SMARTsurg

SMart weArable Robotic Teleoperated surgery

D3.1: On-the-fly 3D reconstruction of the surgical field

Due date: M18

Abstract: The present document is a deliverable of the SMARTsurg project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation programme (H2020). This deliverable aims at presenting the results of Task T3.1 "*On-the-fly 3D reconstruction of the surgical field*" until M18 of the project. This is a preliminary version of the final deliverable that will be provided by M28. It is developed within the scope of WP3, responsible for determining "*Visual Feedback and Teleoperation for Robot-Assisted MIS*" methods. Our main focus is the investigation of state-of-the art methods for 3D reconstruction of surfaces using endoscopic visual data, and their extension in order to achieve real-time performance. The results of this work will be further utilized for the 3D registration of the reconstructed surfaces to pre-operative models of the corresponding anatomical structures.

Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Status

Document Title	On-the-fly 3D reconstruction of the surgical field
Version	1.0
Work Package	3
Deliverable #	3.1
Prepared by	CERTH
Contributors	CERTH
Checked by	POLIMI
Approved by	UWE
Date	29/06/2018
Confidentiality	PU

Contact Points

Coordinator		
University of the West of England BRISTOL	University of the West of England/ Bristol Robotics Laboratory T Building, Frenchay Campus BS16 1QY Bristol, UK	Tel: +44 117 32 81301 E-mail: Sanja.Dogramadzi@brl.ac.uk Website:www.brl.ac.uk/research/researchthemes/medicalrobotics.aspx

Partners		
Information Technologies Institute	Centre for Research and Technology Hellas / Information Technologies Institute Building A - Office 1.1A 6th km Charilaou - Thermi, 57001 Thessaloniki, Greece	Tel.: +30 2311 257777 Fax: +30 2310 474128 E-mail: tzovaras@iti.gr Website: www.iti.gr
POLITECNICO MILANO 1863	Politecnico di Milano Department of Electronics, Information and Bioengineering Building 32.2 Via G.Ponzio 34/5 Milan, Italy	Tel.: +39 022 399 3371 E-mail: giancarlo.ferrigno@polimi.it Website: www.nearlab.polimi.it
Bristol Urological Institute	North Bristol National Health Service Trust/ Bristol Urological Institute Brunel Building, Southmead Hospital BS10 5NB Bristol, UK	Tel.: +44 117 4140898 E-mail: anthony.koupparis@nbt.nhs.uk Website: www.nbt.nhs.uk/bristol-urological-institute
University of BRISTOL	University of Bristol Translational Biomedical Research Centre Senate House, Tyndall Avenue BS8 1TH Bristol, UK	Tel.: +44 117 3423286 E-mail: r.ascione@bristol.ac.uk Website:www.bristol.ac.uk/health-sciences/research/tbrc
IEO Istituto Europeo di Oncologia	European Institute of Oncology Division of Urology Via Ripamonti, 435 20141 Milan, Italy	Tel.: +39 0257489516 E-mail: ottavio.decobelli@ieo.it Website: www.ieo.it
TheMIS THESSALONIKI MINIMALLY INVASIVE SURGERY ORTHOPAEDIC CENTER	TheMIS Orthopaedic Center 6 Adrianoupoleos St. 55133 Thessaloniki, Greece	Tel.: +30 2310 223 113 E-mail: papacostas@the-mis.gr Website: www.the-mis.gr

cybernetix A TechnipFMC Company	Cybernetix 306 Rue Albert Einstein 13882 Marseille, France	Tel.: +33 491 210 484 E-mail: jvandenbosch@cybernetix.fr Website: www.cybernetix.fr
Optinvent	Optinvent R&D Department Avenue des Buttes de Coesmes 80 35700 Rennes, France	Tel.: +33 299871066 E-mail: khaled.sarayeddine@optinvent.com Website: www.optinvent.com
HIT HYPERTECH INNOVATIONS	HIT Hypertech Innovations Agiou Nikolaou 33 2408, Nicosia, Cyprus	Tel./Fax: +357 22251217 E-mail: contact@hit-innovations.com Website: www.hit-innovations.com

Document Change Log

Each change or set of changes made to this document will result in an increment to the version number of the document. This change log records the process and identifies for each version number of the document the modification(s) which caused the version number to be incremented.

Change Log	Version	Date
First draft	0.1	May 15, 2018
Updated	0.3	June 22, 2018
Updated with partner's feedback	0.7	June 28, 2018
Submitted	1.0	June 29, 2018

Table of Contents

List of Figures

List of Tables

Executive Summary

The present document is a deliverable of the SMARTsurg project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation programme (H2020). This deliverable aims at presenting the results of Task T3.1 "*On-the-fly 3D reconstruction of the surgical field*". It is developed within the scope of WP3, responsible for determining "*Visual Feedback and Teleoperation for Robot-Assisted MIS*" methods.

As stated in the GA (1.1.3) the main objective of T3.1 is the 3D reconstruction of the surgical area. The result of the 3D reconstruction procedure is a critical component to the whole SMARTsurg vision system, since its results are closely related to T3.3 "*Augmented reality composite view creation and visualization*" and T5.2 "*Dynamic active constraints enforcement*". If 3D reconstruction fails to produce accurate results or takes large amount of time to produce them, significant challenges may arise. Therefore, in order to create a robust vision foundation, on which other modules can depend, 3D reconstruction component must be built based on two important principles:

- A high quality standard of the 3D reconstructed result must be met, in order to ensure accurate visual representation of the surgical area and useful interoperable data for other modules.
- Minimal time between reconstructed frames is essential for the real-time operation of the system and its deployment in RAMIS tasks.

Based on these principles, research is aimed towards 3D reconstruction methods for MIS or other use cases. For methods originally developed for a different reconstruction scenario to MIS, additional adaptations can be required, e.g. methods of 3D reconstruction that perform well, may often result in increased execution time, making them unsuitable for MIS applications. Therefore, optimized implementations of such methods, achieving real-time or close to real-time speed, can be considered as suitable candidates for MIS 3D reconstruction.

The current document describes the theoretical and practical research, based on the aforementioned principles, conducted up to this stage, regarding state-of-the-art 3D reconstruction[1]. Three basic methods, belonging in the general category of stereoscopy, will be described in detail. The first two methods are mainly targeted at odometry and autonomous

[1] Part of this work has been submitted to IEEE IST 2018 conference for possible publication with the title "Real-Time 3D Reconstruction in Minimally Invasive Surgery with Quasi-Dense Matching".

vehicle applications. However, they are capable of achieving real-time performance, therefore their adaptation to fit into our vision framework is investigated. On the other hand, the third method is aimed at MIS use cases by default, but takes significant time to produce the 3D reconstruction result. Hence, an optimized GPU version of the algorithm is proposed, implemented, and tested achieving real-time performance.

It is worth noting that this a the preliminary version of D3.1, which includes a detailed description of the 3D reconstruction methods, which are currently being investigated or implemented in our system, up to this point. The suitability of the methods for use in the final system is yet to be determined. This requires further research, testing and tuning to achieve better results, while performing tests on more types of data is also essential for covering more generalized MIS use cases. A more thorough presentation and description of the 3D reconstruction methods for the final system will be published in an updated version of the D3.1, which is due in M28.

1 Introduction

This deliverable (D3.1 “On-the-fly 3D reconstruction of the surgical field”) presents methods for 3D reconstruction, as well as their adaptation to real-time, for 3D reconstruction of images captured with a stereo endoscope.

This Deliverable documents mainly the steps and actions performed in task T3.1.

1.1 Objective and Scope

The purpose of this deliverable (D3.1 “On-the-fly 3D reconstruction of the surgical field”) is to provide an insight on the chosen 3D reconstruction methods and their implementation in real-time as well as a custom stereoscopic pre-processing and post-processing framework.

1.2 Document Structure

The document is divided into chapters presenting all the aspects of the research work performed in the context of 3D reconstruction of the surgical field. At first, the two main methods that are currently being investigated will be described, along with the necessary modifications and optimizations for the inclusion to the system. Next, a brief description of the custom stereo processing framework is given, that ensures compatibility and deployment of the 3D reconstruction module with any robotic system without much hassle. An evaluation section is also provided, in order to assess the performance of the 3D reconstruction methods, in various cases, both quantitatively and qualitatively. Finally, a section with plans for future directions for research and alternative implementations is included.

A supplementary annex is also provided. It contains the specifications and detailed description regarding the datasets, the methods have been tested on. Each chapter in the annex contains a complete dataset with specific attributes and its reconstruction challenges, resulting in the coverage of various cases.

1.3 Reference Documents

[1] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidei, D. Stoyanov, 2013. *Optical techniques for 3D surface reconstruction in computer-assisted laparoscopic surgery* -- Medical Image Analysis 17 (2013) p 974–996

[2] Bingxiong L., , Yu S., Xiaoning Q., Dmitry G., Richard G., Yuncheng Y., 2016. *Video Based 3D Reconstruction, Laparoscope Localization, and Deformation Recovery for Abdominal Minimally Invasive Surgery: A Survey*, Int J Med Robot. 2016 Jun;12(2):158-78.

[3] Stoyanov D., Visentini Scarzanella M., Pratt P., Yang G., 2013. *Real-Time Stereo Reconstruction in Robotically Assisted Minimally Invasive Surgery,* Medical Image Computing and Computer-Assisted Intervention - MICCAI 2010, 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part II (pp.275-82)

[4] Bernhardt, S., Abi-Nahid, J., Abugharbieh, R., 2012. Robust, *Robust dense endoscopic stereo reconstruction for minimally invasive surgery*, In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI): Workshop on Medical Computer Vision (MCV), pp. 198–207

[5] Röhl, S., Bodenstedt, S., Suwelack, S., Dillmann, R., Speidel, S., Kenngott, H., Müller-Stich, B.P., 2012. *Dense GPU-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration*. Med. Phys. 39, 1632–1645.

[6] Hernandez-Juarez D., Chacon A., Espinosa A., Vazquez D., Moure J. C., Lopez A. M., 2016. *Embedded real-time stereo estimation via Semi-Global Matching on the GPU,* ICCS 2016. The International Conference on Computational Science, Volume 80, 2016, Pages 143–153

[7] Zbontar J. and LeCun Y., 2014. *Computing the Stereo Matching Cost with a Convolutional Neural Network*. arXiv preprint arXiv:1409.4326.

[8] Ye M., Johns E., Handa A., Zhang L., Pratt P., Yang G.-Z, 2017. *Self-Supervised Siamese Learning on Stereo Image Pairs for Depth Estimation in Robotic Surgery*

[9] Rublee E. Rabaud V., Konolige K., Bradski G., *ORB: An efficient alternative to SIFT or SURF*, ICCV 2011: 2564-2571.

[10] Bay H., Tuytelaars T., and Gool L. V. 2006*. SURF: Speeded up robust features*. In Proc. European Comp Vis. Conf, pages 404–417.

[11] Pablo F. A., Bergasa L. M. Davison A. F., 2012. *Gauge-SURF Descriptors.* Image and Vision Computing Volume 31, Issue 1, January 2013, Pages 103-116

[12] Shi J., Tomasi L.,1994. *Good Features to Track,* Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[13] Kanade T. and Okutomi M., 1994. *A stereo matching algorithm with an adaptive window: Theory and experimet* TPAMI, vol. 16, no. 9, pp. 920–932, 1994

[14] Geiger, Andreas & Roser, Martinand & Urtasun, Raquel,2011, Efficient Large-Scale Stereo Matching, Computer Vision -- ACCV 2010, pp 25-38

[15] Hirschmuller H., 2005. *Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)

[16] Spangenberg, R., Langner, T., Adfeldt, S. & Rojas, 2014. R. *Large scale Semi-Global Matching on the CPU*, Intelligent Vehicles Symposium Proceedings, IEEE, 2014, 195-201

[17] A.S. Ciullo, V. Penza, L. Mattos, E. De Momi, "*Development of a surgical stereo endoscopic image dataset for validating 3D stereo reconstruction algorithms*", 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery, 2016.

[18] Collins T., Bartolli A., 2012. *3D Reconstruction in Laparoscopy with Close-Range Photometric Stereo.* MICCAI 2012: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012 pp 634-642

1.4 Acronyms and Abbreviations

Abbreviation	Definition
D	Deliverable
EC	European Commission
EU	European Union
DMP	Data Management Plan
M	Month
MIS	Minimally Invasive Surgery
WP	Work package
SoA	State of the art
DoA	Description of Action
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
RAMIS	Robot-Assisted MIS
ROS	Robot Operating System
CPU	Central Processing Unit
GPU	Graphical Processing Unit
GPGPU	General Purpose Graphical Processing Unit
ELAS	Efficient Large Scale Stereo Matching
SGM	Semi-Global Matching
ME	Mean Error

2 3D Reconstruction Methods

The problem of reconstructing the 3D geometry from arbitrary scenes or videos is a well-studied field where several algorithms have been developed. However, formulating the problem in the context of MIS, introduces important limitations and constraints. Most of them originate from the environment of MIS, such as the presence of smoke, blood and occlusion and deformation of tissues caused by surgical instruments or other factors. The constraint of real-time performance also reduces the available computation time, which is often required for 3D reconstruction algorithms to perform well.

2.1 State-of-the-art

Given the availability of a stereo endoscope, 3D reconstruction problem is mainly investigated with methods belonging in the stereoscopic category (binocular stereo). Stereoscopic methods try to estimate the 3D structure from a pair of images, which are produced from two camera sensors attached in a single setup. The most critical component in the pipeline of Stereoscopy is establishing stereo correspondences between the images. Once those correspondences have been established, the depth of the 3D points can be estimated [1], [2]. Such correspondences are found by matching pixels or higher level features between the two images so that those matches describe the same points or features in 3D space. Most feature detection and tracking methods take advantage of texture variations of the target surfaces in order to detect their location. If the variation is high enough, features can be detected and matched robustly.

Several approaches have been reported to apply stereoscopic methods to MIS data in the literature. Stoyanov [3] proposed to first establish a sparse set of correspondences of salient features and then propagate the disparity information of those salient features to nearby pixels, assuming small disparity changes between neighboring pixels. Based on this, Bernhardt [4] suggested a similar method, including three stereo matching criteria, in order to remove outliers.

Computing descriptors and correspondences in images is often time consuming. Thus, several implementations have been introduced, which rely on executing heavy computational loads on the GPU. More specifically, Roel [5] proposed a hybrid recursive matching approach, performing a non-parametric transformation on the images. Outside of MIS context, Hernandez-Juarez [6] proposed a Semi-Global Matching approach, fully adapting the algorithm to the GPU, taking advantage of special features of modern GPUs, achieving 3D reconstruction with very fast real-time performance.

In order to increase the accuracy of 3D reconstruction, machine learning methods have also been explored. Convolutional Neural Netowrk frameworks for feature matching and disparity estimation have been discussed in the literature [7], demonstrating promising results. However, the lack of availability of MIS data for training such networks poses a serious challenge for the adoption of such methods, which is often addressed with unsupervised learning approaches [8].

Within the context of SMARTsurg, we chose to investigate two methods, which already achieve promising results and real-time performance. First, Efficient Large Scale Stereo Matching (ELAS) [14] and SGM GPU [6], which are mainly targeted at odometry applications, have been adapted to perform 3D reconstruction of MIS data. On the other hand, Quasi Dense [3] has demonstrated accurate results in MIS datasets, but it is unable to perform in real-time, due to slow processing time. Therefore, we propose and present a set of custom modifications to Quasi Dense method, which offer a significant speed up to the original method, by exploiting the latest features of modern GPUs, achieving real-time performance.

2.2 Efficient Large Scale Stereo Matching

This method is based on the one presented in [14][1]. It is mainly targeted at odometry and autonomous driving applications. These environments are very different from MIS in terms of image structure. They have better lighting conditions, contain more distinct features that are very important in stereoscopic methods and finally refer to a completely different scale. Hence, the method needs to be suitably adapted to work in MIS environment, which to our knowledge, has not been successfully attempted yet.

Efficient Large Scale Stereo Matching or ELAS, starts by building a prior over the disparity space by forming a triangulation on a set of robustly matched correspondences, named ‘support points’. That way, the disparity search space can be reduced and the whole procedure can easily be parallelized. Next, a probabilistic Generative Model approach is used for stereo matching, along with a maximum-a-posteriori estimation for disparity estimation. As a result, accurate disparity maps of high resolution images can be computed at high frame rates.

2.2.1 Support Points

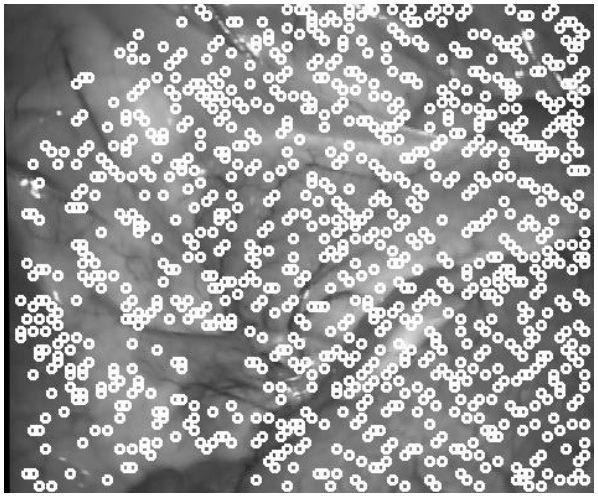

Figure 1: Support Points calculated for ELAS in Porcine Dataset (see Annex I)

In order to facilitate a solid foundation for the probabilistic model, a set of robustly matched stereo correspondences must be computed. For that reason, Sobel matching cost is used for the extraction of support points. Sobel masks of 3 x 3 size and 5 pixel step are chosen, concatenating their horizontal and vertical response over 9 x 9 pixel windows. That way, L1 distance between vectors[2] can be calculated on a regular grid. Since at this initial stage, the disparity search range includes half the image, a set of restrictions must be applied, in order to improve robustness and discard spurious and ambiguous matches. Robustness increases by keeping matches that can be matched both from left to right image and right to left. Next, ambiguity is reduced by discarding matches that have other matches with similarity close to the best match, according to a threshold. Finally, discarding matches that show largely dissimilar values from their surrounding pixels, eliminates spurious matches.

2.2.2 Generative Model for Stereo Matching

Given the image coordinates of the robustly matched support points (u_m, v_m), and their respective disparities d_m, support point is defined as $s_m = (u_m, v_m, d_m)^T$ in $\mathbf{S} = \{\mathbf{s_1}, \dots, \mathbf{s_M}\}$. A set of observations $\boldsymbol{O} = \{\boldsymbol{o_1}, \dots, \boldsymbol{o_N}\}$ is also defined for each image, with each observation $\boldsymbol{o}_n = (u_n, v_n, \mathbf{f}_n)^T$ being the concatenation of its image coordinates and a feature vector $\mathbf{f}^{(l)} or \mathbf{f}^{(r)} \in \mathbb{R}^Q$, with l and r denoting left or right image.

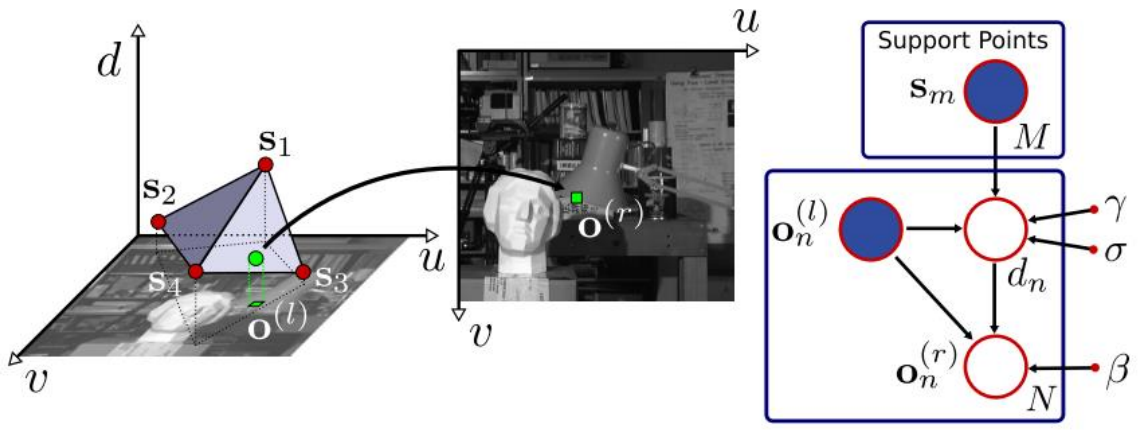

Figure 2: Sampling Process and Graphical Model

Assuming conditional independence between observations $\left\{\boldsymbol{o_n^{(l)}}, \boldsymbol{o_n^{(r)}}\right\}$ and support points $\mathbf{S}$, given the disparities d_n, the joint distribution is factorized

$$p\left(d_n, \boldsymbol{o_n^{(l)}}, \boldsymbol{o_n^{(r)}}, \mathbf{S}\right) \propto p\left(d_n \,|\mathbf{S}, \boldsymbol{o_n^{(l)}}\right) p\left(\boldsymbol{o_n^{(r)}} | \boldsymbol{o_n^{(l)}}, d_n\right)$$

[2] http://mathworld.wolfram.com/L1-Norm.html

with $p\left(d_n|\mathbf{S},\boldsymbol{o}_n^{(l)}\right)$ being the prior and $p\left(\boldsymbol{o}_n^{(r)}|\boldsymbol{o}_n^{(l)},d_n\right)$ the image likelihood, depicted as a graphical model in *Figure 2.*The prior is defined as a combination of a sampled Gaussian and a uniform distribution,

$$p\left(d_n|\mathbf{S},\boldsymbol{o}_n^{(l)}\right) \propto \begin{cases} \gamma + \exp\left(-\frac{\left(d_n - \mu\left(\mathbf{S},\boldsymbol{o}_n^{(l)}\right)\right)^2}{2\sigma^2}\right) & \text{, if } |d_n-\mu| < 3\sigma \ \vee \ d_n \in N_S \\ 0 & \text{, otherwise} \end{cases}$$

where $\mu\left(\mathbf{S},\boldsymbol{o}_n^{(l)}\right)$ a mean function linking the support points and the observations, and N_S the set of all support point disparities in a small 20×20 pixel neighbourhood around $\left(u_n^{(l)},v_n^{(r)}\right)$.

Disparities are calculated by interpolation expressing $\mu\left(\mathbf{S},\boldsymbol{o}_n^{(l)}\right)$as a piecewise linear function, using the Delaunay triangulation, computed on the support points. Image likelihood is expressed as a constrained Laplacian distribution

$$p\left(\boldsymbol{o}_n^{(r)}|\,\boldsymbol{o}_n^{(l)},d_n\right) \propto \begin{cases} \exp\left(-\beta\left\|\mathbf{f}_n^{(l)}-\mathbf{f}_n^{(r)}\right\|_1\right) & \text{if } \begin{pmatrix}u_n^{(l)}\\ u_n^{(l)}\end{pmatrix} = \begin{pmatrix}u_n^{(r)}+d_n\\ u_n^{(r)}\end{pmatrix}, \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{f}_n^{(l)},\mathbf{f}_n^{(r)}$ are 50-dimensional feature vectors of the observations $\boldsymbol{o}_n^{(l)},\boldsymbol{o}_n^{(r)}$ in the left and right image respectively. They contain a concatenation of image derivatives in a 5 x 5 pixel neighbourhood around $(u_n,\ v_n)$, computed from Sobel filter responses.

2.2.3 Disparity Estimation

In order to estimate the disparity map given the left and right images a maximum a-posteriori (MAP) estimation is employed

$$d_n^* = \text{argmax}\, p\left(d_n|\boldsymbol{o}_n^{(l)},\boldsymbol{o}_1^{(r)},\dots,\boldsymbol{o}_N^{(r)},\mathbf{S}\right),$$

where $\boldsymbol{o}_1^{(r)},\dots,\boldsymbol{o}_N^{(r)}$ denotes all observations in the right image which belong in the same epipolar line. It can be factorized as

$$p\left(d_n|\boldsymbol{o}_n^{(l)},\boldsymbol{o}_1^{(r)},\dots,\boldsymbol{o}_N^{(r)},\mathbf{S}\right) \propto \sum_{i=1}^{N} p\left(\boldsymbol{o}_i^{(r)}|\,\boldsymbol{o}_n^{(l)},d_n\right).$$

Next prior and the image likelihood are replaced, with the expressions previously defined. By taking the negative logarithm, results in an energy function that can be easily minimized,

$$E(d) = \beta\left\|\mathbf{f}_n^{(l)}-\mathbf{f}_n^{(r)}(d)\right\|_1 - \log\left[\gamma + \exp\left(-\frac{\left[d-\mu\left(\mathbf{S},\boldsymbol{o}^{(l)}\right)\right]^2}{2\sigma^2}\right)\right]$$

A dense disparity map can be obtained by minimizing the above energy function.

2.3 Semi-Global Matching GPU

Local methods based on correlation as the one described above, are very efficient in calculating accurate disparity maps in regions with varying texture information, at close to real-time speeds. However, they do come with certain pitfalls. First, they assume constant disparities within correlation windows, which often leads to blurred object boundaries. They also fail to calculate correct disparity maps in untextured areas, since stereo matching costs cannot be computed without variance in intensity [15].

Smooth surfaces are very common in MIS environment. Semi-global methods (SGM) have been reported to produce results in these areas, while also being capable or retaining edges. While many variations exist, a typical SGM algorithm starts with a calculation of an initial similarity criterion (Mutual Information, Census Transform). Then, path accumulation of 4 or 8 paths is performed and the path costs are summed up into one disparity cost cube. Then, with a winner-takes-all (WTA) strategy, disparities that are associated with the minimum cost are chosen [16].

However, SGM methods require a lot of time for computations, making them unsuitable for real-time MIS scenarios. Hence, in order to benefit from the –complementary to the local methods- performance of SGM methods, an optimized version of SGM is being evaluated [6]. It is implemented in CUDA programming language, using a sophisticated GPU pipeline, adapted to take advantage of modern GPUs. It also includes important modifications to the original SGM algorithm, to achieve faster computation times. In fact, it is able to achieve almost 200 fps with 640 x 480 image size. The quality of the produced disparity maps is not yet up to the standard of the previous method, but both the adjustment of the method and its combination with the previous one, is under investigation.

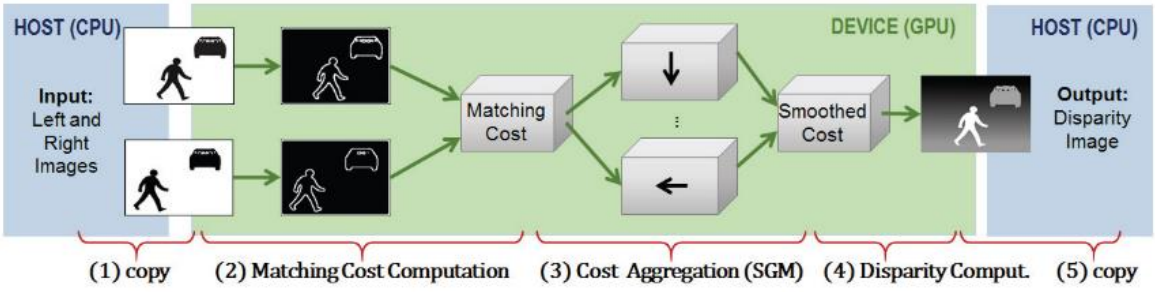

Figure 3: SGM-GPU implementation pipeline

2.3.1 Description

In order to achieve fast computation time, several modifications are introduced to the classic SGM algorithm. Most of them are techniques, which take advantage of the modern GPU technologies, as presented in *2.4.3.2 GPU Parallelization*. However similar to the original SGM pipeline, the matching cost between pixels is computed by taking the Hamming distance of the Center-Symmetric Census Transform (CSCT) of each pixel. CSCT is able to produce results with similar quality with other matching costs, but requires less time to compute. It is also a is

a more “compact” variant of Census Transform, since produces 32 bit feature vectors instead of 64 bit. The CSCT of a pixel (x, y) in image I is calculated as

$$CSCT_{9,7}(I,x,y) = \bigotimes_{i=1}^{4} \begin{cases} \bigotimes_{j=-3}^{3} s\big(I(x+i,y+j), I(x-i,y-j)\big) \\ \bigotimes_{j=1}^{3} s\big(I(x,y+i), I(x,y-j)\big) \end{cases},$$

where $s(u,v)$ is 1 if $u \geq v$ and $\otimes$ is bit-wise operator. The Matching Cost $MC(x,y,d)$ between a pixel (x,y) in the base image and each potentially corresponding pixel in the match image at disparity d is

$$MC(x,y,d) = bitcount\Big(CSCT_{9,7}(I_{base},x,y) \oplus CSCT_{9,7}(I_{match},x-d,y)\Big),$$

where $\oplus$ is bit-wise exclusive-or and $bicount$ counts the number of bits set to 1.Once the 3-dimensional matrix with the Matching Cost is calculated, disparity can be computed by solving a one-dimensional dynamic programming minimization problem.

$$L_r(x,y,d) = MC(x,y,d) + \min \begin{cases} L_r\big(x-r_x, y-r_y, d\big) \\ L_r\big(x-r_x, y-r_y, d-1\big) + P_1 \\ L_r\big(x-r_x, y-r_y, d+1\big) + P_1 \\ \min_i L_r\big(x-r_x, y-r_y, i\big) + P_2 \end{cases} - \min_k L_r\big(x-r_x, y-r_y, k\big),$$

where L_r is defined as the matrix containing the smoothing aggregated cost for path r. $MC(x,y,d)$ is the original matching cost, while the remainder of the equation adds the lowest cost of the previous pixel $\big(x-r_x, y-r_y\big)$ of the path, including the appropriate penalties for small disparity changes P_1 and discontinuities P_2. In order to avoid the values of L' permanently increasing along the path, the minimum path cost $L_r\big(x-r_x, y-r_y, k\big)$ of the previous pixel is subtracted.

2.4 Quasi Dense Stereo Matching

The Quasi Dense Stereo Matching method is built for the reconstruction of 3D information from stereo-laparoscopic images during robotic assisted surgery [3]. It is a novel stereo semi-dense reconstruction algorithm that propagates disparity around a set of candidate feature matches.

In this way, problems with specular highlights and occlusions from instruments can be eliminated. Furthermore, the algorithm can be used with any feature matching strategy allowing the propagation of depth in very disparate views.

Quasi Dense Stereo Matching has two phases: Sparse Matching (presented in Section 2.1.1) and Dense Matching (presented in Section 2.1.2). Sparse matching consists of a sparse 3D reconstruction base on feature matching across the stereo pair. In the dense matching phase, structure is propagated into neighbouring image regions. In addition, some parts of the procedure pipeline can be parallelized in a GPU implementation. Consequently, real time performance can be achieved as well as high quality disparity map computation.

2.4.1 Sparse Matching

The initial step of this method is to recover a sparse set of matches across the stereo-laparoscopic image pair using a feature based technique. This step includes two more sub processes. Firstly, it detects strong feature points in the left image. Detection can be achieved with several feature detection methods, namely ORB [9], SURF [10] and G-SURF [11] as seen in *Figure 4*. After experimentation, it is concluded that the most efficient approach is Good Features to track proposed by Shi-Tomasi [12]. It recovers corners or features in an image based on image intensity gradient.

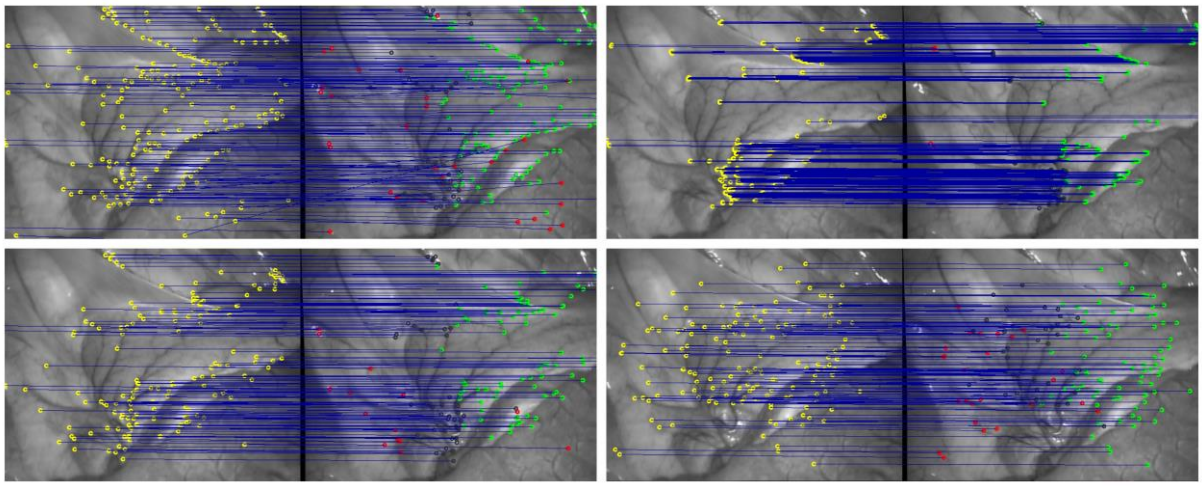

Figure 4: Sparse Matching with different feature matching methods, Good Feature to Track (top-left), ORB (top right), SURF (bottom left) and G-SURF (bottom right)

Secondly, in order to find the corresponding points in the right image, optical flow is estimated using Lucas-Kanade method [13]. This method assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. By combining information from several nearby pixels, the Lucas-Kanade method can often resolve the inherent ambiguity of the optical flow equation.

2.4.2 Dense Matching

As a sparse set of 3D points has been established in the surgical field of view, it is possible to propagate 3D information to cover a semi-dense portion of operating field domain. It should be mentioned that all features correspondences which were calculated in the previous step, are used as seed matches. They are sorted, in descending order, based on the correlation score between their respective templates, and stored using a priority queue structure. After that, the algorithm proceeds to propagate structure around the matches' correlation scores on a best-first basis popping the priority queue. As the algorithm iterates, new matches are added to the queue. When there are no matches to be popped, the algorithm terminates. If a seed match consists of a sparse pixel $p_0 = (x, y)$ in the left image and the corresponding pixel $p_1 = (x', y')$ in the right image, then a spatial neighbourhood $N(p_0, p_1)$ is defined and can be used to enforce a 2D disparity gradient limit as a smoothness constraint. Thus, for each seed pixel, the spatial neighborhoods around them are defined by

$$N(p_0) = \{(x - n_x, y - n_y) : n_x, n_y \in [-N, N]\}$$
$$N(p_1) = \{(x' - n_x - d_x, y' - n_y - d_y) : d_x, d_y \in [-Dg, Dg]\}$$

, where $(x - n_x, y - n_y)$ denotes the coordinates of each pixel within a spatial window of $(2N + 1) \times (2N + 1)$ pixels centered at seed pixel p_0 and can be matched with a candidate pixel $(x' - n_x - d_x, y' - n_y - d_y)$, which is placed within a spatial window of $(2Dg + 1) \times (2Dg + 1)$ pixels centered at $(x' - n_x, y' - n_y)$ in the right image. In conclusion, if (U_0, U_1) denotes a candidate pair of pixels, then the full match propagation neighbourhood is

$$N(p_0, p_1) = \{(U_0, U_1), U_0 \in N(p_0), U_1 \in N(p_1)\}$$

The algorithm uses a dissimilarity measure during propagation in order to determine which pixels to be matched together. A very common and efficient measure is the zero mean normalized cross correlation (ZNCC)[3], which is less prone to illumination bias in homogeneous regions while it is also more indicative in regions with discriminative texture. Given that $U_0 = (u_0, v_0)$ and $U_1 = (u_1, v_1)$ is a candidate pair of pixels, then the average gray value inside a spatial window of size $(2w + 1) \times (2w + 1)$ around U_0 can be calculated by

$$\overline{Img_0}(u_0, v_0, w) = \frac{1}{(2w+1)^2} \sum_{i=-w}^{w} \sum_{j=-w}^{w} Img_0(u_0 + i, v_0 + j),$$

where Img_0 denotes the left image. In addition, the standard deviation inside the same window can be calculated by

[3] https://martin-thoma.com/zero-mean-normalized-cross-correlation/

$$\sigma_0(u_0, v_0, w) = \sqrt{\frac{1}{(2w+1)^2}\left(\sum_{i=-w}^{w}\sum_{j=-w}^{w}(Img_0(u_0+i, v_0+j) - \overline{Img_0}(u_0, v_0, w))^2\right)}.$$

Similarly, these parameters can be calculated for pixel U_1, which is placed in the right image. Combining the above equations, the final dissimilarity measure can be given by

$$ZNCC(Img_0, Img_1, u_0, v_0, u_1, v_1, w) = \frac{\sum_{i=-w}^{w}\sum_{j=-w}^{w}\prod_{t=0}^{1}(Img_t(u_t+i, v_t+j) - \overline{Img_t}(u_t, v_t, w))}{(2w+1)^2 \cdot \sigma_0(u_0, v_0, w) \cdot \sigma_1(u_1, v_1, w)}.$$

The range of the computed value is $[0,1]$. Thus, the higher the ZNCC gets, the more are those two pixels correlated. The propagation stops when no more matches can be achieved, because the correlation scores are lower than a predefined threshold.

2.4.3 Optimization for real-time performance

This sub-section explicitly refers to the modifications that have been made by CERTH to the original method in order to speed it up enabling real-time performance without compromising the quality of the reconstruction.

2.4.3.1 1-D constraint

For performance improvement of the existing method, it is assumed that only rectified images are used. This means that the algorithm focuses on looking for possible matches on the horizontal dimension, which demands fewer calculations and less memory. In this case, the equation that calculates the spatial neighborhood around a seed pixel in the right image is given by

$$N(p_1) = \{(x' - n_x - d_x, y' - n_y) : d_x \in [-Dg, Dg]\},$$

which means that every pixel in the left image, which belongs to $N(p_0)$ can be matched with $(2Dg + 1)$ candidate pixels in the right image.

2.4.3.2 GPU Parallelization

For further improvement in computational performance of the method, it is possible to exploit modern GPU technology to concurrently calculate multiple correlation windows and propagate structure over multiple pixels. Specifically, a CUDA kernel calculates all correlation scores inside a full match propagation neighbourhood $N(p_0, p_1)$ by launching a block of threads for each seed match. In this way, a large number of concurrent threads that run on modern graphic cards are activated.

According to the serial implementation of the method, correlation scores that are being calculated during an iteration of the algorithm refer to just one seed match. Then, the algorithm validates these scores, checks whether any of the pixels related to the potential matches have already been matched and if not, stores them in a priority queue. In order to proceed to the next iteration, the algorithm retrieves the best matching pair from the priority queue and treats

it as a seed match. Subsequently, a respective full match propagation neighborhood around the seed match is calculated and the algorithm tries to find new matches within it. In this way, structure is always propagated around the best seeds. On the other hand, the proposed parallelized implementation of the method calculates the matching scores for the total number of the seeds which are available and have been stored in a simple array. The use of simple array instead of a priority queue is preferable considering the additional overhead that priority queues create because of the sorting procedure which runs in the background. However, bionic sort[4] is applied inside the kernel so it is possible for the algorithm to choose the highest matching score for each pixel within a seed's neighborhood. The following subsections describe the features of a modern GPU and how these are used by the proposed approach in order to achieve real-time performance.

2.4.3.2.1 Shared memory

Shared memory is much faster than local or global memory, because of the fact that it is on-chip. In fact, shared memory latency is roughly 100x lower than uncached global memory latency. As it is illustrated in Figure 5, shared memory is allocated per thread block, so all threads in a block have access to the same shared memory. This means that threads can access data in shared memory loaded from global memory by other threads within the same thread block.

[4] https://www.geeksforgeeks.org/bitonic-sort/

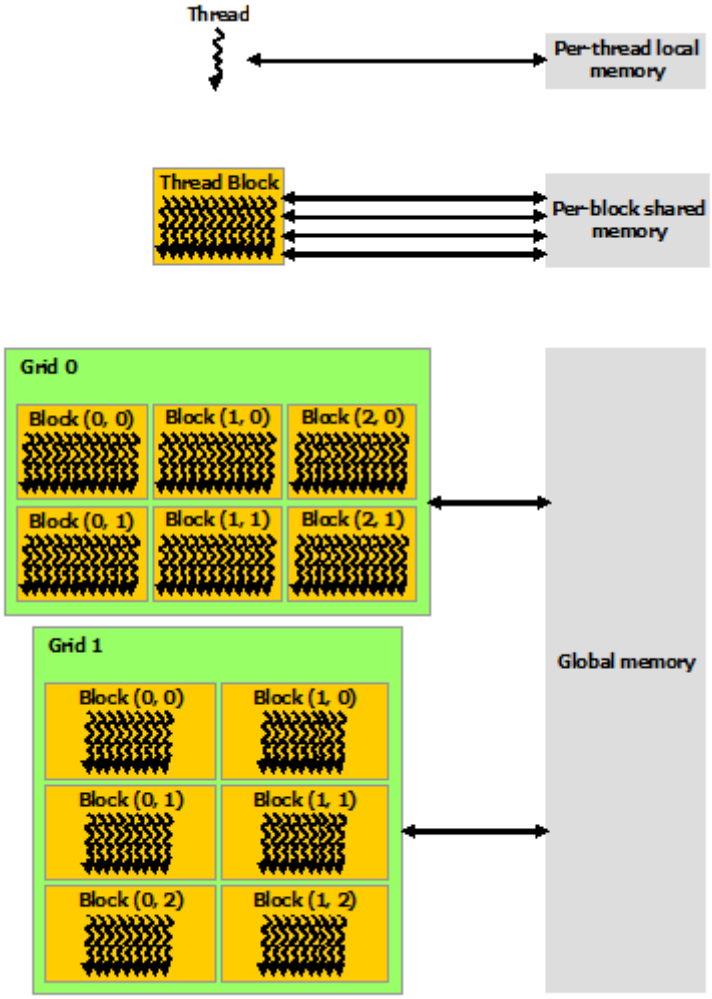

Figure 5: Memory Hierarchy in GPU [5]

Concerning the parallelised implementation of the method, the coordinates of seed pixels are initially stored in global memory. Each block of threads calculates the coordinates as well as the correlation scores for every candidate pixel within the propagation neighbourhood around seed pixels and finally sorts them in respect of correlation scores. Using shared memory to store the results of these calculations significantly accelerates the whole procedure by reducing the total number of access calls in global and local memory.

2.4.3.2.2 Pinned memory

Host (CPU) data allocations are pageable by default and GPU cannot access data directly from pageable host memory. So, when a data transfer from pageable host memory to device memory is invoked, the CUDA driver must first allocate a temporary pinned host array, copy

[5] https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#memory-hierarchy

the host data to the pinned array and then transfer the data from the pinned array to device memory, as illustrated in Figure 6. Cost of the transfer between pageable and pinned host arrays can be avoided by directly allocating host arrays in pinned memory. Doing so, the data transfer rate can be increased although it depends on the type of host system (motherboard, CPU, chipset). Moreover, over-allocating pinned memory can reduce overall system performance because it reduces the amount of physical memory available to the operating system and other programs.

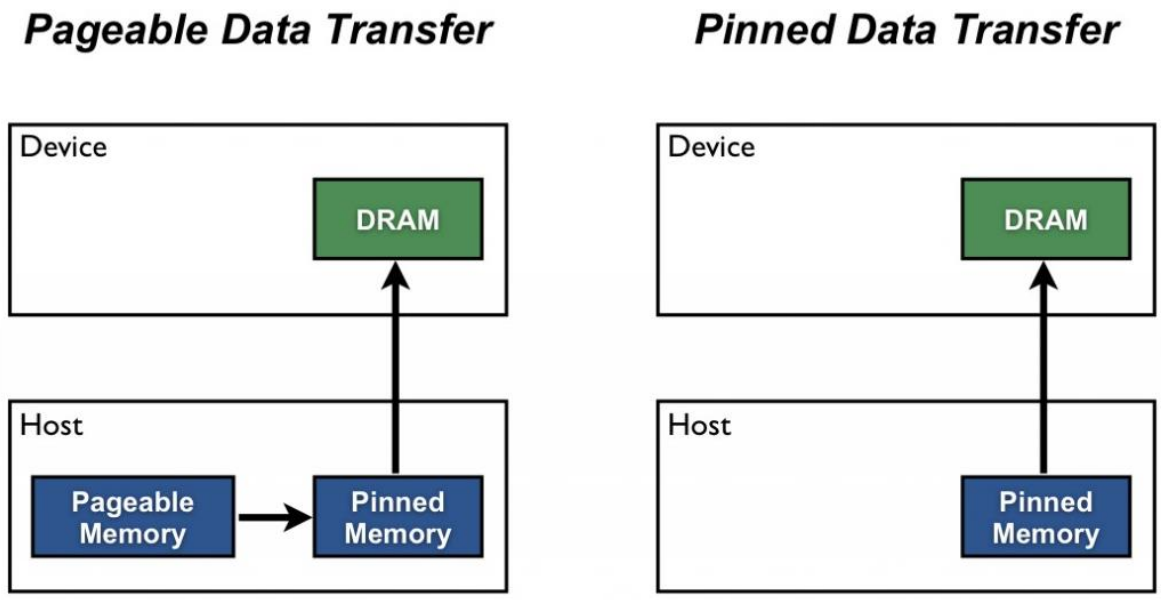

Figure 6: Difference between pageable and pinned data transfer[6]

Because of the fact that during the propagation of the structure around the matches the CUDA kernel can be called many times, data transfers must not dominate the overall execution time. Also, at the end of each kernel execution, coordinates of new potential matches have to be transferred from device to host memory in order to be validated. By allocating the appropriate arrays directly in pinned memory, removes intermediate transfers and decreases the overall execution time correspondingly.

2.4.3.2.3 Overlapping kernel execution and data transfers

A stream in CUDA is a sequence of operations that execute on the device in the order in which they are issued by the host code. While operations within a stream are guaranteed to execute in the prescribed order, operations in different streams can be interleaved and, when possible, they can even run concurrently. In addition, not only modern GPUs give the ability to execute kernel asynchronously but also transfer data. Since all operations are non-blocking with

[6] https://devblogs.nvidia.com/how-optimize-data-transfers-cuda-cc/

respect to the host code, multiple streams can be launched simultaneously separating the total number of calculations into equal pieces.

As illustrated in Figure 7 when only one stream is used, data transfers and kernel execution are served sequentially. On the other hand, in asynchronous version when stream 1 executes the kernel, stream 2 transfers data from host to device memory (H2D). In the next time step, stream 1 transfers data back to host (D2H), stream 2 executes the kernel while stream 3 transfers data to device. Thus, this pattern is followed repeatedly and results in overlapping kernel execution and data transfers reducing the overall execution time.

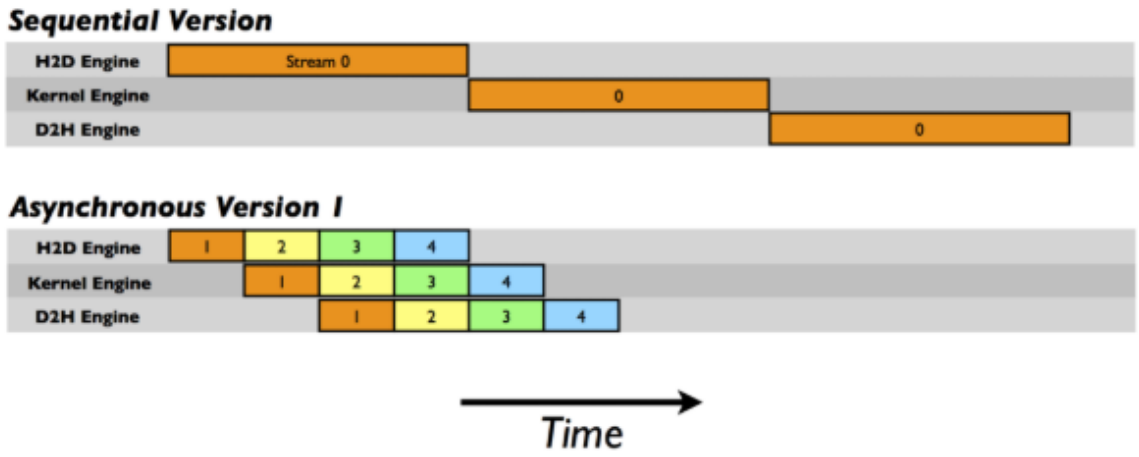

Figure 7: Execution time line[7]

This technique is applied to the parallelized version of the method by dividing the initial number of seed matches by the number of streams and distributing the appropriate amount of data to them. As a result, further performance optimization has been achieved, especially when the number of seed matches is relatively high.

[7] https://devblogs.nvidia.com/how-overlap-data-transfers-cuda-cc/

3 Stereo Processing Framework

3.1 Overview

The presented methods were evaluated with the use of existing datasets (see *Annex I: Datasets*). In the next months we intend to test and evaluate the proposed methods with image sequences taken by a stereo endoscopic camera system acquired by CERTH using phantoms. Pre and post processing was performed on the existing datasets as well.

3.2 Vision Station Setup

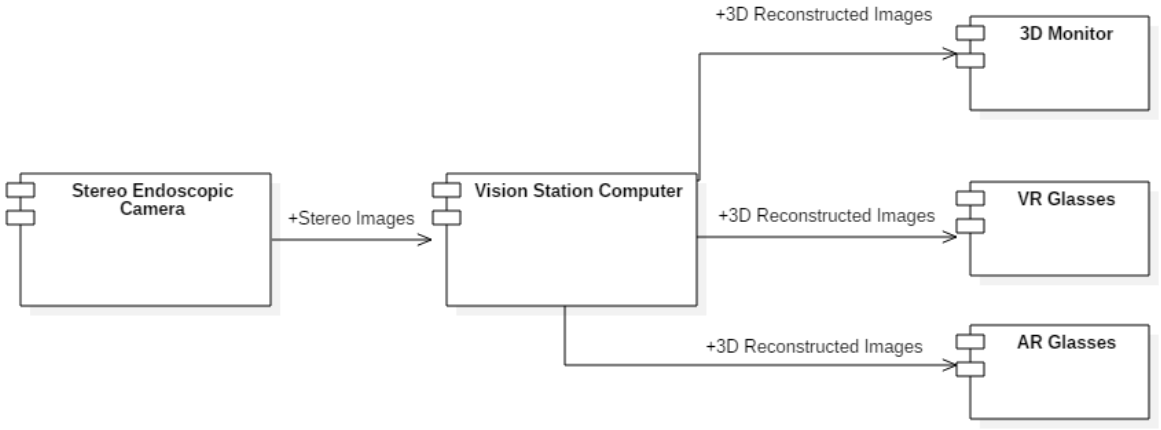

Figure 8: Vision Station Setup

The vision station computer that is intended for the 3D reconstruction of the surgical field and the augmented reality composite view creation and visualisation consists of:

- Nvidia GeForce GTX Titan X (Maxwell architecture), graphics card
- Blackmagic Design Decklink Duo 2, capture and playback card
- 64GB RAM
- Xeon 6-core (dual thread) CPU at 3.5GHz

A stereo endoscopic camera will provide the input for the 3D reconstruction module and the output will be given to the surgeons via a 3D monitor, VR glasses and AR glasses. The vision station computer will use the Blackmagic Design card as both capture and playback card.

3.2.1 Stereo Endoscopic Camera

We intend to use a full HD Stereo ENDOCAM for high detail resolution and for graphic and realistic 3D-visualisation, and specifically Richard Wolf ENDOCAM Epic 3DHD System (*Figure 9*). The output of the camera will be connected to the Capture and playback card.

Figure 9: Stereo Endoscopic Camera, ENDOCAM Epic 3DHD System[8]

3.2.2 Capture and Playback Card

The Blackmagic Design Decklink Duo 2 has 4 independent bi-directional 12-bit SD/HD input/output connectors. We intend to use two of the connectors as input connections, which will do the video capture from the stereo endoscopic camera, and the other two connectors as output connections. Output connections will go to the 3D monitor and the AR glasses.

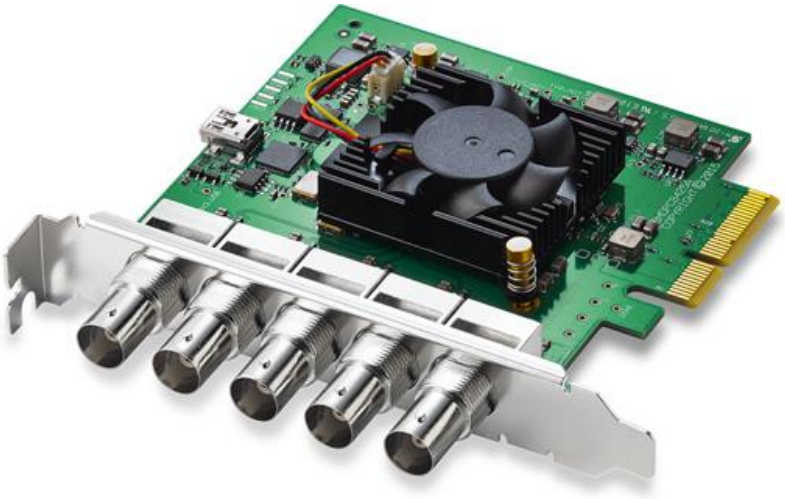

Figure 10: Blackmagic Design Duo 2 Capture and Playback Card[9]

[8] https://www.richard-wolf.com/broschueren/Imaging/A_670_ENDOCAM_Epic_3DHD_XI13_GB.pdf

[9] https://www.blackmagicdesign.com/products/decklink/techspecs/W-DLK-31

Thus, the Blackmagic Design Decklink Duo 2 card will play the role of two capture cards and two playback cards. The playback will contain the result of 3D reconstruction and augmented reality composite view visualisation.

3.2.3 Graphics Card

The vision station computer includes an Nvidia GeForce GTX Titan X graphics card that is intended to be used for achieving real-time 3D reconstruction of the surgical field, as well as provide the output to the VR glasses.

Figure 11: Nvidia GeForce GTX Titan X graphics card[10]

The GPU engine has 3072 NVIDIA CUDA® Cores and 1075MHz boost clock. It has memory of 12 GB and it can achieve memory speed up to 7 Gbps.

3.3 ROS Framework

ROS[11], stands for Robot Operating System, is a collection of software tools and libraries for robot software development, a robotics middleware providing operating system-like functionality on a heterogeneous computer cluster. ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Software in the ROS can be separated into three groups:

- Language-and platform-independent tools used for building and distributing ROS-based software
- ROS client library implementations

[10] https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/

[11] http://www.ros.org/

• Packages containing application-related code which uses one or more ROS client libraries.

ROS includes libraries related to perception, motion planning, hardware drivers, simulation, signal processing and more. ROS has been chosen due the easy to use and customizable communication interface, which allows multiple processes to have access at the same data simultaneously. SMARTsurg vision components have ROS nodes that can communicate with each other through messages, using TCPROS connectivity. TCPROS is a transport layer for ROS messages and services. It uses standard TCP/IP sockets for transporting message data. Using ROS also enables us to have better control over the system as a whole, using integrated error reporting tools and visualisation libraries.

3.4 Algorithm

Our stereo processing framework includes the necessary operations, which must be performed on the raw images that are acquired from the endoscopic stereo camera or loaded from the dataset files. These are standard operations, not depending on the differences in the source of the stereo image input. Thus, our suggested pipeline remains the same and can be very easily adapted to any kind of input/output.

3.4.1 Pre Processing

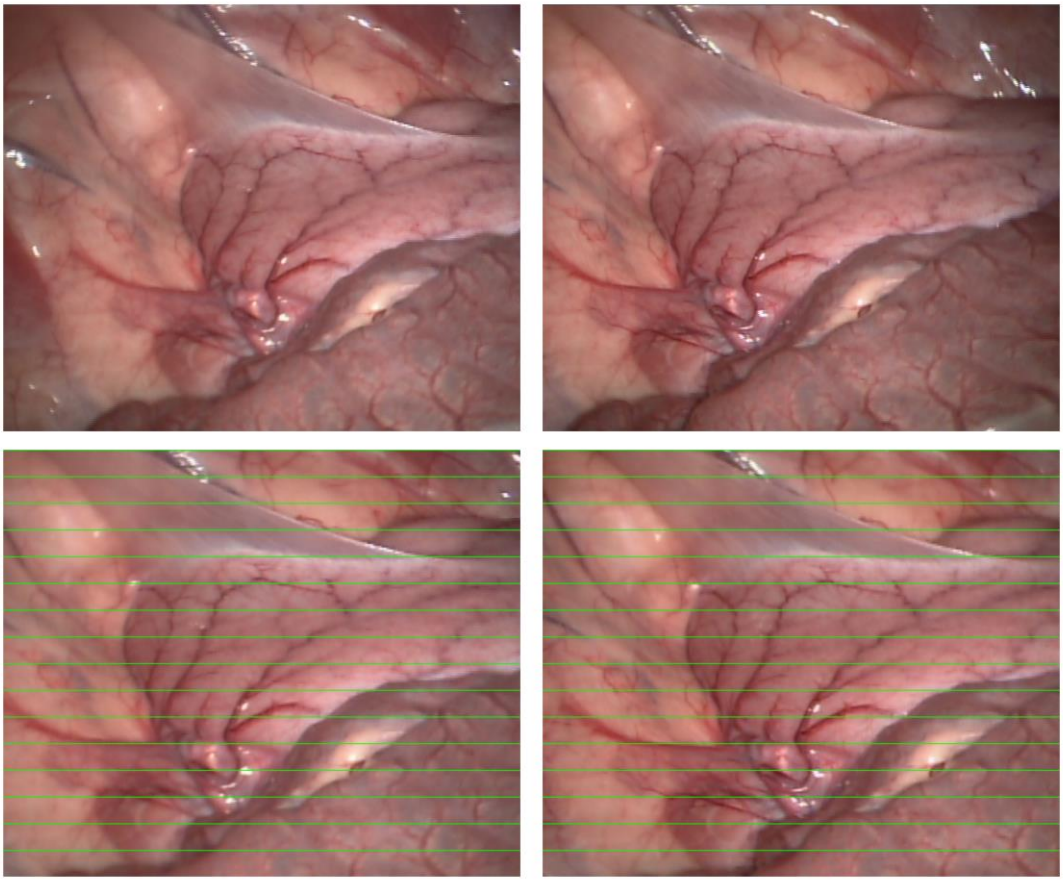

Figure 12: Stereo Rectification procedure, raw images before (top) and after (bottom) rectification

Raw images from endoscopic cameras are by default distorted and depict different regions of the surgical area, due to the distance between the sensors (baseline). Given the calibration parameters of the camera setup, images can be transformed (rectified) to be projected on a common plane. That way, it is ensured that points in the left image and their corresponding match in the right image are always in the same epipolar line, parallel to the horizontal axis. Next, images are converted to grayscale, since it reduces search space without affecting the performance of reconstruction algorithms. Finally, an optional Gaussian Filter is applied in datasets which include noise.

3.4.2 Post Processing

Disparity maps reconstructed by the presented methods, are constructed by point to point correspondences. If a point in the left image is not matched with a point in the right image, then its disparity is not estimated. That results in a hole in the final disparity image. In addition, erroneous matching can also lead in outlier 3D points, causing large errors and visual artifacts.

To address these issues, a set of post processing filters is applied to the estimated disparity maps, in order to improve the 3D reconstruction result.

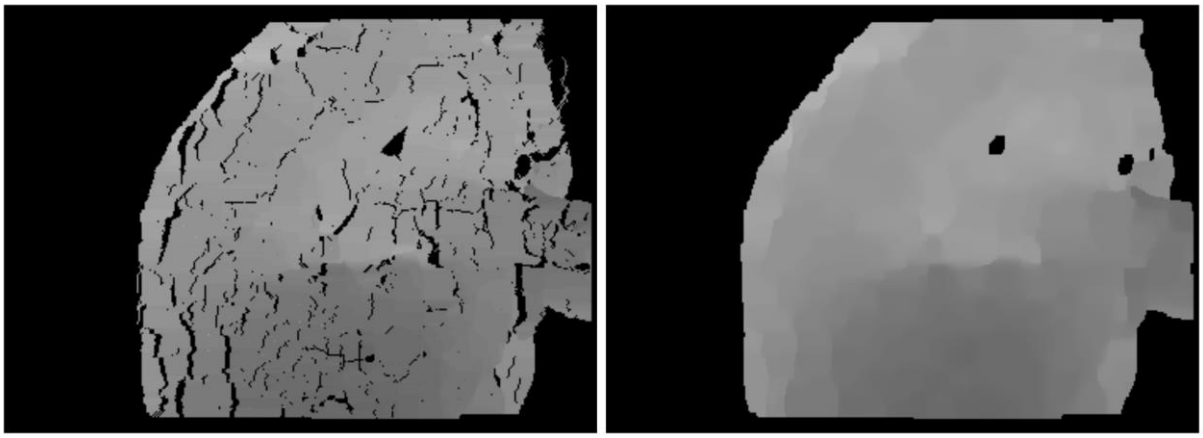

Figure 13: Post processing filtering, estimated disparity map before filtering (left) and after filtering (right)

First, a bilateral filter is applied. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. To preserves sharp edges, the weights, which are based on a Gaussian distribution, depend not only on Euclidean distance of pixels, but also on the radiometric differences. Next, a median filter, which replaces each point with the median of its neighboring points is applied, to remove noise while also preserving edges. Result of the post processing filtering procedure are demonstrated in *Figure 13*.

4 Evaluation

In order to assess the quality, accuracy and performance of the 3D reconstruction methods, an evaluation framework is included. Each method is evaluated over a set of datasets, with specific attributes and challenges. The first dataset is a breathing simulation sequence of a deforming silicon heart. It has an initial resolution of $360x288$ pixels, effectively reduced to $301x227$ after rectification. The second dataset is a kidney phantom captured from various poses and under the presence of smoke, occlusion and bad lighting conditions. This is a challenging dataset for 3D reconstruction provided by POLIMI [17] and it has a resolution of $640x480$ pixels ($388x272$ pixels after rectification). Both those datasets are accompanied with ground truth laser scans, providing ground truth information. The last dataset, is a video sequence of a porcine uterine horn exploration. It has a resolution of $640x480$ pixels, reduced to $480x396$ pixels after rectification. This dataset depicts an in-vivo sequence, therefore no ground truth is available. More information regarding the datasets and their specifications are given in Annex I. This variety in datasets provides us with useful insight on the strengths and weaknesses of each method for each specific use case. The results from our evaluation framework are reported in two sections.

First, Quantitative results are obtained for datasets which include ground truth laser scans (I.1, I.2). After ground truth disparity maps are rectified, Mean Error (ME) [12] and standard deviation is calculated, between them and the ones estimated from the applied methods. ME is calculated as the average absolute difference in pixels between the estimated and the actual disparity of each pixel. It is a basic evaluation metric, able to provide a general performance indicator. More specific and advanced metrics will be included if necessary in the future. Qualitative section, as the name suggests includes the disparity maps estimated for datasets without ground truth (I.3), along with multiple views of the 3D reconstructed point cloud. Finally, experiments are accompanied with results regarding real-time performance, which contain the execution time for the 3D reconstruction of each method over a single frame of the respective dataset.

Results in this chapter will be reported in the same format. In all figures, each row contains results, organized in columns, regarding a single method. The first column shows the disparity map estimated by the method, while the second column includes the depth maps. The third column shows images of the Mean Error (ME) between the estimated disparity map and the ground truth disparity map, supplemented with the dataset. In cases where ground truth is not provided (Qualitative), this column is excluded. Finally, the last column contains a snapshot of the reconstructed 3D point cloud, calculated from the estimated disparity map.

[12] https://en.wikipedia.org/wiki/Mean_absolute_error

4.1 Quantitative

As mentioned above, Quantitative evaluation is performed in datasets, which include ground truth. Therefore, a quantitative error between the estimated disparity map and the ground truth disparity map can be calculated, indicating the accuracy of our 3D reconstruction result. Mean Error is chosen as a simple and effective metric, along with statistical variance of error values. To get a visual representation of the reconstruction error, evaluation maps are constructed, encoding error information in color. Deep blue color indicates minimal error, while red color represents points whose ground truth is not available. Including the number of points which are reconstructed is also important for the evaluation of methods, since it accounts for holes or background areas.

In both datasets, the best performing method, in terms of mean error is Quasi Dense CPU. In fact, in the Deforming Silicon Heart Dataset, it achieves similar reconstruction error but smaller variance of disparities compared to the original work [3], namely 1.39 mean error in pixels and variance of ± 1.72 pixels. It also achieves the best reconstruction in the EndoAbs dataset $(0.97/\pm 0.96)$, while reconstructing 77.09% of the points. Nonetheless, it has the highest execution time from all the methods, with 682 ms and 839 ms for Deforming Silicon Heart and EndoAbs dataset, respectively.

Its proposed GPU implementation performs well in the Deforming Silicon Heart Dataset heart dataset $(1.63/\pm 3.22)$, and in the EndoAbs dataset $(1.60/\pm 4.40)$, but introduces larger variance than its CPU counterpart. However, it achieves a significant speed up to the original CPU method, being about 18 times faster, running at less than 45 ms (or greater than 21 fps) for both datasets. This offers upside, since additional improvements can be introduced in the method, while keeping the execution time within the real-time limits.

ELAS also shows promising performance, mostly in the EndoAbs dataset, achieving error close to Quasi Dense GPU method $(1.63/\pm 2.57)$. It also reconstructs the largest percentage of points in both datasets $(79.40\%$ and $84.79\%)$, compared to Quasi Dense methods, due to their lack of support points in untextured areas. ELAS is a highly configurable method, with various parameters affecting the reconstruction result and execution time. Hence, performance-wise, execution times vary depending on the choice of parameters, but with proper configuration, it is still able to achieve near real-time performance (83 ms in Deforming Silicon Heart Dataset). However, in the EndoAbs dataset its performance is far from real time (943 ms).

SGM GPU is a massively parallel method, achieving incomparable real-time performance to other methods. It can operate in frame rates 20 times faster than the real-time threshold. However, reconstruction results are not quite up to the standard, since mean error for the MIS datasets is almost 5 to 10 times larger from the other methods $(10.85/\pm 16.21, 13.50/\pm 338.33)$.

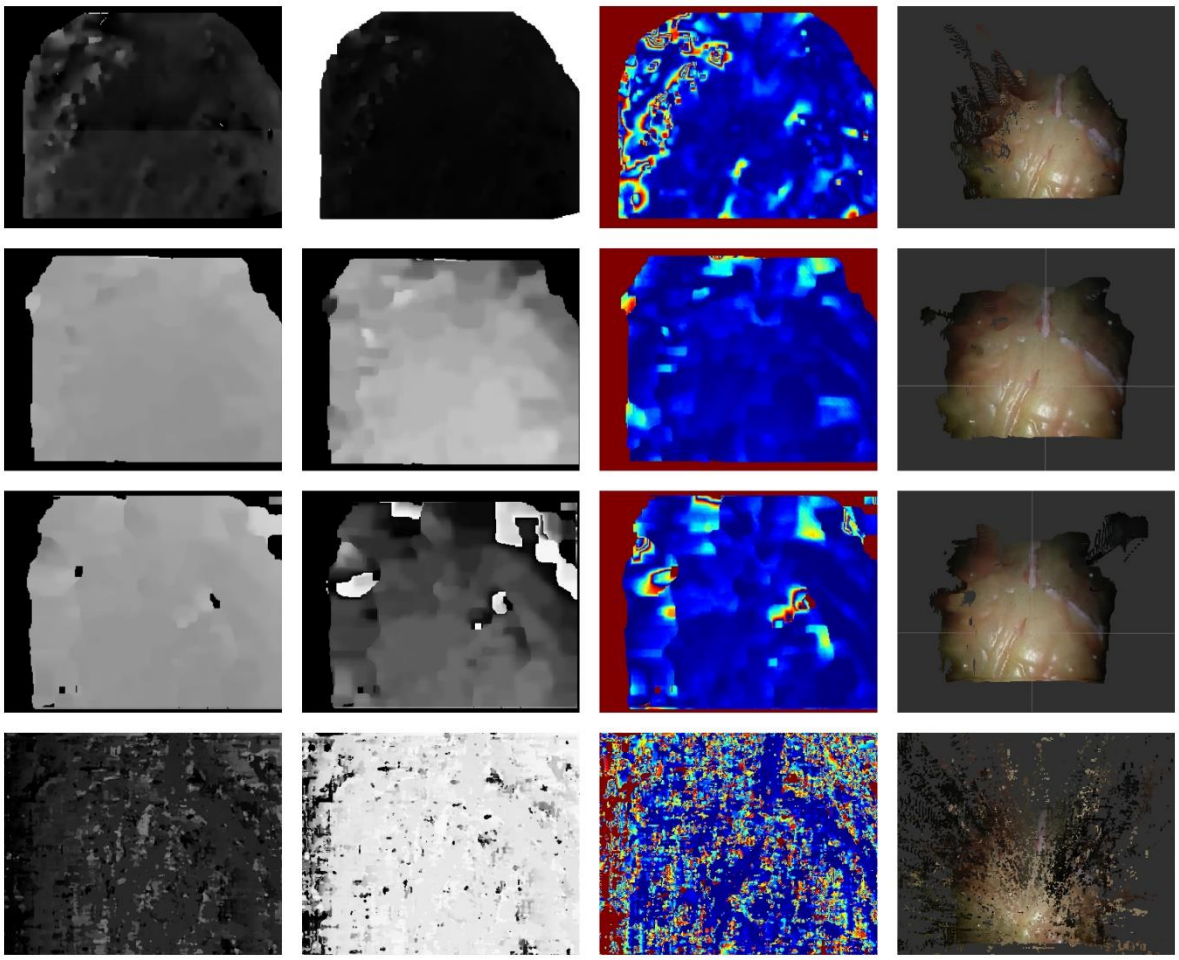

Figure 14: 3D reconstruction results for Deforming Silicon Heart Dataset Dataset (I.1) in rows: 1) ELAS, 2) Quasi Dense CPU, 3) Quasi Dense GPU, and 4) SGM GPU.

Table 1: Execution times for Deforming Silicon Heart Dataset (I.1)

Method	ME	Variance	Reconstructed %	Execution Time
ELAS	2.1	4.3	79.40%	83.3 ms
Quasi Dense CPU	1.39	1.72	75.10%	682 ms
Quasi Dense GPU	1.63	3.22	74.90%	35.36 ms
SGM GPU	10.85	16.21	92.96%	0.2 ms

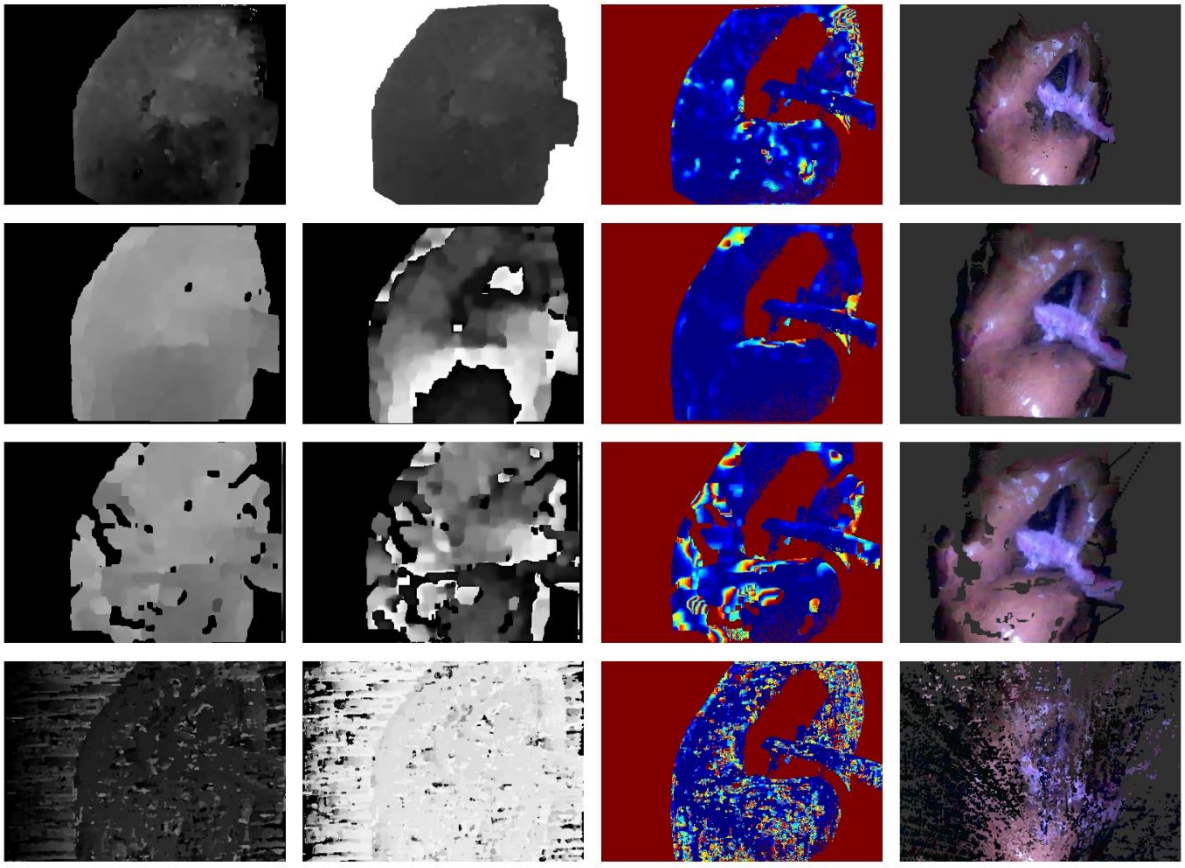

Figure 15: 3D reconstruction results for EndoAbs Kidney Dataset (I.2) in rows: 1) ELAS, 2) Quasi Dense CPU, 3) Quasi Dense GPU, and 4) SGM GPU

Table 2: Execution times for EndoAbs Kidney Dataset (I.2)

Method	ME	Variance	Reconstructed %	Execution Time
ELAS	1.63	2.57	84.79%	943 ms
Quasi Dense CPU	0.97	0.96	77.09%	839 ms
Quasi Dense GPU	1.60	4.4	76.25 %	45.44 ms
SGM GPU	13.50	338.33	96.87 %	0.4 ms

4.2 Qualitative

Datasets accompanied with ground truth are produced by phantoms and their corresponding laser scans. However, that is not the case in real MIS surgery, where in-vivo surgical scenes must be reconstructed in 3D. Thus, the inclusion of an in-vivo dataset from a real operation is important. Such dataset introduces important 3D reconstruction challenges, namely tissue

deformation, reflections, blood, smoke and occlusion from surgical instruments. These challenges need to be addressed by the reconstruction algorithms, towards their adaptation in real MIS procedures. However, since ground truth data are not available, no quantitative error metric can be applied, which results in evaluation of the dataset only from its visual appearance.

Although Porcine Uterine Horn Dataset (I.3) introduces few of the aforementioned 3D reconstruction challenges described above, namely reflections and deformation from respiration, it has stronger texture variations. Thus, features that are more robust can be extracted and more confident matching cost can be computed. Hence, Quasi Dense methods, which are based on such matching costs, were expected to perform best. Once again, Quasi Dense CPU method produces the best-looking result, but this time, a very similar result is obtained by its GPU counterpart. However, Quasi Dense CPU requires almost 2.5 seconds to reconstruct a frame, while Quasi Dense GPU can process a single frame in less than 90 ms. ELAS recovers the geometry quite accurately, but produces erroneous 3D regions especially in points closer to the image borders. Finally, SGM GPU is showing the best performance in execution time (0.4 ms), but fails to estimate accurate disparity maps.

In conclusion, Quasi Dense GPU method is the one to take out from the Quantitative evaluation. Given texture variations in images, the accuracy of stereo matching is high and propagation of matches is performed in the correct paths. Thus, Quasi Dense GPU can achieve similar quality to the 3D reconstructed result of Quasi Dense CPU, while processing frames almost 30 times faster.

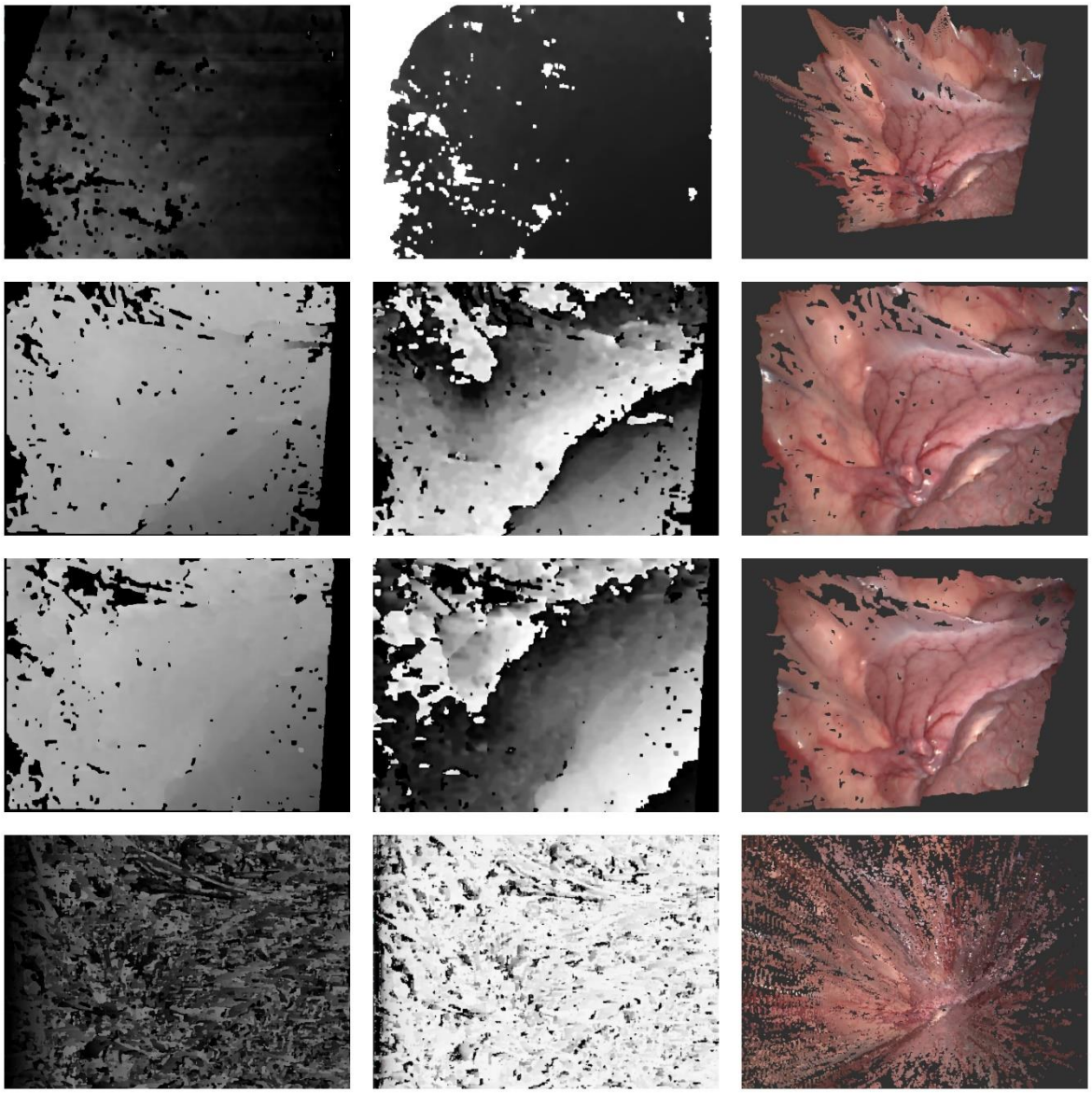

Figure 16: 3D reconstruction results for Porcine Uterine Horn Exploration Dataset (I.3) in rows: 1) ELAS, 2) Quasi Dense CPU, 3) Quasi Dense GPU, and 4) SGM GPU

Table 3: Execution times for Porcine Uterine

Method	Execution Time
ELAS	592 ms
Quasi Dense CPU	2465 ms
Quasi Dense GPU	82 ms
SGM GPU	0.4 ms

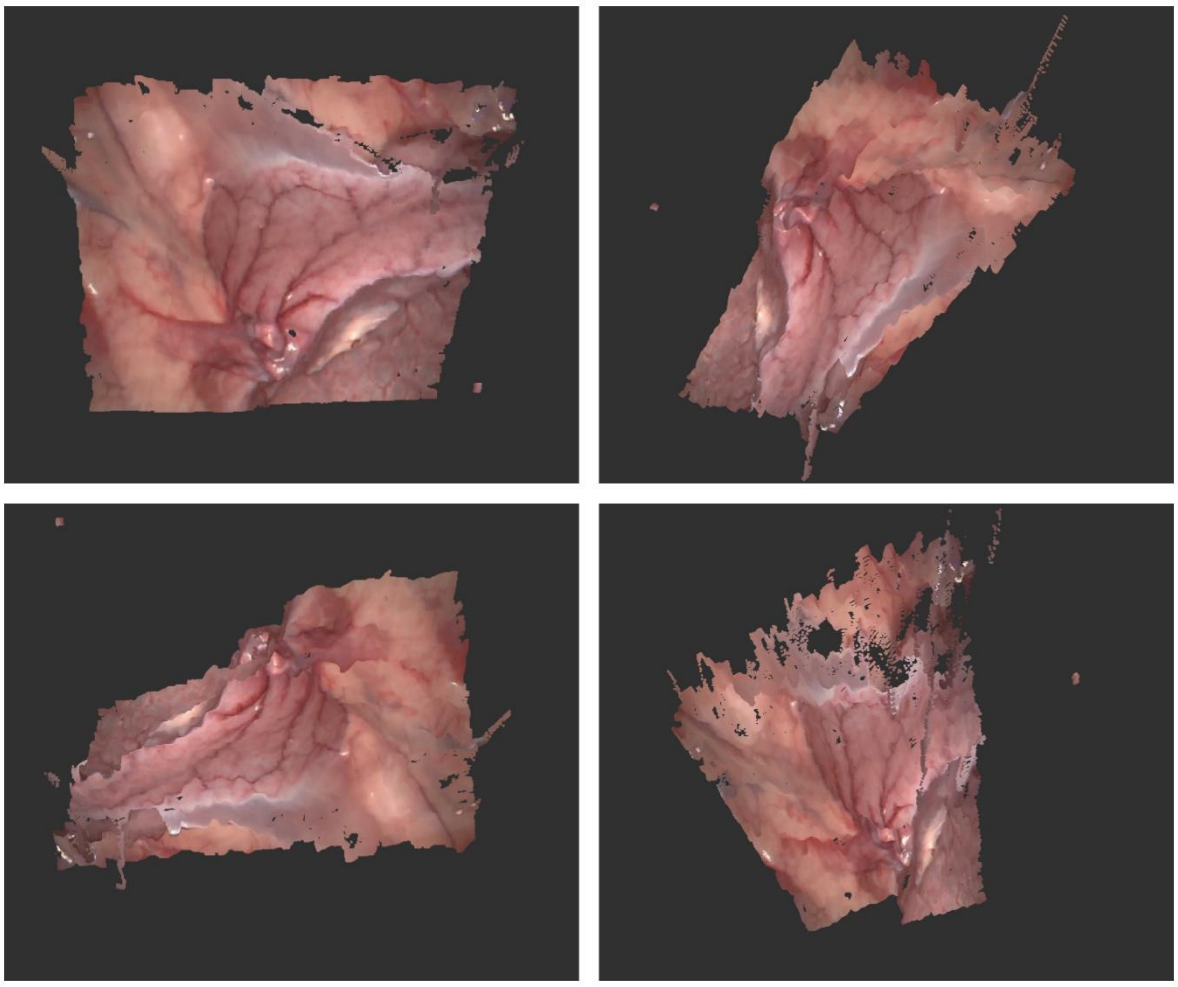

Figure 17: Quasi Dense GPU 3D reconstruction result of Porcine Uterine Horn exploration from different viewpoints

5 Future Work

Up until now, stereoscopic methods have yielded encouraging results. However, further improvement can be achieved by tuning and modifying stereoscopic methods presented in this document. More specifically, SGM-GPU method offers the biggest upside given its very fast execution time. The introduction of a set of robustly matched feature points, as seen in local correspondence methods, could introduce a useful constraint for SGM-GPU, to eventually improve its performance.

Finally, since stereoscopic methods fail in reconstruction of untextured areas due to the lack of good features, 3D reconstruction problem can be tackled with a set of methods, outside the scope of stereoscopy, namely photometric stereo. They try to model the light reflected from the reconstruction target surface, by estimating a dense set of surface normals. However, unlike stereoscopy which requires no additional equipment, photometric stereo setups require additional light sources and/or special filters to be included in the setup.

Therefore, we propose the construction of a custom photometric endoscope, as proposed in [18] by modifying an RGB monocular endoscope. A set of colored light filters will be placed in the tip of the endoscope mounted on a custom 3D printed plastic cover for stability. A concept design of the device is shown in the figures below.

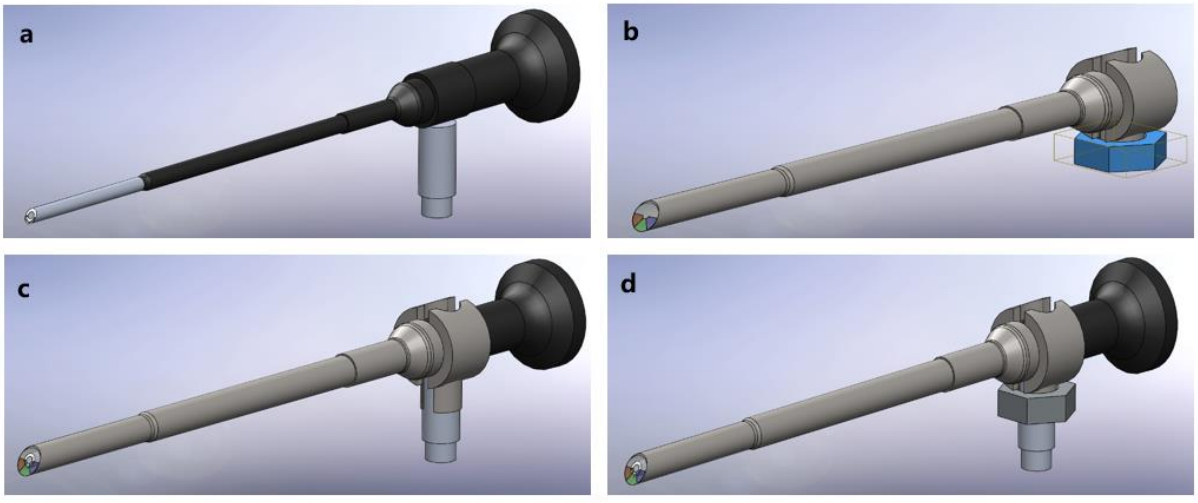

Figure 18: Concept design of RGB monocular photometric endoscope: a) Initial device, b) Custom modification, c) Final device (unlocked), d) Final device (locked)

6 Conclusion

This report presented the SMARTsurg 3D reconstruction module that is currently being implemented as part of T3.1 "*On-the-fly 3D reconstruction of the surgical field"*. It provides a comprehensive insight on the most important components, required to establish a robust and efficient 3D reconstruction module from stereo endoscopic feed. These components include

- A ROS integrated stereo processing framework with custom pre and post processing pipelines, able to incorporate any 3D reconstruction method with minimal effort.
- The 3D reconstruction methods, which are currently being investigated to produce accurate and fast 3D reconstruction from stereo images.
- A quantitative and qualitative evaluation component in order to assess the quality of the 3D reconstruction result.

Since the task is still ongoing, all the components are currently under development, testing, and evaluation for use in the final system. SMARTsurg deliverable D3.1 (M28) will go further in detail on the final version of the 3D reconstruction components and their methodology.

Annex I: Datasets

I.1 Deforming Silicon Heart Phantom Dataset[13]

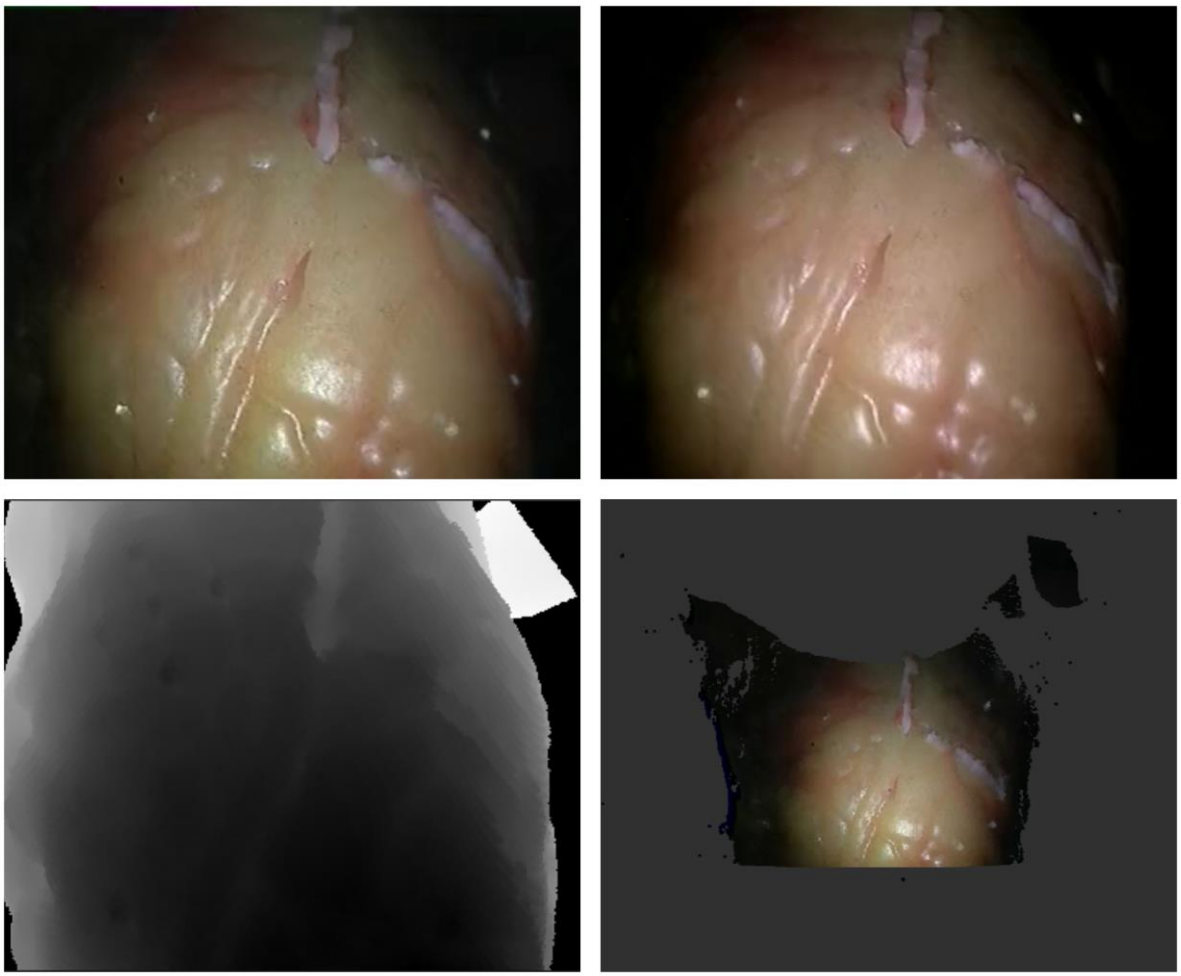

Figure 19: Deforming Silicon Heart Dataset, left and right camera images (top), ground truth disparity map (bottom-left) and corresponding point cloud (bottom-right)

I.1.1 Description

This dataset contains a video sequence of a silicon heart phantom, deforming with cardiac motion and its associated CT scans. The lighting and the resolution are quite low, while the depicted surface has smooth texture, providing a challenging dataset for 3D reconstruction with stereoscopy.

[13] http://hamlyn.doc.ic.ac.uk/vision/

I.1.2 Specifications

Table 4: Deforming Silicon Heart dataset specifications

In-vivo/ Ex-vivo	Ex-vivo Phantom
Static/ Deforming	Deforming
Area or Organ	Heart
Mono/ Stereo	Stereo
Image/ Video	Video
Duration	1.5 minute
Frame Rate	25 fps
Resolution	360x288 pixels
Ground Truth Availability	CT Scans
3D Reconstruction Challenges	Smooth texture, Low lighting

I.2 EndoAbs Dataset[14]

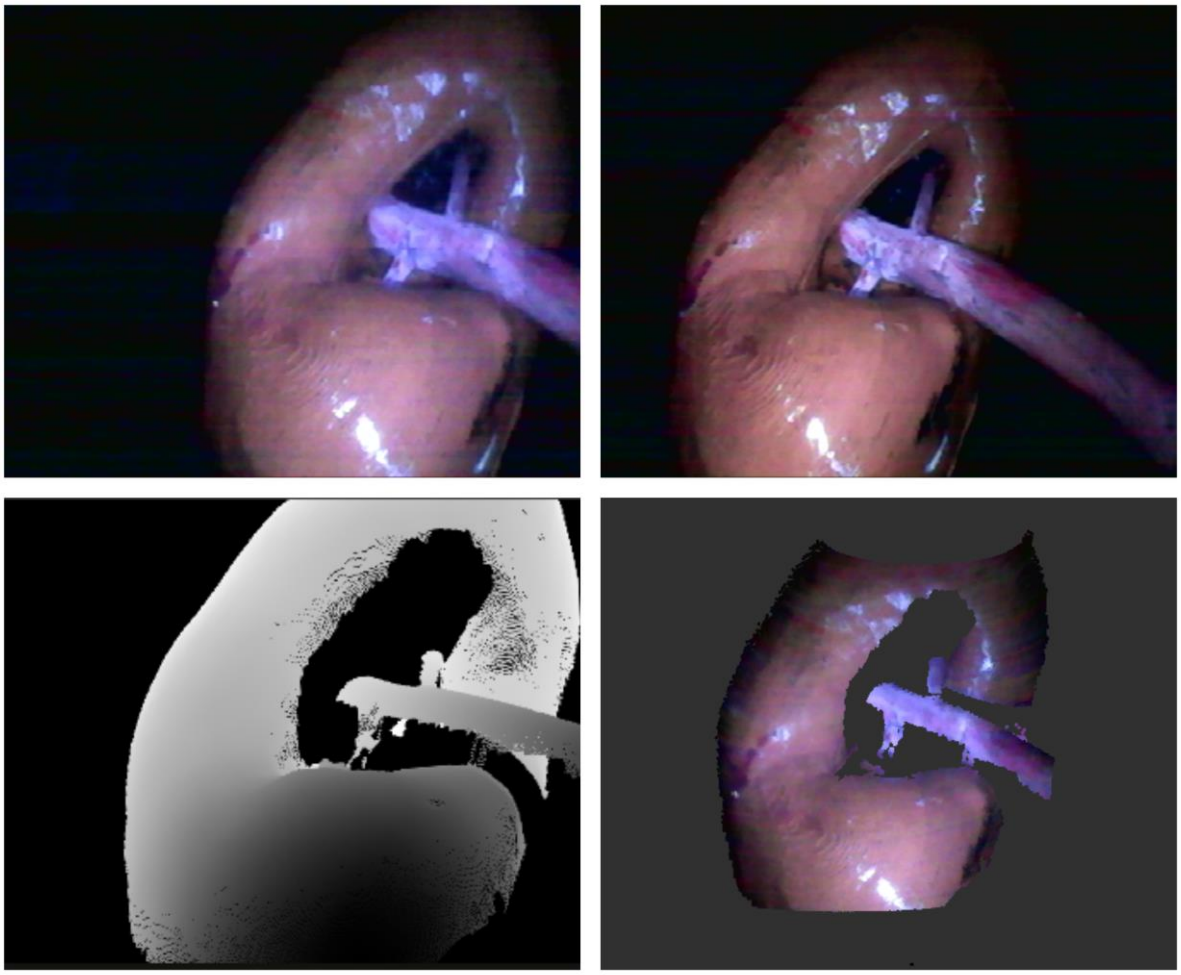

Figure 20: EndoAbs (kidney) Dataset, left and right camera images (top), ground truth disparity map (bottom-left) and corresponding point cloud (bottom-right)

I.2.1 Description

The dataset contains sets of images of organ phantoms (kidney, liver, spleen) in various poses and under different challenging conditions (rest, blood, smoke). The utilization of accompanying ground truth data provides accurate quantitative comparison to the 3D reconstructed results.

[14] http://nearlab.polimi.it/medical/dataset/

I.2.2 Specifications

Table 5: EndoAbs dataset specifications

In-vivo/ Ex-vivo	Ex-vivo Phantom
Static/ Deforming	Static
Area or Organ	Kidney
Mono/ Stereo	Stereo
Image/ Video	Images
Number of Image Pairs	24
Resolution	640x480 pixels
Ground Truth Availability	Laser Scans
3D Reconstruction Challenges	Smooth and reflective texture, Low lighting, smoke, blood

I.3 Porcine Uterine Horn Exploration Dataset[15]

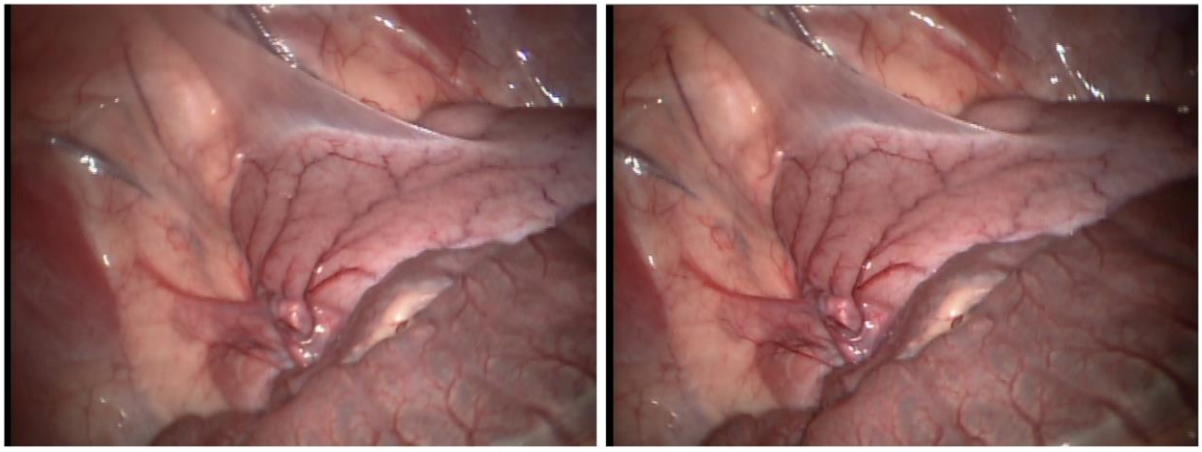

Figure 21: Porcine Uterine Horn Dataset, left and right camera views

I.3.1 Description

This video contains a video sequence of an in vivo porcine procedure, navigating to the Uterine Horn. The stereo laparoscope is rotated around the optical axis causing a change in orientation in the image. The dataset is suitable for suitable for stereo matching algorithms, since frames of the sequence show enough strong features, albeit specular reflections are present, while

[15] http://hamlyn.doc.ic.ac.uk/vision/

VGA resolution at 25 frames per second offers a solid benchmark for real-time 3D reconstruction methods.

I.3.2 Specifications

Table 6: Porcine Uterine Horn Exploration dataset specifications

In-vivo/ Ex-vivo	In-vivo
Static/ Deforming	Deforming
Area or Organ	Uterine Horn
Mono/ Stereo	Stereo
Image/ Video	Video
Duration	25 seconds
Frame Rate	25 fps
Resolution	640x480 pixels
Ground Truth Availability	N/A
3D Reconstruction Challenges	Deformation, Reflections