Towards Motion Correction in Robotized Medical Procedures Using Real-Time 3D Body Measurement

Matej ČAMPELJ *, Matija JEZERŠEK
Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia

https://doi.org/10.15221/25.10

Abstract

Robotized medical procedures rely on high spatial accuracy and repeatability, yet patient motion during treatment remains a critical challenge that can compromise safety and effectiveness. In particular, procedures such as laser skin rejuvenation require continuous adaptation of the robot trajectory to unintentional movements of the patient. To address this, we propose a real-time motion correction framework that combines dense optical flow with 3D point cloud data acquired by a time-of-flight camera. The approach, termed Fusion Flow, transforms pixel-level motion fields into real-world displacements in all three spatial directions and integrates them into the global trajectory of the treatment area.
We evaluated seven optical flow algorithms (DeepFlow, DenseRLOF, DualTVL1, Farneback, PCAFlow, SimpleFlow, and SparseToDense) using a controlled experimental setup with sub-millimeter reference motion. Among them, the Farneback method achieved the best balance of speed and accuracy, operating at 140 FPS with an average trajectory error of only 8 mm over a 2250 mm path.
These results demonstrate that real-time fusion of optical flow and 3D depth data provides clinically relevant precision for motion correction, paving the way for safer and more effective robotized medical procedures.

Keywords: 3D body scanning, optical flow, point cloud, TOF camera

1. Introduction

Robotized medical treatments and surgeries are becoming increasingly popular because they enable faster patient recovery, shorter hospital stays, reduced postoperative pain, fewer wound complications, improved cosmetic outcomes, reduced blood loss, and—with the replication of hand movements—greater precision [1, 2].

During robotized medical treatments, patient movement causes two main problems. The first is breathing, which produces vertical body motion and shifts the z-coordinate of the laser focus out of position (for example, instead of focusing on the skin surface, the laser may shift slightly beneath the skin). The second is transverse motion in the x- and y-directions, which may cause the robot to target the wrong area or even miss the treatment region entirely.

Among other efforts, patient movement has been addressed during MRI brain imaging [3], OCT skin imaging [4], and thermographic imaging during brain surgery [5]. A new method using a two-dimensional B-spline registration network was compared with several methods, such as optical flow, to detect brain movement during craniotomy [6]. An artificial intelligence model was developed for universal motion correction in medical imaging [7]. Optical surface imaging (OSI) was used for real-time respiratory motion tracking, which was modeled as a product of six sine curves [8, 9, 10]. The Lucas-Kanade optical flow algorithm was improved to detect object motion more accurately and to be sufficiently lightweight for an embedded system such as the Raspberry Pi 4 (8 GB RAM) [11]. Optical flow algorithms were also used to separate blood vessels from the surrounding skin [12]. A subpixel algorithm was developed to correct errors in photoacoustic dermoscopy caused by patient movement by fivefold expansion of the cross-correlation matrix [13]. A neural network has also been developed to detect the same local features on two images, enabling the identification of movements within an image. It achieves state-of-the-art on the task of pose estimation and is capable of operating quickly on modern GPUs at speeds of 10 to 15 FPS [14].

To our knowledge, no research has addressed detecting movement across a skin area of approximately 400 × 100 mm (about the size of the human torso) using optical flow, where the camera captures only a portion of the region.

* matej.campelj@fs.uni-lj.si; +386-1-4771-180; www.fs.uni-lj.si

Based on the preceding discussion, it is evident that optical flow algorithms are increasingly used in medical applications. Optical flow describes the apparent motion of objects in the image plane and assumes that pixel intensity remains constant during motion. This is expressed by the optical flow constraint equation 1, where I denotes the image intensity at position (u, v) and time t; u and v are the horizontal and vertical components of the pixel displacement vector $\boldsymbol{d} = (du, dv)$. The partial derivatives represent the spatial and temporal intensity gradients derived from consecutive frames.

$$\frac{\partial I}{\partial t} + du\frac{\partial I}{\partial u} + dv\frac{\partial I}{\partial v} = 0 \quad (1)$$

All optical flow algorithms considered here take two consecutive images as input and compute a displacement vector field. Optical flow methods can generally be divided into two categories. Sparse optical flow algorithms, such as Lucas-Kanade [15], estimate motion at selected feature points (e.g., corners or edges). These methods are computationally efficient but do not provide full-field coverage. By contrast, dense optical flow algorithms estimate motion for every pixel, enabling detailed motion representation at the cost of slower computation. Examples include Farneback, DualTVL1, and DeepFlow. Additionally, hybrid approaches interpolate sparse flow into dense fields (e.g., PCAFlow, SparseToDense) [13].

In this work, we present a novel framework for real-time motion correction in robotized medical procedures by fusing dense optical flow with 3D point cloud data from a time-of-flight camera. We developed a fusion algorithm (“Fusion Flow”) that converts pixel-level optical flow into real-world displacements, enabling motion estimation in all three spatial directions. Furthermore, we conducted a systematic evaluation of seven state-of-the-art optical flow algorithms under realistic motion conditions and identified the Farneback method as the most suitable for this application, achieving high processing speed (140 FPS) with sub-millimeter precision and minimal trajectory error. These results demonstrate that real-time fusion of optical flow and 3D depth data can achieve clinically relevant accuracy for motion correction, thereby enabling safer and more precise execution of robotized laser skin treatments and other medical procedures.

2. Methodology

Fig. 1 shows the experimental setup, in which a linear actuator positioned a time-of-flight (TOF) camera (pmdflexx2, PMD Technologies, Germany) approximately 140 mm above the measured body. The camera, which records grayscale images and point cloud data at a resolution of 224 × 172 pixels and a maximum frame rate of 60 FPS, has a viewing angle of 56 × 44° and a measurement accuracy better than 1% within a depth range of 0.1–7 m [16]. The linear actuator (SMC, Japan, LEFS32A) was employed in place of a robotic arm to reduce movement uncertainty, as its positioning accuracy is ±0.02 mm [17].

Data processing consists of four steps. First, the grayscale image is optionally preprocessed to reduce noise, correct non-uniform illumination, and minimize distortions. Second, optical flow is applied to consecutive grayscale images to calculate the displacement field. Third, this displacement field is transformed from image space to 3D space by fusing optical flow with 3D data. Finally, the resulting displacements are averaged in the x-, y-, and z-directions and time-integrated into a global trajectory, which is compared with the reference position measured by the actuator’s encoder. Each step is described in more detail in the following paragraphs.

2.1 Image preprocessing

Before applying optical flow, several preprocessing methods were tested to improve measurement precision: (i) Gaussian blurring, (ii) illumination correction, (iii) high-pass filtering using a Sobel kernel, and (iv) image undistortion using camera parameters. Illumination correction was performed to remove non-uniformly reflected laser light by first blurring the original image with a large Gaussian filter (kernel size equal to 20% of the image width) and then subtracting the result from the original image.

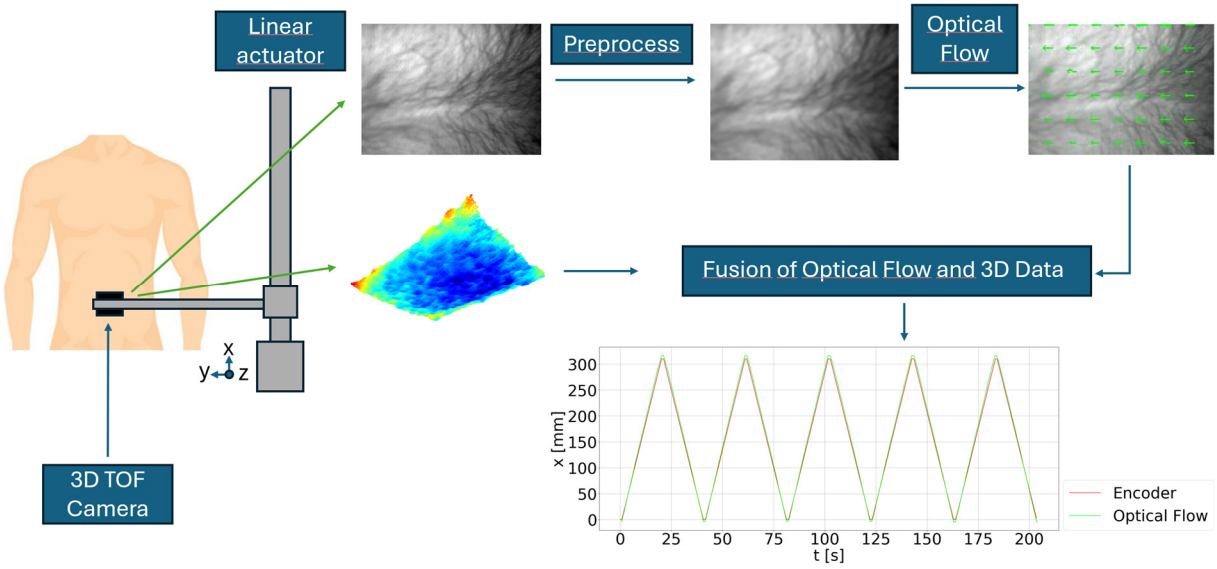

Fig. 1: Experimental setup and data processing workflow of the Fusion Flow algorithm.

2.2 Optical flow

The preprocessed images were then input to optical flow algorithms to estimate the displacement field. The following seven optical flow algorithms, included in the OpenCV library [18], were tested:

- DeepFlow [19]: variational framework suitable for large displacements and dynamic scenes.
- DenseRLOF [20]: robust regression approach designed to reduce noise and outliers.
- DualTVL1 [21]: total variation (TV) regularization with the L1 norm, robust to noise and lighting variations.
- Farneback [22]: polynomial expansion method for dense optical flow estimation.
- PCAFlow [23]: sparse-to-dense interpolation using principal component analysis.
- SimpleFlow [24]: probabilistic model with multiscale bilateral filtering.
- SparseToDense [25]: pyramidal Lucas–Kanade method interpolating sparse features to dense flow.

All algorithms produced a matrix of displacement vectors $\boldsymbol{d}$ for each pixel, where the displacement is measured in pixels.

2.3 Fusion of Optical Flow and 3D data

Since optical flow estimates displacement in the image plane, the next step is to convert pixel-based optical flow into real-world displacements of the measured body to enable future adaptation of robotic movement.

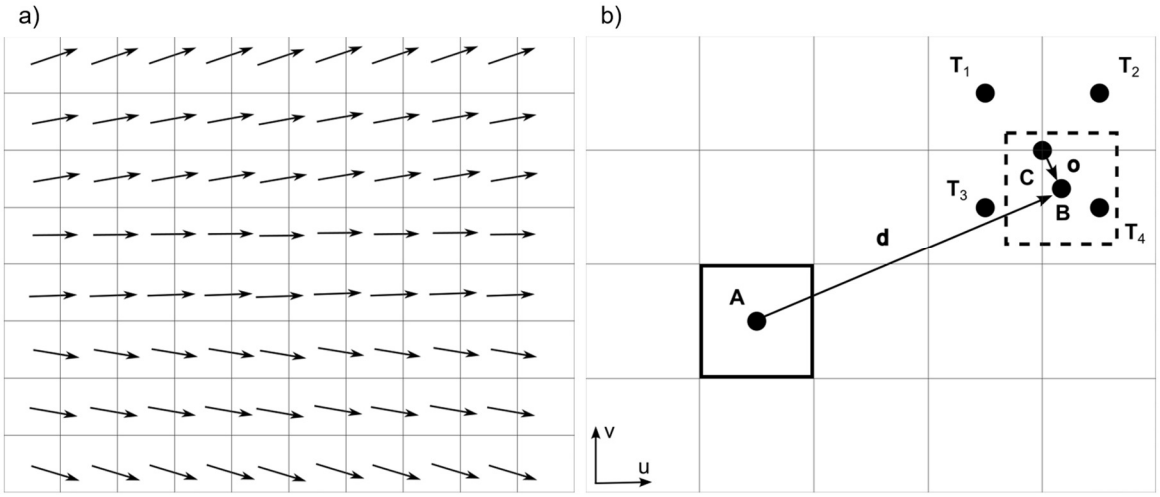

Fig. 2: Optical flow displacement field (a) and displacement of point $\boldsymbol{A}$ to point $\boldsymbol{B}$ with neighboring points ($\boldsymbol{T_1}$, …, $\boldsymbol{T_4}$) used for interpolation in fusion of optical flow and 3D data (b).

The fusion procedure is schematically illustrated in Fig. 2. Fig. 2a shows the vector field of displacements $\boldsymbol{d}(u, v)$, while Fig. 2b presents a magnified area around the original point $\mathbf{A}$, which is displaced to point $\boldsymbol{B}$ by the displacement vector $\boldsymbol{d} = (du, dv)$ in image space (i.e. pixels) with subpixel resolution. Neighboring points ($\boldsymbol{T_1}$, ..., $\boldsymbol{T_4}$), together with their center $\boldsymbol{C}$ ($\boldsymbol{C} = \sum_{i=1}^{4} \boldsymbol{T}_i$) are used for interpolation. Since all points have known coordinates in both image and 3D space, the new position of point $\boldsymbol{B}$ in 3D space can be calculated as:

$$\boldsymbol{B} = \boldsymbol{A} + \Delta\boldsymbol{r} \tag{2}$$

where $\boldsymbol{\Delta r}$ is displacement vector in 3D space, given by:

$$\Delta\boldsymbol{r} = \sum_{i=1}^{4} w_i \cdot (\boldsymbol{T}_i - \boldsymbol{A}) \tag{3}$$

where w_i is the weight of the i-th neighboring point, calculated using:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{bmatrix} -0.5 & 0.5 & -1 \\ 0.5 & 0.5 & 1 \\ -0.5 & -0.5 & 1 \\ 0.5 & -0.5 & -1 \end{bmatrix} \cdot \begin{pmatrix} u_o \\ v_o \\ u_o \cdot v_o \end{pmatrix} + \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \tag{4}$$

where u_o and v_o are the components of the offset vector $\boldsymbol{o} = \boldsymbol{B} - \boldsymbol{C}$ in image space.

2.4 Postprocessing

After the fusion of displacement into 3D space, the camera movement relative to the measured body is calculated in the x-, y-, and z-directions using one of the following statistical parameters of the displacement field $\boldsymbol{\Delta r}(u, v)$: (i) average value, (ii) median value. Using these values, the trajectory is obtained by integrating $\boldsymbol{\Delta r}(u, v)$ over time.

3. Measurement protocol

The camera was moved over the human body from the stomach to the shoulder girdle (clavicle) so that no surrounding borders (e.g., clothing, shoulders, or neck) were included in the frame. The camera was operated in Mode 5, in which each depth frame is computed from five sequential raw frames at a frame rate of 30 FPS [26]. This mode is optimized for short-distance measurements, providing reliable depth data while reducing computational load compared to higher-subframe modes. Room lighting did not affect the recording, as the sensor is equipped with a narrowband IR filter that transmits only the laser light of the TOF camera.

The subject was instructed to remain as still as possible and breathe shallowly. Each recording consisted of moving the camera to a position 225 mm away and back, repeated five times. For statistical analysis, each recording sequence was performed five times. The actuator speed was varied from 20 mm/s to 200 mm/s in increments of 20 mm/s. After each recording, the algorithm described above was used to generate displacement vector fields and estimate camera motion in 3D space, which was then compared with the reference position obtained from the actuator's encoder.

The performance of the algorithms was evaluated in terms of processing speed, precision, and accuracy. Processing speed was assessed using the average FPS, calculated as:

$$FPS = \frac{1}{\overline{\Delta t}} \tag{5}$$

where $\overline{\Delta t}$ is the average time required for the optical flow algorithm to compute the flow between two consecutive frames.

Precision was evaluated using the average standard deviation of the displacement vectors, calculated as:

$$\boldsymbol{\sigma} = \frac{1}{N_{Frames}} \cdot \sum_{i=1}^{N_{Frames}} \sqrt{\frac{1}{N_{Pixels}} \cdot \sum_{j=1}^{N_{Pixels}} \boldsymbol{d}_{i,j} - \overline{\boldsymbol{d}}_i} \tag{6}$$

where N_{Frames} is the number of frames in the recording, N_{Pixels} is the number of pixels in each frame, $\mathbf{d}_{i,j}$ is the displacement vector in the j-th pixel of the i-th frame and $\overline{\boldsymbol{d}}_i$ is the average displacement vector in the i-th frame. For rigid body translation, the displacement field should be uniform, and the average standard deviation should equal zero.

The accuracy was evaluated by comparing measured movement with the reference data from encoder of linear actuator determined by root mean square error (RMSE) method using equation:

$$RMSE = \sqrt{\frac{1}{N_{Frames}} \sum_{i=1}^{N_{Frames}} \left(x_{i.measured} - x_{i.reference}\right)^2} \quad (7)$$

where $x_{i.measured}$ and $x_{i.reference}$ are the measured and reference positions of the camera respectively in the x-direction at i-th frame.

We calculated two statistical parameters: (i) average and (ii) standard deviation for all three evaluation criteria and showed them in the following results.

4. Results

To identify the most suitable optical flow algorithm, we evaluated all methods in terms of processing speed, precision, and accuracy, using recordings acquired at the slowest actuator speed. Importantly, the algorithm with the highest precision or accuracy was not necessarily optimal, as real-time applicability also requires adequate computational speed. In this comparison, no preprocessing was applied, and postprocessing used the average displacement value.

As shown in Fig. 3, the Farneback algorithm achieved the highest processing speed. For real-time integration, a minimum of 30 FPS is required (indicated by the dashed red line in the figure). In addition to Farneback, only DenseRLOF and SparseToDense exceeded this threshold.

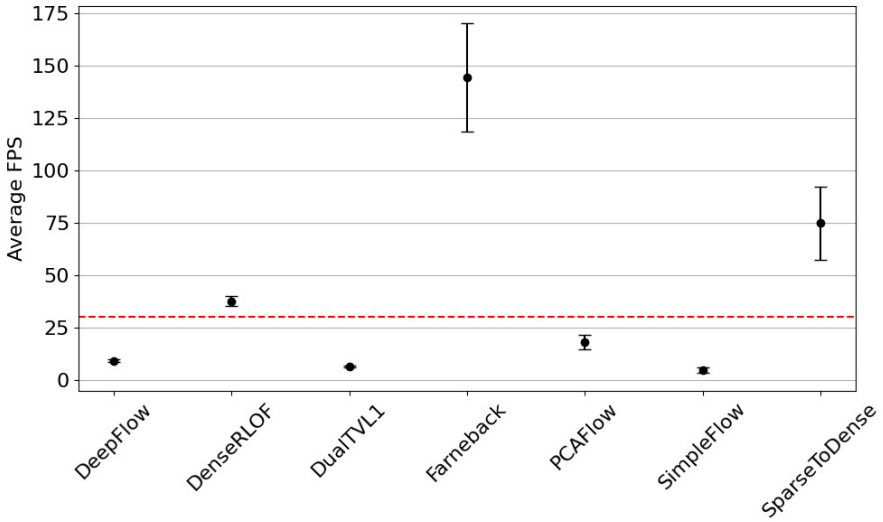

Fig. 3: Average FPS for every algorithm with red dashed line representing the 30 FPS

Fig. 4 shows the precision in terms of the average standard deviations of displacement vectors. Because the camera was moved only in the x-direction, the left graph is more relevant for characterizing algorithm performance during motion, while the right graph presents the standard deviations in the y-direction (no motion). The results indicate that the most precise algorithm was DeepFlow. However, the average standard deviation remained relatively low across all algorithms – approximately 0.28 mm in the x-direction and 0.23 mm in the y-direction – well below the camera's spatial resolution. This demonstrates that all tested algorithms produced displacement fields with sub-resolution consistency, suggesting that precision was not the limiting factor in their performance. The SimpleFlow algorithm failed to calculate the optical flow, resulting in empty columns in the graphs.

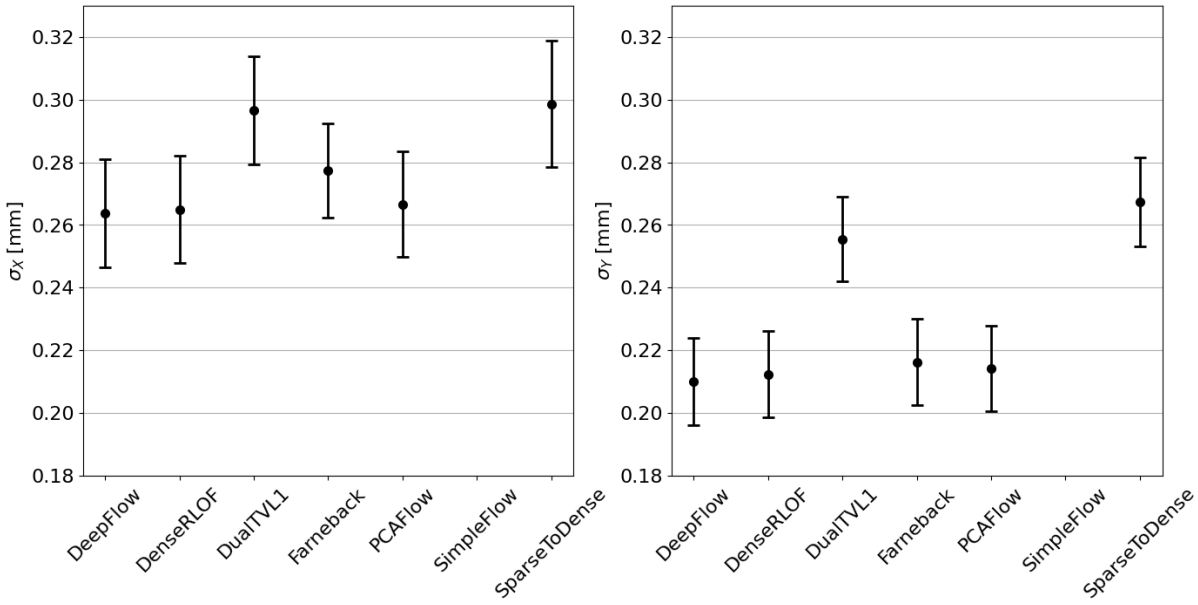

Fig. 4: Averaged standard deviations of algorithms in x- and y-direction.

Finally, Fig. 5 illustrates the measured accuracy of each algorithm in determining camera movement in the x-direction. The results show that the Farneback algorithm had the lowest RMSE.

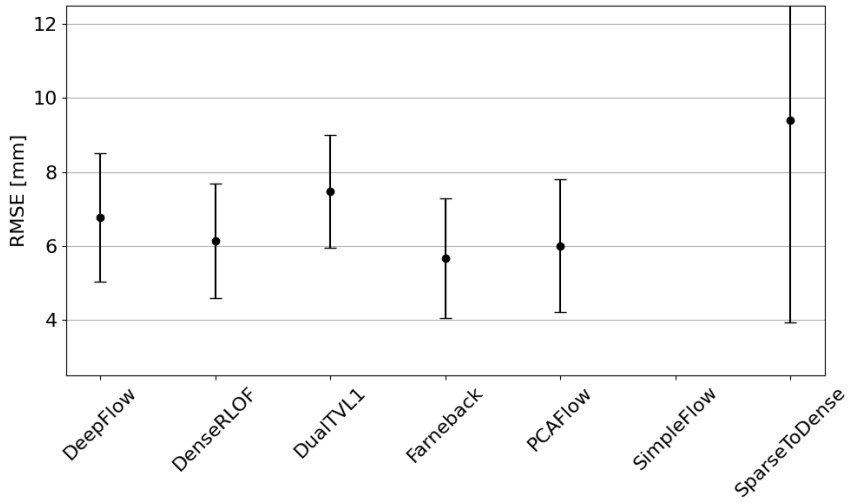

Fig. 5: Root mean square errors of algorithms.

Based on the results shown above, the Farneback algorithm was selected for the next phase of our research, as it was the fastest, moderately precise, and the most accurate.

4.1 Influence of movement speed

Fig. 6 illustrates the deviation of the detected trajectories from the reference as movement speed increases. At lower speeds (20 and 40 mm/s), the measured and reference trajectories align closely, largely overlapping. In contrast, at higher speeds, noticeable discrepancies emerge, and the detected trajectories fail to return to the initial position at x = 0 mm. The graphs present the first repetition of the recording without preprocessing and with averaging applied as postprocessing.

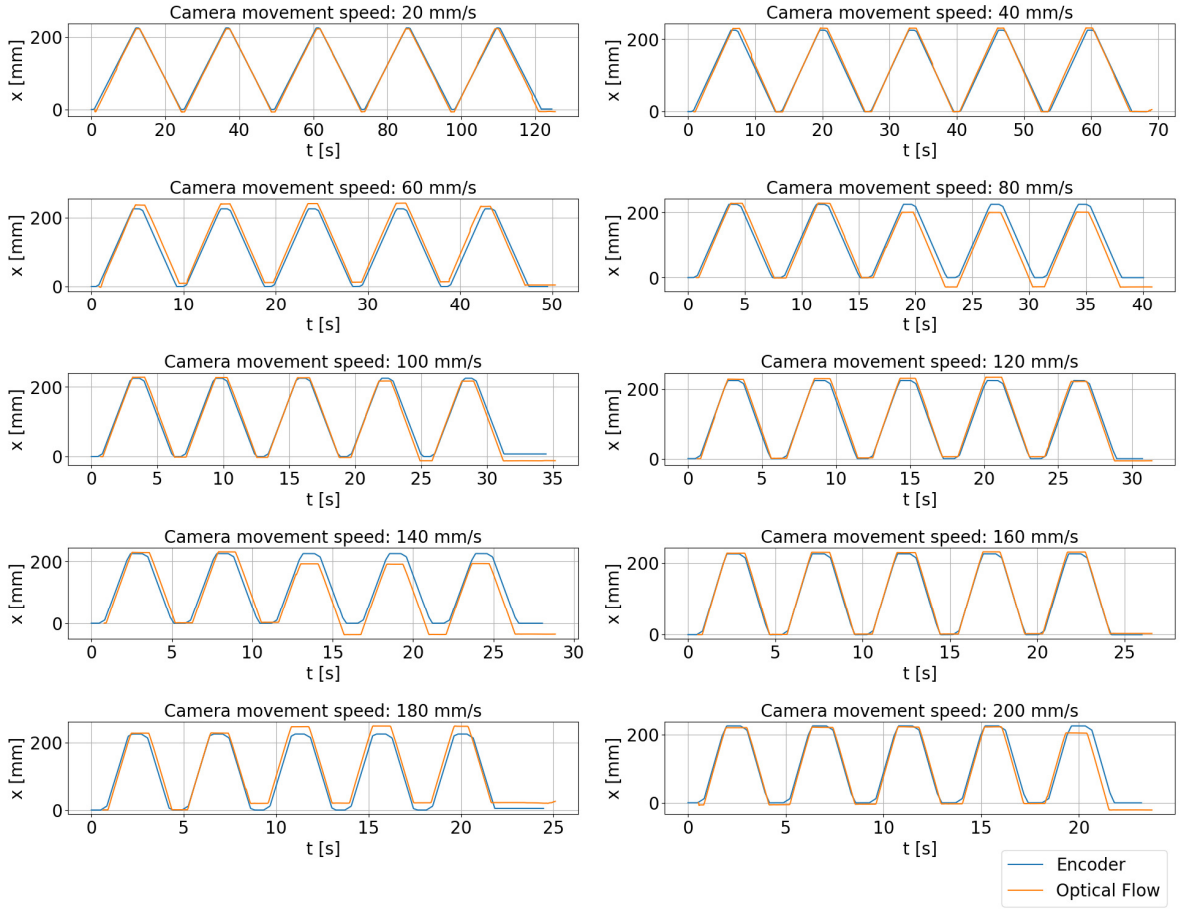

Fig. 6: Trajectories of camera in x direction detected by Farneback and compared with reference data from encoder. Movement speed is increasing from top left to bottom right.

Fig. 7 shows that precision deteriorates linearly with increasing camera movement speed. This effect is particularly evident in the x-direction, the direction of camera motion, where motion blur reduces precision. The graphs indicate that applying a Sobel kernel to the image before inputting it into the Farneback algorithm was the most effective approach, as it produced the smallest increase in the average standard deviation. In contrast, the least effective preprocessing method was image undistortion using the camera's calibration coefficients, which resulted in the steepest slope of average standard deviations in both the x- and y-directions.

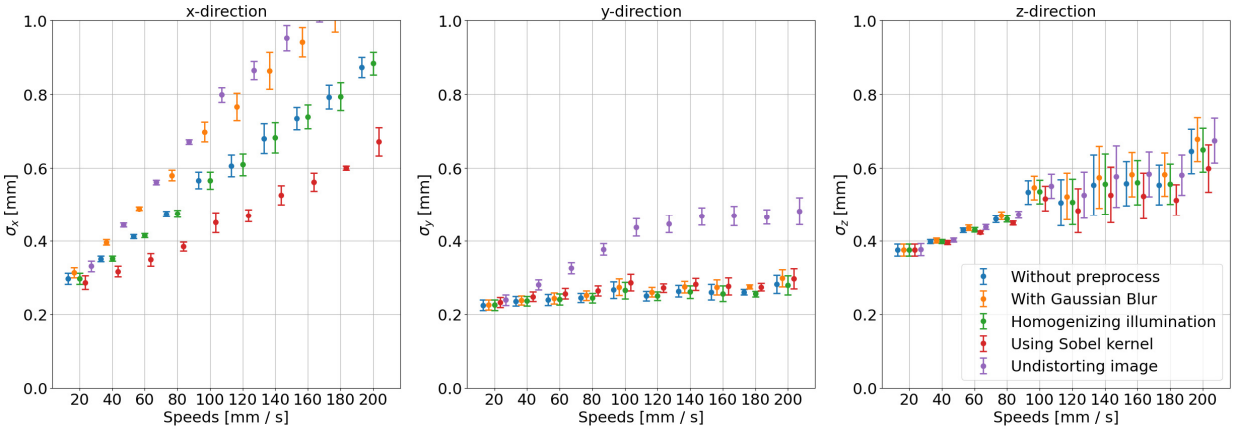

Fig. 7: Averaged standard deviations for different preprocessing methods in relation to speed for all 3 directions.

Fig. 8 shows that the RMSE, which indicates accuracy, increased with increasing camera movement speed. The deviation likewise expanded with higher movement speeds. The results show that all preprocessing methods performed worse at speeds above 60 mm/s. At these higher speeds, the displacement between consecutive frames was larger, resulting in reduced overlap. This poses a problem because optical flow algorithms are designed to detect only small displacements.

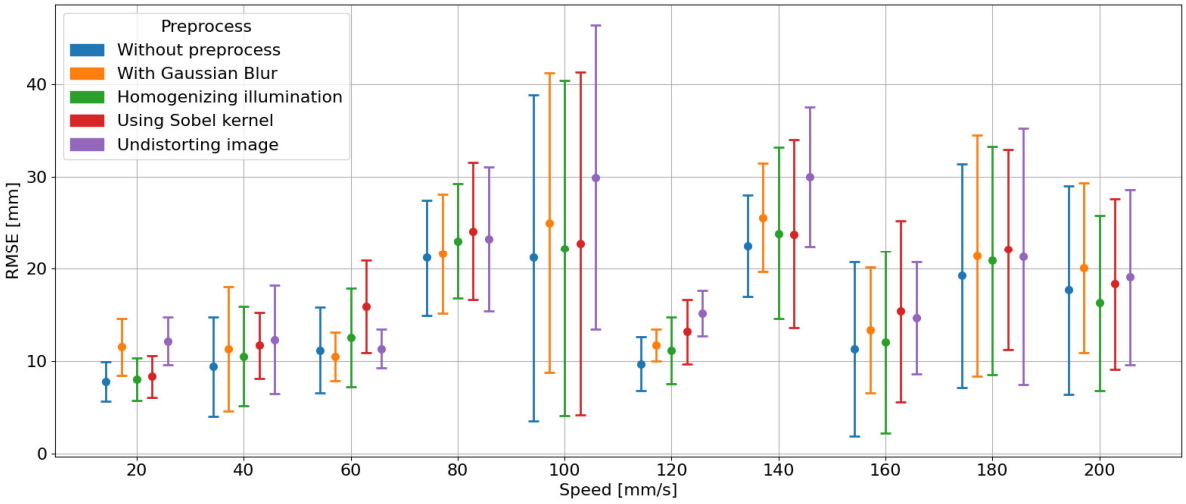

Fig. 8: RMSE for different preprocessing methods in relation to speed for all 3 directions.

Lastly, we evaluated the RMSE values for different postprocessing methods. Fig. 9 shows that there was no significant difference between the two approaches (averaging and median).

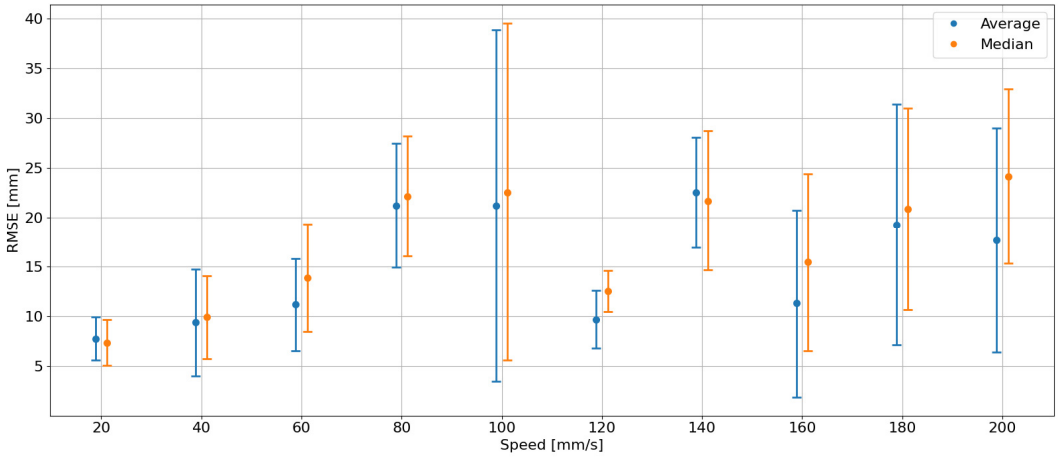

Fig. 9: RMSE for different postprocessing methods in relation to speed for all 3 directions.

To interpret the results shown in Fig. 8 and Fig. 9, we refer to Fig. 6, where it is evident that the integration of $\boldsymbol{\Delta r}$ leads to error accumulation over time. In several recordings (camera movement speeds of 60, 80, 140, 180, and 200 mm/s), sudden deviations occurred, propagating through the remainder of the measurement. If these isolated errors were excluded, the additional drift caused by limited precision, as indicated in Fig. 7, would be substantially smaller. Such errors could be mitigated by applying postprocessing methods that enforce motion smoothness, such as Kalman filtering, which combines a predictive motion model with noisy sensor measurements to generate a smoothed trajectory [26, 27]. Alternatively, they could be addressed using a visual Simultaneous Localization and Mapping (SLAM) algorithm [29], which constructs a global map of the observed scene and corrects local tracking errors by realigning trajectories to the global reference. A hybrid approach that integrates Kalman filtering for local smoothing with SLAM for global consistency may therefore represent a robust strategy for motion correction in robotized medical procedures.

5. Conclusions

We developed a real-time method for detecting 3D body motion by combining grayscale images and 3D point cloud data from a time-of-flight (TOF) camera. Seven optical flow algorithms were evaluated, and the Farneback method was identified as the most suitable, achieving the highest processing speed (140 FPS), moderate precision (standard deviation of 0.28 mm), and the lowest trajectory error (RMSE = 8 mm). Building on these findings, we introduced a fusion algorithm that integrates the optical flow displacement field with depth data to obtain real-world motion estimates. We compared five preprocessing and two postprocessing strategies; applying a Sobel kernel prior to optical flow analysis provided the most robust results, while averaging and median postprocessing performed similarly. With the proposed method, we achieved an average RMSE of 8 mm over a 2250 mm motion path at 20 mm/s, demonstrating its potential for reliable motion correction in robotized medical procedures.

Acknowledgements

The work was supported by the Slovenian Research Agency (ARIS), national program (Grant Nos. P2-0392) and within the framework of the GREENTECH project, co-financed by the European Union – NextGenerationEU.

References

[1] K. Reddy, P. Gharde, H. Tayade, M. Patil, L. S. Reddy, and D. Surya, ‘Advancements in Robotic Surgery: A Comprehensive Overview of Current Utilizations and Upcoming Frontiers’, Cureus, vol. 15, no. 12, p. e50415, https://doi.org/10.7759/cureus.50415.

[2] K. Jeyarajah and B. Morrison, ‘A narrative review of the role of robotics in cancer surgery’, Dig. Med. Res., vol. 3, no. 0, Dec. 2020, https://doi.org/10.21037/dmr-20-85.

[3] L. Zhang et al., ‘Motion Correction for Brain MRI Using Deep Learning and a Novel Hybrid Loss Function’, Algorithms, vol. 17, no. 5, p. 215, May 2024, https://doi.org/10.3390/a17050215.

[4] Y. M. Liew, R. A. McLaughlin, F. M. Wood, and D. D. Sampson, ‘Motion correction of in vivo three-dimensional optical coherence tomography of human skin using a fiducial marker’, Biomed. Opt. Express, vol. 3, no. 8, p. 1774, Aug. 2012, https://doi.org/10.1364/BOE.3.001774.

[5] V. Senger et al., ‘Motion correction of thermographic images in neurosurgery: Performance comparison’, in 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings, Lausanne, Switzerland: IEEE, Oct. 2014, pp. 121–124. https://doi.org/10.1109/BioCAS.2014.6981660.

[6] M. Iorga, M. C. Tate, and T. B. Parrish, ‘A robust motion correction technique for infrared thermography during awake craniotomy’, Int. J. Comput. Assist. Radiol. Surg., vol. 18, no. 12, pp. 2223–2231, May 2023, https://doi.org/10.1007/s11548-023-02953-8.

[7] J. Wang, R. Faghihpirayesh, D. Joca, P. Golland, and A. Gholipour, ‘UniMo: Universal Motion Correction For Medical Images without Network Retraining’, Sept. 21, 2024, arXiv: arXiv:2409.14204. https://doi.org/10.48550/arXiv.2409.14204.

[8] L. Chen et al., ‘Accuracy of real-time respiratory motion tracking and time delay of gating radiotherapy based on optical surface imaging technique’, Radiat. Oncol., vol. 15, no. 1, p. 170, Dec. 2020, https://doi.org/10.1186/s13014-020-01611-6.

[9] K. Povsic, M. Flezar, J. Mozina, and M. Jezersek, ‘Laser 3-D measuring system and real-time visual feedback for teaching and correcting breathing’, J. Biomed. Opt., vol. 17, no. 3, p. 036004, Mar. 2012, https://doi.org/10.1117/1.JBO.17.3.036004.

[10] K. Povšič, M. Jezeršek, and J. Možina, ‘Real-time 3D visualization of the thoraco-abdominal surface during breathing with body movement and deformation extraction’, Physiol. Meas., vol. 36, no. 7, p. 1497, May 2015, https://doi.org/10.1088/0967-3334/36/7/1497.

[11] A. Ammar, H. B. Fredj, and C. Souani, ‘Accurate Realtime Motion Estimation Using Optical Flow on an Embedded System’, Electronics, vol. 10, no. 17, p. 2164, Sept. 2021, https://doi.org/10.3390/electronics10172164.

[12] A. Aminfar, N. Davoodzadeh, G. Aguilar, and M. Princevac, ‘Application of optical flow algorithms to laser speckle imaging’, Microvasc. Res., vol. 122, pp. 52–59, Mar. 2019, https://doi.org/10.1016/j.mvr.2018.11.001.

[13] Z. Cheng, H. Ma, Z. Wang, W. Zhang, F. Yang, and S. Yang, ‘Subpixel and On-Line Motion Correction for Photoacoustic Dermoscopy’, IEEE J. Sel. Top. Quantum Electron., vol. 27, no. 4, pp. 1–8, July 2021, https://doi.org/10.1109/JSTQE.2020.3024247.

[14] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, ‘SuperGlue: Learning Feature Matching with Graph Neural Networks’, Mar. 28, 2020, arXiv: arXiv:1911.11763. https://doi.org/10.48550/arXiv.1911.11763.

[15] B. D. Lucas and T. Kanade, 'An iterative image registration technique with an application to stereo vision', in Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, in IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1981, pp. 674–679.

[16] 'pmd flexx2 3D Camera Development Kit'. Accessed: July 28, 2025. [Online]. Available: https://3d.pmdtec.com/en/3d-cameras/flexx2/

[17] 'SMC LEFS32A-700 actuator, electric, slider, ELECTRIC ACTUATO', SMCpneumatics.com. Accessed: July 29, 2025. [Online]. Available: https://www.smcpneumatics.com/LEFS32A-700.html

[18] M. Kuklin (Xperience.AI), 'Optical Flow in OpenCV (C++/Python) | LearnOpenCV #'. Accessed: July 29, 2025. [Online]. Available: https://learnopencv.com/optical-flow-in-opencv/

[19] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, 'DeepFlow: Large Displacement Optical Flow with Deep Matching', in 2013 IEEE International Conference on Computer Vision, Sydney, Australia: IEEE, Dec. 2013, pp. 1385–1392. https://doi.org/10.1109/ICCV.2013.175.

[20] T. Senst, V. Eiselein, and T. Sikora, 'Robust Local Optical Flow for Feature Tracking', IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 9, pp. 1377–1387, Sept. 2012, https://doi.org/10.1109/TCSVT.2012.2202070.

[21] C. Zach, T. Pock, and H. Bischof, 'A Duality Based Approach for Realtime TV-L1 Optical Flow', in Pattern Recognition, F. A. Hamprecht, C. Schnörr, and B. Jähne, Eds, Berlin, Heidelberg: Springer, 2007, pp. 214–223. https://doi.org/10.1007/978-3-540-74936-3_22.

[22] G. Farnebäck, 'Two-Frame Motion Estimation Based on Polynomial Expansion', in Image Analysis, J. Bigun and T. Gustavsson, Eds, Berlin, Heidelberg: Springer, 2003, pp. 363–370. https://doi.org/10.1007/3-540-45103-X_50.

[23] J. Wulff and M. J. Black, 'Efficient sparse-to-dense optical flow estimation using a learned basis and layers', in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA: IEEE, June 2015, pp. 120–130. https://doi.org/10.1109/CVPR.2015.7298607.

[24] M. Tao, J. Bai, P. Kohli, and S. Paris, 'SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm', Comput. Graph. Forum, vol. 31, no. 2pt1, pp. 345–353, May 2012, https://doi.org/10.1111/j.1467-8659.2012.03013.x.

[25] J.-Y. Bouguet, 'Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm'.

[26] 'pmd_flexx2_VGA-Getting_Started_102023.pdf'. Accessed: Aug. 21, 2025. [Online]. Available: https://media.automation24.com/manual/nl/pmd_flexx2_VGA-Getting_Started_102023.pdf?utm_source=chatgpt.com

[27] A. Arjas et al., 'Neural Network Kalman Filtering for 3-D Object Tracking From Linear Array Ultrasound Data', IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 69, no. 5, pp. 1691–1702, May 2022, https://doi.org/10.1109/TUFFC.2022.3162097.

[28] S. Liu, 'Object Trajectory Estimation Using Optical Flow'.

[29] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, 'A Comprehensive Survey of Visual SLAM Algorithms', Robotics, vol. 11, no. 1, p. 24, Feb. 2022, https://doi.org/10.3390/robotics11010024.