + n \|t^C\|^2. \quad (\text{A.6})$$

The above is minimized for $t^C = 0$ as the first term is independent of t^C and the second term is always non-negative. It follows that the optimal translation equals the difference between the centroid of B and the scaled and rotated centroid of

A , or

$$t = \overline{r_B} - sR(\overline{r_A}) . \quad (\text{A.7})$$

Using $t^C = 0$ and the fact that lengths are preserved under rotation, Equation A.6 can be expanded to

$$\sum_{i=1}^n \| r_{B,i}^C \|^2 - 2s \cdot \sum_{i=1}^n r_{B,i}^C \cdot R_\theta(r_{A,i}^C) + s^2 \sum_{i=1}^n \| r_{A,i}^C \|^2 . \quad (\text{A.8})$$

It is straightforward to see that Equation A.7 is minimized when the term

$$\sum_{i=1}^n r_{B,i}^C \cdot R_\theta(r_{A,i}^C) \quad (\text{A.9})$$

is at its maximum, independent of the scale factor. This allows to determine scale and rotation separately. In this report, the discussion on finding the optimal rotation is limited to the case where the two sets A and B are coplanar. The rotated centroid coordinates in set A will further be referred to as $\tilde{r}_{A,i}^C$ with θ denoting the angle of rotation and α_i the angle between the points $r_{A,i}^C$ and $r_{B,i}^C$. Rotating $r_{A,i}$ by θ decreases the angles α_i . It is easy to see that the angle between $\tilde{r}_{A,i}^C$ and $r_{B,i}^C$ becomes $\alpha_i - \theta$. Using the geometric definition of the scalar product (and again preservation of lengths under rotation), Equation A.9 can be rewritten as

$$\sum_{i=1}^n \| r_{B,i}^C \| \| r_{A,i}^C \| \cos(\alpha_i - \theta) = C \cos \theta + S \sin \theta \quad (\text{A.10})$$

with

$$C = \sum_{i=1}^n \| r_{B,i}^C \| \| r_{A,i}^C \| \cos \alpha_i = \sum_{i=1}^n r_{B,i}^C \cdot r_{A,i}^C \quad (\text{A.11})$$

and

$$S = \sum_{i=1}^n \| r_{B,i}^C \| \| r_{A,i}^C \| \sin \alpha_i = \sum_{i=1}^n (r_{B,i}^C \times r_{A,i}^C) \cdot n_A , \quad (\text{A.12})$$

where n_A denotes the normal vector to the plane containing A and B . Equation A.10 then obtains extrema for $C \sin \theta = S \cos \theta$ with its maximum at

$$\sin \theta = \frac{S}{\sqrt{S^2 + C^2}} \quad \text{and} \quad \cos \theta = \frac{C}{\sqrt{S^2 + C^2}} . \quad (\text{A.13})$$

Minimizing Equation A.8 with respect to s results in a scale factor that is

asymmetrical to exchanging the sets A and B (refer to [47] for details). As this is generally not desirable, it is better to define the residual error symmetrically as

$$e_i = \frac{1}{\sqrt{s}} r_{B,i}^C - \sqrt{s} R(r_{A,i}^C) , \quad (\text{A.14})$$

which results in a total error term of

$$\frac{1}{s} \sum_{i=1}^n \| r_{B,i}^C \|^2 - 2 \cdot \sum_{i=1}^n r_{B,i}^C \cdot R(r_{A,i}^C) + s \sum_{i=1}^n \| r_{A,i}^C \|^2 . \quad (\text{A.15})$$

This can be written as $s^{-1}S_B - 2D + sS_A$ or, completing the square in s , as

$$(\sqrt{s}S_A - \frac{1}{\sqrt{s}}S_B)^2 + 2(S_AS_B - D) \quad (\text{A.16})$$

Finally, Equation A.16 is minimized for

$$s = S_B/S_A = \left(\frac{\sum_{i=1}^n \| r_{B,i}^C \|^2}{\sum_{i=1}^n \| r_{A,i}^C \|^2} \right)^{\frac{1}{2}} , \quad (\text{A.17})$$

If the expected error is not the same for all point pairs, it is advantageous to introduce the weight w_i for the i -th pair, such that Equation the total error computes as

$$\sum_{i=1}^n w_i \| e_i \|^2 . \quad (\text{A.18})$$

This only changes the solution slightly and Equations A.4, A.11, A.12, and A.17 become

$$\overline{r}_A = \frac{\sum_{i=1}^n w_i r_{A,i}}{\sum_{i=1}^n w_i} , \quad \overline{r}_B = \frac{\sum_{i=1}^n w_i r_{B,i}}{\sum_{i=1}^n w_i} , \quad (\text{A.19})$$

$$C = \sum_{i=1}^n w_i r_{B,i}^C \cdot r_{A,i}^C \quad (\text{A.20})$$

$$S = \sum_{i=1}^n w_i (r_{B,i}^C \times r_{A,i}^C) \cdot n_A , \quad (\text{A.21})$$

and

$$s = \left(\frac{\sum_{i=1}^n w_i \| r_{B,i}^C \|^2}{\sum_{i=1}^n w_i \| r_{A,i}^C \|^2} \right)^{\frac{1}{2}} . \quad (\text{A.22})$$

B Performance of the ICP variants

Table 1 shows the numeric results of the performance evaluation of the different ICP variations (standard deviation indicated in brackets).

Table 1: Performance of the ICP variants.

	Exponential time constant [1/steps]	Exponential fit: R^2	Average final error [%]	Average convergence time [s]
Variant 1	1.666	0.998	29.4 ± 18.5	1.58 ± 3.38
Variant 2	3.354	0.991	25.7 ± 21.4	3.17 ± 6.54
Variant 3	6.323	0.987	21.7 ± 16.1	4.43 ± 9.55
Variant 4	-	-	18.7 ± 17.1	52.06 ± 104.60
Variant 5, 10%	5.695	0.985	25.1 ± 17.4	2.55 ± 5.47
Variant 5, 50%	4.372	0.980	23.9 ± 17.0	1.98 ± 4.12