IDIAP
Martigny - Valais - Suisse

Combining multiple tracking algorithms for improved general performance

Kim Shearer [a] Kirrily D. Wong [b]
Svetha Venkatesh [c]

IDIAP–RR 00-13

À paraître dans
Pattern Recognition

Institut Dalle Molle d'Intelligence Artificielle Perceptive • CP 592 • Martigny • Valais • Suisse

téléphone +41 − 27 − 721 77 11
télécopieur +41 − 27 − 721 77 12
adr.él. secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP
[B] Electrical and Electronic Engineering, The University of Western Australia
[c] School of Computing, Curtin University of Technology

Rapport de recherche de l'IDIAP 00-13

Combining multiple tracking algorithms for improved general performance

Kim Shearer Kirrily D. Wong Svetha Venkatesh

à paraître dans
Pattern Recognition

Résumé. Automated tracking of objects through a sequence of images has remained one of the difficult problems in computer vision. Numerous algorithms and techniques have been proposed for this task. Some algorithms perform well in restricted environments, such as tracking using stationary cameras, but a general solution is not currently available. A frequent problem is that when an algorithm is refined for one application, it becomes unsuitable for other applications.
This paper proposes a general tracking system based on a different approach. Rather than refine one algorithm for a specific tracking task, two tracking algorithms are employed, and used to correct each other during the tracking task. By choosing the two algorithms such that they have complementary failure modes, a robust algorithm is created without increased specialisation.

1 Introduction

The task of tracking regions of interest is a fundamental problem of computer vision. There are a number of problem domains where a solution to the tracking task is desirable, such as traffic and security monitoring and image and video databases. In some applications there are simplifications which make the task simpler and allow at least a partial solution. When considering fixed security cameras such as might be used for the interior of a building, many of the complicating factors of tracking are removed. In this case simple frame differencing can provide much of the tracking, with further intelligence built in to perform detection of important situations. In the area of video indexing, we wish to specify one or more image regions, which represent objects of interest, and track these regions through the video. This allows the annotation of video by object position, and video retrieval by qualitative spatial reasoning. This requires a more general solution to the tracking problem.

The goal of the work in this paper is to produce a user supervised tracking scheme which requires minimum user intervention. This means tracking regions for as long as possible, without continuing when tracking is incorrect. Complete automation of object tracking through video sequences is currently not possible. Even when the objects of interest are initially specified by the user, the task of object tracking can only be fully automated in a small number of cases. The main problem is that characteristics that determine object boundaries are difficult to specify in absolute terms. Boundaries that seem quite clear to human perception often show little variation in the digital image data.

When using the system described in this paper the user is expected to define the objects of interest, or key objects, in the initial frame of a video. The tracking system then tracks these key objects as far through the video sequence as possible, before once again asking for user input to confirm or correct the detected object position. Thus the tracking system should not only track as well as possible, but it is also important that the algorithm can maintain an indication of confidence in its tracking, so that user intervention can be requested at the correct time.

The format of this paper is as follows. Section 2 describes the approaches used to track the object of interest through a video sequence. In section 3 the method of combining the results of the two algorithms is presented. The results section then shows the improvement in performance gained by the combined algorithm, and finally conclusions are presented.

2 Approaches to tracking

There are numerous methods which have been proposed for tracking objects in image sequences. These vary from simple methods such as frame differencing, to complex methods such as layered segmentation based on motion fields.(1)–(4) In general, tracking methods can be placed in one of two broad categories. These two categories are region trackers and edge trackers.

A region tracker identifies a region of the image, for which it uses a similarity measure to decide on the best matching region in the next image of a sequence. The region is taken to contain some object of interest, with the boundary often being a bounding box, or simple polygon. This category of algorithm suffers from problems from sources such as

1. illumination variation,
2. change in object size due to movement with respect to the image plane,
3. change in object size due to rotation of the object,
4. surface reflectance.

The difficulty with region tracking algorithms is that they need to adapt to gradual changes in conditions in the image, but should not be permitted to slowly drift from the tracked region onto the background.

Edge trackers attempt to follow edges, or locations of high luminance or colour change, through an image. The edges tracked are usually boundaries of objects of interest within an image sequence. This type of algorithm relies on the theory that the boundary of an object of interest will have a strong edge variation in colour or general illumination. Edge tracking algorithms struggle in low illumination, as the changes in colour and luminance are small, and where the background displays strong texture. Where there are strong edges in the background of an image it is difficult to provide a generally applicable rule to decide which edge to track. The major problem with edge tracking algorithms is how to decide when an edge is part of the object, and when to discard a strong edge as part of the background.

Tracking remains a difficult problem partly because for each individual tracking algorithm a simple example can be presented for which the algorithm will fail. This paper describes a novel approach to the tracking problem in which two tracking algorithms, with complementary failure modes, are combined to provide a more robust composite tracking algorithm. It is intended that the composite tracker will allow better performance than either algorithm would achieve in isolation. The algorithms chosen to examine the idea of composite tracking are: correlation as the region based algorithm, and adaptive contours as the edge based algorithm.

The region based tracking system employs simple correlation of rectangular image regions to track objects. In many cases this simple algorithm performs well, requiring intervention only four or five times in a typical video shot of 20 to 25 seconds. There are, however, a number of quite simple configurations which are almost impossible to track using correlation. The simplest case of this is a long, thin object that is angled diagonally to the principal axis of the frame. In this case a bounding region aligned with the principal axis will often contain more background scenery than object, and so the tracked region will follow the background rather than the object as it moves through a sequence of images. Difficulty is also encountered when objects change size, as again the background can dominate tracking, and the bounding box will be less useful even if tracking continues. This type of problem is very difficult, if at all possible, to remedy using correlation. Providing a more detailed region boundary, such as an arbitrary polygon can reduce the background included in the tracking region, but is even more sensitive to changes in object shape. Such an idea is only applicable to rigid bodies, and is therefore too restrictive for use in our video indexing application.

The edge tracker is based on the adaptive contour by Blake, Curwen and Zisserman,(5),(6) which is a closed B–spline defined by a small number of control points. In our implementation, the initial contour is determined by fitting a least squares B–spline to an arbitrary number of boundary points specified by the user. The user also provides the number of spans or control points in the spline, and the position of a reference point on the object in each of the first two frames, from which an estimate of the initial velocity in calculated. The control points of the least squares spline become the state variables for a Kalman filter, which predicts the most likely position of the object in successive frames. For each frame of the video sequence, the tracker searches for the object boundary along normals to the predicted contour at a fixed number of sample points per span. The Kalman filter provides an estimate of the uncertainty in the position measurement, and this estimate can be used to adjust the scale of the search in each frame. For each sample point, the search scale increases when uncertainty is high, and decreases when uncertainty is low. Since only the edges of the object are tracked, the contour is able to follow changes in object size and shape, and the problem of tracking background regions does not occur. Instead, the tracker can become distorted when strong features appear in the background.

The difficulty with edge trackers lies in identifying the *correct* boundary, which may not be the strongest feature in the region of interest. A number of measures for selecting the correct edge were compared. The simplest method locates the strongest colour gradient in the search region, however, using this search scheme the contour is easily distracted by strong edges in the background, and quickly distorts. Colour correlation along normals one pixel wide was evaluated but proved ineffective. An alternative method combines local colour correlation, distance from the predicted location, and colour gradient, to produce a score for each pixel in the search region. This produced improved results. The most reliable method, however, was to select the pixel closest to the predicted location at which a local maximum in colour gradient occurs. Each point on the contour is considered “locked” if this local gradient maximum is above an empirically determined threshold.

The problems caused by strong background edges can be reduced by coupling the edge tracker to a shape template. If we assume that the object of interest is a rigid body at a distance from the camera much greater that the object size, then an affine projection model known as weak perspective can be used. Under this projection model, the boundary of the object of interest in any frame is assumed to be an affine transformation of the original boundary. Using this assumption an affine subspace can be constructed of all affine transformations of the original contour. The Mahalanobis distance measure is then applied to measure how far from this affine subspace a measured shape is, and hence how far the shape has deformed from the original shape of the object.

The assumption of weak perspective is used in the following manner. For each step of the filter, the actual observation of the object's location is followed by an additional *virtual* observation. The virtual observation uses a point within the subspace of affine transformations of the original contour to force some shape selectivity into the contour. The shape used for the virtual observation is the projection into the affine subspace of the contour, or equivalently, the shape within the affine subspace of transformations of the original contour that is closest (minimum Mahalanobis distance) to the current contour. The relative confidence the filter has in the actual and virtual observations can be adjusted, so that the contour will either tend to retain its original shape, or to respond more rapidly to new features. The Mahalanobis distance of the current contour can also be used as a measure of confidence in the trackers performance.

Note that *a priori* knowledge about the object of interest could be used to construct different subspaces.

3 Combining Tracking Algorithms

The initial state of the tracking algorithms is provided by the user, who specifies a number of points on the object boundary. These initial points have a B–spline contour fitted to them using least squares regression. An estimate of the object's initial velocity is then obtained by requiring the user to indicate the position of a reference point on the object in the first and second frame. The correlation algorithm then takes the bounding box of the initial B–spline contour as the region to track, and both algorithms use the initial velocity to improve the initial tracking estimate. For the purposes of testing this system, both tracking algorithms are run on each frame. Tracking is performed from one frame to the next using both algorithms, and then the results are compared.

One important aspect required for this work is a reliable measure of tracking confidence for each algorithm. In order to combine the two algorithms in a meaningful way it is necessary to know when each algorithm is tracking well, and when each fails.

The measure used for success of tracking for the correlation algorithm is consistency of object velocity. During tracking, once the position of the object in the next frame has been determined, the displacement from the current frame to the next is compared against the average displacement for the previous three frames. This has been found to be a very accurate measure for the application of video indexing. Under the general assumption that the frame rate of the video is sufficient for object tracks to be well behaved, this measure performs very well. For the video used in our experiments, a threshold of eight pixel difference was sufficient to distinguish accurately between success and failure for tracking. The videos used in testing were all of a similar size in pixels (384 by 288), so a single threshold for the displacement was suitable.

The adaptive contour uses two indicators to decide on tracking success or failure. The adaptive contour fails if the Mahalanobis distance remains above a preset threshold for a specified number of frames, or if a significant proportion of the sample points along the contour are not locked onto a feature. Both of these measures provide a measure of confidence in the accuracy of tracking, rather than a simple tracking or not tracking assessment.

In some cases this scheme may allow the adaptive contours to deform from the correct boundary, and snap onto an incorrect boundary that provides strong edges. The comparison of the location of the contour boundary with the region tracked by the correlation should allow detection of this case, as it is highly unlikely that the snake and correlation will choose to track the same incorrect location. With the contour tracker attracted to strong edges and correlation tracking by region colour, it would

Tab. 1 –

require an object of similar colour and shape to the object of interest to tempt both algorithms away simultaneously. As velocity is also taken into account during tracking, the likelihood is reduced even further.

There are five possible scenarios for success and failure of the two algorithms, which are given in table 1.

Case 1 is simplest to deal with, as it indicates all is well. Case 2 and 5 both indicate that something is amiss, and in both of these cases the user is asked for intervention. The intervention takes the form of respecifying some subset of the points on the boundary of the object of interest, where the contour is unlocked. In the worst case, the user can completely respecify the object boundary, as in the first frame.

It is in cases 3 and 4 that we see the advantage of the composite approach. In each of these cases we have one tracking algorithm that is confident of its success, and one which has lost the object. In each of these cases we can correct the parameters of tracking for the algorithm that has failed from the algorithm that is successful.

Passing of parameters from adaptive contours to the correlation algorithm is simple. The bounding box of the current adaptive contour is passed to the correlation tracker, and the velocity is calculated from the new position. When the adaptive contour fails and the correlation succeeds, the contour determines its predicted location from the correlation bounding box rather than the Kalman filter. This is done by projecting the original contour shape to fit the interior of the bounding box, then searching for the object boundary, and applying a measurement as usual. The results of this simple approach can be seen in figures 1 and 3 in the results section.

A much better approach to this problem is to prevent either tracking algorithm from reaching a point at which it will fail. This can be accomplished in many cases by using the results of various events in tracking to reduce errors in the tracking algorithms before the errors become too large. The two algorithms cooperate to improve each other's tracking performance.

The clearest example of this is when the bounding box for an object changes. If the adaptive contour algorithm is tracking well, that is most control points are locked and the Mahalanobis distance is consistently small, and the bounding box for the contour has changed significantly, then it is sensible to alter the bounding box used for the correlation tracker. This is particularly true if the positions predicted by the two trackers are mostly in agreement. In this case the bounding box for the correlation can be updated to reflect the new shape detected by the adaptive contour algorithm, thus preventing the loss of tracking likely to occur.

This improves the inherent problem with correlation, which is that there is no sensible way to change the size of the correlation region. By introducing input from an edge based tracker, suitable changes can be made.

One method of improving tracking for the adaptive contour is to incorporate the position predicted by the correlation into the filter when tracking for the contour is uncertain. If the number of control points that have lost lock increases too much or the Mahalanobis distance is increasing, then an observation can be made at both the position predicted by the filter, and the position prediced by the correlation tracker. Whichever of these positions yields the better lock can then be used as the new position, leading to a reduction in tracking failure due to lost control points. This can aid the adaptive contour algorithm in maintaining a reasonable, although not ideal, lock on the object, rather than losing lock completely.

The final sequence in the results section shows the ability of the combined tracking approach.

Fig. 1 –

Fig. 2 –

4 Results

The results are based upon two video sequences. The initial results use a sequence showing a boat approaching the shore of a river. As the boat approaches the shore the sail turns in the wind, giving a clear example of failure for the correlation algorithm. The remainder of the results show portions of a video clip of a person, whom we shall call the guide, walking around the campus of Curtin University. There are a number of difficulties in tracking the guide through this clip. Firstly the guide is not a rigid object, this causes problems for both the adaptive contour algorithm and the correlation. In addition to this the background of the clip is of a solid consistent colour, which causes difficulties for correlation based algorithms. There are also strong edges which cause difficulties for the adaptive contour algorithm.

Figures 1 and 3 show the results of tracking the sail of a yacht near a jetty, and are consecutive frames from a tracking sequence. With the sail being strongly different from the background, this would seem a simple sequence to track. However, the sail turns with the wind part way through the sequence (frame 3(c)), changing its apparent shape. At this point the correlation tracker loses track of the object, usually becoming stranded on a building face in the background. In figure 1(c) the correlations bounding box has been corrected from the adaptive contour, giving a smaller bounding box in the correct position. The correlation fails partly because the region to be tracked no longer appears as it did originally, causing a reduction in strength of correlation.

Figure 3 shows the adaptive contour tracking the sail, with the contour following the shrinking boundary. In figure 1 we can see that the correlation has used the contour to produce a new bounding box when it is lost. This new bounding box is better fitted to the target than the original bounding box. This shows not only that the correlation tracker can be corrected for position from the contour, but that a better bounding box may be acquired from the contour where the object changes size.

The problem for the correlation that is caused by the reduction in the size of the tracked object is shown in figure 2. This figure shows two graphs of the correlation space from the sequence shown in figure 1. The graph 2(a) shows the correlation space in the early part of the sequence, and has a well defined peak. Graph 2(b) shows the correlation space at the point at which the bounding box is corrected from the active contour, this displays a broad, flat peak in correlation space. Here the peak is elongated along the x direction, leading to ambiguity in location of the correct bounding box.

The results in figures 6 to 11 show excerpts from an extended tracking sequence. This video clip causes either tracking system applied in isolation to fail quickly. Figure 4 shows five frames taken from the tracking using correlation alone. Here the bounding box gradually slides off the object to the background. In figure 5 the results of using the adaptive contour alone are shown. This figure shows the adaptive contour losing the object in the first six frames of the sequence. The new tracking system is able to follow the object well into the clip without losing the objects position, in fact tracking is still reliable after 50 frames of the sequence. The sequence is sampled at 14 frames per second, and segments are shown in figures 8, 9, 10 and 11.

When the composite tracking algorithm is applied to the guide video sequence the two algorithms interact in the following manner. The adaptive contour algorithm loses tracking at the seventh frame of the sequence (frame 6(c) in figures 6 and 7) and is reset from the correlation. The contour then successfully regains lock on the object. The correlation algorithm slowly slides off the back of the object and loses track at the 23rd frame (frame 9(c) in figures 8 and 9). The bounding box is corrected from

Fig. 3 –

FIG. 4 –

FIG. 5 –

the adaptive contour and also adjusted to a better size. The correlation has more success tracking with the new bounding box. Figures 10 and 11 show the adaptive contour once again losing track of the guide at frame 35 of the sequence (frame 10(a) in the figures). The correlation is again used to provide a corrected position. While the position given by the correlation is not perfect, figure 10 shows that the adaptive contour is able to adjust from the corrected position to a positive track of the guide. Thus this sequence shows the two algorithms, failing at different times during the sequence, and correcting each other for far better tracking performance than either could achieved by itself.

5 Conclusion

The results in the previous section show the advantages of this approach to tracking. While there are still cases which defeat the composite tracker, it has been clearly shown that this tracker performs better than either tracker in isolation. Most importantly the tracking performance is improved without specialisation of the tracking algorithms for a specific task. Frequently the performance of such algorithms is improved at the expense of generality.

It remains to develop more sophisticated measures for tracking success. The measure used for correlation is effective, but returns only a binary result. Given the used supervised nature of this method, a less discrete measure is desirable. Future work will examine rate of change of correlation, and shape of peaks in correlation, as possible better measures. Either of these measures could be used to give an estimate of confidence, rather than a simple success or failure result. Similarly, the success of the adaptive contour is determined by preset thresholds on the colour gradient, the fraction of unlocked points, and the Mahalanobis distance.

Currently, the parameters of the adaptive contour are set at compile time. These include the covariance of the actual and virtual observations, and steady state search scale, all of which are critical in determining how the contour reacts to occlusion or distraction. Increasing the reaction speed of the tracker makes the contour more easily distorted, but increasing its robustness means that it reacts slowly to changes in velocity, and may lose lock on the object. Again, it will be useful to determine values for these parameters according to the specific sequence being tracked.

Références

1. J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic, “Automatic and semi-automatic methods for image annotation and retrieval in QBIC,” *SPIE Proceedings of Storage and Retrieval for Image Video Databases III*, pp. 24–35, 1995.
2. S. Ayer and H. S. Sawhney, “Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding,” in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, June 1995.
3. J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, “Hierarchical model-based motion estimation,” in *European Conference on Computer Vision*, pp. 237–252, May 1992.
4. D. Daneels, D. Van Campenhout, W. Niblack, W. Equitz, R. Braber, E. Bellon, and F. Firens, “Interactive outliner: An improved approach using active geometry features,” in *IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases II*, pp. 226–233, 1993.

FIG. 6 –

Fig. 7 –

Fig. 8 –

5. A. Blake, R. Curwen, and A. Zisserman, “A framework for spatiotemporal control in the tracking of visual contours,” *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127–145, 1993.

6. A. Blake, R. Curwen, and A. Zisserman, “Affine-invariant contour tracking with automatic control of spatiotemporal scale,” in *Proceedings of the 4th International Conference on Computer Vision*, pp. 66–75, 1993.

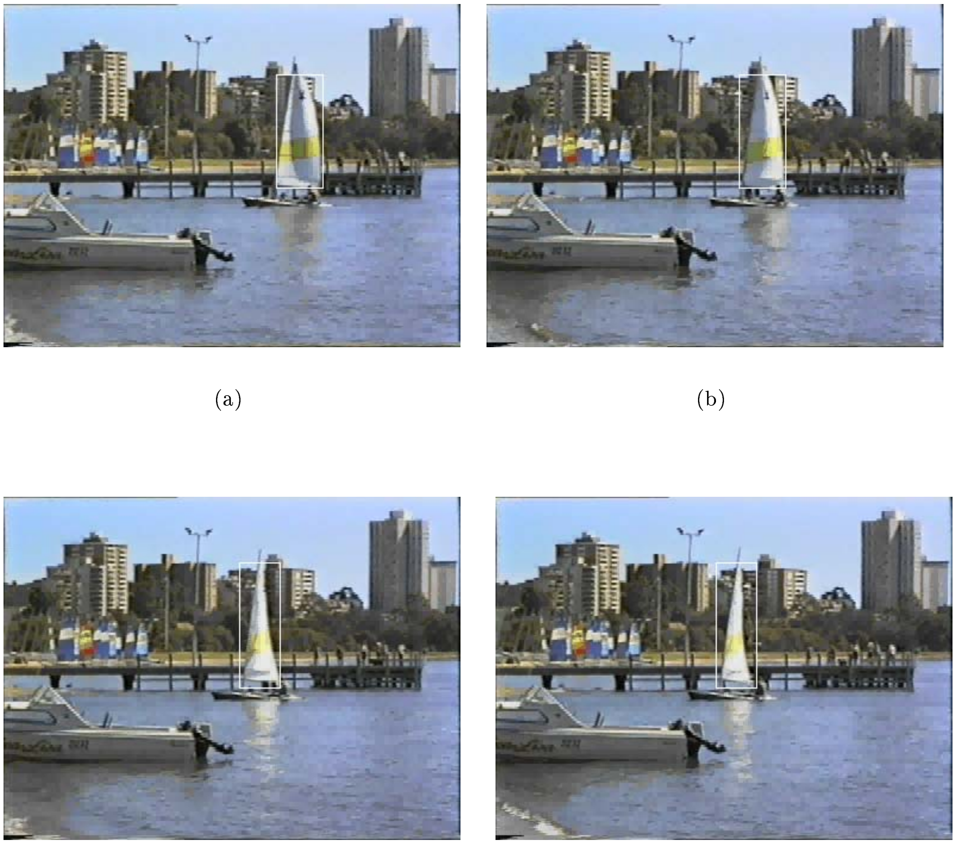
Fig. 9 –

Fig. 10 –

Fig. 11 –

Case	*Region*	*Edge*	*Bounding box*
1	Success	Success	Agree
2	Success	Success	Disagree
3	Success	Failure	
4	Failure	Success	
5	Failure	Failure	

Tab. 1 – *Table of possible tracking outcomes*



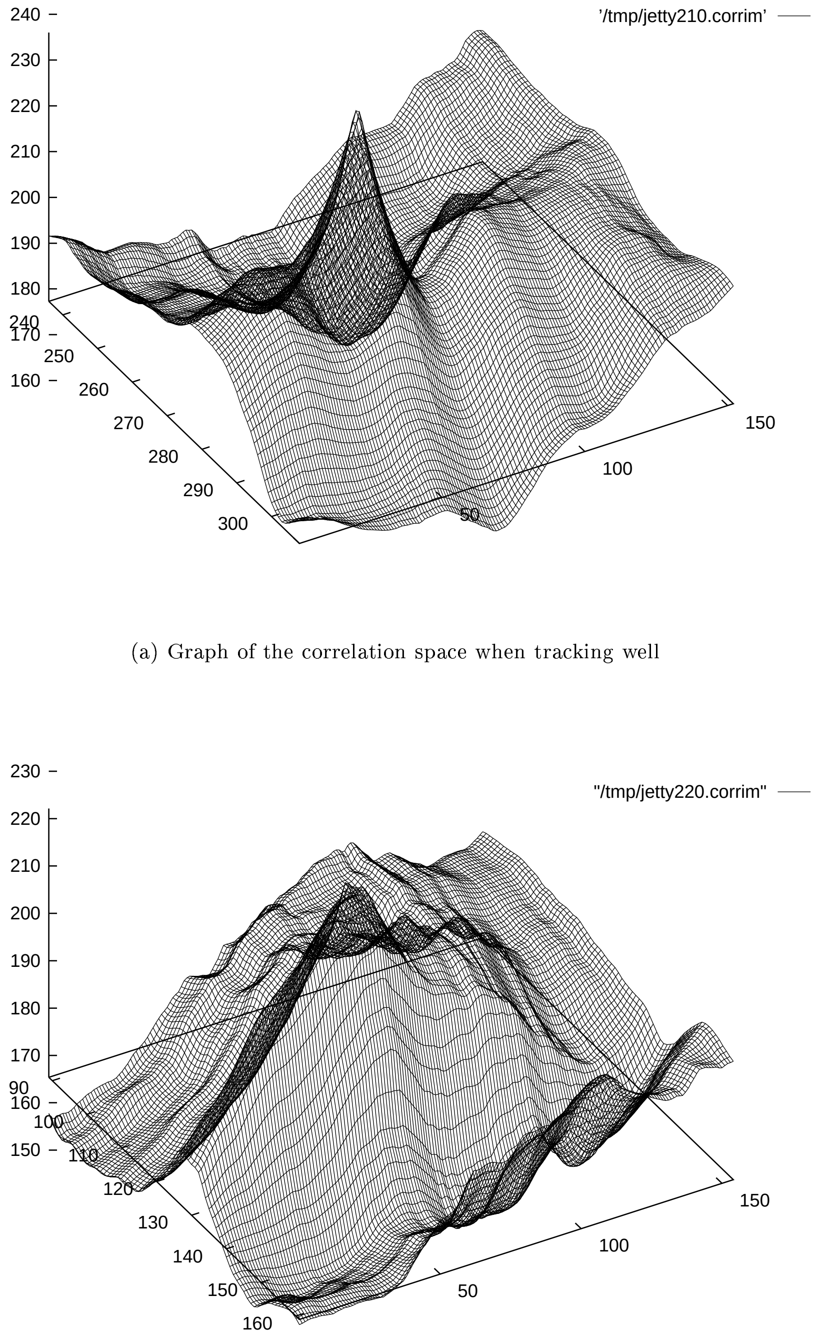
(a)

(b)

(c)

(d)

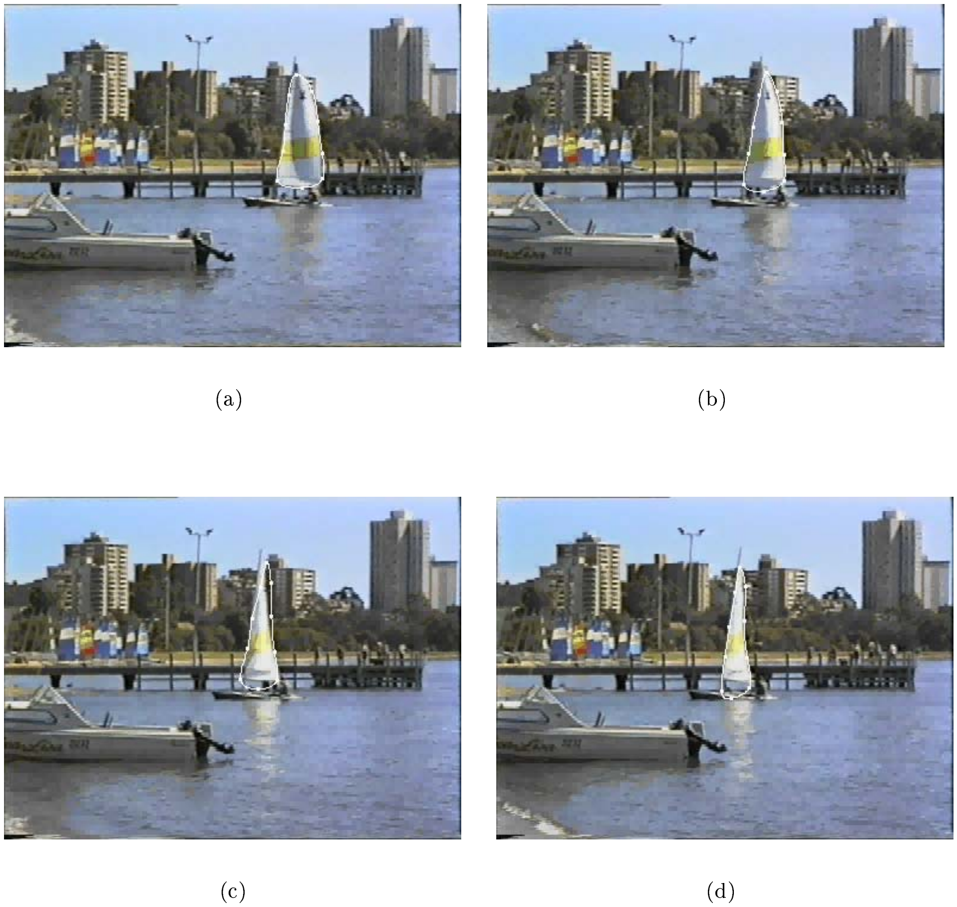
Fig. 1 – *Correlation box with snake correction*



(a) Graph of the correlation space when tracking well

(b) Graph of the correlation space when tracking poorly

FIG. 2 – *Graphs of correlation space when tracking*



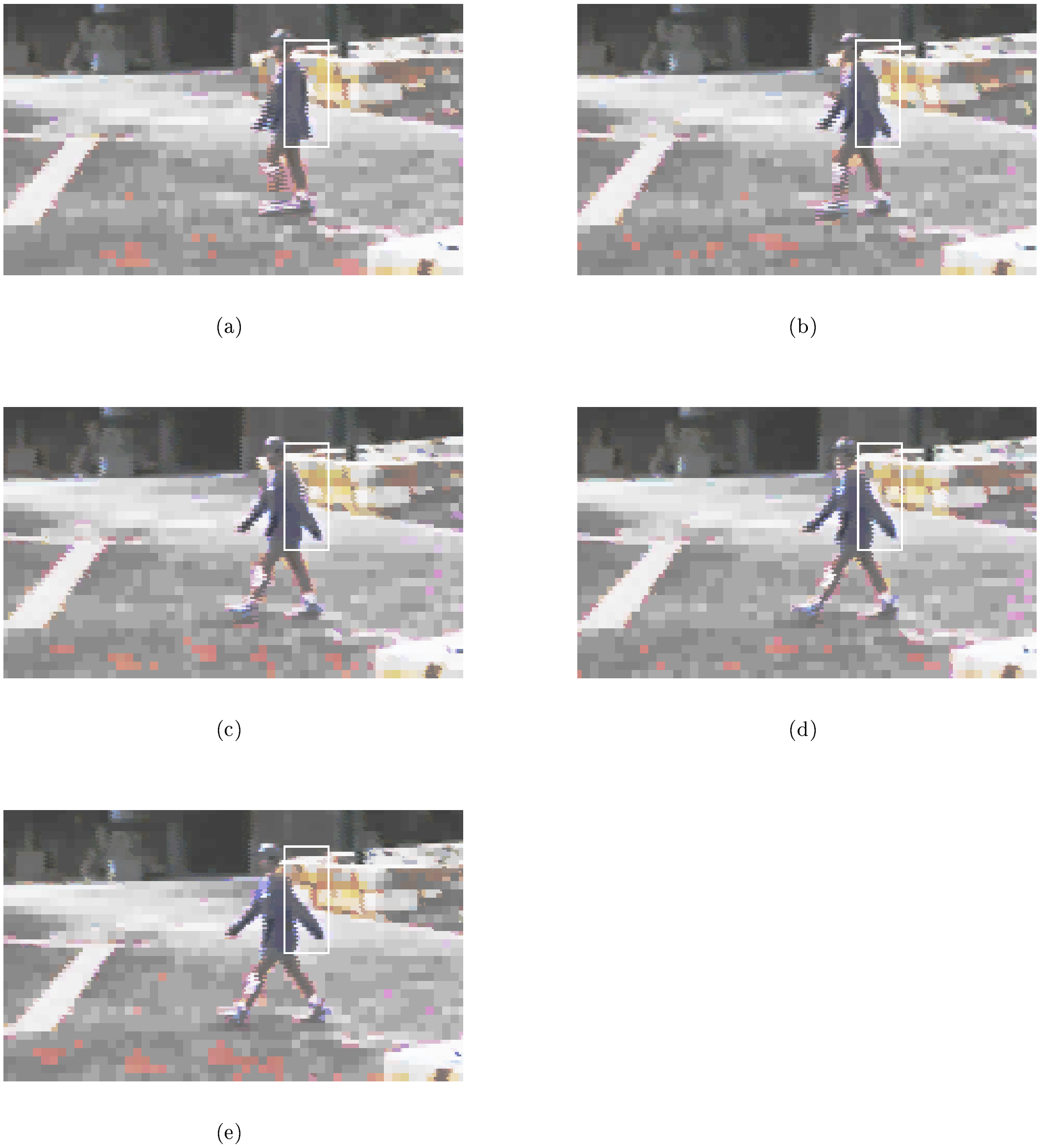
(a)

(b)

(c)

(d)

Fig. 3 – *Snake tracking of sail*



(a)

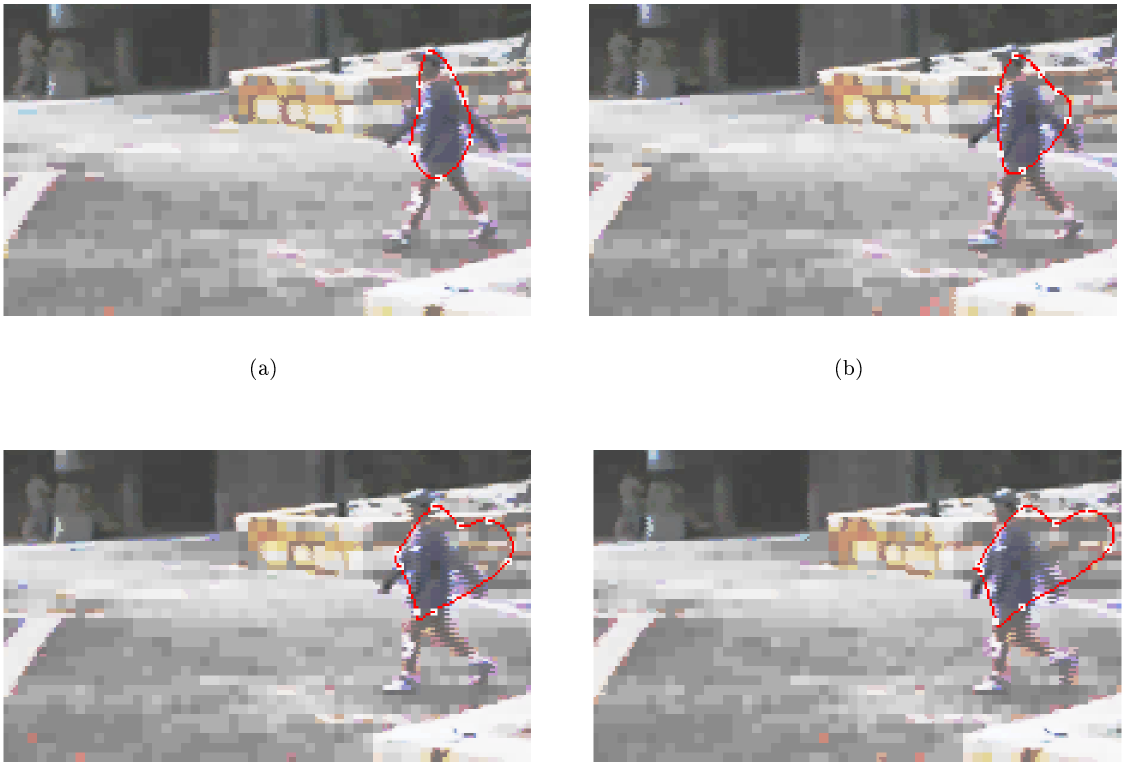
(b)

(c)

(d)

(e)

Fig. 4 – *Correlation tracking alone*



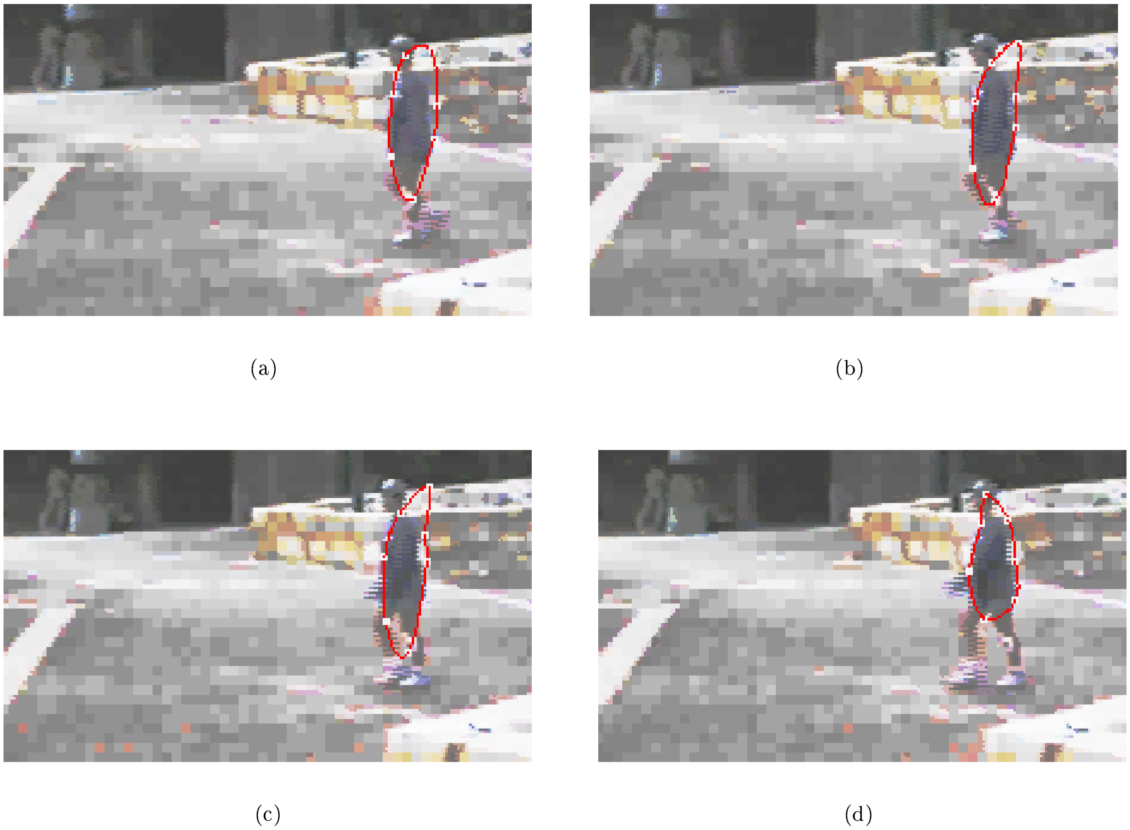
(a)

(b)

(c)

(d)

Fig. 5 – *Adaptive contours tracking alone*



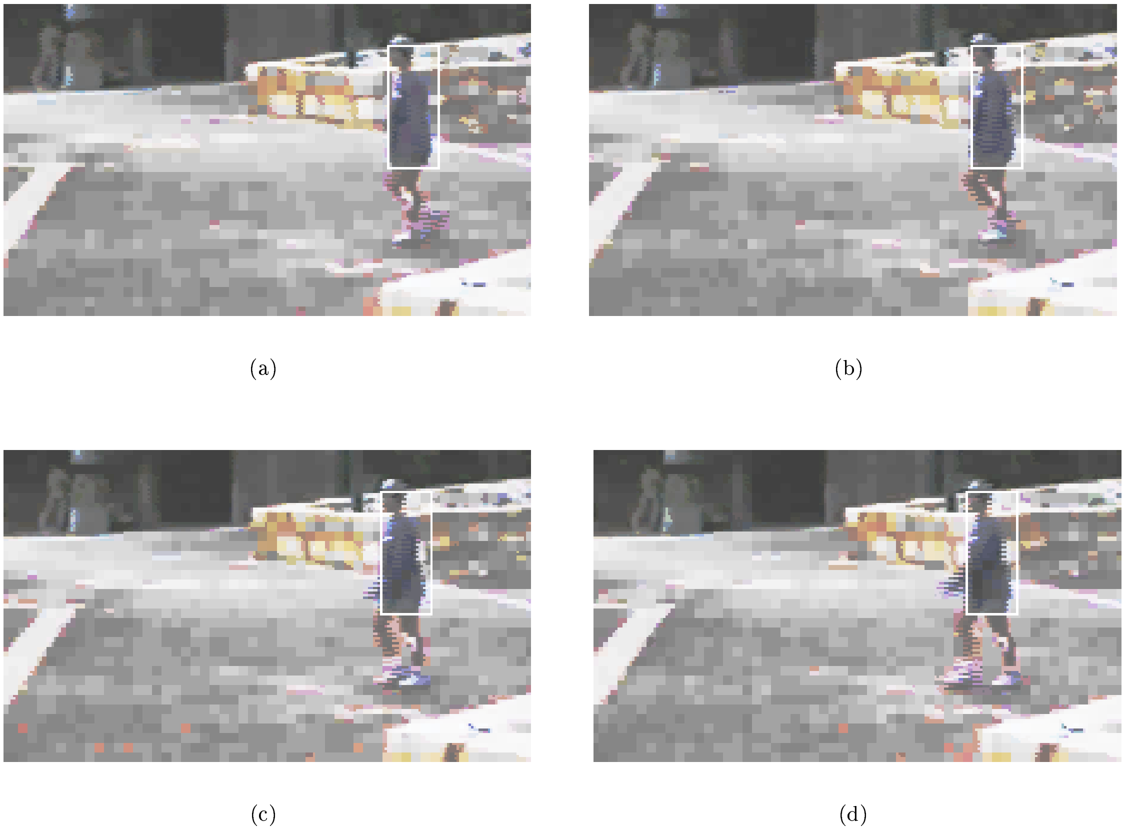
(a)

(b)

(c)

(d)

Fig. 6 – *First correction of adaptive contour, displaying contour*



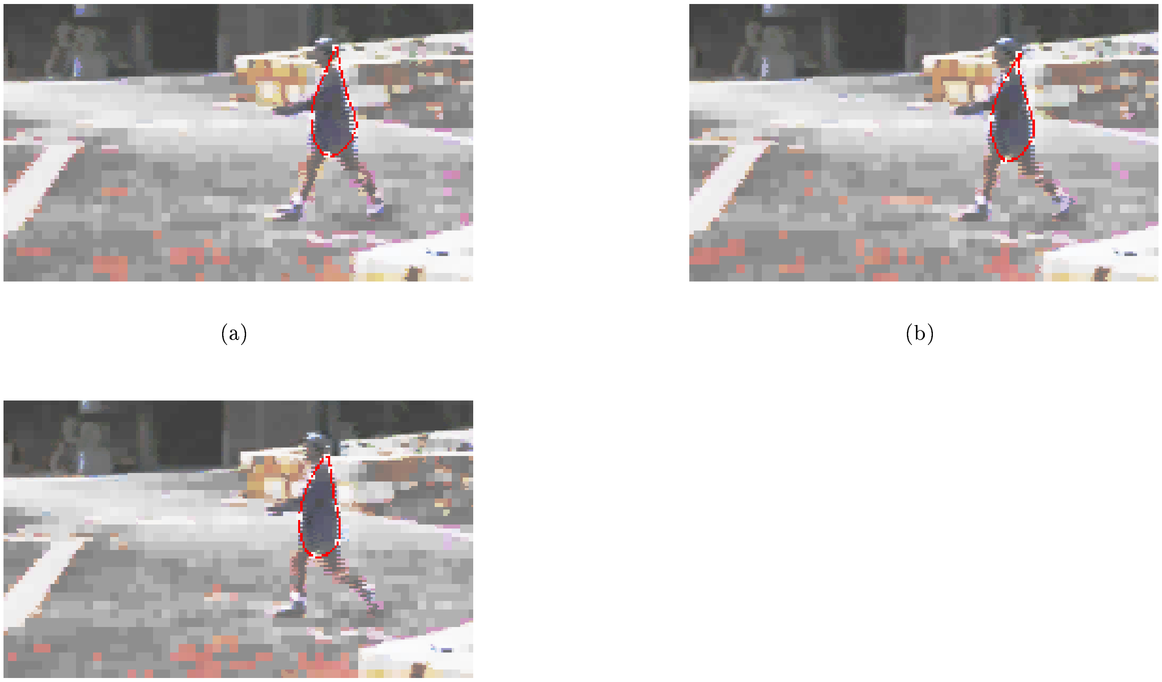
(a)

(b)

(c)

(d)

Fig. 7 – *First correction of adaptive contour, displaying correlation*

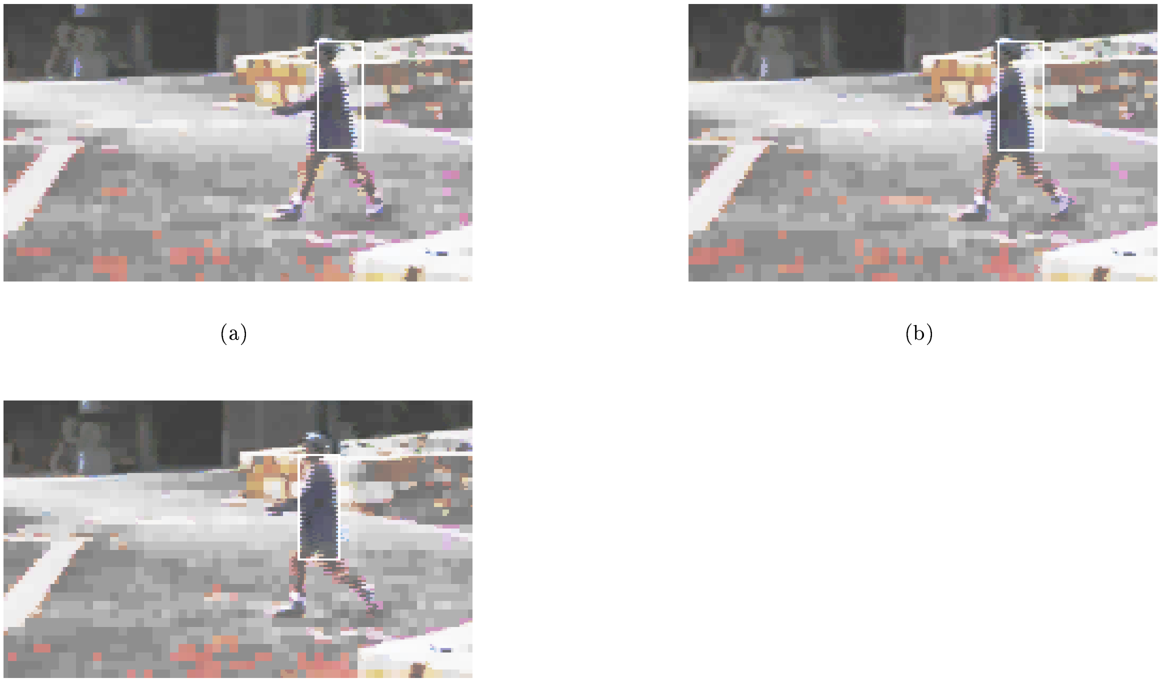


(a)

(b)

(c)

Fig. 8 – *First update of correlation, displaying adaptive contours*

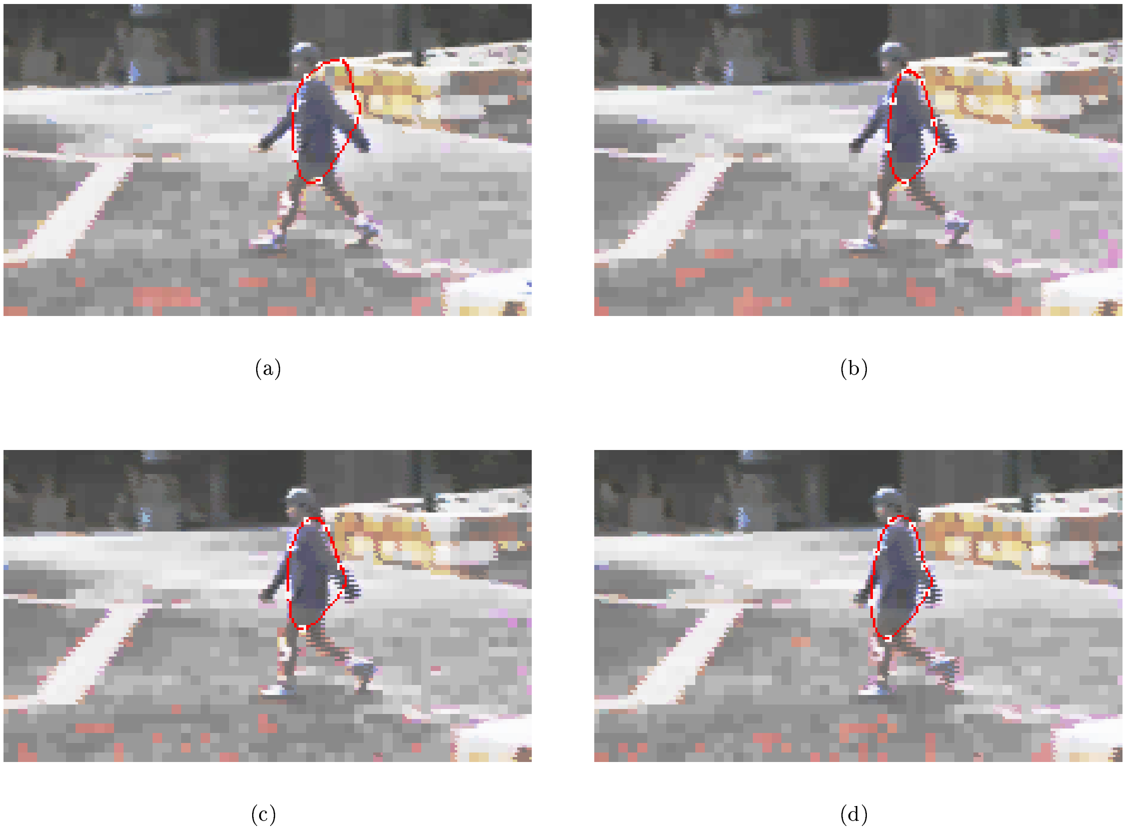


(a)

(b)

(c)

Fig. 9 – *First update of correlation, displaying correlation*



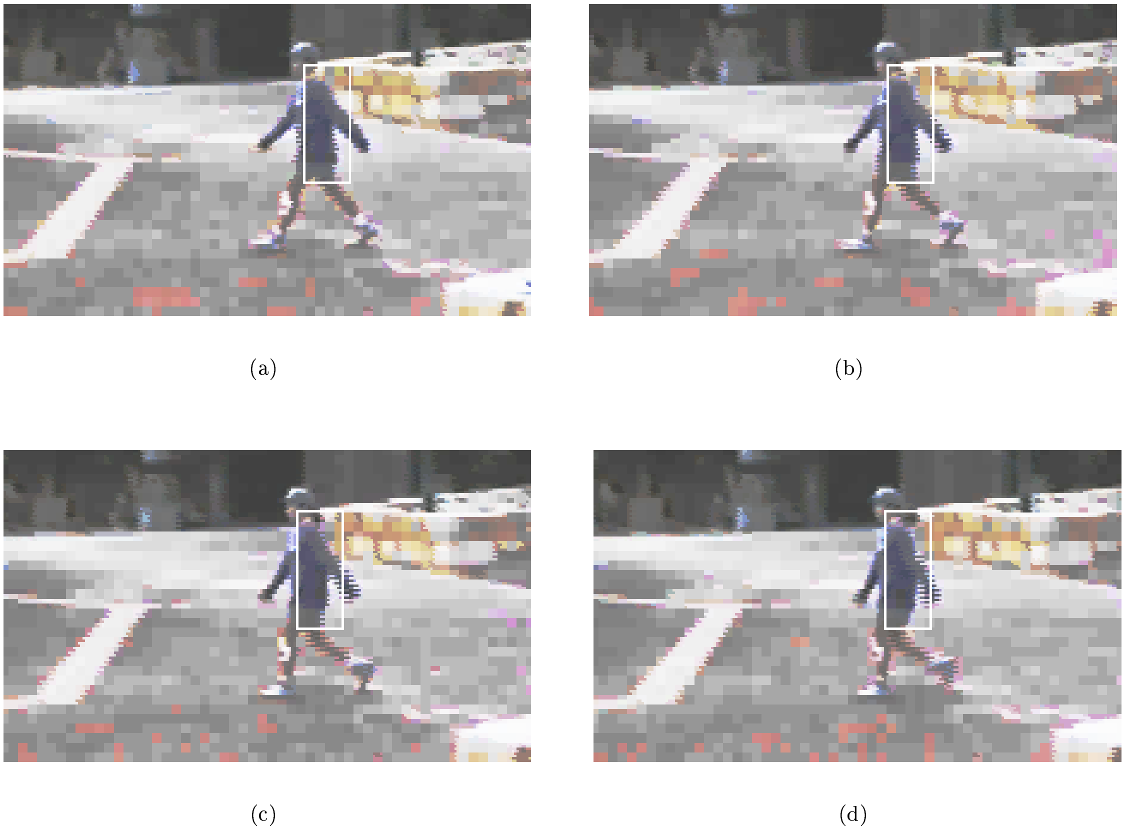
(a)

(b)

(c)

(d)

Fig. 10 – *Second correction of contours, adaptive contours displayed*



(a)

(b)

(c)

(d)

Fig. 11 – *Second correction of contours, correlation displayed*