Universidad Zaragoza
1542

Final Master Thesis

Optical flow estimation in systems with co-located light and camera for monocular endoscopy

Estimación de flujo óptico en sistemas con luz y cámara solidarios para endoscopia monocular

Author

Diego Royo Meneses

Supervisor

José María Martínez Montiel

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2022

Acknowledgements

Many thanks to my supervisor, Prof. José María Martínez Montiel, for his dedicated support and guidance throughout this project. Also, thanks to other members of the EndoMapper project that helped with this master thesis: Anita Rau, Javier Morlana and, most importantly, Víctor Martínez and Prof. Juan Domingo Tardós Solano.

Finally, thanks to my family, friends and fellow master's degree colleagues for their continued support. You have made this journey a worthy one.

This project has received funding from the European Union's
Horizon 2020 research and innovation program under
grant agreement No 863146

Abstract

The European EndoMapper project aims to develop real-time mapping of the interior of the human body using only footage from an exploration procedure. This technology can enable novel medical operations that include robotized autonomous interaction and live augmented reality. All of this comes down to two principles that need to be overcome: first, generating a map of the human body, and then, being able to locate oneself within it. Simultaneous Localization And Mapping (SLAM) is a computer vision problem that tries to perform both tasks at the same time. For the case of an endoscopic procedure, the input to the SLAM algorithm is a monocular video sequence that is captured during exploration. Usually, Visual SLAM (VSLAM) algorithms are based on image matches i.e. they try to find parts of the environment that appear on two or more frames.

In this Master Thesis, we explore and build on optical flow estimation, that is, algorithms that attempt to compute how each pixel moves between two images. Pixel motion is able to give dense correspondences for a video sequence. Most existing methods assume that the brightness of each pixel is constant regardless of where the camera is located. This is a good choice in most cases, for example, in outdoor scenes where diffuse objects are illuminated by the sun. In an endoscopy, the light and camera are co-located: changes in the position of the camera are correlated with changes of illumination. Moving the camera closer to an object makes it appear brighter.

Our work explores two approaches to solve this problem. First, we develop a photometric model of light transport with a co-located light and camera. We introduce this model in an existing estimation algorithm and we are able to extract more precise image matches along with additional information from depth and surface normals. Secondly, we explore learning-based approaches. They have the great advantage of not requiring a hand-crafted illumination model, and their high-dimensional parameters are able to be learned using a large amount of training data. With the current technology, it is not possible to obtain enough ground truth optical flow in real endoscopy sequences. We explore different simulation environments and find that using a combination of real and synthetic data is key. With this, we obtain a 40% error reduction on optical flow estimation when evaluating on simulated data, and a 15% on captured data. Additionally, we show that mixing both types of training data produces much better qualitative results for other scene points whose ground truth is not available.

Index

Chapter 1

Introduction

When presented with a video or a sequence of images, computer vision seeks to understand and automate tasks that the human visual system can do, even trying to surpass its abilities. One of the most thoroughly studied problems of this field is Simultaneous Localization And Mapping or SLAM. As its name implies, it combines the tasks of generating a map of an unknown environment while trying to locate an agent inside it at the same time (see Figure 1.1). There exist multiple algorithms able to generate real-time approximate solutions using different kinds of sensors.

This Master Thesis is part of the EndoMapper project, which aims to develop the fundamentals of real-time localization and mapping inside the human body using only the video stream provided by a standard endoscope. Solving the SLAM problem opens the door for medical augmented reality, novel medical procedures, and robotized autonomous interaction with superhuman accuracy.

Endoscopes used nowadays are only equipped with a monocular camera. For this reason, the EndoMapper project is focused on Visual SLAM algorithms that work with a video sequence. They are based on image matches, that is, they try to find parts of the environment that appear on two or more frames. Understanding how each pixel changes and moves between images is our main goal, a problem known as optical flow.

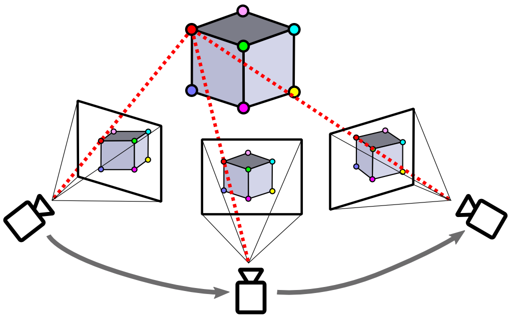
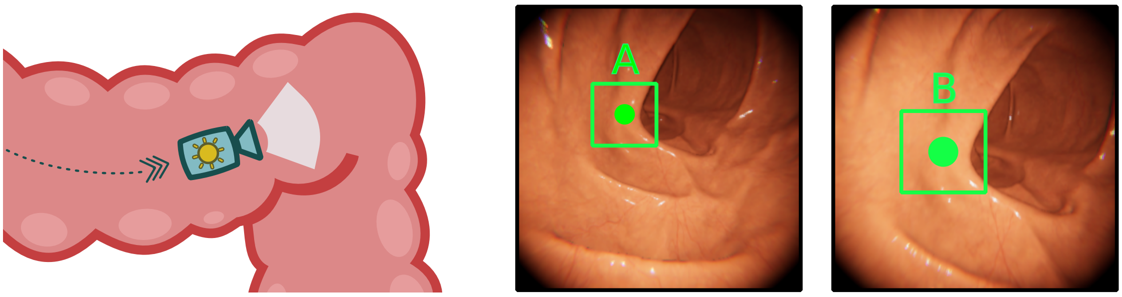


Figure 1.1: An agent is travelling through a scene. Using only the video frames, Visual SLAM algorithms can generate a map of the cube and locate the camera with respect to it. It works by finding common matching points between images.



(a) Co-located light and camera. (b) Matching patches (green), different illumination.

Figure 1.2: During an endoscopy procedure, moving the camera instrument also moves the light source. The illumination changes are correlated with this motion.

With this, the main objective of this thesis is the estimation of optical flow in real medical sequences. To obtain a solution, traditional theoretical models usually assume that the illumination for the scene remains constant [6]. While this is a good premise for many indoor and outdoor sequences, it describes poorly what happens on an endoscopy procedure. In this situation, the light source is fixed with the camera, known as co-located light, and produces notable illumination changes (see Figure 1.2). There are existing Shape-from-Shading and photometric stereo methods that model the illumination to reconstruct scene geometry in medical environments [3, 5, 14, 19]. In our work, we propose novel algorithms and approaches for optical flow estimation that also model light propagation inside the human body. Using a photometric approach does not only add robustness to illumination changes, but it can extract further information about the scene and the camera.

We mainly focus on learning-based methods, which have gained popularity over the recent years due to their great success with optical flow estimation from a pair of images [4, 8]. Our working hypothesis is that they can be trained to include a much more complex illumination and motion model that any hand-crafted approach could deliver. As a large amount of input data is needed, we rely on captured and simulated endoscopy procedures, and show how to combine them to obtain superior results.

1.1 Objectives

To achieve our goal of estimating optical flow in real captured endoscopy sequences, we pursue the next objectives. They are structured in the following chapters:

- In Chapter 2 we propose an illumination model for a medical endoscopy setting, explaining how an image is formed with a co-located light source and camera. Finally, a calibration procedure is performed using real captured data.

- In Chapter 3, we explore and develop analytical and learning-based methods for optical flow estimation. We present different training datasets of captured and simulated colonoscopy procedures and analyze how well they fit the illumination and motion changes that happen on a real endoscopy.

- In Chapter 4 we study RAFT for optical flow, a deep neural network, to understand how it produces results when used in an environment with a co-located light and camera.

- In Chapter 5, we introduce an evaluation metric with quantitative data from simulated and, most importantly, captured sequences, and use it to evaluate the different presented algorithms.

Chapter 2

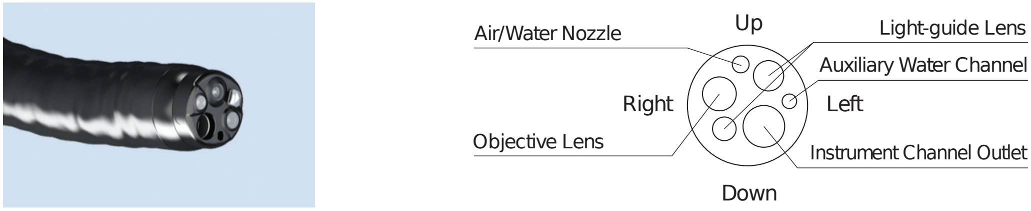
Photometric model

The objective of this section is to present a model that is capable of describing light emission, propagation and capture in a medical endoscopy setting. Understanding the image formation process is a key part of our work, allowing us to extract dense information directly from pixel brightness. Other feature-based algorithms work on a sparse set of interest points which typically select high-contrast patches. Instead, a photometric approach is especially useful for our domain, as the appearance of the interior of the human body is very low-textured, with many similar areas.

Figure 2.1 shows one of the endoscope models used in the EndoMapper project. The distal end contains the light-guide and objective lenses. The first one illuminates the scene, and the second one captures the reflected light to produce an image. They are very closely placed, given the insertion tube has a diameter of 9.2mm.

2.1 Co-located light and camera model

The model can be divided into three parts: light emission, surface reflectance and image capture. The main idea is to look at direct light transport to explain how the brightness values in the final image are formed. Note that interreflections in the scene (i.e. global illumination) are not taken into account, as they cannot be easily modelled and the direct component has proven to give good results. Figure 2.2 shows an overview of the different parts of our model.



(a) Insertion tube.

(b) Distal end.

Figure 2.1: Olympus EVIS EXERA III GIF-H190 endoscope.

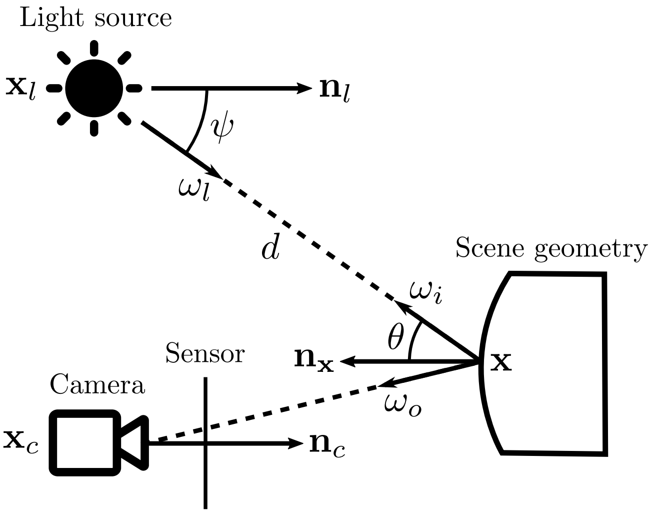


Figure 2.2: Overview of the co-located light and camera model.

Light emission model. One of the simplest ways to parametrize an emitter is with a point light source, represented as an infinitesimally small point $\mathbf{x}_l$ which sends light out in all directions equally. It is subject to the inverse-square law, that is, it decays proportionally to the square of the distance travelled. With this, the radiance L_e received by a point $\mathbf{x}$ in the scene is modelled as

$$L_e(\mathbf{x}) = \frac{\sigma_e}{\|\mathbf{x} - \mathbf{x}_l\|^2}, \tag{2.1}$$

where σ_e is the intensity of the source, which can be seen as the radiance measured at unit distance. However, the light-guide lens of the endoscope does not emit the same radiance in all directions, and thus an isotropic point light source is limited for our case. We extend the emission model to a spotlight which adds falloff $\mu \in [0, 1]$ based on the direction ω_l from the light to the illuminated point $\mathbf{x}$. The spotlight is parametrized with a point $\mathbf{x}_l$ and a direction $\mathbf{n}_l$ where light is emitted with maximum intensity. Consider the direction ω_l when illuminating $\mathbf{x}$:

$$\omega_l = \frac{\mathbf{x} - \mathbf{x}_l}{\|\mathbf{x} - \mathbf{x}_l\|^2}. \tag{2.2}$$

The value of the spread function $\mu(\psi)$ depends on the angle ψ between ω_l and $\mathbf{n}_l$. Typically, a cosine falloff is used

$$\mu_{\boldsymbol{\alpha}}(\psi) = \cos^{\alpha} \psi, \quad \psi = \angle\left(\omega_l, \mathbf{n}_l\right), \tag{2.3}$$

or a polynomial decay, more general, for which we prefer to use,

$$\mu_{\boldsymbol{\alpha}}(\psi) = 1 + \alpha_1 \psi^2 + \alpha_2 \psi^4 + \alpha_3 \psi^6 \tag{2.4}$$

where $\boldsymbol{\alpha}$ adjusts the spread. With this, the radiance emitted by a spotlight is

$$L_e(\mathbf{x}) = \mu_{\boldsymbol{\alpha}}(\psi) \cdot \frac{\sigma_e}{\|\mathbf{x} - \mathbf{x}_l\|^2} \tag{2.5}$$

with four parameters: the emission σ_e and spread $\boldsymbol{\alpha}$ of the light source, as well as its position $\mathbf{x}_l$ and orientation $\mathbf{n}_l$. For simplicity, it can be safely assumed that the light source is exactly co-located with the camera, leaving only two parameters. However, as seen in Section 2.2, performing extra tuning yields better results in our test case by adding a small separation from the light to the camera.

Surface reflectance. Human tissue has many complex interactions with light and has been thoroughly studied in computer graphics. In the general case, it is parameterized using the concept of the Bidirectional Reflectance Distribution Function, shortened as BRDF, $f_r(\omega_i, \omega_o)$ which determines the ratio of incident light with direction ω_i that gets redirected in the outgoing direction ω_o at point $\mathbf{x}$ (see Figure 2.3). Defining this function is not trivial for the interior of the human body, as multiple phenomena happen, like subsurface scattering between multiple layers of tissue. One reasonable simplification for this is the Lambertian BRDF model

$$f_r(\omega_i, \omega_o) = \frac{\rho}{\pi}, \tag{2.6}$$

that defines an ideal matte or diffusely reflecting surface. Note that the incident and outgoing directions ω_i, ω_o do not matter. The fraction of reflected light is modelled by the albedo ρ, the only parameter. The π term is necessary for normalization over the hemisphere of possible values for ω_i and ω_o.

The main limitation of this model comes from specular highlights, caused by mirror-like properties of the surface. This happens in points of the image where ω_o is close to the perfect specular direction

$$\omega_o \approx \omega_r, \quad \omega_r = 2(\omega_i \cdot \mathbf{n}_\mathbf{x})\mathbf{n}_\mathbf{x} - \omega_i, \tag{2.7}$$

where $\mathbf{n}_\mathbf{x}$ is the surface normal at point $\mathbf{x}$. Note that, because the light and the camera are co-located, ω_o is very close to ω_i, if not assumed to be equal. As such, Equation (2.7) holds true when $\omega_i \approx \mathbf{n}_\mathbf{x}$ (and thus, $\omega_o \approx \mathbf{n}_\mathbf{x}$). We could extend the model to account for these reflections [11], however, we choose to discard points where $\omega_o \approx \omega_r$.

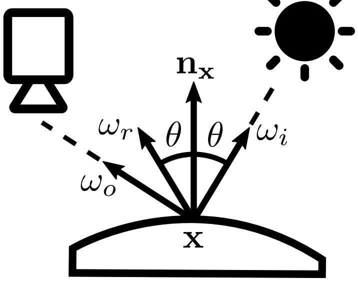


Figure 2.3: Surface reflectance overview. Note the perfect specular direction ω_r.

Finally, there is one last term to take into account. Varying the incident angle ω_i also changes the area of the projected solid angle of the light source, and thus affects the amount of reflected light L_o at $\mathbf{x}$ with a cosine weight of the angle θ:

$$L_o(\mathbf{x}, \omega_i, \omega_o) = L_e(\mathbf{x}) \cdot f_r(\omega_i, \omega_o) \cdot |\mathbf{n}_\mathbf{x} \cdot \omega_i| \tag{2.8}$$

where $|\mathbf{n}_\mathbf{x} \cdot \omega_i| = \cos\theta$, and L_e is the emitter radiance from Equation (2.5).

Image capture model. The objective lens focuses the incoming light to create an image based on the camera processing pipeline. Due to the shape and physical properties of this focusing system, not all points of the sensor receive the same amount of radiance proportionally. This is an effect known as vignetting, and it causes pixels further from the optical center to appear dimmer. As it can be quite complex, we model it using a polynomial decay, similar to Equation (2.4), based on the pixel distance r of each pixel $\mathbf{u}$ to the optical center $\mathbf{c}$, normalized to the $[0, 1]$ range. For example, $r = 0$ when $\mathbf{u} = \mathbf{c}$, and $r = 1$ when $\mathbf{u}$ is the pixel furthest away from $\mathbf{c}$.

$$V_{\boldsymbol{\beta}}(\mathbf{u}) = 1 + \beta_1 r^2 + \beta_2 r^4 + \beta_3 r^6, \quad r = \|\mathbf{u} - \mathbf{c}\| \in [0, 1] \tag{2.9}$$

where $\boldsymbol{\beta}$ adjusts the radial attenuation. The optical center $\mathbf{c}$ can be obtained through geometric calibration and is not necessarily located in the center of the image.

When light reaches the sensor, it first captures incoming photons for a set exposure time, resulting in an analog signal which is then amplified (ISO control) to produce brightness values that are in the correct range and do not appear too dark or too bright. Exposure time and ISO settings can be both modelled as a multiplying factor g_t for the incoming radiance

$$S_t(\mathbf{u}) = g_t \cdot V_{\boldsymbol{\beta}}(\mathbf{u}) \cdot L_o(\mathbf{u}) \tag{2.10}$$

where $L_o(\mathbf{u})$ represents the incoming irradiance at pixel $\mathbf{u}$, similar to Equation (2.8). Strictly speaking, $L_o(\mathbf{u})$ integrates the incoming radiance over the area of the pixel $\mathbf{u}$ instead of just a single point $\mathbf{x}$. However, the area of the pixel remains constant as another multiplying factor. S_t represents the sensor state of the image at time t, as the camera parameters like g_t can change over time, even if L_o remains the same.

To obtain the final result, the signal is converted from analog to digital brightness values, which are post-processed to enhance the quality of the image. The specific digital signal processing operations of each camera model are specific to each manufacturer and typically private. It can include white balance, color correction, JPEG encoding and others. However, we only account for gamma encoding in our model, as the others do not produce significant brightness changes for our case. This step increases the perceived dynamic range of the image using the non-linear manner in which humans perceive color. Thus, we introduce a gamma parameter γ such that

$$I_t(\mathbf{u}) = S_t(\mathbf{u})^{1/\gamma} \tag{2.11}$$

where I_t is the final, captured image.

2.1.1 Complete model

Combining every part, the resulting model explains how an image I_t at time t is formed:

$$I_t(\mathbf{u}) = \left(\mu_{\boldsymbol{\alpha}}(\psi) \cdot \frac{\sigma_e}{\|\mathbf{x}_l - \mathbf{x}\|^2} \cdot \frac{\rho}{\pi} \cdot |\mathbf{n}_{\mathbf{x}} \cdot \omega_i| \cdot g_t \cdot V_{\boldsymbol{\beta}}(\mathbf{u}) \right)^{1/\gamma} \tag{2.12}$$

where $\mathbf{x}$ is the world point, with normal $\mathbf{n}_{\mathbf{x}}$, which projects to the image at pixel $\mathbf{u}$. This is agnostic to the projection model and can be used, for example, in pinhole and fisheye cameras.

The parameters of the model are $\Omega = \{\sigma_e, \alpha_i, \mathbf{x}_l, \mathbf{n}_l, \rho, g_t, \beta_j, \gamma\}$ with $i, j \in [1, 3]$ parameters for light spread and vignetting, and $t \in [1, T]$, as the gain can change over time, where T is the number of images or time steps in the video sequence.

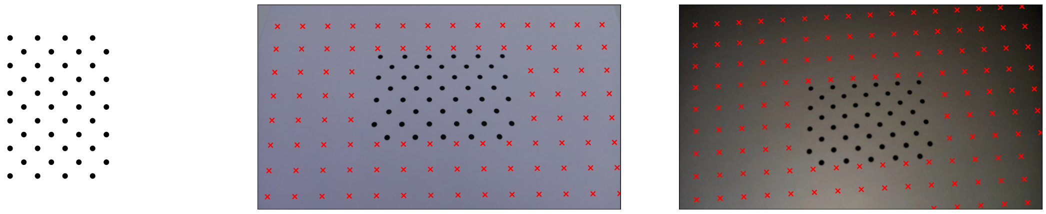
2.2 Photometric calibration

The calibration process is used to obtain a set of parameters Ω that best fit our model to a set of captured images I'. More formally, we want to obtain

$$\Omega^* = \arg\min_{\Omega} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t \in [1,T]} \mathcal{L}\left(I_t(\mathbf{u}; \Omega_t) - I'_t(\mathbf{u})\right) \tag{2.13}$$

where $I_t(\Omega_t)$ denotes the synthetic image generated by our model for the time step t. All the model parameters Ω remain constant through time, except for g_t. We use a set of world points $\mathcal{X}$, each of which $\mathbf{x}$ projects to a pixel $\mathbf{u}$ in the modelled and captured images. Points that fall outside of the image are not taken into account, that is, $I_t(\mathbf{u}) = I'_t(\mathbf{u}) = 0$ for pixels $\mathbf{u}$ outside the image. The result of the robust error function $\mathcal{L}$ is minimized using a Levenberg–Marquardt algorithm.

A known black and white asymmetric circleboard pattern is used to locate the light and camera. This serves two purposes. First, depth and cosine attenuation (see Equations (2.5) and (2.8), respectively) can be calculated. Second, it can safely be assumed that the albedo in the white squares is constant. The black squares are not used in the optimization as they do not reflect nearly as much light. We select evenly spaced points in the white areas, as seen in Figures 2.4b and 2.4c.



(a) Circleboard pattern. (b) Uniform illumination. (c) Camera flash illumination.

Figure 2.4: Pattern and scene points used for calibration, with the two setups used. Images (b) and (c) show the subset of $\mathcal{X}$ (red crosses) that is visible with white albedo.

Removing the ambiguities that come from the depth, cosine and albedo terms allows the model to be calibrated. This only leaves one scale factor, as the emitted radiance σ_e is coupled with the albedo ρ. For example, if the radiance received by the sensor is halved by two, it can be due to a dimmer emitter or a less reflective surface. However, this is not a problem for our purpose of optical flow estimation. In any case, there exist materials with known albedo, but they are more expensive than a sheet of paper. For our case, we assume $\rho = 1$ for all points.

Both the light spread and lens vignetting use similar formulas, and can become coupled if the light source and camera are closely located. As this is the case with our model, we perform a two-step calibration. First, the circleboard pattern is photographed under direct, uniform sunlight with the camera flash turned off, as seen in Figure 2.4b. This allows to remove the light source from the equation, especially α_{1-3}, leaving β_{1-3} for vignetting with no coupling problems. A second set of calibration images is taken in a dark room only illuminated by the camera flash, as seen in Figure 2.4c. The vignetting parameters β_{1-3} remain fixed from the previous estimation, which lets us optimize for α_{1-3} independently.

2.2.1 Calibration results

As a first approach, we use a smartphone as a capture device. For our purposes, it is very similar to an endoscope as the camera flash can also be modelled using a spotlight that is co-located with the capture lens. Concretely, it has a Sony IMX686 Exmor RS sensor with a dual-LED flash which we model as a single light source. This model has also been used to calibrate a real endoscope [11], yielding good results.

For the robust loss function $\mathcal{L}$, we use a Huber loss based on a parameter δ

$$\mathcal{L}_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \leq \delta, \\ \delta\left(|x| - \frac{1}{2}\delta\right) & \text{otherwise,} \end{cases} \tag{2.14}$$

which is quadratic for small values of x, and linear for large values. We calculate δ as the 10% of the largest brightness value out of all the input images.

Previous steps. The photometric calibration process requires the camera to be geometrically calibrated with a set of image and world space point matches.

The intrinsic parameters include focal distances, optical center $\mathbf{c}$ and distortion coefficients to undistort the input images. Extrinsics include the camera position $\mathbf{x}_c$ and rotation matrix R_c, obtained using a Perspective-n-Point (PnP) [2] method.

Vignetting calibration. As mentioned earlier we use a set of 20 images, similar to those in Figure 2.4b, to separate β_{1-3} from α_{1-3}. Using dedicated software in the target smartphone allows capturing RAW images, which have ground truth values for the gain g_t and gamma encoding γ parameters. Thus, we only optimize for β_{1-3}. The RAW images have been minimally processed to remove the Bayer filter [1] and converted to grayscale (see Figure 2.5a).

As we are using ambient illumination instead of a spotlight source, we estimate the outgoing radiance L_o in Equation (2.8) from the brightness values in the first image by applying the inverse operations for the camera model

$$L_o(\mathbf{x}) = \left(\frac{I'(\mathbf{u})}{g_t \cdot V_{\boldsymbol{\beta}}(\mathbf{u})} \right)^{\gamma}. \tag{2.15}$$

Note that g_t also needs to account for the conversion from pixel irradiance to radiance. The values of $L_o(\mathbf{x})$ calculated from the first image are used to estimate brightness values $I(\mathbf{u})$ on the rest of the calibration set.

Natural vignetting is typically approximated by $\cos^4 \theta$ of the incident angle θ of the light in the sensor array, which provides a good initial estimation. Figure 2.5b shows the first and last steps in the optimization. Results will be used in the next step.

We obtain a mean absolute error of 1.53 brightness values or 3.6%, with a standard deviation of 1.61 or 3.8%. As seen in Figures 2.5c and 2.5d, there exist outliers coming from specular reflections which cannot be removed, hence the need for robust loss.

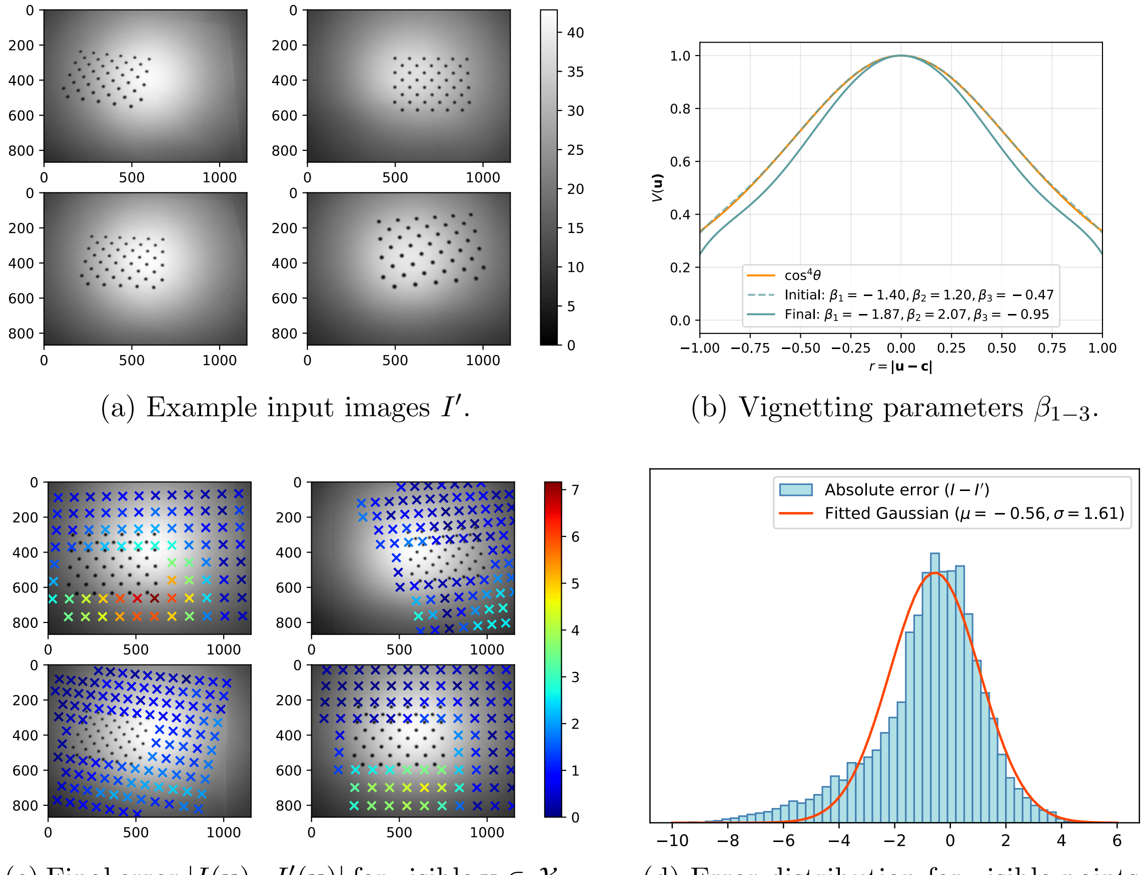
Complete calibration. For the last step a set of 20 images is taken in a dark room illuminated by the camera flash. Knowing the position $\mathbf{x}_c$ and rotation matrix R_c of the camera, the parameters of the light source $\mathbf{x}_l$, $\mathbf{n}_l$ can be formulated as

$$\mathbf{x}_l = \mathbf{x}_c + \Delta\mathbf{x}, \quad \Delta\mathbf{x} \in \mathbb{R}^3, \tag{2.16}$$

$$\mathbf{n}_l = R_c \begin{pmatrix} \sin\Delta\psi \cos\Delta\phi \\ \sin\Delta\psi \sin\Delta\phi \\ \cos\Delta\psi \end{pmatrix}, \quad \Delta\psi \in [0, \pi], \Delta\phi \in [0, 2\pi), \tag{2.17}$$

where $\Delta\psi$ and $\Delta\phi$ represent spherical coordinates. They are initialized to zero, such that $\mathbf{n}_l$ corresponds to the camera's forward direction. The position $\Delta\mathbf{x}$ is initialized to $(-1.25\,\text{mm}, -0.95\,\text{mm}, 0\,\text{mm})^{\mathrm{T}}$, obtained from smartphone measurements. This is represented in Figure 2.6.

Adding to the five parameters in Equations (2.16) and (2.17), we also optimize four more variables. First, the light spread α_{1-3} with values close to a typical cosine-weighted model [5]. Finally, light emission σ_e can be initialized to any positive value. Figure 2.7 shows the final results obtained. During the optimization, the camera is moved less than a millimetre from its original position, i.e. $\Delta\mathbf{x}$ does not change, and is rotated with $\Delta\psi = 0.043$ rad towards the camera. For the final complete model, we obtain a mean absolute error of 0.23 brightness values or 1.6%, with a standard deviation of 1.6 or 1.3%. As we can remove most of the specular points following Equation (2.7), these metrics are lower when compared to the vignetting calibration. We can confidently state that our model fits the camera well. The next sections do not use the calibrated parameters, as the generated datasets employ different capture setups. Nevertheless, the main properties of the photometric model remain the same.



(a) Example input images I'.

(b) Vignetting parameters β_{1-3}.

(c) Final error $|I(\mathbf{u}) - I'(\mathbf{u})|$ for visible $\mathbf{x} \in \mathcal{X}$.

(d) Error distribution for visible points.

Figure 2.5: Vignetting calibration results. We project all world points $\mathbf{x} \in \mathcal{X}$ to all 20 input images I'_{1-20}, and filter points that are visible in both I'_1 and I'_i, $i \in 2, 20$. (a) shows brightness values for some sample images. (b) shows the initial and final vignetting parameters based on the normalized distance r to the optical center $\mathbf{c}$. The error distribution for the final parameters can be seen in (c) and (d), displayed on the the previously filtered points. Specular reflections cause some outliers with high negative error.

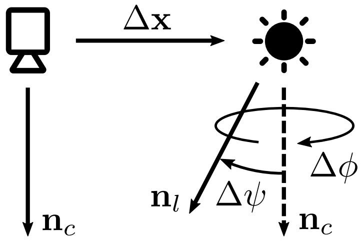
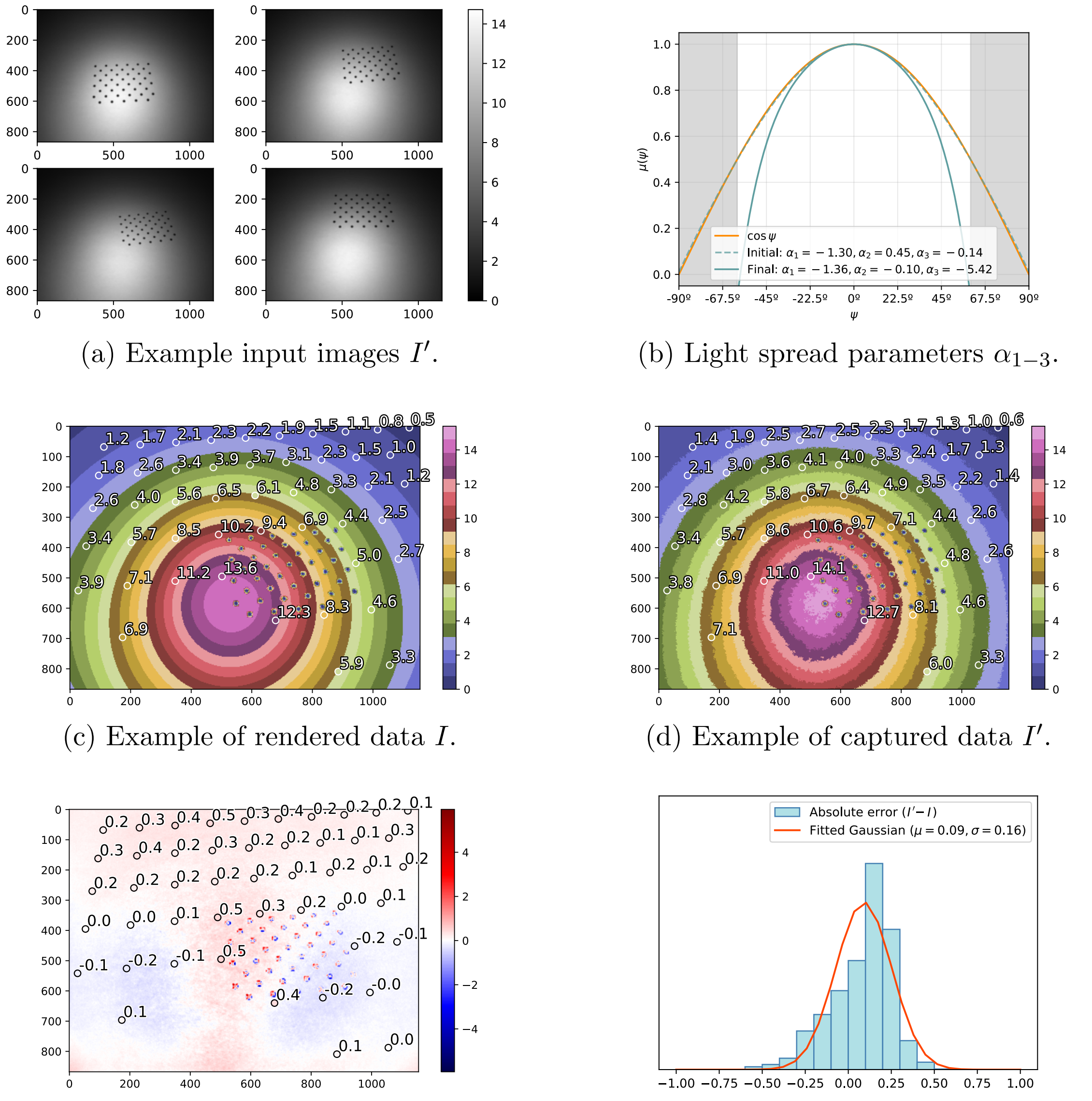


Figure 2.6: Relative position ($\Delta\mathbf{x}$) and rotation ($\Delta\psi$, $\Delta\phi$) used in the optimization.



(a) Example input images I'.

(b) Light spread parameters α_{1-3}.

(c) Example of rendered data I.

(d) Example of captured data I'.

(e) Example of error $I' - I$.

(f) Error distribution for visible points.

Figure 2.7: Complete calibration. We project all world points $\mathbf{x} \in \mathcal{X}$ to all 20 input images I'_{1-20}, and filter points that are visible in each image. (a) shows brightness values for some sample images. (b) shows the initial and final light spread parameters based on the angle ψ of the ray with respect to the camera. The grey area corresponds to values of ψ that are not present in the input data i.e. outside of the field of view of the light. The error distribution for the final parameters, subtracting points of (c) from (d), can be seen in (e) and (f).

Chapter 3

Optical flow

Consider a video sequence where the camera and the scene objects can move in any way. This relative 3D motion is projected to the image and results in 2D displacements for each pixel, which define the optical flow field for each frame. However, there is no guarantee that this motion can be measured. A video sequence only allows the analysis of the motion of specific brightness patterns on each image, which then are used to compute the optical flow for each pixel. This is a long-standing vision problem that remains unsolved [17], with challenges as large displacements, occlusions and illumination changes [7]. It has many applications such as autonomous driving and navigation [13] or video analysis [18, 9], which can be directly applied to endoscopy procedures.

Equation (3.1) shows how 3D motion $\mathbf{m}_\mathbf{x}$ for a point $\mathbf{x}$ translates to 2D $\mathbf{m}_\mathbf{u}$ when projected to the sensor at pixel $\mathbf{u}$, as detailed in Figure 3.1. More than the specific details, it shows multiple key aspects of this relationship. First, note that $\mathbf{m}_\mathbf{x}$ represents relative motion of either the camera $\mathbf{x}_\mathbf{c}$ or the scene point $\mathbf{x}$. Second, points further away move slower in the image, with a larger z coordinate in the denominator, due to the parallax effect. Lastly, looking at the cross products, a point moving parallel to $\mathbf{r}_\mathbf{x}$ does not change its position in the image. This can be seen in Figure 3.1.

$$\mathbf{m}_\mathbf{u} = f \, \frac{\mathbf{n}_c \times (\mathbf{r}_\mathbf{x} \times \mathbf{m}_\mathbf{x})}{(\mathbf{r}_\mathbf{x} \cdot \mathbf{n}_c)^2} \tag{3.1}$$

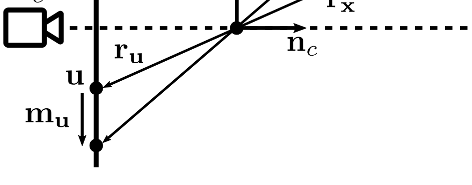
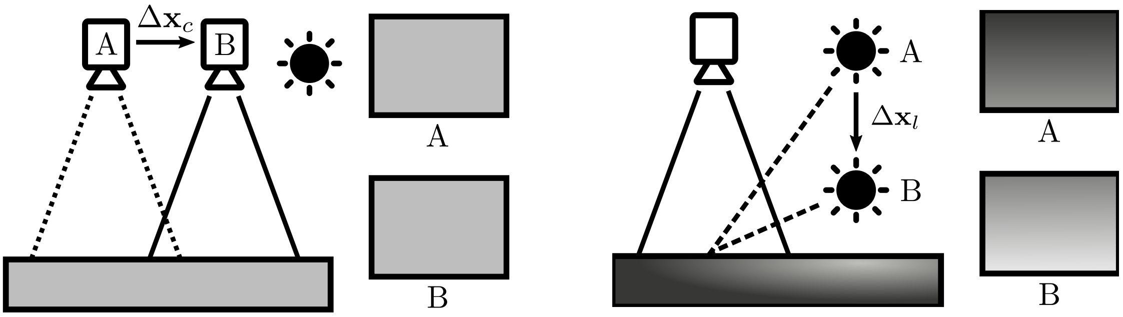


Figure 3.1: Projection of 3D velocity in a 2D image, defining optical flow as $\mathbf{m}_\mathbf{u}$.

Still, motion of the brightness patterns does not always translate to relative motion between the camera and the scene. Figure 3.2a shows an example with a moving camera. Because of the non-textured surface with constant albedo, its parallel movement and fixed lighting, it cannot be measured using only visual information. On the other side, Figure 3.2b shows an example where the change in illumination produces a large displacement in the brightness patterns of the image. Yet, neither the scene geometry nor the camera have moved, so the real optical flow is zero.

There are many individual solutions for optical flow estimation, as changes in the brightness patterns can be interpreted as different flow motions. One of the most common assumptions to solve this problem is that the brightness for each scene point remains constant through all frames. This is called the brightness constancy constraint, and holds for scenes with no illumination changes that contain diffuse surfaces i.e. they reflect light in all directions equally. Most indoor and outdoor scenes follow these assumptions. There are caveats, like occlusions or light sources placed near moving objects but, in the general case, it provides a good estimation of the optical flow.

One of the central parts of our work is that the co-located light and camera model does not follow the brightness constancy assumption. For an endoscopy sequence, the relative motion between the camera and the scene translates to relative motion for the light source, which results in varying brightness for the same points. In this section, we present various methods that attempt to solve this problem. On one hand, analytical methods require careful, hand-crafted modelling of the changes in illumination. Even then, it is practically impossible to include all the complex interactions that happen in a real endoscopy, like global lighting and occlusions. On the other hand, learning-based methods do not need to make any assumptions about the illumination model. This is a key insight of our work as, with a very large number of parameters, the training process is able to adapt the network to include many of the phenomena that occur in the input data as long as there exist enough samples of them.



(a) Motion with no brightness change.

(b) Brightness change with no motion.

Figure 3.2: Examples where brightness changes do no reflect motion in the scene.

3.1 Analytical methods

Traditionally, optical flow estimation has been approached as an optimization problem. Algorithms attempt to minimize a data term (e.g. brightness constancy) and a regularization term. This last part attempts to deal with the limitations of the data term. One such example is that areas with low texture should move very similarly to neighbouring, more textured patches, where it is much easier to estimate the correct optical flow. This can be implemented as a minimization of the optical flow gradient, a technique formally known as Total Variation (e.g. TV-L1). However, instead of using the regularization term to estimate optical flow in problematic areas, our approach is to modify the data term and only estimate the optical flow on pixels where it is well-posed. The data term includes the photometric model detailed in Section 2. Nonetheless, as mentioned earlier, it is only an approximation of what happens in a real endoscopy. We can use this approach to generate high-quality data in synthetic and real images that can be later used by learning-based algorithms.

Brightness constancy assumes that, given a pixel $\mathbf{u}$ with optical flow $\mathbf{m_u}$, its value does not change over time between images I_1 and I_2, that is,

$$I_2(\mathbf{u} + \mathbf{m_u}) = I_1(\mathbf{u}). \tag{3.2}$$

Trying to solve for $\mathbf{m_u}$ is an underconstrained problem, as there are two unknowns (optical flow in the X and Y axes) with only one equation. There are many approaches to solve this problem. The Lucas-Kanade refinement algorithm, which we will use and modify in this section, assumes that the flow is constant in the local neighbourhood of a pixel. It is a patch alignment algorithm that uses all the points in the region to generate enough constraints to solve the problem. Note that for this approach to work the patch needs to be textured, to avoid cases similar to Figure 3.2a.

However, Equation (3.2) does not hold for our photometric model. Patches in the image can appear brighter or dimmer when the camera (and light source) moves. From Equation (2.12), this can be due for three reasons:

- The camera gain g_t.
- The distance of point $\mathbf{x}$ to the light source, noted as $d = \|\mathbf{x}_l - \mathbf{x}\|^2$, due to the inverse square falloff.
- The cosine term $|\mathbf{n_x} \cdot \omega_i| = \cos\theta$, determined by the angle θ of the light source w.r.t. the surface normal at point $\mathbf{x}$.

Now, for a pixel $\mathbf{u}$, the relationship between $I_1(\mathbf{u})$ and $I_2(\mathbf{u} + \mathbf{m_u})$ is

$$I_2(\mathbf{u} + \mathbf{m_u}) = I_1(\mathbf{u}) \cdot \frac{g_2}{g_1} \cdot \frac{d_1^2}{d_2^2} \cdot \frac{\cos\theta_2}{\cos\theta_1} = I_1(\mathbf{u}) \cdot g_{21} \cdot d_{21}(\mathbf{u}) \cdot c_{21}(\mathbf{u}) \tag{3.3}$$

as detailed in Figure 3.3. The values d_{21} and c_{21} encompass point-specific depth and angle changes, whereas g_{21} is a global factor between the two images. We simplify the previous equation with a gain term $k_{\mathbf{u}} = g_{21} \cdot d_{21}(\mathbf{u}) \cdot c_{21}(\mathbf{u})$, and add it to the standard Lucas-Kanade equation system. Thus, we can estimate three variables: a 2D motion vector $\mathbf{m}_{\mathbf{u}}$ and a gain parameter $k_{\mathbf{u}}$. Section 5.1 shows results on real and synthetic sequences, where this algorithm is able to obtain sub-pixel accuracy in points with enough texture, improving the previously calculated ground truth.

3.2 Learning-based methods

Hand-crafted optical flow estimation is only able to model part of what happens in a real endoscopy procedure. The co-located light and camera model does not include complex phenomena, like global illumination, as they cannot be expressed with a closed-form solution. Learning-based models, on the other hand, do not need to make any assumptions on the illumination model. We use deep neural networks for their large number of parameters, which are able to adapt to almost any environment.

The problem then changes: instead of hand-crafting a model that follows a real endoscopy setting, our goal is to obtain training data for it. However, ground truth optical flow data is not available for real procedures. Size is a limiting factor inside the human body, and current endoscopes only include a monocular camera along with a light source. Current state-of-the-art Structure-from-Motion (SfM) algorithms can process a very small part of endoscopy video sequences. They are only able to provide information on less than 1% of the pixels, which is not enough to learn a real model. Thus, synthetic datasets become a key tool to solve this problem, as they can provide large amounts of accurate data, but they need to be as close as possible to an actual endoscopy sequence i.e. need to have a narrow simulation gap.

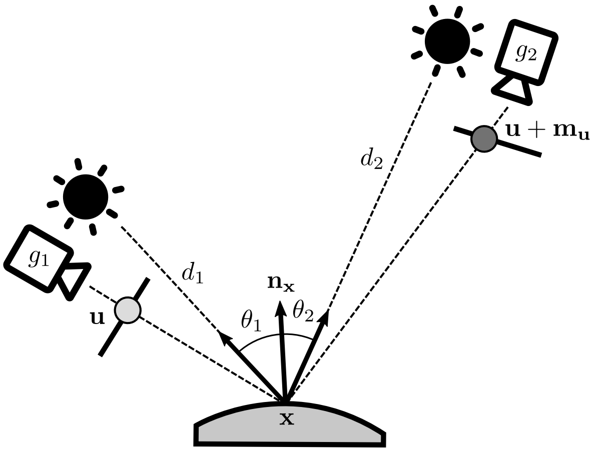


Figure 3.3: Co-located light and camera assumption for optical flow estimation.

In this section, we first present multiple synthetic datasets, how they were processed, and how well they fit a real endoscopy. Second, we show what information can be extracted from real datasets. Finally, we use both to train an existing state-of-the-art network for optical flow estimation.

We also explored a self-supervised training approach that built on the presented photometric model. The main idea is to use dense optical flow matches to estimate all the parameters of the model that cannot be directly calibrated. That includes the depth, surface normal and camera pose information. With this, the network should be able to compare brightness values, similarly to other unsupervised approaches that work using the brightness constancy assumption. However, we found that parameter estimation is very sensitive to noise, which does occur during training, and the network was not able to learn from the input data.

3.2.1 Optical flow computation

Our learning model uses synthetic sequences with ground truth optical flow. However, this information is not directly available in any of the presented datasets and must be pre-calculated. Consider a video sequence for which we have depth and pose information. To obtain the optical flow for each pair of frames, we use depth for each pixel $d_i = \mathbf{d}(\mathbf{u}_i)$ and the geometrical calibration of the camera Θ_t, which contains its intrinsic and extrinsic parameters over time. This last part determines the projection model of the camera π, which projects world points $\mathbf{x}$ to image points $\mathbf{u}$ as

$$\mathbf{u} = \pi(\mathbf{x}; \Theta_t), \quad \mathbf{x} = \pi^{-1}(\mathbf{u}, d; \Theta_t). \tag{3.4}$$

Note that unprojecting a 2D pixel to its 3D position requires knowing its depth d. Also, both operations π and π^{-1} need a calibrated camera, with intrinsic and extrinsic parameters Θ_t at time t. For a pinhole camera, this corresponds to its focal distance, optical distance, distortion parameters, position and orientation. Typically, only the last two change.

To obtain the optical flow from a pair of images I_1 and I_2, each pixel $\mathbf{u}_1$ is unprojected from the first image and then projected to the second one

$$\mathbf{u}_2 = \pi(\pi^{-1}(\mathbf{u}_1, d_1; \Theta_1); \Theta_2) \tag{3.5}$$

such that each point has a flow vector $\mathbf{m}_{\mathbf{u}_1} = \mathbf{u}_2 - \mathbf{u}_1$. Note that we can estimate $\mathbf{m}_\mathbf{u}$ for every pixel $\mathbf{u}$ regardless of whether if it is occluded or if it falls outside the image.

3.2.2 Synthetic datasets

We employ an end-to-end deep learning approach that estimates optical flow from a pair of images. This section presents different datasets, consisting of pairs of images with ground truth optical flow, and evaluates how well their training samples follow a real endoscopy procedure.

FlyingChairs and FlyingThings3D. Both these datasets contain almost no illumination changes, and thus mostly follow the brightness constancy assumption. However, they are especially relevant as they form simple cases which are used to train an initial model of the RAFT network.

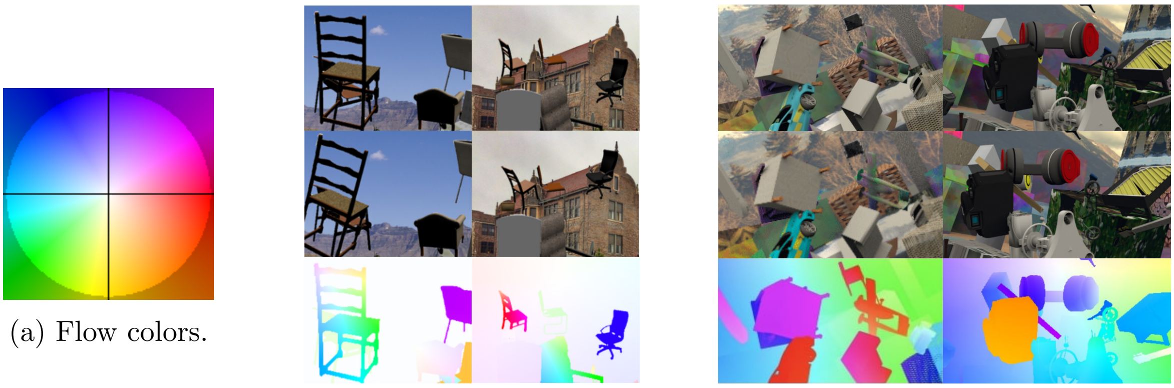
FlyingChairs [4] is a synthetic dataset that contains nearly 23.000 renderings of 3D chair models moving in front of random backgrounds obtained from Flickr. The motion of both chairs and background is purely planar, with results in some simple occlusions. On the other side, FlyingThings3D [12], as its name implies, contains random everyday objects flying along randomized 3D trajectories. This produces more complex motions, occlusions, and small illumination changes.

Figure 3.4 shows a comparison between the two datasets. Note the color wheel on Figure 3.4a, which will be used to represent any 2D field of vectors on an image, optical flow in this case. For a given color, its hue encodes the direction of the flow vector, and the saturation encodes its length.

ColonoscopyDepth. This is a dataset being currently developed by a member of the EndoMapper project [15]. It simulates image acquisition during an endoscopic exploration procedure. A high-fidelity model of real human intestines is obtained via CT scans, and is explored from start to end. A small agent with a camera, simulating an endoscope, travels inside the intestine model from start to end in ten different sequences.

They have been generated with the Unity framework and its High Definition Render Pipeline (HDRP), including physically based rendering that mimics real tissue. The Unity AOV Recorder is used to extract Arbitrary Output Values during the render process, which may include depth, normals, albedo and other useful information. However, the motion field cannot be directly extracted from this plugin and must be calculated afterwards, as was explained in Section 3.2.1.

To account for small and large motions, we generate datasets where the frames used for optical flow are separated at distinct distances. For example, a distance of one frame means that we calculate the optical flow from one frame to the next. In total, we generate six versions where the frames are separated by one, two or three frames in the forward or backwards direction of the video sequence (see Figure 3.5).



(a) Flow colors.

(b) FlyingChairs [4].

(c) FlyingThings3D [12].

Figure 3.4: Motion comparison between FlyingChairs and FlyingThings3D. The color wheel on (a) encodes the flow vectors shown at the bottom of (b) and (c), which represent pixel movement from the upper to the lower image.

Given that ten sequences are available, we split six for training and validation, leaving the rest for test. While the intestine walls use a simplified material with no subsurface scattering, it follows a complex illumination model with global lighting. For the movement, the camera is mostly focused on the vanishing point of the scene, moving forwards without looking at the intestine walls. Thus, the resulting motion field does not vary much between frames. Finally, it is important to note that this dataset follows a pinhole camera model, while images captured by an endoscope have fisheye distortion. This difference is especially notable on the edges of the image, where the pinhole model has much larger motions. While it may not matter for other applications, it can lead a neural network to overfit this specific case.

3.2.3 Real endoscopy datasets

We use the EndoMapper dataset, a collection of real endoscopy exploration sequences ranging from 15 to 45 minutes, captured in a hospital environment. The only sensor available is a monocular camera, so it is not possible to obtain a ground truth for most parts of the data. Some of the procedures have been processed using a state of the art Structure-from-Motion (SfM) method. The COLMAP algorithm [16] is a general-purpose SfM system that is able to extract sparse 3D information about scene points and camera poses. Normally, a standard method tries to estimate optical flow from a pair of consecutive frames. Even if it is able to obtain multiple image matches between the two frames, it is a hard problem to solve accurately. Consider the triangle formed between a world point and two camera positions. Because the camera moves very little between two frames, trying to triangulate the position of a point is very sensitive to noise, as the problem tries to join two almost parallel lines with a very

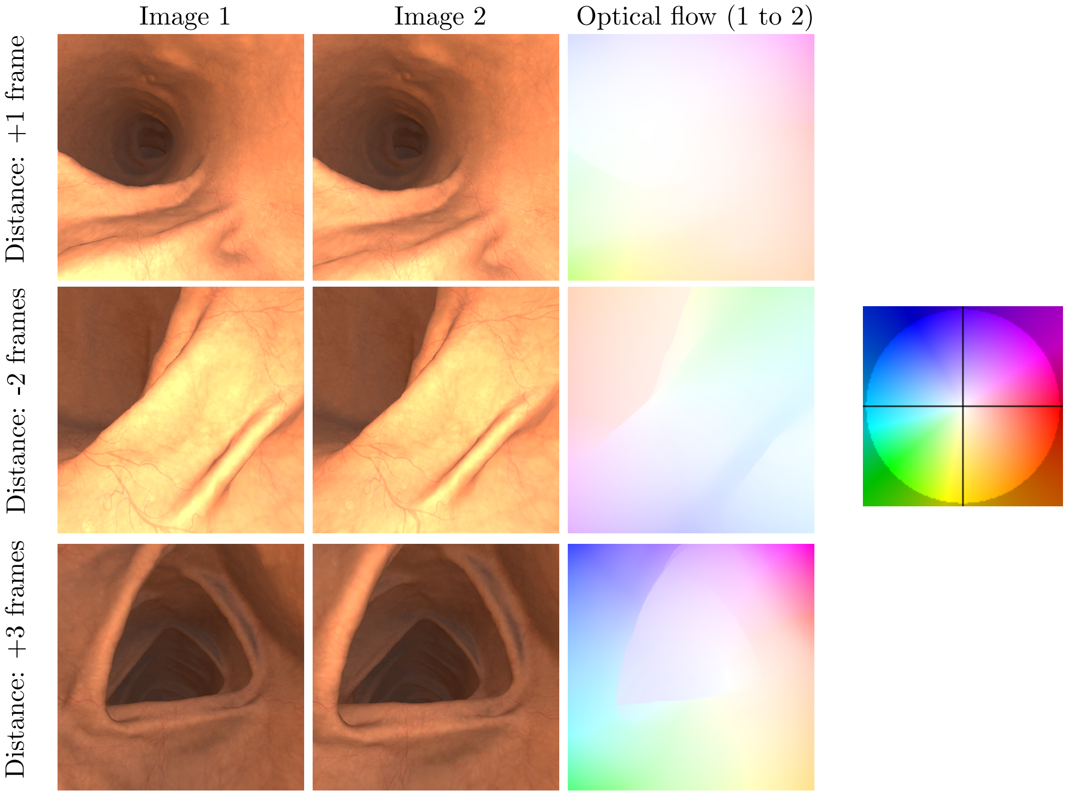


Figure 3.5: Generated optical flow at different distances using ColonoscopyDepth.

small angle i.e. low parallax. The reason that COLMAP is able to work is that it has information from all the frames in the sequence, being able to use multiple high-parallax views of the same point. This computation, however, is very expensive, and only produces results for very few pixels. As such, it is far from seeing real-time medical use, but the key is that it is able to provide ground truth data in real sequences.

Concretely, we work with three sequences, labeled SEQ_33, SEQ_34 and SEQ_327. They had good results with the COLMAP algorithm as the explored environment is relatively clean, and the medical procedure did not have any problems. Each one contains multiple clusters, that is, groups of images where COLMAP was able to find any matches in common. We take the 3D information, which was previously processed by other members of the EndoMapper project, and use it as input to calculate optical flow using a similar method as described in Section 3.2.1. While handling the data is different, the main points remain the same. However, we do not calculate motion at different distances. The translation between consecutive frames can vary a lot, as COLMAP can filter out images from the sequence. Thus, motion is calculated from consecutive frames, discarding pairs where the translation is in the top 10% of the largest ones. The camera can remain still during the exploration procedure so, in most frames, the pixel motion is fairly small. Also, it can contain wrong, outlier information, so the top 10% of the largest flow vectors are discarded.

Figure 3.6 shows training samples for each of the three sequences, which were used to create the final dataset. Table 3.1 also contains details about the train/test split, number of frames and valid points where COLMAP was able to compute enough information to estimate optical flow. There exists notable estimation error on some cases due to inaccuracies of the COLMAP algorithm. This can be corrected using the optical flow refinement algorithms shown earlier, but it would require some fine tuning on each individual frame. Nevertheless, Section 5.2 shows that this dataset provides a good method of training and evaluating on endoscopy sequences.

Table 3.1: Sequences used for optical flow estimation in the EndoMapper dataset.

Sequence ID	**Type**	**# frames**	**Resolution**	**Pixels with valid motion**	
				# pixels	**% pixels**
SEQ_00033	Train	2.519	480x360 px	1.873.002	0,430%
SEQ_00034	Train	5.011	480x360 px	8.118.058	0,937%
SEQ_00327	Test	4.117	480x360 px	4.122.494	0,579%

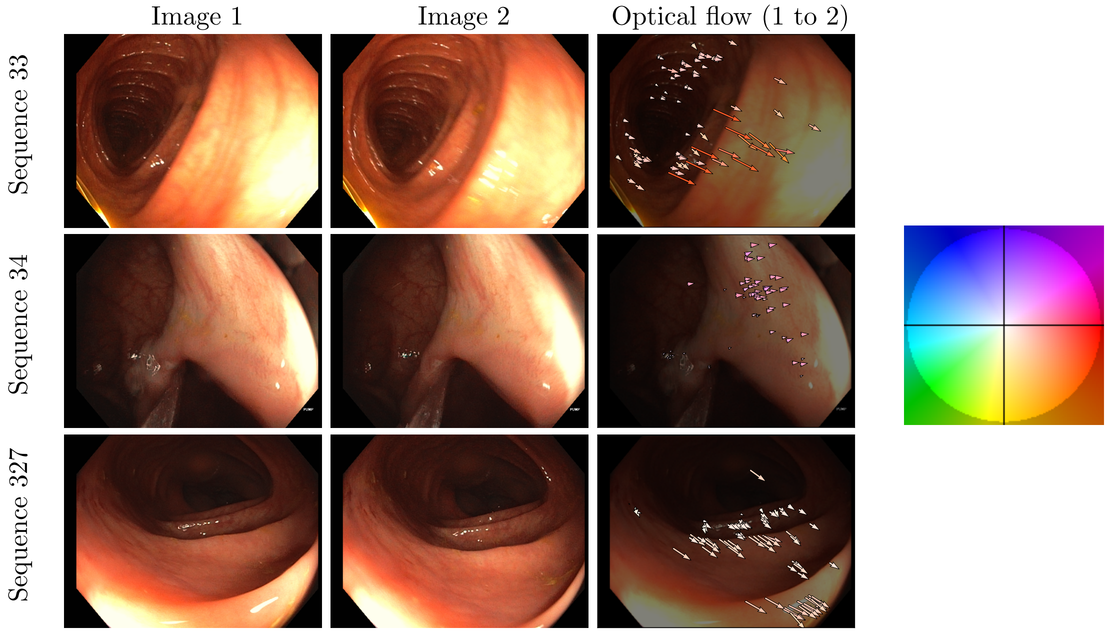


Figure 3.6: Generated optical flow for different sequences of the EndoMapper dataset. Because it can not be computed on every pixel, we show the motion vectors as colored arrows using the same color scheme as before.

Chapter 4

RAFT for optical flow

For our work, we decided to use the Recurrent All-pairs Field Transforms (RAFT) [17] deep neural network for optical flow estimation. At the time of selection, it provided state-of-the-art results with an available implementation, and now it still is one of the top-performing approaches. Its architecture consists of three main components, seen in Figure 4.1. First, a convolutional encoder extracts a feature vector for each pixel. Second, a correlation layer produces a 4D volume for all pairs of pixels, with subsequent pooling to produce lower-resolution versions. Finally, it uses a recurrent GRU-based update operator [2], an architecture similar to long short-term memory used in other recurrent networks. It retrieves values from the correlation volumes and iteratively updates a flow field initialized at zero.

This update operator mimics the steps of an iterative optimization algorithm with an important difference: it uses learned instead of handcrafted priors. As mentioned before, it is one of the key advantages of deep learning. We do not expect the network to exactly learn the photometric model shown in Section 2. Real data presents more phenomena (e.g. occlusions, complex materials or global illumination) and motion priors (movement inside a tubular scene, where there are no objects that can be individually segmented). Given that we are able to obtain ground truth optical flow even in these hard cases, the network can exploit this to obtain better results.

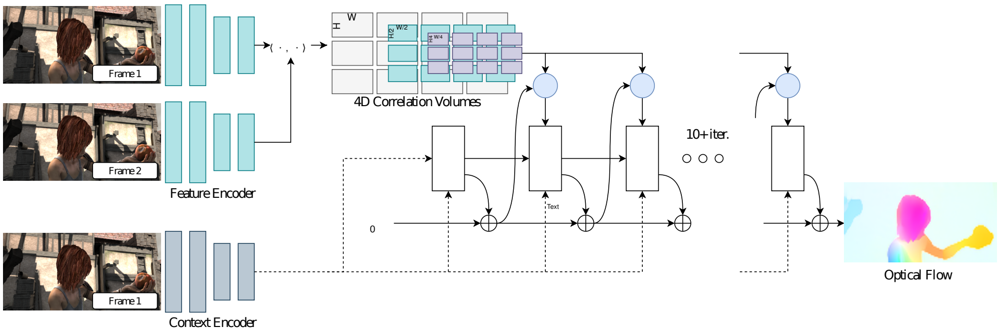
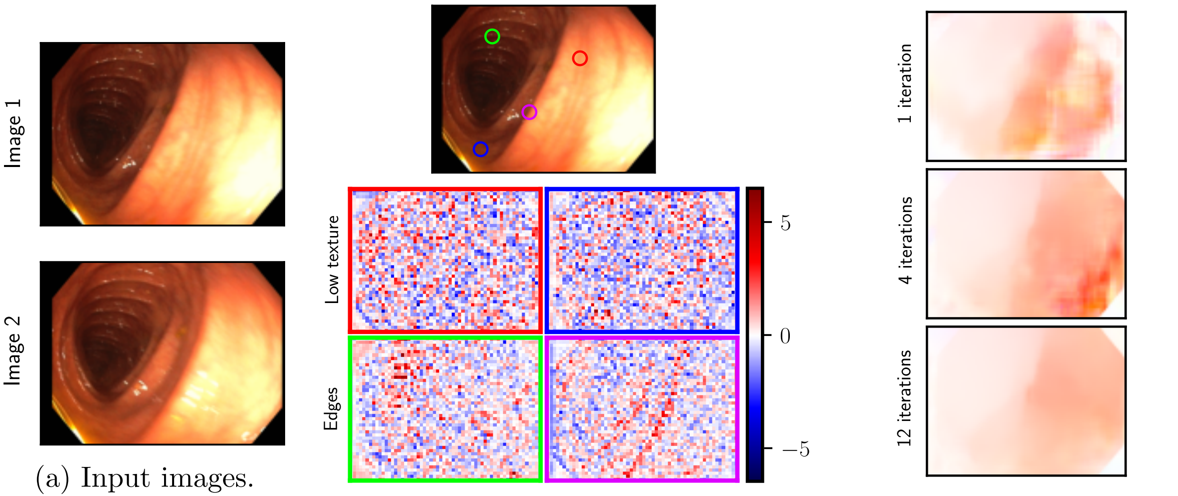


Figure 4.1: RAFT network architecture. Figure taken from the original paper [17].

Figure 4.2 shows an overview of how RAFT estimates optical flow in a medical setting. For this example, we have chosen an image pair with significant illumination changes. First, the network extracts feature vectors from each pixel and matches them, generating the correlation volume. Up until this point, the network does not have a way of relating the movement of the light and camera with the changes of brightness. Because the network cannot match patches based on their radiance, the best that it can do is to extract brightness-independent information. Looking at Figure 4.2b, RAFT computes high correlation values mostly on textured parts of the image e.g. edges. For low texture patches, the correlation values seem more random. This limitation can be overcome by adding more elements to its architecture (e.g. simultaneous pose estimation), but they were not implemented due to time limitations. Still, the GRU operator has a great ability to learn motion priors specific to the input data, as will be seen in Section 5.



(a) Input images.

(b) Correlation volume slices.

(c) Optical flow iterations.

Figure 4.2: Example of a forward pass in RAFT for a pair of endoscopy images.

Chapter 5

Experimental validation

In this chapter, we evaluate the presented analytical and learning-based optical flow algorithms. For this, we perform tests on both synthetic and real datasets. Most importantly, we are able to give quantitative error metrics on captured endoscopy sequences.

5.1 Analytical estimation

Lucas-Kanade with photometric gain. We evaluate an optical flow estimation algorithm based on Equation (3.3). We perform a test on two frames of the ColonoscopyDepth dataset with significant illumination changes. Note that Lucas-Kanade is an optical flow refinement algorithm, and thus requires an initial estimation. In this case, it can be computed from e.g. brightness-invariant feature matching algorithms [10]. We optimize three variables: two for the 2D motion $\mathbf{m_u}$ and one gain factor $k_{\mathbf{u}} = g_{21} \cdot d_{21}(\mathbf{u}) \cdot c_{21}(\mathbf{u})$, and also use a 15x15 patch around $\mathbf{u}$ to obtain enough constraints to solve the problem.

Figure 5.1 shows multiple iterations of the least squares minimization done by Lucas-Kanade algorithm, showing how the two patches converge to very similar appearances. One of the most important advantages of Lucas-Kanade refinement is that it is able to detect sub-pixel motion during the alignment. This technique could be used to generate ground truth data and improve existing training samples. However, it requires a small degree of manual supervision: due to time contraints it was not possible to apply it to a large amount of data. The obtained result of $k_{\mathbf{u}} = 1.09$ is consistent with the forward camera motion, as the second image should appear brighter than the first. Finally, Figure 5.2 shows a second test done on two frames of the EndoMapper dataset, showing its great potential for this task.

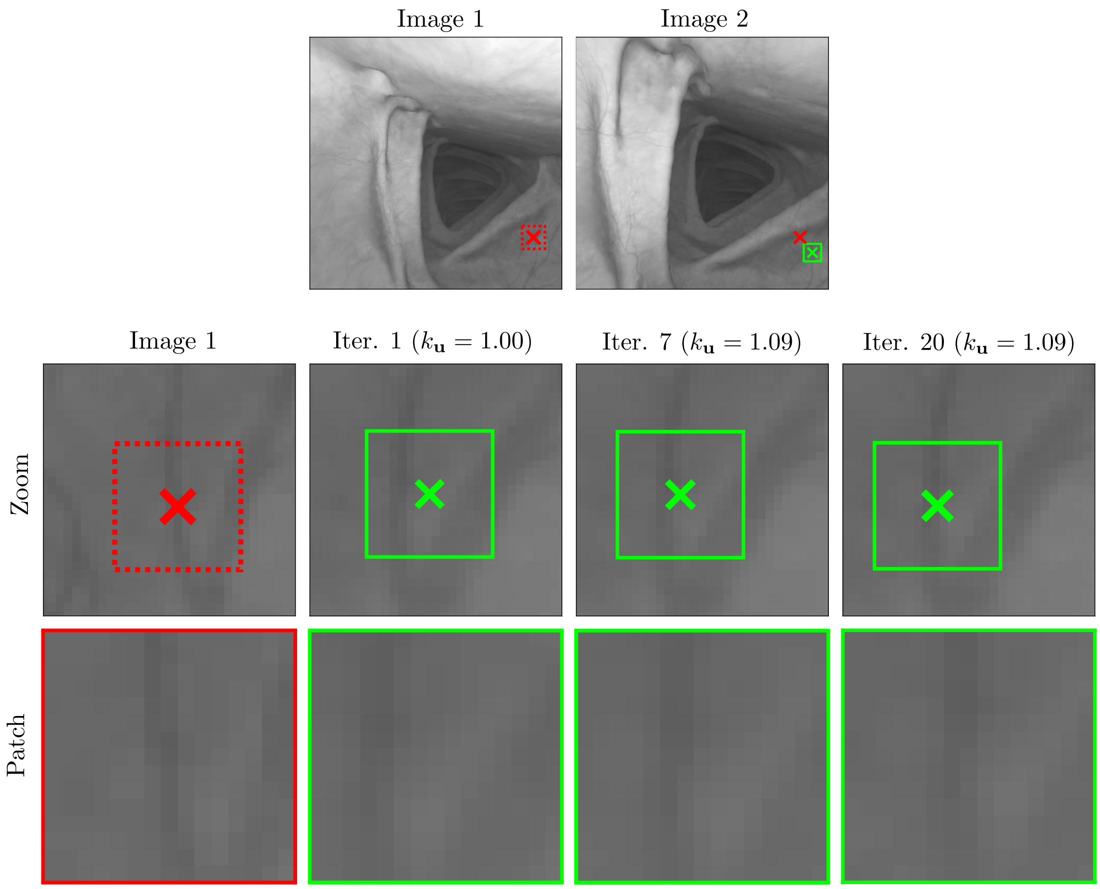


Image 1
Image 2
Image 1
Iter. 1 ($k_{\mathbf{u}} = 1.00$)
Iter. 7 ($k_{\mathbf{u}} = 1.09$)
Iter. 20 ($k_{\mathbf{u}} = 1.09$)
Zoom
Patch

Figure 5.1: Optical flow refinement with photometric gain on ColonoscopyDepth data.

5.2 RAFT training schedules

Table 5.1 shows an overview of the multiple datasets that were presented previously. Each one is identified by a letter, which is used to denote the different trained models. For example, C+T+D denotes a model that has been trained using a combination of the FlyingChairs, FlyingThings3D and ColonoscopyDepth2 datasets.

All the trained models are summarized in Table 5.2, along with the number of steps and hyperparameters. The provided code for RAFT has built-in data augmentation capabilities that include cropping, stretching and hue changes for the images. We use the Average Endpoint Error (AEE) metric for model evaluation

$$\text{AEE} = \frac{1}{n} \sum_{i}^{n} \left\| \mathbf{m}_{\text{est}}^{(i)} - \mathbf{m}_{\text{gt}}^{(i)} \right\|, \tag{5.1}$$

which corresponds to the mean pixel distance between the estimated flow vector $\mathbf{m}_{\text{est}}$ and the ground truth one $\mathbf{m}_{\mathbf{gt}}$ for all pixels of the image. Note that this error metric depends on the resolution of the dataset used, which is mostly similar on our case, and the amount of displacement between frames.

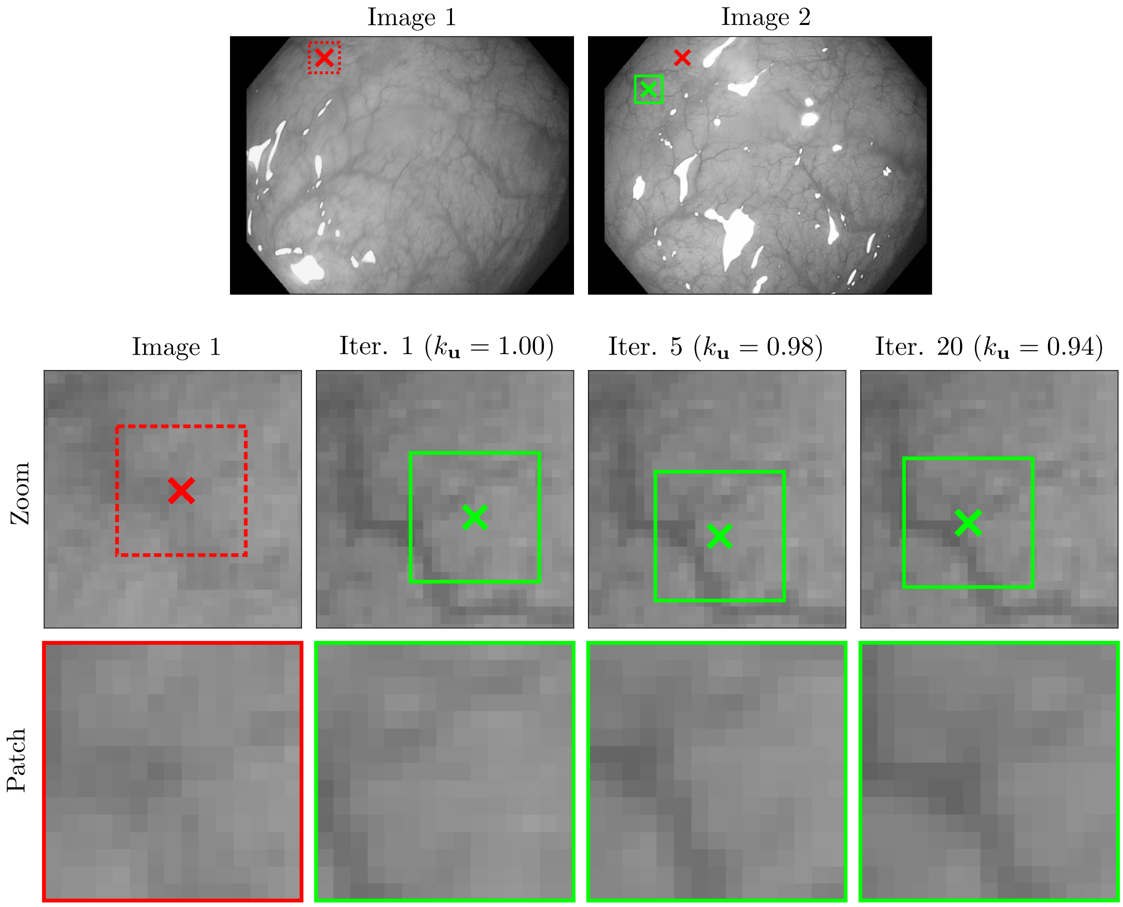


Image 1
Image 2
Image 1
Iter. 1 ($k_{\mathbf{u}} = 1.00$)
Iter. 5 ($k_{\mathbf{u}} = 0.98$)
Iter. 20 ($k_{\mathbf{u}} = 0.94$)
Zoom
Patch

Figure 5.2: Optical flow refinement with photometric gain on EndoMapper data.

Evaluation in real and synthetic endoscopic sequences. Tables 5.3 and 5.4 show AEE results obtained when evaluating the trained models on the EndoMapper and ColonoscopyDepth sequences, respectively. Our main and most important goal is to perform better on real sequences. However, as will be shown later, evaluating synthetic datasets also proves to be very helpful. Instead of evaluating on a set of sparse points only, they are able to give a bigger picture of how the network is estimating optical flow, adding important qualitative results.

We now analyze the error metrics on real datasets. First, the C+T model only trained with the FlyingChairs and FlyingThings3D datasets obtains good results, proving that RAFT generalizes well. Note that these results are obtained only on pixels filtered by COLMAP, which were well-posed for a Structure-from-Motion computation. The best results are obtained by the C+T+M model, trained on real data. However, we show that these metrics are not as good in reality. Figure 5.3 displays the evaluation of a training sample on the simulated ColonoscopyDepth dataset. Because the C+T+M model is only trained on a few pixels per frame, it fails to generalize to the whole image. Thus we show that, by adding synthetic data to the training mix, the results do not suffer from this problem, keeping the AEE lower in both cases.

Table 5.1: Datasets used for training and evaluation of RAFT. They are ordered according to their fidelity to the photometric model, from FlyingChairs (i.e. constant illumination) to EndoMapper (i.e. real data). Each dataset has a short name, denoted by a letter, that will be used later. †Some points do not have depth information.

Dataset name (ID)		Type	Split	Samples	Resolution	% valid px
FlyingChairs	C	Synthetic	Train	22 800	512x384 px	100%
FlyingThings3D	T	Synthetic	Train	39 000	960x540 px	100%
ColonoscopyDepth	D	Synthetic	Train	36 000	475x475 px	>99%†
ColonoscopyDepth	D	Synthetic	Test	28 800	475x475 px	>99%†
EndoMapper	M	Real	Train	7 530	480x360 px	<1%
EndoMapper	M	Real	Test	4 117	480x360 px	<1%

Table 5.2: Training hyperparameters for the different implemented RAFT variations. Learning rate and weight decay are adjusted to help the model converge.

Model	**Training details**		
Datasets	**# steps**	**Learning rate**	**Weight decay**
C+T (base)	960 000	$2.5 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$
+D	200 000	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$
+M	50 000	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$
+D+M	200 000	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-5}$

Second, training on simulated colon sequences does not improve the error when compared the C+T model. Looking at the optical flow examples shown in Figure 3.5, the largest motions are located on the edges of the image. As was mentioned before, the ColonoscopyDepth dataset follows a pinhole camera model which produces this effect, as opposed to the fisheye model found in the real endoscope. RAFT learns a prior for the kinds of motion that happen in the image and can be seen in Figure 5.4. Two identical images are passed as input, with exactly zero optical flow. RAFT does a good job of estimating zero motion on textured areas. However, the corners of the image are black, so they have a high correlation with other black pixels. Thus, they are highly affected by the model of the network. For C+T+D, it is especially notable. Again using a mix of real and synthetic data (i.e. C+T+D+M) fixes this issue.

Table 5.3: Evaluation of different trained RAFT models in real data.

Training data	**Average Endpoint Error (AEE) in px**		
	Train		**Test**
	Sequence 33	**Sequence 34**	**Sequence 327**
C+T	**2.02**	**3.76**	3.83
C+T+D	2.16	4.74	4.15
C+T+M	(1.56)	(3.12)	**3.08**
C+T+D+M	(1.68)	(3.48)	3.39

Table 5.4: Evaluation of different trained RAFT models in synthetic data.

Training data	**Average Endpoint Error (AEE) in px**	
	ColonoscopyDepth	
	Train	**Test**
C+T	**0.71**	0.76
C+T+D	(0.20)	**0.26**
C+T+M	5.47	5.40
C+T+D+M	(0.41)	0.47

Ground truth error. As can be seen in Figure 5.5, the inaccuracies of the COLMAP algorithm are reflected on the quality of the ground truth data. However, models trained on real data do not seem to overfit to this noise, so the AEE metrics remain consistent between different schedules. Even so, the error can be corrected using a refinement algorithm as displayed in Figure 5.1, with current data as an initial estimation and some manual fine-tuning.

Mixing real and synthetic data. The combined C+T+D+M model uses a combination of real and synthetic data. Adding less synthetic data makes the model to lose generality when estimating dense flow, and adding less real data makes it perform worse on the target sequences. We tried multiple configurations and found that a 15% of synthetic samples is enough for a good balance. However, note that each sample of the ColonoscopyDepth dataset contains much more information in each frame. Finally, we give an overview of what improves optical flow estimation on real sequences. As can be seen in Figure 5.5, adding any domain-specific training data improves the network's estimations on specular reflections. More importantly, the C+T+M model has the same generalization problem as mentioned before and estimates the wrong motion on the right side of the image. This is fixed by adding synthetic data to the C+T+D+M model. Looking at Figure 5.6, the C+T+D model overestimates the motion near the marked discontinuity. Again, adding captured data improves its estimation, with the best results obtained by the C+T+D+M model.

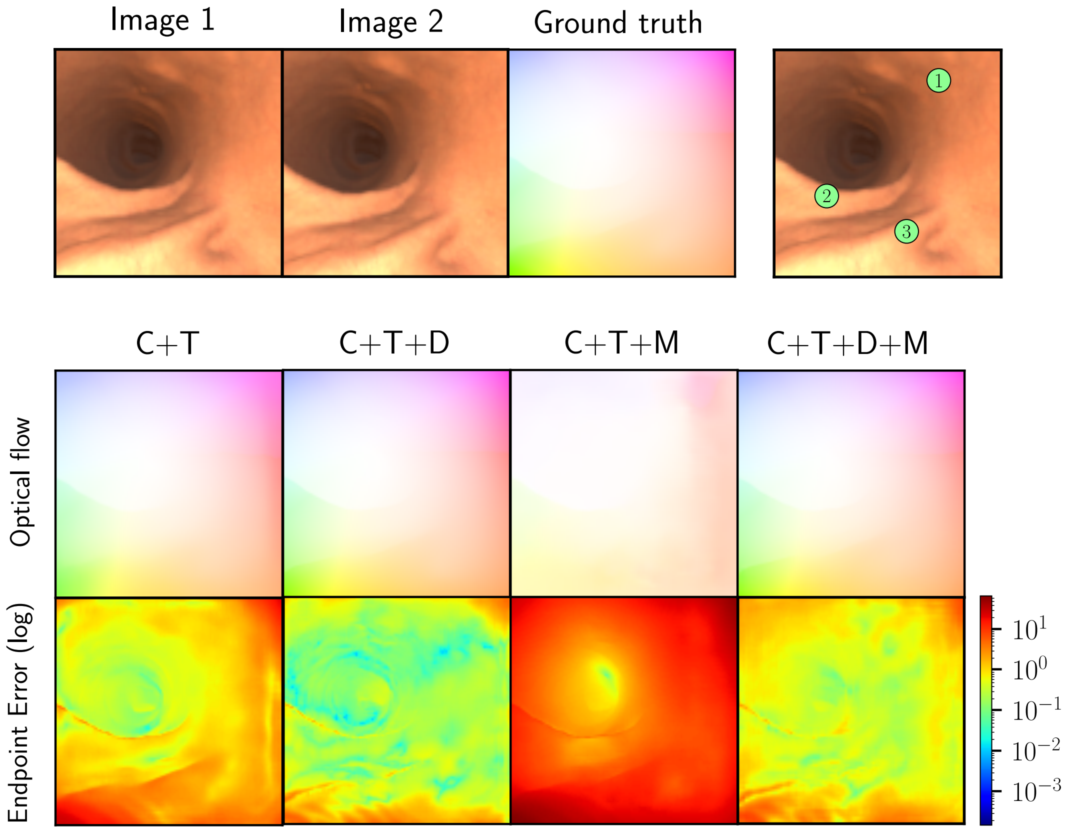


Figure 5.3: Optical flow estimation error for all trained models in two frames of the ColonoscopyDepth dataset. Although the model trained in real data (i.e. C+T+M) obtains lower error in sparse real sequences, it also has a much higher error when evaluated on dense synthetic sequences. On the other hand, models trained with synthetic data (i.e. C+T+D and C+T+D+M) do not suffer from these problems, and are able to improve in areas that are problematic in an endoscopy when compared to the default model (i.e. C+T). Looking at the green dots marked 1 to 3, these patches contain large displacements, illumination changes and occlusions respectively.

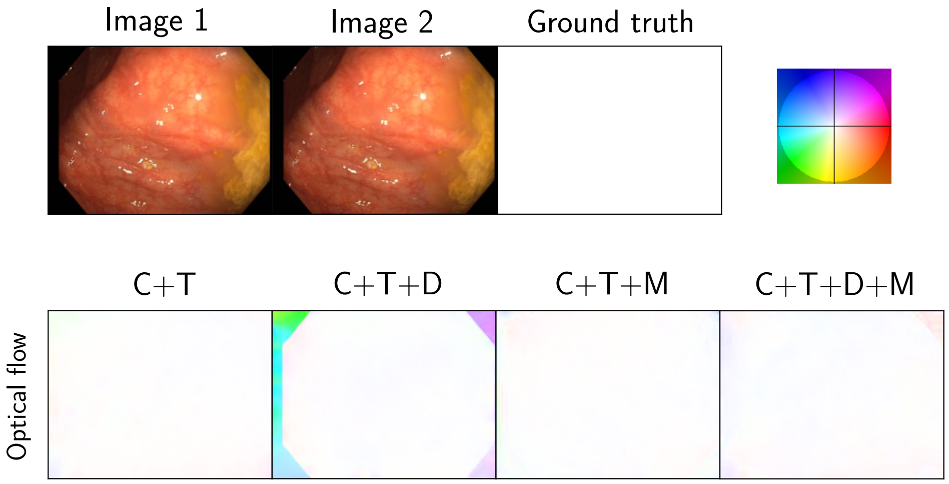


Figure 5.4: Optical flow estimation error in a trivial test case where the same image is used two times. As it does not move, the resulting optical flow is zero for all points. This gives a good insight of what the network is doing for pixels which are less affected by the correlation layer, which happens on the black pixels of the image edges.

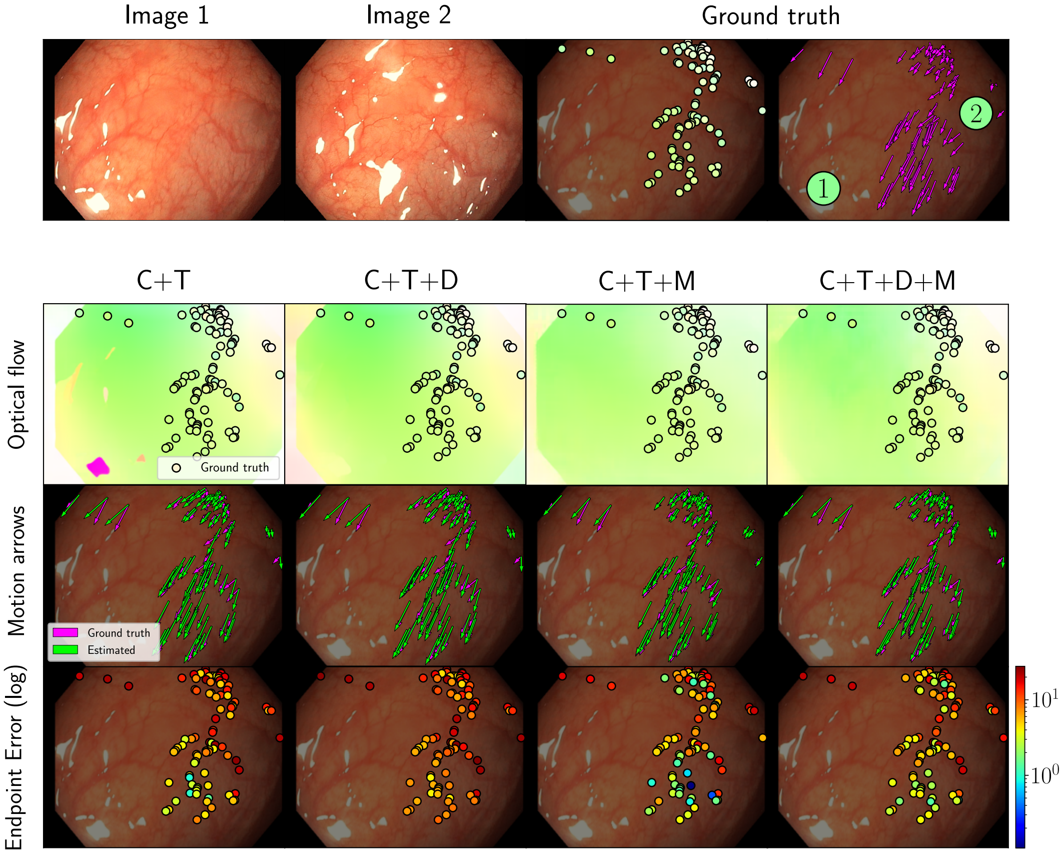


Figure 5.5: Error metrics for the EndoMapper dataset. The difference between ground truth and estimated motion can be seen in three ways by comparing (top to bottom) color, arrows or endpoint error. For the color case, the sparse ground truth flow colors are displayed on top of the dense estimated flow. Looking at the green circles, we show (1) a specular reflection that is wrongly interpreted by the C+T model, fixed by adding synthetic or real training data and (2) an area that is wrongly estimated by the C+T+M model, and is fixed by adding more synthetic data in the C+T+D+M model.

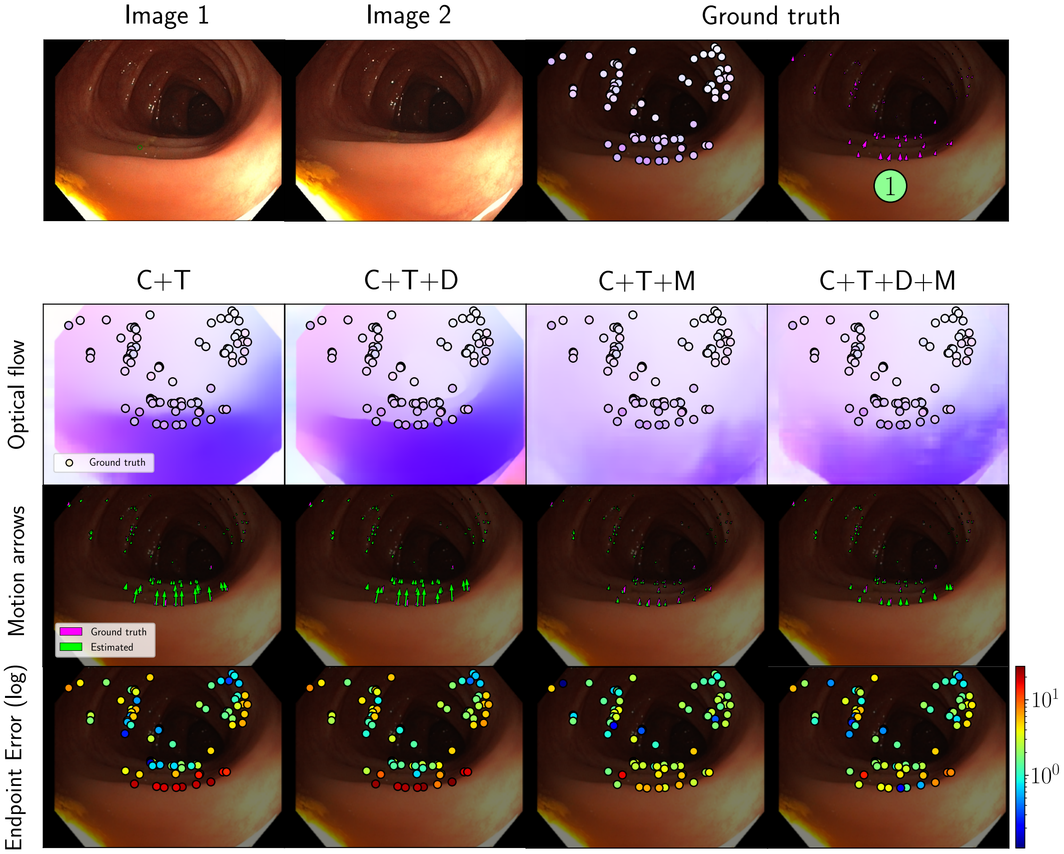


Figure 5.6: Error metrics for the EndoMapper dataset. The difference between ground truth and estimated motion can be seen in three ways by comparing (top to bottom) color, arrows or endpoint error. For the color case, the sparse ground truth flow colors are displayed on top of the dense estimated flow. Looking at the green dot (1), the discontinuity above is an interesting case. The C+T+D model does a good job at defining the edge, but overestimates the motion. This is partially corrected by the C+T+D+M model, obtaining a lower error overall.

Chapter 6

Conclusions

In this work, we have explored the advantages of learning-based methods for optical flow in the case of an endoscopy exploration procedure. Even if we can create a hand-crafted photometric model, it is not possible to include all the complex phenomena that occur inside the human body, as there is no closed-form solution. Also, there might not be a solution available for all points, leaving sparse data. Deep neural networks include high-dimensional parameter spaces which are better at adapting to a real scenario. However, they need a large amount of training data, but this gives them the ability to densely estimate optical flow in all pixels.

Given the limitations of the endoscope, it is not possible to obtain the ground truth motion on most points. We explore two possibilities. First, we filter well-posed points where optical flow estimation is feasible. They can be processed with different techniques, like COLMAP, a state of the art Structure-from-Motion algorithm, or Lucas-Kanade refinement. Using this approach produces results only on a few points which are not enough. Thus, we explore simulated environments as a second approach. They are especially useful, as ground truth can be available even for the most complex occlusions or illumination changes, which can be learned by the network model.

RAFT for optical flow [17] is a neural network that offers great performance in endoscopy sequences, even when trained with data from a different domain. By training with sparse points on sparse real sequences, the network loses the bigger picture and produces qualitatively worse motion estimations when evaluated on the whole image. On the other hand, current synthetic environments offer simplified photometric and motion models, with a notable simulation gap. As one of the main results of our work, using a combination of real and simulated models can overcome most of the mentioned issues. We obtain a 40% error reduction when evaluating on synthetic sequences, and a 15% on real sequences. Additionally, we show that mixing both types of training data produces much better qualitative results for other scene points whose ground truth is not available.

6.1 Future work

Our work has shown the benefits of learning-based optical flow estimation methods, and how mixing real and synthetic training data produces better overall results. Future work could bring improvements under different lines of research:

More realistic simulation environments. There exists a notable simulation gap between current simulated and captured environments. For example, real endoscopy procedures have fisheye distortion and more varied camera movements. Additionally, human tissue exhibits different light transport phenomena such as specular reflections or subsurface scattering.

More complex neural network architectures. Currently, the RAFT network uses feature-based matching. With this, the correlation volume does not store information about the brightness change, and thus it cannot be used to infer additional details about the motion. For example, dimmer pixels could mean that the camera has moved away from the scene. Modifying the architecture could be used to obtain a better performance in scenes that follow the presented photometric model.

Processing real endoscopy sequences. Techniques explored in this work can process less than 1% of the pixels of the sequence, even with state of the art Structure-from-Motion algorithms. Lucas-Kanade refinement has proven to obtain good results in well-posed points. It could be used to obtain more ground truth datasets or process existing ones for more points.

Bibliography

[1] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Gastone Ciuti, Marco Visentini-Scarzanella, Alessio Dore, Arianna Menciassi, Paolo Dario, and Guang-Zhong Yang. Intra-operative monocular 3d reconstruction for image-guided navigation in active locomotion capsule endoscopy. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 768–774. IEEE, 2012.

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[5] Yang Hao, Jing Li, Fei Meng, Peisen Zhang, Gastone Ciuti, Paolo Dario, and Qiang Huang. Photometric stereo-based depth map reconstruction for monocular capsule endoscopy. *Sensors*, 20(18):5403, 2020.

[6] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[7] Junhwa Hur and Stefan Roth. Optical flow estimation in the deep learning age. *CoRR*, abs/2004.02853, 2020.

[8] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[9] Junghwan Ko and Jungsuk Lee. Stereo camera-based intelligence surveillance system. *Journal of Automation and Control Engineering Vol*, 3(3), 2015.

[10] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] Víctor Martínez Batlle and Juan Domingo Tardós Solano. Real-scale 3d reconstruction from monocular endoscope images. Final master thesis, Universidad de Zaragoza, 2021.

[12] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[13] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[14] Takayuki Okatani and Koichiro Deguchi. Shape reconstruction from an endoscope image by shape from shading technique for a point light source at the projection center. *Computer vision and image understanding*, 66(2):119–131, 1997.

[15] Anita Rau, PJ Eddie Edwards, Omer F Ahmad, Paul Riordan, Mirek Janatka, Laurence B Lovat, and Danail Stoyanov. Implicit domain adaptation with conditional generative adversarial networks for depth prediction in endoscopy. *International journal of computer assisted radiology and surgery*, pages 1–10, 2019.

[16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[17] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020.

[18] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3899–3908, 2016.

[19] Chenyu Wu, Srinivasa G Narasimhan, and Branislav Jaramaz. A multi-image shape-from-shading framework for near-lighting perspective endoscopes. *International Journal of Computer Vision*, 86(2):211–228, 2010.

Appendix A

Project management

This project was carried out with a grant from the EndoMapper Project (EU-H2020 grant 863146). Table A.1 shows the total number of hours spent on the project, along with a breakdown for each of the performed tasks.

Table A.1: Breakdown for the dedication on each project task.

Task	**Time**
Literature review	25 h
Photometric model and calibration	195 h
Optical flow estimation algorithms	
Analytical methods	45 h
Learning-based methods	220 h
Dataset processing	135 h
Thesis document	105 h
Group meetings and management	55 h
Total	780 h

All of the code relevant to this work is available on the following website:

- `https://github.com/UZ-SLAMLab/optical-flow-for-monocular-endoscopy`