A Multi-Scale Approach to Corner Tracking

Farahnaz Mohanna
Farzin Mokhtarian

Department of Electronic and Electrical Engineering
University of Surrey, Guildford, UK, GU2 7XH
{f.mohanna}{f.mokhtarian}@surrey.ac.uk

ABSTRACT

This paper presents a multi-scale corner tracking algorithm based on a multi-scale corner detector. To extract corners from each frame of video sequences, the enhanced CSS corner detector using different scales of smoothing is applied. In the matching stage two-frame correspondence combined with three-frame based monitoring is considered. We monitor tracked corners from the third frame of input sequence in matching stage. Proposed three-frame monitoring helps to ensure that the number of tracked corners and their tracked positions among frames become more robust. Since the proposed corner tracker has enough robust corners based on multi-scale corner detector, it is practical and efficient. In matching stage, among similarity functions, standard cross-correlation, zero-mean cross-correlation, sum of squared differences, and χ^2-test measurements are tested. Well-known real video databases depicting translation, scaling, rotation and affine transformation with different lighting and different camera motions are used as experiments. All experiments confirm that the performance of the proposed tracker is high and reliable due to monitoring matched corners among frames.

Keywords: corner detection, curvature scale space, Canny edge detector, corner matching, correspondence

1 INTRODUCTION

Feature tracking especially corner tracking is an important issue in computer vision and its applications. Interest in tracking process; actually corner tracking is for multimedia applications so it should be robust, efficient and practical. Realtime corner tracking needs robust corners which can be identified and tracked from frame to frame in image sequences. Tracked corners must be located in each frame using the information about the corners in the previous frames. However due to projective, affine and rotational transformations between two successive frames, even the position of good features can vary so far that no correspondence can be found within the match window. While enlarging the match window not only can not solve the problem but also incorrect correspondences will be found which will result in the corner tracker losing its target. Therefore the best solution for the mentioned problem is that the number of corners and also their positions among frames in image sequences become robust. In other words, the number of corners in successive frames should be constant and their positions should change slightly, if at all. In this paper we propose a multi-scale corner tracker based on a multi-scale corner detector. By using a multi-scale corner detector, the number of tracked corners among the frames in image sequences, even when transformations occur, becomes more stable. Accordingly, during the matching process, the number of correspondences between each two successive frames becomes more robust. Furthermore we monitor tracked corners among each three successive frames in image sequences due to occlusion or sudden change in the drift of tracker. Then we mark missed corners during this stage using the translation model between two successive frames. We use translation model based on slight motion between two successive frames. Afterwards during the matching process between the corners of current frame and next frame, if for each marked corner, no correspondence is found, those marked corners should be removed in next frame. Otherwise they will be matched with other corners in next frame. Corners that are marked through monitoring process help the matching process to find new tracked corners. Then the monitoring process exceeds the possibility of being more robust in tracking the number of corners. Overall, multi-scale corner tracker consists of three stages, multi-scale corner detector, removing similar corners marked twice or thrice in multi-outputs of first stage, and corner matching in parallel with monitoring tracked corners. The following is the organisation of the remainder of this paper. Section 2 presents an overview of recent research on feature tracking. Section 3 describes the multi-scale corner detector briefly. Section 4 explains the corner matching process, followed by monitoring procedure for marking missed corners in section 5. In section 6, some of the results of multi-scale tracker that was applied to a number of real video databases are demonstrated. The conclusions are presented in section 7.

2 LITERATURE SURVEY

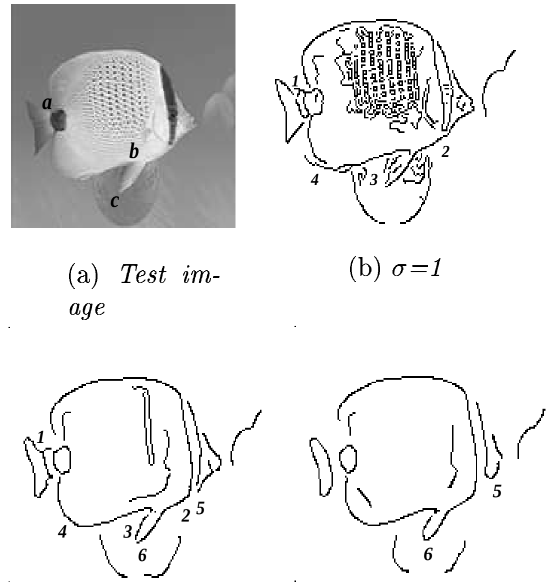
Feature tracking specially corner tracking has attracted considerable attention by computer vision researchers in recent years. We review a number of proposed methods in this section. In [Tomma98a], a feature tracker based on an efficient outlier rejection scheme, suitable for feature tracking in subsea video sequences was proposed. Their work is actually the extension of Shi-Tomasi-Kanade scheme[Shi94] by introducing a technique for rejecting spurious features. Roberts and Charnley [Rober94] used corners as object tokens which are tracked independently. To extract corners, Harris corner detector was applied only in areas of image plane containing useful information. The correspondence problem was solved using a function which compares the normalised magnitude of the difference vectors between each candidate corner vector and the tracker's current corner vector. Shapiro et al.[Shapi92] presented an algorithm to track moving objects in the image using corner points. They used corners which are stable and well-localised that overcome the problem of *flashing* corners, noise and occlusion. Their algorithm has a minimum threshold for surpassing the values of correlation from a certain minimum which makes it less robust. In [Smith98], the problem of obtaining a good initial set of corner matches between two images was tackled and several different matching metrics were evaluated. Shi and Tomasi[Shi94] proposed a method for feature selection, tracking algorithm based on a model of affine image changes. They presented a numerical way of determining affine changes by a Newton-Raphson style minimisation procedure, in the style of what Lucas and Kanade[Lucas81] did for the pure translation model. Tommasini et al.[Tomma98b] extended the well-known Shi-Tomasi-

Kanade tracker[Shi94] by introducing an automatic scheme for rejecting spurious features. Isard and Blake[Isard98] proposed CONDENSATION; Conditional Density Propagation algorithm to establish a stochastic framework for tracking curves in visual clutter, using a sampling algorithm. In [Black98] an approach was described for tracking rigid and articulated objects using a view-based representation. The approach extended work on eignspace representations, estimation techniques, and parameterised optical flow estimation. Smith and Brady[Smith95] proposed a system employing corner features which incorporates cluster tracking using an affine model but no examples of non-rigid motion or non-smooth motion are given.

3 MULTI-SCALE CORNER DETECTOR

To extract corners from each frame of image sequences, we can apply enhanced curvature scale space corner detector to these frames. The enhanced CSS corner detector; ECSS is an improvement of the Curvature Scale Space (CSS[1]) corner detector. Details about the ECSS outline and its results can be found in [Mokht01]. The degree of the ECSS robustness in comparison with some well-known detectors was reported in [Mohan01]. The positions of some corners extracted by the ECSS can change from frame to frame. The positions of corners on an image, extracted by the ECSS depend on the edge contours of that image and their local maxima. Therefore the problem that corner positions are changeable can be addressed effectively. We can minimise this problem using different levels of smoothing in the stage of Canny edge detector of the ECSS corner detector. By applying Canny edge detector with different scales of smoothing, which is named ,

[1]The CSS-based shape descriptor has been selected for MPEG-7 standardisation

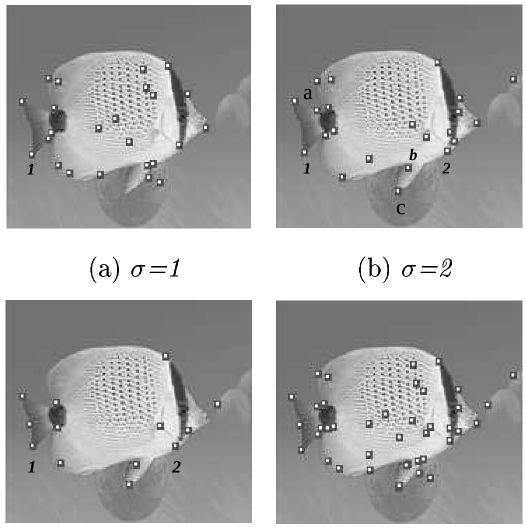


(a) *Test image* (b) σ=*1*

(c) σ=*2* (d) σ=*3*

Figure 1: *Edge contours of test image (a) using Canny edge detector with different levels of smoothing*

multi-scale edge detector, we consider all changes that may happen in edge contours of a frame in comparison to its next frame in image sequences due to any transformation between these two frames or variation of illumination or moving camera. Therefore we select the local maxima of all edge contours of that frame as corner candidates. The ECSS corner detector which includes multi-scale edge detector is named multi-scale corner detector. In Fig.1, three different binary images corresponding to three levels of smoothing (σ); 1, 2, and 3 are produced from test image; Fig.1(a) which is an example grey level image. From this figure it is visible, at a low level of σ such as 1, the output is rather noisy, and edges tend to be cut more frequently, but more details from the image edges can be grabbed. At higher levels of σ such as 2, and 3, however, longer edges are observed but some details disappear. We believe that by taking into account both lower and higher levels of smoothing more information can be extracted from the image which is used later on in the matching stage. We also emphasise that selecting σ larger than three, does not



(a) σ=*1* (b) σ=*2*

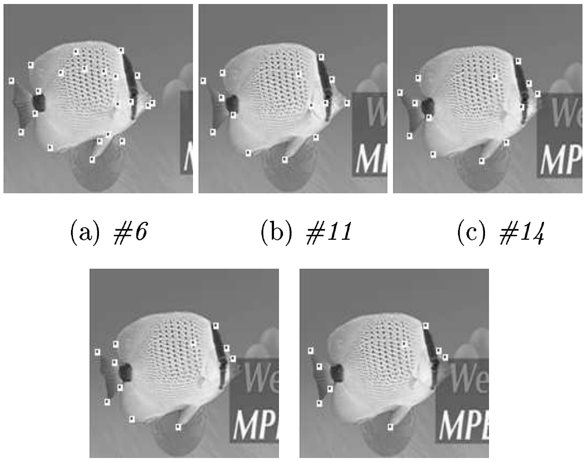
(c) σ=*3* (d) *Final output*

Figure 2: *Three output images of the ECSS corner detector and its final output on test image Fig.1(a); using multi-scale edge detector.*

yield more information about edge contours as we obtained from previous levels of σ. By using multi-scale edge detector, due to filling gaps at points 1, 2, 3, and 4 in Fig.1(c) in comparison to Fig.1(b) and also filling gaps at points 5, and 6 in Fig.1(d) in comparison to Fig.1(c), we do not lose the corners of test image at points a, b, and c in Fig.1(a) as it has been shown in Fig.2(b) for σ=2. In Fig.2, three output images of the ECSS corner detector using multi-scale Canny edge detector with three values of σ have been illustrated. Therefore by considering all corners which are marked in these three images as the output of multi-scale corner detector on test image, we can have all possible corners in this image. This procedure will be done for all frames of an image sequence. Then due to slight motion between two successive frames in image sequence, we can have enough stable corners among the frames in image sequence for matching and tracking process. The final results of multi-scale corner detector can be seen in Fig.2(d). In fact if we compare the corners that have been shown on three images of Fig.2, some of them are marked twice (corner at point 2) or thrice (corner at point 1). Those corners which are marked more than once should be removed from the final corners of multi-scale corner detector. Therefore we consider a small window around each corner of Fig.2(a) and compare this corner and its eight neighbours with all corners of Fig.2(b) and (c). Each corner of Fig.2(b) and (c) which has the same coordinates with that corner or one of its eight neighbours will be removed. Finally the corners of Fig.2(a) and all the corners of Fig.2(b) and Fig.2(c) that are not removed will be considered as the output of multi-scale corner detector which has been shown in Fig.2(d).

4 CORNER MATCHING

Corner matching is commonly referred to as the *correspondence problem* . The problem is how to automatically match corresponding corners from two images, while no incorrect matches are assigned at the same time. The matcher first receives the corners of frame0 and frame1 from the output of multi-scale corner detector. For every corner in frame0, we construct a window match with size 11×11 in frame1 centred at the same position of that corner in frame0. Then all corners of frame1 lying in this window are match candidates for that corner in frame0. If more than one corner lie in this window, we need to compute a similarity function for each of them. Afterwards the winner candidate is one with the highest score of similarity. For computation of similarity function, we consider a small match window with size 3×3 around each corner of frame0 that has more than one match and also around each of its match candidates in next frame. Our selection among different similarity functions are *standard cross-correlation* (SCC), *zero mean cross-correlation* (ZMCC), *sum of squared differences* (SSD), and χ^2-*test*. We applied these four similarity functions to all frames of the bream se-



(a) *#6* (b) *#11* (c) *#14*

(d) *#15* (e) *#16*

Figure 3: *Results of matching process on a number of frames in bream sequences using χ^2-test*

quence and also all frames of CVSSP[2] and CMU/VASC[3] video databases. Overall, we found that SSD, χ^2-test and SCC perform better than ZM-CC in tested video databases. Also SSD and χ^2-test perform better than SCC in tested video databases. In Fig.3, at least three successive frames are considered due to comparing the results of matching process itself (using χ^2-test similarity function) to the results of matching together with monitoring process. Then in the next step using the same frames give us a good clue of what monitoring process is doing in proposed corner tracking algorithm. As it can be seen from the results of matching process in Fig.3 ; simple matching even using the best similarity function, sometimes can not assign a correct match, or even any match between output corners of two successive frames of the multi-scale corner detector. In this way, in long video sequences, it may happen that no match corner remains at final frame. Therefore a good solution to this problem is to monitor tracked corners through the matching process in order to have robust numbers of matched corners among the output of the multi-scale corner detector.

All the corners of the first output of the multi-scale corner detector; output0 are marked in frame0 of tested sequence as the first output of matcher. For second output of the matcher unit; the matched corners between output0 and output1 of the multi-scale corner detector are marked in frame1 of tested sequence as the second output of matcher unit. Then the matched corners between frame1 and output2, the third output of multi-scale corner detector are marked in frame2 of tested sequences as the third output of matcher unit; and this procedure will be continued for all outputs of matcher unit (refer to Fig.5). Therefore if we wanted to compute the translation coordinates of each corner between two successive output images of matcher; it is possible by computing the differences between the *x-coordinate* of a corner in an output of the matcher and the *x-coordinate* of a corner in its previous output where those corners were matched. This difference is named *x-translation* of that corner in current output of the matcher unit. The *y-translation* of each corner in an output of the matcher can be computed in a similar way.

5 CORNER MONITORING

We monitor the matched corners among three successive output images of the matcher unit as following: consider frame0, frame1, and frame2 of input sequences as three first frames for starting the monitoring process. It was mentioned in section 4 that the matched corners between output0 and output1 of the multi-scale corner detector are marked in frame1 and the matched corners between frame1 and output2 are marked in frame2. For each matched corner that is marked through the matching process not through the monitoring process, we set a flag equal to 1. For corners that are not matched through the matching process but marked

[2]http://vssp-www.ee.surrey.ac.uk/data/video/index.html

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

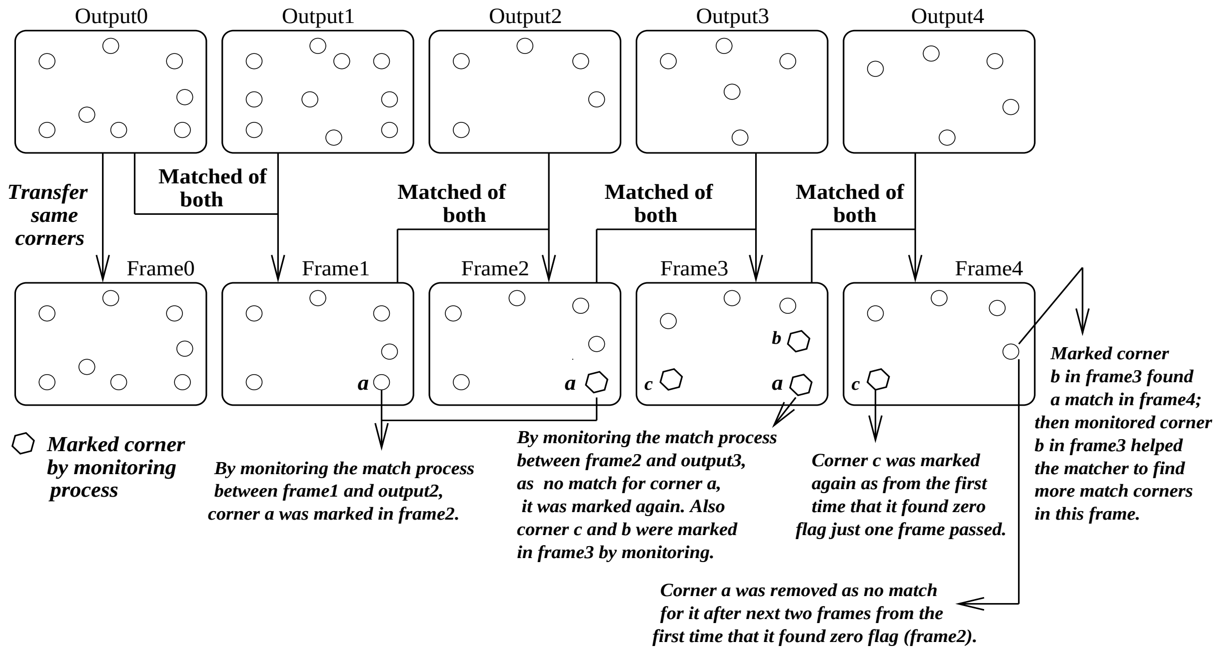


Figure 5: *Scheme of monitoring process*

by monitoring, we set the flag to zero. This helps to separate these corners for further process in the monitoring stage. We also have x and y translations of each corner in frame1 and frame2. The monitoring process starts to mark unmatched corners from frame2 of the matcher unit. Each unmatched corner in frame1 (that we can not find a match for between the corners of output2 of the multi-scale corner detector) is marked in frame2 with zero assigned flag. Zero flag for this corner means that it was marked during monitoring process. Therefore it is not a real match that is derived from the matching process. In continuing the matching process; frame2 and output3 of the multi-scale corner detector are considered while some marked corners on frame2 have zero flags and the others have set flags. Again each unmatched corner in frame2 (that we can not find a match for this corner between the corners of output3) is marked in frame3 but with zero assigned flag. If we continue this procedure to the end of sequence, we will find many falsely marked corners. Therefore we must have some criteria that in such cases prevents marking of unmatched corners. An alternative is after frame3 only those unmatched corners become marked that have set flags not zero flags. In other words, if two previous matched corners of a corner in a frame of an image sequence have zero flag and still in the next frame we could not find for that corner a match candidate, that corner should be removed from marked corners in next frame and also from the list of tracked corners. Therefore any marked corners by monitoring process, automatically will be removed if it can not find a match through matching process in next two frames. The only question that remains is, where we should mark the monitored corner in next frame through this process. Actually we use a translational model between two successive frames which is practical due to slight motion between these frames. In Fig.5 the monitoring process between fourth successive frames is illustrated schematically. In Fig.4, the results of matching process using χ^2-test similarity function combined with monitoring process are illustrated in twelve frames of the bream sequence starting from frame0. By comparing the same frames such as 6,11,14,15 and 16 in Fig.3 and Fig.4, it is visible that applying the

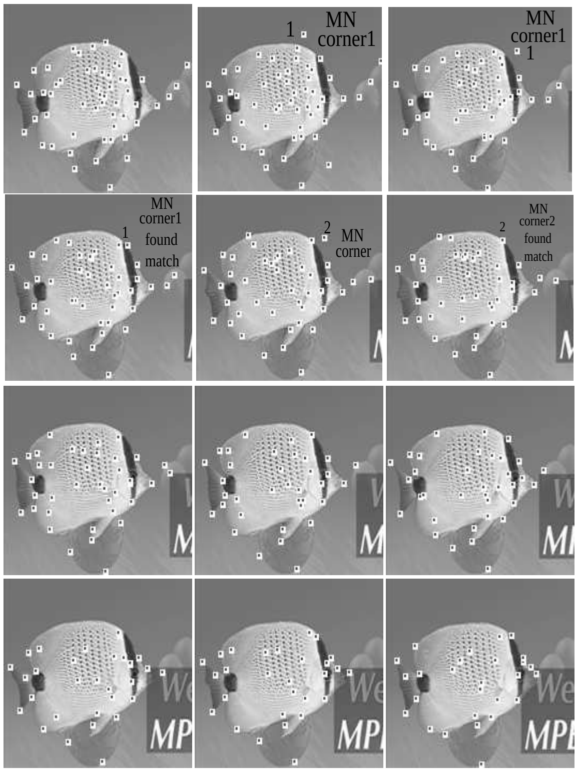


Figure 4: *Results of monitoring tracked corners in twelve frames of the bream sequence starting from frame0 using χ^2-test.*

proposed monitoring procedure makes a great differences to the results of corner tracking. In frames of Fig.4, only two monitoring corners; MN corner1 and MN corner2 were selected to show how they appear in one frame and they are removed from or matched with the corners in next frame or next two frames in the sequence. MN stands for monitor corner. Also in this figure it is visible why we should have a criterion for removing falsely marked corners, otherwise MN corners such as MN corner1 which is a falsely marked corner will remain in the list of tracked corners. Overall, the number of tracked corners using the combined matching and monitoring process is more robust than using only a matching process.

6 EXPERIMENTAL RESULTS

We have tested the proposed algorithm on a wide range of real video sequences. Real video sequences are sequences that have a slight motion between each two successive frames. Selected video sequences for testing included all transformations, such as translation; Fig.4(bream sequence), scaling; building sequence, rotation; Claire sequence and affine transformation; hotel sequence in Fig.6. In all sequences different lighting and different camera motion existed. Furthermore in some sequences such as the bream sequence we have non-rigid motion of object as the tail of bream can go left or right while the whole of body goes forward. The size of comparing window for removing similar corners in the output of multi-scale corner detector, match window, and comparing window for removing more than one match for each corner in matching process was held constant for all sequences. The black point at one end of white path of tracked corners in frame30 of Claire and building sequences shows the previous position of each moving corner from the first frame of these sequences. The movement of corners in Claire sequence is concentrated in the face and the head of Claire. Note that in the building sequence, camera has forward motion. Comparison of the left and right sides of frame0 with frame30 in this sequence shows that due to scaling transformation, corners move from middle of these images toward left, right, up and down directions.

7 CONCLUSIONS

This paper presented a multi-scale corner tracker based on the multi-scale corner detector using two-frame corresponding matching combined with three-frame monitoring. Multi-scale corner detector extracted more robust corners as good features for tracking with arbitrary transformations among frames. Despite having problems finding the best correspondence, using three-frame monitoring from third frame in each video sequence made more stable the number of tracked corners among frames. Also by applying some of the well-known similarity functions, it was found that SSD, χ^2-test and SCC per-

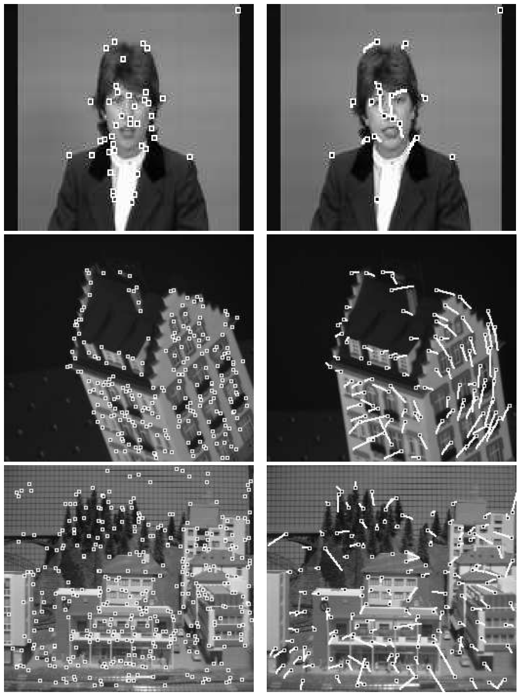


Figure 6: *Results of multi-scale corner tracking in the frame0 and frame30 of the Claire, Hotel and Building sequences respectively*

form better than ZM-CC in tested video databases. Furthermore SSD and χ^2-test perform better than SCC in tested video databases. Application of multi-scale corner tracker showed that this tracker can be used successfully for retrieving moving objects through video databases based on having enough robust corners on and around boundary of moving objects and keeping them robust through frames of video sequences.

REFERENCES

[Black98] M. Black and A. Jepson. Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation. *Journal of Computer Vision*, 26(1):63–84, 1998.

[Isard98] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1(29):5–28, 1998.

[Lucas81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc of International Joint Conferences on Artificial Intelligence*, 1981.

[Mohan01] Farahnaz Mohanna and Farzin Mokhtarian. Performance evaluation of corner detection algorithms under similarity and affine transforms. In *Proc. British Machine Vision Conference*, pages 353–362, Manchester, UK, 2001. Selected as one of the best papers of the conference.

[Mokht01] F. Mokhtarian and F. Mohanna. Enhancing the curvature scale space corner detector. In *Proc 12th Scandinavian Conference on Image Analysis*, pages 145–152, Bergen, Norway, 2001.

[Rober94] J M Roberts and D Charnley. Parallel attentive visual tracking. *Engineering Applications of Artificial Intelligence*, 7(2):205–215, 1994.

[Shapi92] L S Shapiro, H Wang, and J M Brady. A matching and tracking strategy for independently moving objects. In *Proc British Machine Vision Conference*, pages 306–315, 1992.

[Shi94] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[Smith95] S M Smith and J M Brady. Asset-2: Real-time motion segmentation and shape tracking. *IEEE Trans Pattern Analysis and Machine Intelligence*, 17(8):814–820, 1995.

[Smith98] P. Smith. Effective corner matching. In *Proc. British Machine Vision Conference*, pages 545–556, 1998.

[Tomma98a] T. Tommasini, A. Fusiello, and V. Roberto. Robust feature tracking in underwater video sequences. In *Oceans'98 IEEE Conference*, 1998.

[Tomma98b] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good feature track better. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.