ROBUST CORNER TRACKING FOR UNCONSTRAINED MOTIONS

F. Mohanna and F. Mokhtarian

University of Surrey, Guildford, GU2 7XH, UK
Email:[F.Mohanna, F.Mokhtarian]@ee.surrey.ac.uk

ABSTRACT

This paper presents a robust corner tracking method for unconstrained motions. The aim is to track corners in video sequences and identify best correspondences under any transformation, perspective distortion and intensity changes. To extract corners from each video frame, the multi-scale enhanced CSS corner detector is applied. In the matching stage a new two-frame correspondence algorithm using multiple matches is employed. Experiments have been carried out on a wide range of real video sequences depicting translation, scaling, rotation and affine transformation including non-smooth motions with different lighting and different camera motions. All the results were compared to the results of the popular Tommasini[1] feature tracking algorithm. Experiments and comparison of the results confirm that the proposed corner tracker is more efficient, and more robust for unconstrained and non-smooth motions. These properties make it specially useful for video database retrieval and multimedia applications.

Keywords: CSS corner detection, curvature scale space, correspondence, multiple hypotheses, feature tracking.

1. INTRODUCTION

Our interest in corner tracking is for video database retrieval and multimedia applications. Such tracking should be robust, efficient and practical. Feature tracking algorithms which make some important assumptions for tracking, are successful in tracking feature points as long as those assumptions are satisfied. As there is no guarantee that all video sequences have the same or even close motion to their assumed motion model, they are not practical for multimedia applications and actually they fail to retrieve a query when unconstrained motions exist. To tackle unconstrained motions, we need a feature tracker which does not make any limiting assumptions and does not use any restrictive motion models. The underlying principles of a robust algorithm should be general and useful for any kind of video sequences.

This paper presents a multiple-match corner tracking algorithm with all properties discussed above for real-time feature tracker used in video database retrieval. To extract corners from each video frame the multi-scale enhanced Curvature Scale Space (CSS[2]) corner detector [1] is applied. In the matching stage two-frame correspondence using standard cross-correlation (SCC) is used. Since the match with the highest matching score using SCC is not always the best match, all the match candidates of each tracked corner are considered. However by considering multiple matches, many false matches will join the tracked corners which cause tracker to lose its target. Therefore a criterion has to be applied for removing false matches. As a solution, after corner extraction in the off-line stage of the proposed tracker, a family is defined for each corner of the first frame. These families include all the matches that can be found for the corners of first frame when tracking through other frames. Furthermore, among the members of each family, the match with the highest score of the SCC matching is considered as a reference point for that family. Then the distances of all the family members are computed to the reference point of those families. Afterwards all the family members which have distances more than a threshold to their reference points will be removed from the list of tracked corners. By applying this procedure to the end of the input video sequence, we always find (as far as possible) the closest and the minimum matches that we need to continue tracking. We applied the multiple-match corner tracker to many video sequences of CVSSP[3], CMU/VASC[4] and SAMPL[5] databases. All the results were compared to the Tommasini feature tracker results. Results show that the proposed tracker is general purpose, practical and very efficient for multimedia applications. The following is the organisation of the remainder of this paper. Section 2 presents a very brief overview of well-known feature tracking algorithms. Section 3 describes the multi-scale enhanced CSS corner detector briefly. The corner matching process using multiple hypotheses is explained in section 4. In section 5, the experimental results and their comparison to the Tommasini

[1]http://mvl.dimi.uniud.it/Respro/Tracker

[2]The CSS-based shape descriptor has been selected for MPEG-7.

[3]CVSSP in University of Surrey

[4]http://www.cs.cmu.edu/afs/cs/project/cil/ftp/ html/v-images.html

[5]http://sampl.eng.ohio-state.edu/ sampl/data/motion/index.htm

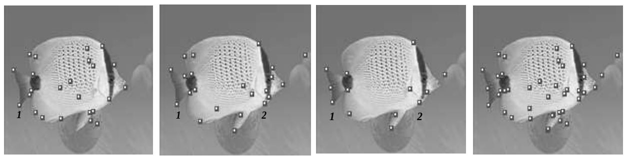
tracker are demonstrated. Discussion and conclusions are presented in section 6.

2. FEATURE TRACKING SURVEY

Several feature point trackers have been proposed. Lucas and Kanade [2] have worked on the tracking problem and proposed a method for registering two images for stereo matching based on a translation model between images. From the initial work of Lucas and Kanade, Tomasi and Kanade [3] developed a feature tracker based on the sum of squared intensity differences (SSD) matching measure, using a translation model. Then, Shi and Tomasi [4] proposed an affine translation model. Their system classified a tracked feature as good (reliable) or bad (unreliable) according to the residual of the match between the associated image region in the first and current frames. If the residual exceeded a user-defined threshold, the feature was rejected. Visual inspection of the results demonstrated good discrimination between good and bad features, but the authors did not specify how to reject bad features automatically. Tommasini et al. [5] proposed a robust tracker based on the work in [4] by introducing an automatic scheme for rejecting spurious features. In Tommasini et al.'s algorithm a feature is defined as a region that can be tracked easily from one frame to the other. Also in their framework, a feature can be tracked reliably if they can find a numerically stable solution to the equation of their assumed motion model. Note that in the Tommasini tracking algorithm, feature extraction and tracking are combined but in our tracker, feature extraction is separate from the tracking algorithm. We are tracking the corners which have been extracted by the ECSS corner detector. Therefore these two trackers are completely different but we compared our proposed tracker with Tommasini tracker as it is a popular feature tracker. Therefore having better results for our proposed tracker in comparison to the Tommasini tracker confirms better performance of the proposed tracker. Finally these techniques and many of the others do not address the problem of recovering points lost during the tracking. The proposed technique in this paper takes a step towards this problem.

3. MULTI-SCALE ECSS CORNER DETECTOR

To extract corners from each video frame, we apply the multi-scale enhanced curvature scale space corner detector to these frames. The enhanced CSS corner detector; ECSS is an improvement of the CSS (Curvature Scale Space) corner detector. The ECSS corner detector outline is as following: The first step is to compute the absolute values of curvature of image edge contours at different scales. The next stage is locating curvature maxima in absolute curvature of short contours and in smoothed absolute curvature of long



(a) σ=*1* (b) σ=*2* (c) σ=*3* (d) *Final*

Fig. 1. *Three outputs of the multi-scale corner detector and its final output on bream image*

contours. Then initial corners are extracted from these local maxima after comparing them to their neighbouring local minima. The final step is to track initial corners at multiple lower scales to improve localisation. More details about the ECSS detector outline and its results can be found in [6]. The degree of ECSS robustness in comparison to some well-known detectors was reported in [7]. It was shown that the robustness of the ECSS detector is higher than the tested detectors. Nevertheless, by applying the ECSS detector to the frames of a video sequence, the positions of some corners can change from frame to frame. Therefore as the ECSS corners positions depend on the edge contours of an image and their local maxima, we minimise these changes by applying the ECSS detector to the frames in a video sequence using three different levels of smoothing (e.g. σ=1, 2, and 3). Then Corners which are marked at least in one of these three outputs (outputs a,b, and c in Fig.1) are the final output (output d in Fig.1) of the multi-scale ECSS corner detector. By applying this procedure to the frames of a video sequence, we consider all changes that may occur in the edge contours of a frame in comparison to its next frame due to any transformation between these two frames or variation of illumination or moving camera. The ECSS corner detector which includes the multi-scale edge detector is referred to as multi-scale ECSS corner detector [1].

4. TWO-FRAME MULTIPLE MATCHING

After extracting the corners of all frames in the off-line stage of the proposed tracker, two-frame corner matching will be applied. In this stage, two-frame correspondence using standard cross-correlation (SCC) is used. Since the match with the highest matching score using SCC is not always the best match, all the match candidates of each tracked corner are considered. However by considering multiple matches, many false matches will join the tracked corners which cause tracker to lose its target. Therefore we need a criterion to remove all false matches that will appear among the tracked corners. As a solution, a family is defined for each corner of the first frame. These families include all the matches that can be found for the corners of the first frame while tracking through other frames. Furthermore, among each family's members the match with highest score of the SCC matching is considered as a reference point for that

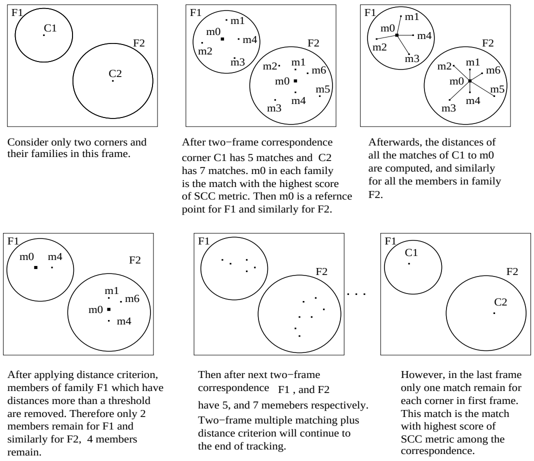


Fig. 2. *An example of the reference points in the families and the distances of other members to these points.*

family. Then the distances of all the family members are computed to the reference point of those families (Fig.2). Afterwards all the family members which have distances more than a threshold to their reference points will be removed from the families and from the tracked corners. As a result of using the described distance criterion, we can have the closest and the minimum number of matches to continue tracking without making any assumptions on motion models and without using any information on camera movement or camera calibration. A remaining issue is corners that multiple-matching algorithm can not find any match for. We solved this problem by monitoring tracked corners when applying multiple-matching to the other frames. For each matched corner through multiple-matching stage, we set a flag to 1. The corners in frame0 also have set flags. Furthermore, we have x and y translations of the corners between each two frames. Then monitoring process will start to mark unmatched corners from frame2. Each unmatched corner in frame1 will be marked in frame2 with zero assigned flag. Zero flag for each corner means that it was marked during monitoring process. Therefore it is not a real match that is derived from the multiple-matching process. Hence, through multiple-matching process between frame2 and frame3, each unmatched corner in frame2 will be marked in frame3 with zero assigned flag while some marked corners on frame2 have zero flags and the others have set flags. Accordingly in frame3, some corners have zero flags and the others have set flags. However during occlusion or any situation in which a corner might be missed (for example the object including this corner does not exist in the remainder of the sequence any more), marking corners by monitoring process should be stopped. Therefore marking of missed corners can continue for two or three frames after occlusion, and not more. This is because if a chance remains for these corners to appear again, they should appear in two or three frames after occlusion. As a result, an alternative is after frame3 only those unmatched corners become marked that have set flags not zero flags. The positions of monitored corners are calculated using translation due to slight motion between two successive frames of a video sequence. Over all, by applying two-frame multiple matching combined with three-frame monitoring the number of tracked corners stays more robust.

5. EXPERIMENTAL RESULTS

We have tested the proposed method on a wide range of real video sequences as mentioned in section 1. In all the sequences different lighting and different camera motion existed. Selected video sequences included all transformations and unconstrained motions; such as translation and occlusion in bream sequence, rotation in Claire and mom-daughter sequences, scaling in cil sequence, affine transformation in hotel sequence and sudden motions in children, tennis, trevor, boxw, heart, mug, and MOVI sequences. In OSU-1 and OSU-2 sequences, all transformations and non-smooth motions are included together and these are good examples of complicated tasks for corner tracking. Bream sequence can be used also as an example of non-rigid motion of the object. The tail or the fin of the bream can move left or right while the whole of the body moves forward. Claire sequence is example of face and head movements. In cil sequence, camera has forward motion. Experiments were performed as follows: first, the multi-scale corner detector automatically extracted a number of corners in first frame of a video sequence for both trackers based on a selected σ. Number of frames in a video sequence was held constant as well. Therefore they started with same corners in first frame and same conditions. We also attempted to obtain the best possible results for each tracker by searching for parameter values that appeared to yield the best results. Our criterion for performance evaluation (PE) of trackers is defined as the number of tracked corners in each frame (n_t) divided by the number of corners in first frame (n_o),

$$PE = \frac{n_t}{n_o} \qquad (1)$$

PE can vary from 0 to 1. An algorithm is the best tracker which has PE=1. Due to lack of space, some of the experimental results have been demonstrated in Fig.3 and Fig.4. In Fig.5 , the first frames of all tested sequences have been shown.

6. DISCUSSION AND CONCLUSIONS

This paper proposed a robust tracker based on two-frame correspondence using a multiple-matching algorithm. The problem of recovering points lost during the tracking was also addressed. Despite having problems finding the best correspondence, applying multiple matching together with

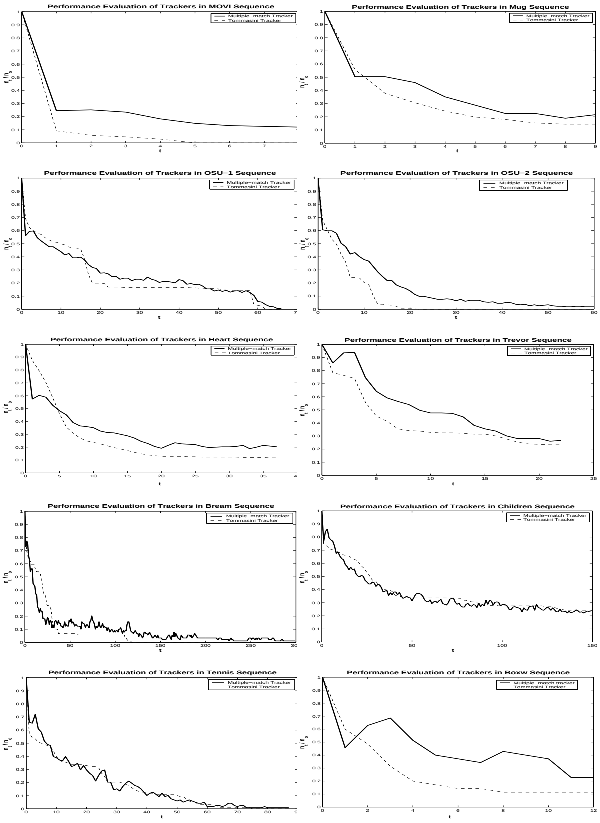


Fig. 3. *Comparison of the Results of the multiple-matching corner tracker and Tommasini feature tracker in MOVI, mug, OSU-1, OSU-2, heart, trevor, bream, children, tennis, and boxw sequences*

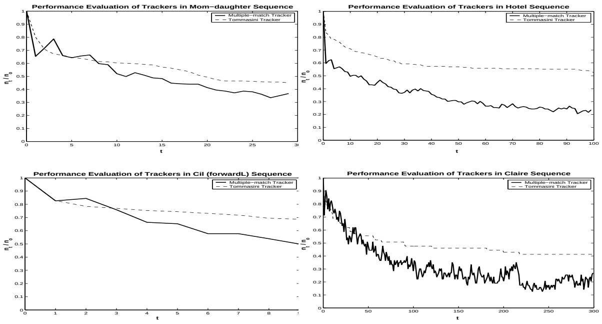


Fig. 4. *Comparison of the Results of the multiple-match and Tommasini trackers in mom-daughter, hotel, Claire, and cil sequences*

Fig. 5. *The first frame of children, hotel, tennis, MOVI, bream, heart, Claire, cil, mug, mom-daughter, boxw, Trevor, OSU-1, and OSU-2 sequences from up-left to down-right. For OSU-1 two frames have been shown.*

distance thresholding helps to increase the chance of finding as close as possible at least one match for each tracked corner. The proposed tracker was applied to a wide range of real video sequences. In any video sequence which has unconstrained motion, non-rigid motion and any motion not anticipated in Tommasini tracker's motion model (such as sequences in Fig.3), Tommasini tracker can not track robustly and as the sequence continues this tracker will lose more corners, while our tracker can continue tracking more robustly. In video sequences such as sequences in Fig.4 which have a motion model very close or the same as Tommasini tracker's motion model, this tracker works more robustly while our tracker loses more corners, but it is still suitable for multimedia applications due to having enough corners on query for retrieving. Overall, application of multiple matching tracker showed that this tracker can be used successfully for retrieving moving objects through video databases in a more practical and efficient way based on having sufficient robust corners on moving objects. In fact, it is a more general purpose tracker.

7. REFERENCES

[1] F. Mohanna and Mokhtarian, "Fusion of corners from multiple scales for robust tracking," in *Proc. International Conference on Computer Vision, Pattern Recognition and Image Processing (part of Proc. Joint Conference on Information Sciences)*, Durham, North Carolina, 2002, pp. 765–768.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc of International Joint Conferences on Artifi cial Intelligence*, 1981, pp. 674–679.

[3] C. Tomasi and T. Kanade, "Detection and tracking of feature points," Tech. Rep., Carnegie Mellon University Technical Report CMU-CS-91-132, Pittsburgh,PA, 1991.

[4] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[5] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good feature track better," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[6] F. Mokhtarian and F. Mohanna, "Enhancing the curvature scale space corner detector," in *Proc 12th Scandinavian Conference on Image Analysis,* Norway, 2001, pp. 145–152.

[7] F. Mohanna and F. Mokhtarian, "Performance evaluation of corner detection algorithms under similarity and affi ne transforms," in *Proc. BMVC Conf.,* Manchester, UK, 2001, pp. 353–362.