COMPARATIVE STUDY OF FEATURE LOCALIZATION METHODS FOR ENDOSCOPY IMAGE MATCHING

Ana Urdapilleta and Antonio Agudo

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 08028, Barcelona, Spain

ABSTRACT

The purpose of this work is to determine which is the best general method of feature localization for image matching in endoscopy images. To accomplish this, we conduct an exhaustive analysis of ten well-known feature detectors, descriptors, and learned algorithms, such as SIFT, FAST, SURF, ORB, BRIEF, BRISK, FREAK, HARRIS, DFM, and LoFTR. The analysis is performed across six challenging medical datasets, including cardiorespiratory endoscopy, human laparoscopy, bronchoscopy, gastroscopy, rabbit laparoscopy, and pig laparoscopy. This framework is highly diverse, containing a variety of textures, camera motions, tissue deformations, and visual barriers. To determine which technique is the best on average, we perform a qualitative analysis of the inliers and a quantitative analysis using the number of keypoints, number of matches, number of inliers, computational cost, sparsity of the inliers, recall and 1-precision. To complete the study, we considered the sequential and template modes, as they are highly used in computer vision. Furthermore, we examine how those features may be exploited in the reconstruction of 3D shapes from visual cues.

***Index Terms*—** Endoscopic Images, Feature Extraction and Matching, Data-driven Features.

1. INTRODUCTION

Endoscopies are Minimally Invasive Surgeries (MIS) used for diagnosing and treating diseases, conducting direct interventions, and performing surveillance [1] [2]. Their advantages include reducing recovery time, postoperative pain, hospital stays and possible complications [3]. Nevertheless, tunnel vision of the endoscopic images usually increases the duration and risk of the operation in order to obtain information on the scene, e.g., 3D data [4, 5] or the real motion of the endoscope to revisit a particular region [6]. Recently, computer vision techniques have begun to be used in MIS for tasks such as image classification [7], shape clustering [8], object detection, and 3D reconstruction [9], which can help to further reduce the procedures' time and possible diagnostic and technical errors caused by humans. All of the aforementioned computer vision tasks employ image matching, i.e., finding correspondences between images. Consequently, finding accurate features of the images and the right feature matching between sequential endoscopy images in real-time is essential [5, 6].

# frames	Resolution	T	# frames	Resolution	T	# frames	Resolution	T
Cardiopulmonary Endoscopy			Human Laparoscopy			Bronchoscopy		
79	360×288	9	1,589	1,280×720	449	78	1,280×1,024	1
Gastroscopy			Rabbit Laparoscopy			Pig Laparoscopy		
96	413×504	63	462	220×220	280	1,275	1,300×1,800	81

Table 1. **Dataset information.** For all sequences, the table reports the number of frames and image resolution. In addition, it is included T to denote the number of frame chosen as template for every case.

Endoscopic images are highly challenging since there are many types of endoscopies, which implies having images with diverse textures and challenging visual scenarios, such as textureless areas, occlusions, or elastic deformations. These characteristics affect matching accuracy and lead to areas without detected features and mismatchings [5]. Therefore, this work aims to study and find the most appropriate average feature localization technique (including both handcrafted and learned approaches) for non-specific medical settings.

2. METHODOLOGY

In this paper we use up to six different monocular endoscopy sequences (see Table 1), including cardiopulmonary endoscopy (heart) [10], laparoscopy (human uterus) [11], bronchoscopy, gastroscopy, laparoscopy (rabbit) [6], and laparoscopy (pig liver). Each of them contains different visual obstacles such as blood, foam, fluids and tools. Moreover, some elastic deformations due to the motion of the organs or the pressure made by the tools can be seen in the images. The data also include textureless areas, as well as specularities and changes of intensity caused by the lighting. Moreover, the camera performs large motions, by means of rotations and translations, producing large changes in the point of view.

2.1. Algorithms

To find image features there are many well-known methods, which can be classified into handcrafted and learned [12].

Traditional feature detectors and descriptors follow the

This work has been supported by the project MoHuCo PID2020-120049RB-I00 funded by MCIN/AEI/10.13039/501100011033.

next scheme: detection of interest points, description, and matching the descriptors. The interest points must be well localized in the image and should be likely to match points in other images [13]. Matching success mainly depends on the properties of the keypoints and their descriptors. To choose appropriate detector and descriptor algorithms the nature and the expected deformations of the image should be considered, since some algorithms are more robust than others to specific variations such as brightness or scale [14].

Regarding the well-known handcrafted detectors and descriptors, some of the most used in the last decades are SIFT [15], FAST [16], BRIEF [17], ORB [18], SURF [19], BRISK [20], HARRIS [21] and FREAK [22, 23]. For learned feature matching algorithms, we will focus on LoFTR [24], DFM [25] and COTR [26].

HARRIS [21] is a computer vision algorithm designed to detect corners –discrete, meaningful, and reliable feature points–. It has been proved that it performs consistently on natural images. SIFT [15] algorithm efficiently computes local features invariant rotation, and is partially invariant to 3D camera viewpoint, illumination, some affine distortions, and noise addition. Unlike HARRIS [21], SIFT [15] is also scale invariant. SIFT [15] keypoints are remarkably distinctive, which allows finding matches with high probability. Using the scale space to detect local features permits matching small and highly occluded objects for small keypoints. In addition, it permits matching blurred and noisy objects for large keypoints. Even though HARRIS [21] and SIFT [15] obtain high-quality features, they are computationally demanding for real-time applications. Therefore, FAST [16] algorithm was developed to cover the need for high-speed feature detection. SURF [19] is a rotation and scale invariant, robust, and distinctive feature detector and descriptor. Overall, SURF [19] performs similarly to SIFT [15], while being around three times faster. Indeed, it successfully handles blurred and rotated images but does not deal well with illumination and viewpoint changes. BRIEF [17], ORB [18] and BRISK [20] compute binary descriptors and they were proposed as computationally efficient alternatives to SIFT [15] and SURF [19]. FREAK [22] descriptor is based on the retina, i.e., the human visual system. The aim was to make a faster, more compact, and more robust to scale, rotation, and noise algorithm. It has been proven to be competitive for embedded applications, having a low memory load, as well as being faster and more robust than state-of-the-art descriptors such as SIFT [15], SURF [19], and BRISK [20].

COTR [26] and LoFTR [24] are deep-learning based algorithms that use transformers on attention layer in order to find correspondences. One of the main benefits of LoFTR [24] is that it produces dense matches even in indistinctive regions such as low-texture areas, motion blur, or repetitive patterns. DFM [25] is a deep-learning-based image-matching algorithm that uses features extracted by an off-the-shelf pretrained deep neural network [27] without additional training. It was motivated by the mental rotation paradigm. All three algorithms have shown state-of-the-art results.

	SIFT	FAST	BRIEF	ORB	SURF	BRISK	HARRIS	FREAK
Detector	SIFT	FAST	STAR	ORB	SURF	BRISK	HARRIS	STAR
Descriptor	SIFT	BRISK	BRIEF	ORB	SURF	BRISK	SIFT	FREAK

Table 2. **Detectors and descriptors considered in this study.** When both are not the same it is displayed in gray.

2.2. Implementation

First of all, we tune the parameters for all methods, and then use the same values for all our experiments. For the eight Python-code handcrafted algorithms, we use the open source computer vision (OpenCV) library. Since BRIEF [17] and FREAK [22] need another algorithm as a detector and FAST [16] and HARRIS [21] need another algorithm as a descriptor, we use the configuration in Table 2. The machine we use to execute the scripts has a Debian GNU/Linux 11 operating system with a processor Intel Core i5-8250U CPU at 1.60GHz×8. Furthermore, learned algorithms are also executed in Google Collab to analyze the computational time when accelerating the methods using GPU.

In our experiments, we execute the code for two different image matching scenarios: 1) *sequential mode*, where consecutive images are considered and, 2) *template mode*, where all images are matched with respect to the same image. For the template case, we select a particular reference frame T (see Table 1, third column) for each dataset taking into account visual characteristics such as the sharpness of the image, occlusions, and motion of the camera. We consider that strategy to be coherent, as it is standard in optical flow algorithms.

Regarding learned algorithms, we first compare COTR [26] and LoFTR [24] by using the Heart dataset in sequential mode. Since COTR [26] is 1,012 times more computationally expensive than LoFTR [24], while producing 10 times fewer inliers, for the rest of the datasets we just analyze LoFTR [24]. Moreover, learned algorithms apparently tend to be sensitive to input image resolution since the architecture is optimized for the training resolution. Consequently, we execute DFM [25] and LoFTR [24] for the Uterus dataset in two different scenarios: with the original video resolution and with the corresponding training one. To compare them, we crop the frames to patches, maintaining the aspect-ratio of the corresponding training resolution. Analyzing the number of inliers directly would not be adequate since the frames with higher resolution would likely have more inliers. Therefore, to analyze the performance of the methods we should consider the concentration of inliers computed by $\sigma = \frac{\text{average number of inliers}}{\text{number of pixels in image}}$. Table 3 shows that the metric is higher for the training resolution than for the original one. Therefore, the most convenient option is to execute them with the image resolution used in training. In any case, as we seek the best general algorithm, no assumption can be made since each video could be acquired by a different system,

	DFM [25]	LoFTR [24]
Video resolution	$1.01 \cdot 10^{-2}$	$1.06 \cdot 10^{-2}$
Training resolution	$3.77 \cdot 10^{-2}$	$1.20 \cdot 10^{-2}$

Table 3. **Resolution Comparison.** The table reports σ to denote the number of inliers per pixel for both input video and training resolutions.

under specific conditions, etc. and, therefore, we will use the original video resolution with no variation. Otherwise, we would have to deal with different image patches for each method and the comparison would be unfair.

2.3. Quantitative Metrics

The metrics we use for the quantitative analysis of algorithm performance are the number of keypoints, number of matches, number of inliers, the computational time in seconds, the inlier sparsity, the recall and 1-precision. The inliers are obtained by applying RANSAC [28] to the handcrafted algorithms and MAGSAC [29] to LoFTR [24]. In the case of DFM [25], the algorithm already uses hierarchical refinement, so we do not apply any further steps to filter the matches. For the inlier sparsity metric, if we divide an image into equal size areas, ideally there should be the same number of inliers in each area, at least, from a theoretical point of view. However, that observation could fail especially in endoscopy images, where the content in the image could not be well distributed. Therefore, we divide the frames into 16 areas of the same size. The metric is computed as $\beta_i = \frac{\alpha_i - \lambda}{\theta}$, where $\lambda = \frac{\theta}{16}$, α_i represents the number of inliers in the i-th area, λ indicates the ideal number of inliers for each area, and θ is the total number of inliers in the frame. The metric calculates normalized values of the average sparsity across the whole dataset. The metric's result is a value between -1 and 1 for each of the 16 areas. The ideal result would be to obtain zero for each of them. A negative value indicates that there are fewer inliers than expected in the area, while a positive one indicates a larger concentration of inliers.

3. EXPERIMENTAL RESULTS

In this section, we provide our experimental results to evaluate all the feature methods in our endoscopic videos. We also test the use of some 2D points to infer 3D information.

3.1. Feature evaluation

Table 4 shows the average number of keypoints and matches that the methods produce for each dataset. While for some methods such as ORB [18], SURF [19], and BRISK [20] the difference between the number of keypoints and the number of matches is not very significant, for other methods the amount might reduce up to a 40%.

Next, we evaluate the number of inliers together with the computational cost for both sequential and template modes in Table 5. For the sequential case, the method that obtains the most sparse results is LoFTR [24] with a maximum deviation of 0.06, followed by SURF [19] (0.17) and DFM [25] (0.19). In contrast, the least sparse ones would be HARRIS [21] and BRISK [20], reaching up to a maximum deviation of 0.47 and 0.46, respectively. Moreover, LoFTR [24] is one of the algorithms with a higher number of inliers. The difference in the number of inliers compared to the rest of the algorithms is especially noticeable for the bronchoscopy and gastroscopy sequences, two very challenging scenarios. Indeed, LoFTR [24] produces on average 3.5k and 995 inliers, respectively, while the rest of the algorithms obtain at most 102 and 58 inliers. In the pig laparoscopy, the computational time of most algorithms is extremely high for real-time applications having average values between 6.75 and 182.81 for BRIEF [17] and FAST [16], respectively. The only computationally efficient option for that dataset is ORB [18]. Nevertheless, when a GPU to accelerate learned algorithms can be used, both DFM [25] and LoFTR [24] would be good alternatives for real-time applications. Furthermore, LoFTR [24] is by far the algorithm with more inliers. The numbers of inliers are much smaller for the template mode, which is mainly caused by the camera motion –including both translations and rotations–, and the elastic deformations of the organs. Nevertheless, LoFTR [24] is still the most sparse algorithm, with more inliers and the computationally cheapest when using GPU acceleration.

Table 6 shows the recall and 1-precision metrics computed for both sequential and template modes. The algorithms with the highest recall for most datasets are BRIEF [17] and

	Cardiopulmonary		Human Laparoscopy		Bronchoscopy		Gastroscopy		Rabbit Laparoscopy		Pig Laparoscopy	
	# kp	# Matches	# kp	# Matches	# kp	# Matches	# kp	# Matches	# kp	# Matches	# kp	# Matches
SIFT [15]	969	624	3,626	2,292	970	534	458	262	380	242	18,073	10,695
FAST [16]	3,044	1,776	19,449	10,251	2,746	1,372	1,418	747	2,201	1,342	37,381	20,283
BRIEF [17]	320	240	2,848	2,036	1,656	901	315	197	111	81	10,228	6,796
ORB [18]	412	407	491	491	498	486	437	434	405	404	500	500
SURF [19]	736	726	3,870	3,868	2,708	2,633	829	819	523	522	9,368	9,363
BRISK [20]	782	773	1,657	1,657	473	461	593	592	330	329	7,751	7,744
HARRIS [21]	3,851	2,335	5,801	4,110	1,643	726	3,182	1,460	1,516	1,322	8,320	5,514
FREAK [22]	302	204	2,822	1,701	1,665	854	308	174	111	75	10,213	5,629
DFM [25]		2,033		11,095		100		302		781		3,996
LoFTR [24]		1,183		11,271		7,668		2,272		566		11,199

Table 4. **Quantitative comparison in terms of keypoints and matches.** The table reports the average number of keypoints and matches across the whole sequence.

	Cardiopulmonary			Human Laparoscopy			Bronchoscopy			Gastroscopy			Rabbit Laparoscopy			Pig Laparoscopy		
	# inliers	t[s]	t[s] GPU	# inliers	t[s]	t[s] GPU	# inliers	t[s]	t[s] GPU	# inliers	t[s]	t[s] GPU	# inliers	t[s]	t[s] GPU	# inliers	t[s]	t[s] GPU
Sequential mode																		
SIFT [15]	321	0.46		1,254	4.31		64	1.91		60	0.67		131	0.29		2,944	62.02	
FAST [16]	708	1.08		2,942	46.38		9	0.93		59	0.31		616	0.61		429	182.81	
BRIEF [17]	150	0.07		1,253	0.58		83	0.30		61	0.08		52	0.05		3,250	6.75	
ORB [18]	141	0.05		178	0.09		64	0.12		60	0.06		149	0.05		150	0.13	
SURF [19]	313	0.53		1,687	5.25		102	4.91		122	1.08		226	0.36		1,900	16.53	
BRISK [20]	299	0.13		700	0.47		71	0.18		66	0.11		117	0.06		1,730	8.76	
HARRIS [21]	105	5.45		187	11.46		36	2.45		67	5.15		8	1.75		124	19.92	
FREAK [22]	121	0.08		726	1.09		11	0.47		32	0.09		48	0.06		624	12.41	
DFM [25]	2,033	1.91	0.1	11,095	16.55	0.52	100	22.85	0.67	302	4.93	0.17	781	1.81	0.08	3,996	36.80	0.81
LoFTR [24]	1,008	2.10	0.06	9,842	28.66	0.65	3,498	27.85	0.67	1,423	7.38	0.14	494	2.14	0.06	11,199	55.04	1.00
Template mode																		
SIFT [15]	76	0.46		98	4.58		44	0.09		15	0.71		54	0.31		77	43.53	
FAST [16]	111	1.08		34	40.39		7	0.05		11	0.31		157	0.67		11	160.64	
BRIEF [17]	37	0.07		99	0.59		56	0.09		16	0.09		19	0.05		102	5.88	
ORB [18]	26	0.05		17	0.09		42	0.30		23	0.06		35	0.05		21	0.15	
SURF [19]	60	0.53		81	5.49		46	0.06		30	1.25		71	0.34		47	14.19	
BRISK [20]	67	0.13		35	0.56		58	0.10		21	0.11		45	0.06		71	5.82	
HARRIS [21]	34	5.45		40	14.55		31	0.15		23	6.42		1	1.69		2	19.58	
FREAK [22]	26	0.08		29	1.06		6	0.08		8	0.10		16	0.06		13	11.20	
DFM [25]	312	1.72	0.1	465	15.27	0.51	35	22.80	0.12	53	4.92	0.17	112	1.75	0.07	153	36.53	0.82
LoFTR [24]	360	1.83	0.06	1,245	22.93	0.68	2,261	26.72	0.07	410	5.78	0.16	204	2.04	0.06	340	54.47	1.06

Table 5. **Quantitative comparison in terms of inliers and computational cost.** The same information is provided for sequential and template modes. The average number of inliers and computational time (in seconds) are computed across the whole sequence. The computational cost for the GPU accelerated execution of learned methods is also provided.

	Cardiopulmonary		Human Laparoscopy		Bronchoscopy		Gastroscopy		Rabbit Laparoscopy		Pig Laparoscopy	
	recall	1-precision	recall	1-precision	recall	1-precision	recall	1-precision	recall	1-precision	recall	1-precision
Sequential mode												
SIFT [15]	0.33	0.49	0.35	0.46	0.07	0.88	0.12	0.79	0.35	0.46	0.16	0.73
FAST [16]	0.24	0.60	0.16	0.72	0	0.99	0.04	0.92	0.28	0.54	0.01	0.98
BRIEF [17]	0.47	0.39	0.44	0.39	0.05	0.91	0.18	0.72	0.47	0.35	0.31	0.53
ORB [18]	0.34	0.65	0.36	0.64	0.13	0.87	0.14	0.86	0.37	0.63	0.30	0.70
SURF [19]	0.43	0.57	0.44	0.57	0.04	0.96	0.14	0.86	0.43	0.57	0.20	0.80
BRISK [20]	0.38	0.62	0.43	0.57	0.15	0.85	0.12	0.88	0.35	0.65	0.22	0.78
HARRIS [21]	0.03	0.96	0.03	0.95	0.02	0.95	0.02	0.95	0.01	0.99	0.02	0.98
FREAK [22]	0.40	0.42	0.26	0.58	0.01	0.99	0.10	0.83	0.43	0.37	0.06	0.89
DFM [25]		0		0		0		0		0		0
LoFTR [24]		0.15		0.13		0.55		0.38		0.13		0.28
Template mode												
SIFT [15]	0.09	0.86	0.03	0.95	0.05	0.92	0.04	0.94	0.14	0.77	0.01	0.99
FAST [16]	0.05	0.92	0	1	0	1	0.02	0.97	0.07	0.88	0	1
BRIEF [17]	0.13	0.81	0.03	0.94	0.04	0.95	0.06	0.92	0.17	0.72	0.01	0.98
ORB [18]	0.08	0.92	0.04	0.96	0.08	0.91	0.06	0.94	0.09	0.91	0.04	0.96
SURF [19]	0.09	0.91	0.02	0.98	0.02	0.99	0.05	0.96	0.13	0.87	0.01	0.99
BRISK [20]	0.10	0.89	0.02	0.98	0.13	0.87	0.05	0.95	0.13	0.90	0.02	0.99
HARRIS [21]	0.01	0.97	0.01	0.97	0.02	0.95	0.01	0.97	0	1	0	1
FREAK [22]	0.10	0.85	0.01	0.98	0	0.99	0.04	0.95	0.14	0.78	0	1
DFM [25]		0		0		0		0		0		0
LoFTR [24]		0.58		0.76		0.71		0.72		0.45		0.85

Table 6. **Quantitative comparison in terms of recall and 1-precision.** See caption in Table 5. The recall cannot be computed for DFM [25] and LoFTR [24] since these learned methods do not calculate the keypoints for each of the images. The 1-precision metric is always 0 for DFM [25] since due to its refinement layer we consider that all of the computed matches are indeed inliers.

SURF [19]. In addition, for the bronchoscopy and gastroscopy datasets, the recall values are very low –with a maximum of 0.15 and 0.18, respectively. This demonstrates that matching features is an extremely challenging task for this kind of images. According to 1-precision, the best algorithms are DFM [25] and LoFTR [24], since they obtain a smaller proportion of false matches.

To sum up, we can conclude that even though SIFT [15], SURF [19], BRIEF [17] and DFM [25] could be viable options for endoscopy images with texture, the most promising algorithm for any kind of endoscopy image would be LoFTR [24]. This method obtains more inliers all over the images, including textureless areas and additionally, it is one of the most computationally efficient (using a GPU). Moreover, it is one of the algorithms with less false matches. Although our results show a great performance of this method in this type of sequences not considered in training, as future work a training with endoscopic data could be considered in order to evaluate the final performance and compare it with the current general purpose model.

3.2. Application: 3D shape reconstruction

Feature matching can be used in many computer vision applications, such as the 3D shape reconstruction along with the camera trajectory from 2D point tracks. This is a highly challenging task for endoscopic images as the image views that are captured by the monocular endoscope are far from ideal for shape reconstruction (type of motion, some constraints in the shape deformation, strong occlusions, etc.). In this work, we apply the matches that we have obtained in this study to solve the problem by means of COLMAP [30], an incremental structure-from-motion algorithm. COLMAP [30] was originally implemented for SIFT [15] features, but it offers the option to use matches obtained with other approaches. The input images should be overlapping images taken from different viewpoints. The dataset that better fulfills these constraints is the uterus video since it has translation around the organ (note that not all datasets we use in this paper can be used for 3D estimation, due to the lack of camera motion). Therefore, we apply COLMAP [30] for a subset of the dataset by applying the matches obtained with each of the algorithms. According to the results that we have obtained, SIFT [15] and LoFTR [24] have the best 3D reconstructions even though some inconsistencies are detected. For both algorithms, we have some outlier points. Then, for SIFT [15] we have that there are some gaps in the reconstruction since the inliers were more concentrated in some areas (see first and second columns in Fig. 1). For LoFTR [24] the reconstruction is considerably denser as there was a large number of inliers in all the areas of the images (see third and fourth columns in the same figure). However, there is a part of the reconstruction in which the depth is not well recovered. This is probably because due to the translation, there are fewer images that capture that area and therefore there might not be enough information –motion parallax– to better determine the depth. Moreover, that part of the images is much darker than the rest of the areas, making matching more complex and inaccurate. Regarding the camera trajectory, SURF [19], SIFT [15] and LoFTR [24] obtain the most accurate camera trajectory –at least from a qualitative point of view. Nevertheless, it is hard to determine to which extent is correct since a 3D ground truth is not available for quantitative evaluation.

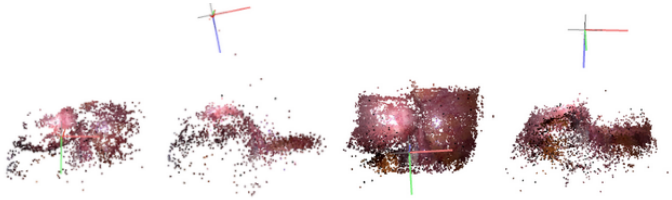

Fig. 1. **Uterus 3D reconstruction.** Two novel views (XY & XZ) of the 3D reconstruction obtained by SIFT [15] and LoFTR [24], respectively.

4. CONCLUSION

In this work we have proposed an exhaustive analysis where both handcrafted and learned algorithms are considered in endoscopic images with different particularities. In general, our sequences include a wide variety of complex points to handle in this context, such as changes of illumination, deformations, motion blur, noisy observations, occlusions due to bubbles or fluids, and so on. The only method that was eligible for all of the videos –for non-specific endoscopic images and assuming no specific training data– was LoFTR [24]. It obtained the most sparse inliers and it was able to find good inliers in areas that none of the other methods could, such as textureless or blurred areas. Moreover, this method was also exploited properly to infer 3D information from endoscopic images.

5. REFERENCES

[1] V. X. Nguyen, V. T. L. Nguyen, and C. C Nguyen, "Appropriate use of endoscopy in the diagnosis and treatment of gastrointestinal diseases: up-to-date indications for primary care providers," *Int. J. Gen. Med.*, vol. 3, no. 11, pp. 345–357, 2010.

[2] E.J. Kuipers and J. Haringsma, "Diagnostic and therapeutic endoscopy," *J. Surg. Oncol*, vol. 92, no. 3, pp. 203–209, 2005.

[3] A. Darzi and Y. Munz, "The impact of minimally invasive surgical techniques," *Annual Review of Medicine*, vol. 55, pp. 223–237, 2004.

[4] A. Agudo, "Spline human motion recovery," in *ICIP*, 2022.

[5] S. Liu, J. Fan, D. Ai, H. Song, T. Fu, Y. Wang, and J. Yang, "Feature matching for texture-less endoscopy images via superpixel vector field consistency," *Biomedical Optics Express*, vol. 13, no. 4, pp. 2247–2265, 2022.

[6] A. Agudo, "Total estimation from RGB video: On-line camera self-calibration, non-rigid shape and motion," in *ICPR*, 2020.

[7] D. Mukhtorov, M. Rakhmonova, S. Muksimova, and Y. I. Cho, "Endoscopic image classification based on explainable deep learning," *Sensors*, vol. 23, no. 6, pp. 3176, 2023.

[8] A. Agudo, "Segmentation and 3D reconstruction of non-rigid shape from RGB video," in *ICIP*, 2020.

[9] D. Kitaguchi, N. Takeshita, H. Hasegawa, and M. Ito, "Artificial intelligence-based computer vision in surgery: Recent advances and future perspectives," *Annals of gastroenterological surgery*, vol. 6, no. 1, pp. 29–36, 2022.

[10] A. Agudo, "Piecewise bézier space: Recovering 3D dynamic motion from video," in *ICIP*, 2021.

[11] A. Malti and A. Bartoli, "Combining conformal deformation and cook-torrance shading for 3D reconstruction in laparoscopy," *TBE*, vol. 61, no. 6, pp. 1684–1692, 2014.

[12] S. Zeng, Y. Zhao, and S. Li, "Comparison between the traditional and deep learning algorithms on image matching," in *ICIS*, 2022.

[13] R. Szeliski, *Computer vision: algorithms and applications*, Springer London, 2010.

[14] D. Tyagi, "Introduction to feature detection and matching," *Medium*, 2020.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91â110, 2004.

[16] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*, 2006.

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *ECCV*, 2010.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *ICCV*, 2011.

[19] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *ECCV*, 2006.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *ICCV*, 2011.

[21] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988.

[22] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *CVPR*, 2012.

[23] S. Isik, "A comparative evaluation of well-known feature detectors and descriptors," *IJAMEC*, vol. 3, pp. 1–6, 2014.

[24] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *CVPR*, 2021.

[25] U. Efe, K. G. Ince, and A. Alatan, "DFM: A performance baseline for deep feature matching," in *CVPRW*, 2021.

[26] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. Yi, "COTR: Correspondence transformer for matching across images," *arXiv*, 2021.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2015.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381â395, 1981.

[29] D. Barath, J. Matas, and J. Noskova, "MAGSAC: Marginalizing sample consensus," in *CVPR*, 2019.

[30] J. L. Schönberger and J. M. Frahm, "Structure-from-Motion Revisited," in *CVPR*, 2016.