On Feature Tracking in X-Ray Images

Moritz Klüppel[1], Jian Wang[1], David Bernecker[1], Peter Fischer[1,2], Joachim Hornegger[1,2]

[1] Pattern Recognition Lab, FAU Erlangen-Nürnberg
[2] Erlangen Graduate School in Advanced Optical Technologies (SAOT)
moritz.klueppel@medtech.stud.uni-erlangen.de

Abstract. Feature point tracking and detection of X-ray images is challenging due to overlapping anatomical structures of different depths, which lead to low-contrast images. Tracking of motion in X-ray sequences can support many clinical applications like motion compensation or 2D / 3D registration algorithms. This paper is the first to evaluate the performance of several feature tracking and detection algorithms on artificial and real X-ray image sequences, which involve rigid motion as well as external disturbances. A stand-alone application has been developed to provide an overall test bench for all algorithms, realized by *OpenCV* implementations. Experiments show that the Kanade-Lucas-Tomasi tracker is the most consistent and effective tracking algorithm. Considering external disturbances, template matching provides the most sufficient results. Furthermore, the influence of feature point detection methods on tracking results is shown.

1 Introduction

In interventional radiology, fluoroscopy provides real-time two-dimensional (2D) X-ray images. To enhance the 2D X-ray images with additional planning information, the images can be overlaid with pre-operative three-dimensional (3D) CT or MRI images. The registration of 3D volume and real-time 2D fluoroscopic images is one of the key challenges for image-guided interventional radiology [1]. However, motion of the patient or distinct organs can occur, which needs to be compensated for an accurate 2D/3D overlay. Therefore, motion has to be detected and characterized first. One way to achieve this step is to apply feature point tracking algorithms, which find 2D correspondences between neighboring frames.

Previous work on patient motion compensation mostly falls into two categories, namely model-driven [2] and data-driven [3]. An example for the methods used in the latter category is the optical-flow based Kanade-Lucas-Tomasi tracker (KLT-Tracker) [4], which is the starting point of our evaluation.

In computer vision, most of the feature tracking methods were designed for natural images. Tracking in X-ray images is a more challenging task due to the lower contrast. Furthermore, overlapping structures of different depths lead to a low distinction. To authors' knowledge, no previous work has been published so far, which assesses the standard behavior of tracking algorithms on X-ray images. This work presents an evaluation of state of the art feature tracking approaches for estimating the patient motion from 2D X-ray images. The tracking methods are evaluated on both digitally reconstructed radiography (DRR) image sequences and clinical X-ray images.

2 Materials and Methods

Feature tracking relies on the robust detection of salient points or regions, which appear over several frames. For this task, several methods are evaluated in the following section. An implementation of all methods can be found in the open-source computer vision library *OpenCV*. Although no direct comparison between the *OpenCV* algorithms and the reference implementations has been performed, this library is widely used in the field of computer vision and is permanently enhanced. The next section then describes how the detected feature points are tracked over time.

2.1 Feature Detection and Description

In this evaluation, the following feature point detection methods are considered.

- Good Features To Track (GFTT) [4]
- Features from Accelerated Segment Test (FAST) [5]
- Scale Invariant Feature Transform (SIFT) [6]
- Speeded Up Robust Features (SURF) [7]
- Binary Robust Invariant Scalable Keypoints (BRISK) [8]

Out of the five methods, GFTT was developed for the use in conjunction with a specific tracking algorithm, the KLT-Tracker. GFTT aims at finding feature points which exhibit optimal characteristics for this tracking method. Since the KLT-Tracker uses patches around the feature points for tracking, GFTT is a pure feature point detector. In contrary, FAST, SIFT, SURF and BRISK do not only detect feature points, but also encode information about the surrounding region of the feature point. These descriptor values, which are usually invariant to scaling and rotation, can be used to establish correspondences between feature points in successive frames.

2.2 Feature Tracking

For the actual tracking of the detected feature points over the neighboring frames, three different methods are considered.

The Kanade-Lucas-Tomasi tracker (KLT-Tracker) uses small image patches around the detected feature points to compute the sparse optical flow between two neighboring frames. One drawback of this method is that it can only cope with small displacements of the individual feature points. As comparison, the descriptor values obtained from the each feature points can be used for descriptor matching between the two successive frames. For computing the correspondences between the feature points, we either use a brute-force nearest neighbor algorithm, or the Fast Library for Approximate Nearest Neighbors (FLANN) [9]. As a third tracking method, we have used template matching, where relative larger patches surrounding the feature points are used. Each patch is compared to its surrounding in the next frame using normalized sum of squared differences (SQD), normalized correlation coefficient (COEF) or normalized cross correlation (NCC). The displacement for the feature point is then given by the displacement which exhibits the smallest difference or highest correlation measure of the image patch.

2.3 Experiments

With the application of interventional imaging in mind, the feature tracking algorithms need to be evaluated under the aspect of real-time capability and the actual quality and accuracy of the tracking results. Following performance metrics and experimental results are based on these two evaluation points.

Performance Metrics The computational efficiency is affected by operations performed on the whole image (e.g. detecting feature points) and operations performed for each detected point (e.g. descriptor calculation). The rate between the number of tracked feature points and the costs for tracking each of these points can be described by the relative computational efficiency

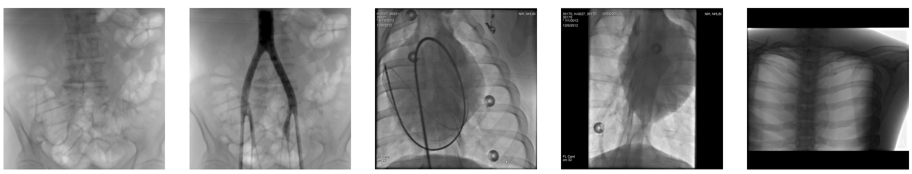
$$t_{rel} = \frac{T}{\#\text{ of detected/tracked features}} \tag{1}$$

where T is the overall execution time for one frame.

The consistency of tracking results is a decisive factor for a successful tracking method. The proposed modified Tracking Rate (mTR) computes the ratio of feature points in I_0 and the number of actually tracked feature points in I_t

$$\text{mTR} = \frac{\#\text{ of tracked features in } I_t}{\#\text{ of features in } I_0} \tag{2}$$

For the case of known ground-truth movement $\boldsymbol{u}_i^*$ of a set of points, the Object Tracking Error (OTE) is used for measuring tracking accuracy. Based



(a) DRR (b) DRR_{cfd} (c) $\text{XRAY}_{\text{dia1}}$ (d) $\text{XRAY}_{\text{dia2}}$ (e) XRAY_{pha}

Fig. 1. The different datasets which are used for experimental evaluation. Each set provides a sequence of images, acquired under motion.

Table 1. Computational efficiency of feature detection and tracking procedures. Mean computation time as well as corresponding standard deviation are presented in milliseconds (ms).

Detection	GFTT	FAST	SIFT	SURF	BRISK		
t_{rel}[ms]	1.095	0.009	5.573	0.433	0.181		
$\sigma\,(t_{rel})$[ms]	0.11	0.001	5.515	0.106	0.019		
Tracking	KLT	tmSQD	tmNCC	tmCOEF	dmSIFT	dmSURF	dmBRISK
t_{rel}[ms]	0.068	1.728	1.733	1.749	61.636	7.866	4.381
$\sigma\,(t_{rel})$[ms]	0.027	2.863	2.857	2.837	69.712	4.689	4.238

on the tracked feature points, the error of the estimated motion $\hat{\boldsymbol{u}}_i$ of all the N frames is combined in the modified Tracking Error:

$$\text{mTE} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{u}_i^* - \hat{\boldsymbol{u}}_i\|^2 \quad . \tag{3}$$

This measure describes tracking accuracy and the consistency of the tracking result at the same time.

Experimental Materials The evaluation is performed on artificial and real-world data. An illustrative overview of the used image sequences is given in Fig. 1. For the simulated digitally reconstructed radiography (DRR) images the ground truth motion is known. The motion consists of translations and rotations in 3D which were projected to 2D motion in the images. For both kind of motions, one dataset was evaluated respectively. For tracking algorithms in a clinical background, external disturbances in image sequences are common. Interventional devices or contrast agent can suddenly appear in the image and cause sustainable changes of the image data. Therefore, the behavior of the feature point tracking algorithms is also evaluated for such sequences in one dataset (DRR_{cfd}), where blood flow was simulated by Computational Fluid Dynamics (CFD). The real image sequences are acquired using a C-arm CT and have unknown ground truth data. For these datasets, only the computational efficiency and the mTR can be calculated. Two datasets ($\text{XRAY}_{\text{dia1}}$, $\text{XRAY}_{\text{dia2}}$) provide X-ray images of real patients, and one sequence (XRAY_{pha}) consists of images of a phantom model. In this dataset patient motion was simulated by moving the phantom during image acquisition.

3 Results

Tab. 1 presents the results of different feature detection methods in milliseconds (ms). As expected, SIFT is much more time-consuming than BRISK and FAST. The high variance of SIFT performance leads to a wide gap in efficiency while detecting a small and large number of features. Despite its scale-space approach, SURF provides higher computational efficiency. Except for SIFT, all methods are in an acceptable range, also for large numbers of feature points. Similarly, Tab. 1 shows the timing results of tracking procedures. BRISK gives the best result, while the descriptor matching algorithms are very slow. The brute-force method is applied for SIFT (dmSIFT) and SURF (dmSURF), while FLANN is applied for BRISK (dmBRISK). Compared to the other algorithms, the KLT-Tracker has the highest computational efficiency. Meanwhile, all three comparison methods (tmSQD, tmNCC, tmCOEF) of template matching also provide high efficiency.

The experimental results of $\overline{\mathrm{mTR}}$ for the KLT-Tracker are not substantially different from the template matching algorithms good results. All are in the range of more than 97%. In contrast, the descriptor matching approaches provide bad tracking rates of 5-80%. Furthermore, the respective standard deviations are high, which means no consistency is established. A reason for this behavior could be the qualitative ranking of the matches by the respective algorithms, so the matches are admittedly computed correctly, but are not followed constantly over each frame.

The tracking error mTE is shown in Fig. 2. It shows that the descriptor matching procedures fail while following the feature points. In comparison, the KLT-Tracker provides the best results as illustrated by the first five bar sets. However, the error is increased when combined with the highly efficient FAST detector. In contrast, the features detected by SIFT are well tracked, but the low efficiency of this detector (Tab. 1) has to be considered. For the CFD dataset, the template matching algorithms provide the best results. For the KLT-Tracker, the feature points get 'washed' away if the blood flow appears, but the templates can be tracked again when the disturbance is gone.

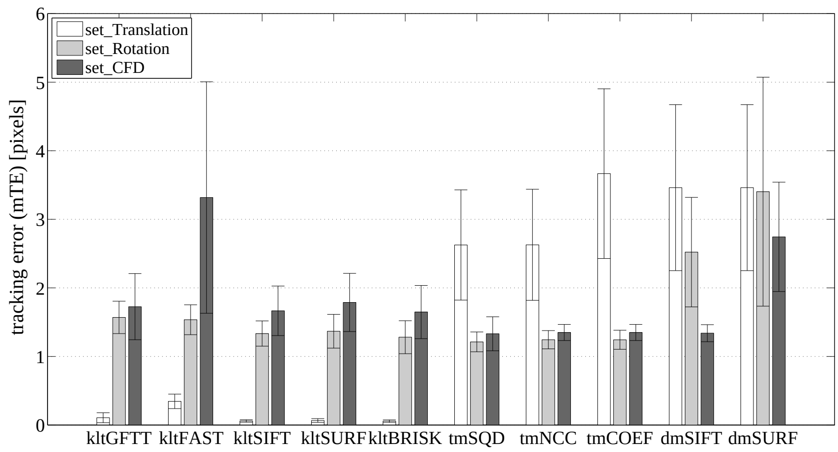


Fig. 2. The tracking error mTE for each DRR dataset is illustrated in this figure. Each tracking procedure has been evaluated. The deficient results of the descriptor matching algorithm in combination with BRISK had been skipped for illustrative purpose.

4 Discussion

In this paper, different popular feature tracking algorithms are evaluated on X-ray images. The evaluation is performed in a single framework that provides an identical test environment and I/O parameters. Experiments are designed to investigate the behaviors of anatomical feature detection and tracking in both simulated (DRR) and real X-ray images. Different 3D motion patterns are considered in the experiments. In general, the computational efficiency decreases with increasing detection accuracy of the feature detection algorithms. The KLT-Tracker is highlighted as an efficient, accurate and consistent tracking approach. Furthermore, template matching provides sufficient results during external disturbances in the image sequences. As the mTR and mTE metrics showed, the descriptor matching algorithms generally failed, which means extensions are needed in further investigations.

As the presented results showed the success and efficiency of optical flow, realized by the KLT-Tracker, future work will have to focus on further implementations of optical flow algorithms. Baker *et al.* present a benchmark of multiple optical flow implementations [10]. To counter the tradeoff between high quality features and low computational efficiency, further SIFT implementations could be improved by parallelization. In addition, future experimental evaluation procedures should be able to use real ground truth 2D motion of the X-ray image to provide more precise test results.

References

1. Markelj P, Tomaževič D, Likar B, et al. A review of 3D/2D registration methods for image-guided interventions. Med Image Anal. 2012;16(3):642–661.
2. Cao Y, Wang P. An adaptive method of tracking anatomical curves in x-ray sequences. In: Med Image Comput Comput Assist Interv. Springer; 2012. p. 173–180.
3. Wang J, Borsdorf A, Hornegger J. Depth-layer-based patient motion compensation for the overlay of 3D volumes onto X-ray sequences. Proc BVM. 2013; p. 128–133.
4. Tomasi C, Kanade T. Detection and tracking of point features. School of Computer Science, Carnegie Mellon University; 1991.
5. Rosten E, Reitmayr G, Drummond T. Real-time video annotations for augmented reality. In: Advances in Visual Computing. Springer; 2005. p. 294–302.
6. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91–110.
7. Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. In: European Conference on Computer Vision. Springer; 2006. p. 404–417.
8. Leutenegger S, Chli M, Siegwart RY. BRISK: Binary robust invariant scalable keypoints. In: Proc IEEE Int Conf Comput Vis; 2011. p. 2548–2555.
9. Muja M, Lowe DG. Fast approximate nearest neighbors with automatic algorithm configuration. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications; 2009. p. 331–340.
10. Baker S, Scharstein D, Lewis J, et al. A database and evaluation methodology for optical flow. Int J Comput Vis. 2011;92(1):1–31.