Pattern Recognition, 26th DAGM Symposium, Springer-Verlag, Berlin, pages 326-333

Efficient Feature Tracking for Long Video Sequences

Timo Zinßer*, Christoph Gräßl*, and Heinrich Niemann

Chair for Pattern Recognition, University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
zinsser@informatik.uni-erlangen.de

Abstract. This work is concerned with real-time feature tracking for long video sequences. In order to achieve efficient and robust tracking, we propose two interrelated enhancements to the well-known Shi-Tomasi-Kanade tracker. Our first contribution is the integration of a linear illumination compensation method into the inverse compositional approach for affine motion estimation. The resulting algorithm combines the strengths of both components and achieves strong robustness and high efficiency at the same time. Our second enhancement copes with the feature drift problem, which is of special concern in long video sequences. Refining the initial frame-to-frame estimate of the feature position, our approach relies on the ability to robustly estimate the affine motion of every feature in every frame in real-time. We demonstrate the performance of our enhancements with experiments on real video sequences.

1 Introduction

Feature tracking provides essential input data for a wide range of computer vision algorithms, including most structure-from-motion algorithms [1]. Other important applications that depend on successful feature tracking are, for example, camera self-calibration [2] and pose estimation for augmented reality [3].

The well-known Shi-Tomasi-Kanade tracker has a long history of evolutionary development. Its basic tracking principle was first proposed by Lucas and Kanade in [4]. For tracking a feature from one frame to the next, the sum of squared differences of the feature intensities is iteratively minimized with a gradient descent method. The important aspect of automatic feature detection was added by Tomasi and Kanade in [5].

Shi and Tomasi introduced feature monitoring for detecting occlusions and false correspondences [6]. They measure the feature dissimilarity between the first and the current frame, after estimating an affine transformation to correct distortions. If the dissimilarity exceeds a fixed threshold, the feature is discarded. This method was further refined in [7], where the X84 rejection rule is used to automatically determine a suitable threshold.

* This work was partially funded by the European Commission's 5th IST Programme under grant IST-2001-34401 (project VAMPIRE). Only the authors are responsible for the content.

Baker and Matthews propose a comprehensive framework for template alignment using gradient descent [8], as employed by the Shi-Tomasi-Kanade tracker. In contrast to the algorithm of Lucas and Kanade, they suggest estimating the inverse motion parameters and updating them with incremental warps. Their *inverse compositional approach* facilitates the precomputing of essential operations, considerably increasing the speed of the algorithm.

As its motion estimation is completely intensity-based, the feature tracker is very sensitive to illumination changes. Jin *et al.* developed a method for simultaneous estimation of affine motion and linear illumination compensation [9]. Our first contribution is the combination of Jin's method with Baker's inverse compositional approach. We evaluate our new algorithm by comparing it with the intensity distribution normalization approach suggested in [7].

Due to small parameter estimation errors, features tracked from frame to frame will slowly drift away from their correct position. We propose to solve the *feature drift problem* by incorporating the results of the affine motion estimation. Another solution with respect to the tracking of larger templates is put forward by Matthews *et al.* in [10].

After a short overview of our tracking system in the next section, we present the combined motion estimation and illumination compensation algorithm in Sect. 3. Our approach for solving the feature drift problem is detailed in Sect. 4. Finally, we demonstrate experimental results in Sect. 5.

2 Tracking system overview

Our goal of real-time feature tracking for long video sequences not only led to the enhancement of key components of the Shi-Tomasi-Kanade tracker, but also required a careful arrangement of the remaining components. In this section, we will shortly explain these additional design considerations.

We employ the feature detector derived in [5]. It was designed to find optimal features for the translation estimation algorithm of the tracker. Tomasi and Kanade also discovered that detected corners are often positioned at the edge of the feature window [5]. As this phenomenon can lead to suboptimal tracking performance, we use smaller windows for feature detection than for feature tracking. Consequently, even if a corner lies at the edge of the detection window, it is well inside the actual tracking window. Another possibility is to emphasize the inner pixels of the detection window by applying Gaussian weights. Unfortunately, this method did not further improve the tracking in our experiments.

When feature tracking is performed on long video sequences, losing features is inevitable. As we want to keep the number of features approximately constant, lost features have to be replaced regularly. In order to retain the desired real-time performance, we devised a hierarchical algorithm which successively selects the best features according to the ranking provided by the interest image. After the selection of one feature, only a local update of the algorithm's data structure is required. Additionally, the algorithm is able to enforce a mini-

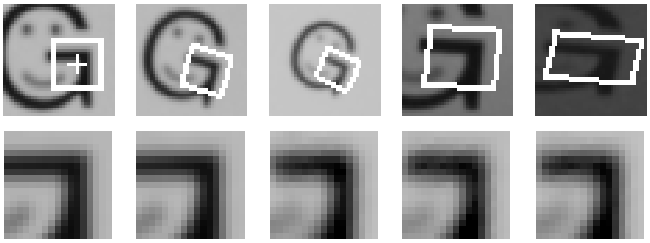

Fig. 1. In the top row, five instances of a feature that was tracked with the proposed algorithm are shown. The respective reconstructions in the bottom row illustrate the performance of the affine motion estimation and the linear illumination compensation.

mum distance between each new feature and all other features, which prevents wasting computational resources on tracking highly overlapping features.

The main task of feature tracking is to estimate the translation of a feature from one frame to the next. Lucas and Kanade observed that the basin of convergence of their gradient descent algorithm can be increased by suppressing high spatial frequencies [4]. The amount of smoothing is bounded by the size of the feature windows, because at least some structure has to remain visible for a meaningful registration. In order to increase the maximum displacement that can be tolerated by the tracker, we employ a Gaussian image pyramid coupled with a coarse-to-fine strategy for translation estimation. Usually working with three levels of downsampled images, we can considerably extend the basin of convergence. Another important addition is the linear motion prediction, which is especially beneficial when a feature moves with approximately constant velocity.

After the affine motion estimation, which is discussed in the next section, outliers have to be detected and rejected. Although the dynamic threshold computation in [7] is promising, we rely on a fixed threshold for the maximum SSD error. In our experience, the gap between correctly tracked features and outliers is sufficiently large when illumination compensation is performed. Jin *et al.* discard features whose area falls below a given threshold [9]. We extend this method by observing the singular values of the affine transformation matrix, which represent the scale of the feature window along the principal axes of the affine transformation. This way, we can also reject features that are extremely distorted, but have approximately retained their original area.

3 Efficient feature tracking

After estimating the translation of a feature from one frame to the next, we compute its affine motion and the illumination compensation parameters with respect to the frame of its first appearance. By continually updating these parameters in every frame, we are able to successfully track features undergoing strong distortions and intensity changes, as illustrated in Fig. 1. In addition, this approach allows us to discard erroneous features as early as possible.

In order to achieve real-time performance, we adopt the inverse compositional approach for motion estimation proposed in [8]. The traditional error function is

$$\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - f_t(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p} + \boldsymbol{\Delta p})) \right)^2 , \tag{1}$$

where $f(\boldsymbol{x})$ and $f_t(\boldsymbol{x})$ denote the intensity values of the first frame and the current frame, respectively. In our case, the parameterized warp function $\boldsymbol{g}$ is the affine warp

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p}) = \begin{pmatrix} 1 + p_1 & p_2 \\ p_3 & 1 + p_4 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} p_5 \\ p_6 \end{pmatrix} , \tag{2}$$

where $\boldsymbol{x}$ represents 2-D image coordinates and $\boldsymbol{p}$ contains the six affine motion parameters. By swapping the role of the frames, we get the new error function of the inverse compositional algorithm

$$\sum_{\boldsymbol{x}} \left(f(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\Delta p})) - f_t(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p})) \right)^2 . \tag{3}$$

Solving for $\boldsymbol{\Delta p}$ after a first-order Taylor expansion yields

$$\boldsymbol{\Delta p} = \boldsymbol{H}^{-1} \sum_{\boldsymbol{x}} \left(\nabla \boldsymbol{f}(\boldsymbol{x}) \frac{\delta \boldsymbol{g}}{\delta \boldsymbol{p}} \right)^T \left(f_t(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p})) - f(\boldsymbol{x}) \right) \tag{4}$$

$$\text{with} \quad \boldsymbol{H} = \sum_{\boldsymbol{x}} \left(\nabla \boldsymbol{f}(\boldsymbol{x}) \frac{\delta \boldsymbol{g}}{\delta \boldsymbol{p}} \right)^T \left(\nabla \boldsymbol{f}(\boldsymbol{x}) \frac{\delta \boldsymbol{g}}{\delta \boldsymbol{p}} \right) .$$

The increased efficiency of the inverse compositional approach is due to the fact that matrix $\boldsymbol{H}^{-1}$ can be precomputed, as it does not depend on the current frame or the current motion parameters. The new rule for updating the motion parameters is

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p}_{\text{new}}) = \boldsymbol{g}(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\Delta p})^{-1}, \boldsymbol{p}) . \tag{5}$$

We combine the efficient inverse compositional approach with the illumination compensation algorithm presented in [9], in order to cope with intensity changes, which are common in video sequences of real scenes. They can be caused by automatic exposure correction of the camera, changing illumination conditions, and even movements of the captured objects.

The linear model $\alpha \boldsymbol{f}(\boldsymbol{x}) + \beta$, where α adjusts contrast and β adjusts brightness, has proven to be sufficient for our application (compare Fig. 1). With this illumination compensation model, our cost function becomes

$$\sum_{\boldsymbol{x}} \left(\alpha f(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\Delta p})) + \beta - f_t(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p})) \right)^2 . \tag{6}$$

Computing the first-order Taylor expansion around the identity warp $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{0})$ gives us

$$\sum_{\boldsymbol{x}} \left(\alpha f(\boldsymbol{x}) + \alpha \nabla \boldsymbol{f}(\boldsymbol{x}) \frac{\delta \boldsymbol{g}}{\delta \boldsymbol{p}} \boldsymbol{\Delta p} + \beta - f_t(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p})) \right)^2 . \tag{7}$$

With the introduction of the new vectors

$$\boldsymbol{q} = (\alpha\Delta p_1 \;\; \alpha\Delta p_2 \;\; \alpha\Delta p_3 \;\; \alpha\Delta p_4 \;\; \alpha\Delta p_5 \;\; \alpha\Delta p_6 \;\; \alpha \;\; \beta)^T\,, \tag{8}$$

$$\boldsymbol{h}(\boldsymbol{x}) = (\; xf_x(\boldsymbol{x}) \;\; yf_x(\boldsymbol{x}) \;\; xf_y(\boldsymbol{x}) \;\; yf_y(\boldsymbol{x}) \;\; f_x(\boldsymbol{x}) \;\; f_y(\boldsymbol{x}) \;\; f(\boldsymbol{x}) \;\; 1 \;)^T, \tag{9}$$

we can rewrite Equation (7) as

$$\sum_{\boldsymbol{x}} \left(\boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{q} - f_t(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\right)^2. \tag{10}$$

Solving this least-squares problem finally results in

$$\boldsymbol{q} = \left(\sum_{\boldsymbol{x}} \boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x})^T\right)^{-1} \left(\sum_{\boldsymbol{x}} \boldsymbol{h}(\boldsymbol{x}) f_t(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\right). \tag{11}$$

As can easily be seen, the 8×8 matrix composed of dyadic products of vector $\boldsymbol{h}(\boldsymbol{x})$ is still independent of the current frame and the current motion parameters. Therefore, it only has to be computed and inverted once for each feature, which saves a considerable amount of computation time. Additionally, the simultaneous estimation of motion and illumination parameters promises faster convergence.

4 Feature drift prevention

There are several reasons why the feature windows in two frames will never be identical in video sequences of real scenes:

- image noise,
- geometric distortions (rotation, scaling, non-rigid deformation),
- intensity changes (illumination changes, camera exposure correction),
- sampling artefacts of the image sensor.

Although these effects are usually very small in consecutive frames, it is obvious that frame-to-frame translation estimation can never be absolutely accurate. Consequently, using only translation estimation will invariably cause the feature window to drift from its true position when the estimation errors accumulate.

As the feature drift problem only becomes an issue in long video sequences, it was not considered in early work on feature tracking [4, 5]. Feature monitoring and outlier rejection as described in [6, 7] can only detect this problem. Once the feature has drifted too far from its initial position, the affine motion estimation fails to converge and the feature is discarded. If subsequent algorithms require highly accurate feature positions, this shortcoming can be problematic. Jin *et al.* use affine motion estimation exclusively, thus giving up the much larger basin of convergence of pure translation estimation [9].

We propose to solve the feature drift problem with a two-stage approach. First, pure translation estimation is performed from the last frame to the current frame. Then, the affine motion between the first frame and the current frame

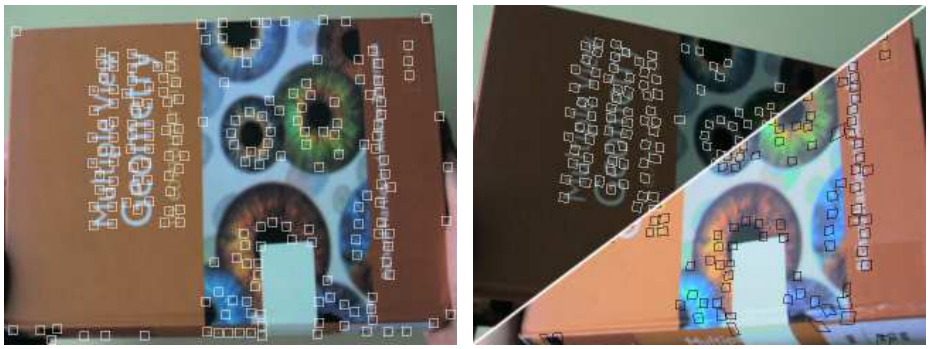

Fig. 2. Illumination compensation test sequence with 100 frames and 200 features. Left image: frame 0. Right image: frame 50 (lower right) / 99 (upper left).

is estimated. Hereby, the newly computed translation parameters and the four affine distortion parameters of the preceding frame are used as initialization. The translation parameters of the new affine motion parameters constitute the final feature position for the current frame. Our solution requires affine motion estimation, preferably with illumination compensation, in every frame. This can now be done in real-time thanks to the efficient algorithm put forward in Sect. 3. Because the coordinate system for estimating the affine motion is always centered on the original feature, small errors in the computation of the affine distortion matrix will not negatively affect the translation parameters in our approach.

5 Experimental evaluation

All experiments in this section were performed on a personal computer with a Pentium IV 2.4 GHz cpu and 1 GB main memory. The video images were captured with a digital firewire camera at a resolution of 640×480. The feature detector, the translation estimation, and the affine motion estimation worked with window sizes of 5×5, 7×7, and 13×13, respectively.

We compared our new affine motion and linear illumination compensation algorithm of Sect. 3 with the photometric normalization approach suggested by Fusiello *et al.* [7]. They normalize the intensity distribution of the feature windows with respect to the mean and the standard deviation of the intensities. Their approach is limited to alternating estimation of motion and illumination.

The test sequence illustrated in Fig. 2 contains 100 frames and exhibits strong intensity changes created by small movements of the test object. 200 features had to be tracked without replacing lost features. The number of successfully tracked features is 162 for our algorithm and 156 for the distribution normalization algorithm. Most of the lost features were close to the edge of the object and left the field of view during the sequence. As confirmed by this experiment, in general the robustness of both approaches is very similar. The great advantage of our algorithm is the lower average number of required iterations, which is

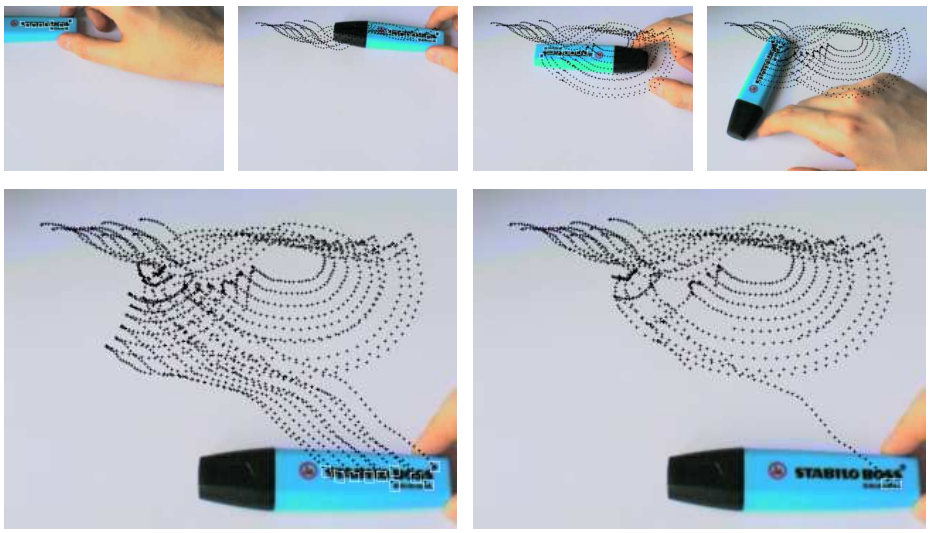

Fig. 3. Feature drift prevention test sequence with 220 frames and 10 features. The upper row shows frames 0, 60, 120, and 150 with feature drift prevention. The lower row shows frame 219 with (left) and without (right) feature drift prevention.

2.21 iterations compared to 3.58 iterations for the distribution normalization algorithm. Consequently, with 20.9 ms against 23.9 ms overall computation time per frame, our tracking algorithm has a notable speed advantage.

The feature drift prevention experiments illustrated in Fig. 3 and Fig. 4 were performed on a test sequence with 220 frames. 10 features were chosen automatically with the standard feature detector described in Sect. 2. The standard approach only tracked one feature over the whole sequence, whereas the proposed feature drift prevention enabled the tracker to successfully track all 10 features. The close-up views of selected frames shown in Fig. 4 confirm the explanations given in Sect. 4. The small errors of the frame-to-frame translation estimation accumulate over time, finally preventing the affine motion estimation used for feature rejection from converging. On the other hand, using the translation parameters of the affine motion estimation as final feature positions yields very accurate and stable results.

6 Conclusion

We proposed and evaluated two enhancements for efficient feature tracking in long video sequences. First, we integrated a linear illumination compensation method into the inverse compositional approach for affine motion estimation. The resulting algorithm proved to be robust to illumination changes and outperformed existing algorithms in our experiments. Furthermore, we overcame the feature drift problem of frame-to-frame translation tracking by determining the

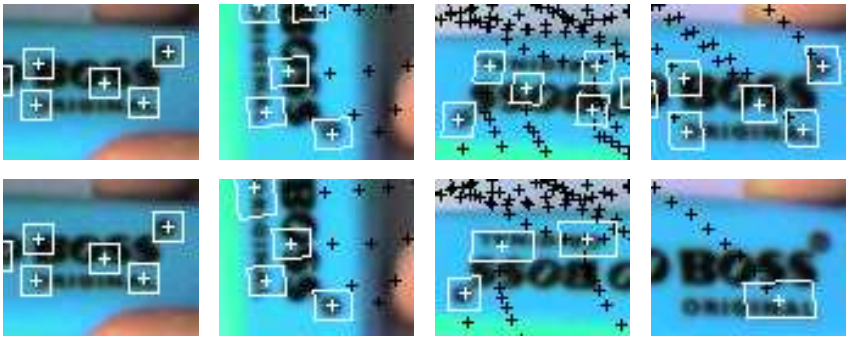

Fig. 4. Close-up views of feature tracking with (upper row) and without (lower row) feature drift prevention are shown for frames 0, 80, 120, and 219 of the test sequence.

final feature position from the translation parameters of the affine motion estimation. We demonstrated the increased accuracy and robustness of this approach in our experiments. With the described enhancements, our tracking system can robustly track 250 features at a rate of 30 frames per second while replacing lost features every five frames on a standard personal computer.

References

1. Oliensis, J.: A Critique of Structure-from-Motion Algorithms. Computer Vision and Image Understanding **84** (2001) 407–408
2. Koch, R., Heigl, B., Pollefeys, M., Gool, L.V., Niemann, H.: Calibration of Handheld Camera Sequences for Plenoptic Modeling. In: Proceedings of the International Conference on Computer Vision, Corfu, Greece (1999) 585–591
3. Ribo, M., Ganster, H., Brandner, M., Lang, P., Stock, C., Pinz, A.: Hybrid Tracking for Outdoor AR Applications. IEEE Computer Graphics and Applications Magazine **22** (2002) 54–63
4. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence. (1981) 674–679
5. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University (1991)
6. Shi, J., Tomasi, C.: Good Features to Track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA (1994) 593–600
7. Fusiello, A., Trucco, E., Tommasini, T., Roberto, V.: Improving Feature Tracking with Robust Statistics. Pattern Analysis and Applications **2** (1999) 312–320
8. Baker, S., Matthews, I.: Equivalence and Efficiency of Image Alignment Algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, USA (2001) 1090–1097
9. Jin, H., Favaro, P., Soatto, S.: Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination. In: Proceedings of the International Conference on Computer Vision, Vancouver, Canada (2001) 684–689
10. Matthews, I., Ishikawa, T., Baker, S.: The Template Update Problem. In: Proceedings of the British Machine Vision Conference. (2003)