Real-time Active Visual Tracking with Level Sets

Warakorn Gulyanon, Claire Morand, Neil. M. Robertson and Andrew. M. Wallace

Heriot-Watt University, Edinburgh, UK
{wg35, c.morand, n.m.robertson, a.m.wallace}@hw.ac.uk

Keywords: Visual tracking, level sets, robot head, active vision.

Abstract

This paper presents a new real-time active visual tracker which improves standard mean shift tracking by using level sets to extract contours from the target. We use colour and the disparity map computed from a stereo camera pair which prove to be powerful features for tracking in an indoor surveillance scenario. To combine the features in the level sets process, we enhance Chen's *et al* appearance model of [5] by using a probabilistic model determined via Expectation-Maximization (EM) clustering. The level set result is used as the weighting kernel which improves the accuracy of the similarity measurement in the mean shift method. Finally a Kalman filter deals with complete occlusions.

1 Introduction

Visual tracking is the process of estimating the location of an object of interest over time in an image sequence using single or multiple cameras [3]. Visual tracking becomes one of the essential feature for a machine (e.g. robot) to sense and understand the surrounding environment [1]. The surveillance system should detect the interesting object, track that object in an image sequence and recognize its behaviour [17].

The aim of this work is to track a single person in an in-door environment from a stereo pair of cameras placed on a Pan-Tilt (PT) head attached to a robot. Thus, the method must be designed to be robust to dynamic background and motion blur. To do so, we choose to continue from the work of Chen and Wallace [5] which uses the mean shift tracker with the level sets as an adaptive kernel. The use of active contours solves the principal weakness of the mean shift tracker which results from the tracker's inability to deal with the object's postural change because the kernel is fixed in size and shape. It also improves the performance of the mean shift by allowing to compute the target's model from only the object's features. Our main improvements from [5] consist in 1) adding a feature - the disparity map, 2) changing foreground and background model to a probabilistic model, 3) adding the dynamic choice of a Kalman filter in complement to the mean shift. We show that increasing the complexity of the approach does not affect the capacity of the method to be used in real-time applications while leading to more robust results.

First, the simple use of colours in [5] is not sufficiently discriminative when working with a real in-door environment where the background is not homogeneous. By using the disparity map in addition, the active contour segmentation is made more robust in the presence of a cluttered background. This technique may equally be applied outdoors subject to the range of the disparity map. Second, the use of a probabilistic model (vs a deterministic one) allows to better deal with the active contour evolution. Third, the main drawback of the mean shift is that it cannot deal with significant occlusions. Therefore a Kalman filtering step was added to solve cases of occlusions.

1.1 Related work

In 1988, Kass proposed a model using an energy minimization as a framework which became a well-known classical active contours techniques, called "snake" [9]. This energy function consists of features that describe the object such as edge, line and intensity. It can be interpreted as three forces: internal force, image force and external constraint force. However, because the snake method is based on edge and corner, this basic model is not working well on real images due to high detail, texture, illumination, noise, etc. The minimization of the energy function can easily fall in a local minimum. For these reasons, Chan *et al.* [4] proposed another classical method of active contours which does not depend on edges. In their paper, they proposed a model in which the stopping term of the active contour is based on the segmentation of the image instead of the gradient of the image like in snakes. However, this model is based on a single representing colour for the entire region which obviously is not enough in real images.

Yilmaz and Lankton [18, 10] proposed an energy function that combines the advantages of both the geodesic active contours (e.g. snakes) and the region-base active contours. The former work used the Bayesian framework to derive many features, e.g. colour, shape, gradient, motion, into a probabilistic image of object and background. The segmentation is done by minimizing the energy function using variational approach. The latter approach as-

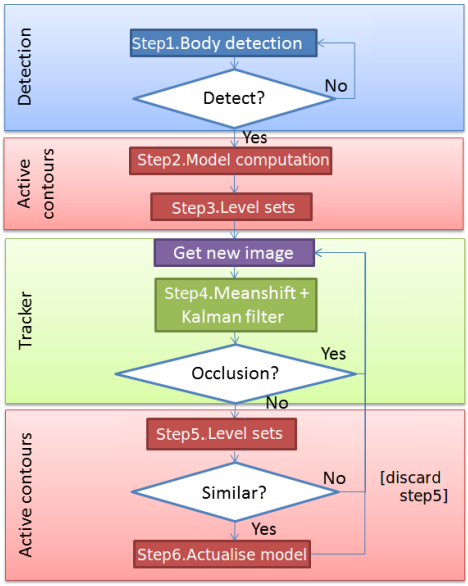


Figure 1: Schematic of the proposed tracking algorithm. Each step is described in the text.

sumes that nearby points inside and outside of the true edge of an object can be well modelled by the mean intensity of the corresponding local region. One drawback is that the initial state has to be near to the true boundary, otherwise the curve can get aligned to another local minimum energy. Moreover, the parameters have to be tuned for each image individually.

Rousson [14] introduced a feature from shape by training from the implicit function ϕ. The training method is called the voxel-wise probabilistic level set formulation where ϕ, resulting from any segmentation method, is kept in the Probability Density Function (PDF) form. However, this method is not guaranteed to succeed when the person can present any side and be in any posture.

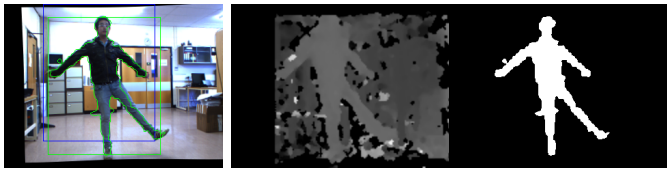
Unger [15] proposed a unique way of object tracking by segmenting the object in 2D+t volume instead of in a single 2D frame. Only the RGB colour histogram is used as a feature. The occlusion is represented as disjunction in the volume. The implementation is not real-time because the size of the volume keeps increasing which affects the performance of the segmentation.

Brox [2] used a feature vector composed of three colour channels, three texture channels and two motion channels. Non-linear diffusion is applied on the features to enhance them before doing the segmentation.

2 Technical approach

The combination of mean shift, Kalman filter, and active contour is implemented in this work in order to track a person using a pan/tilt robot head. The flow of the proposed method is shown in Figure 1.

First, in **steps 1 to 3**, the tracker is initialized. This begins with finding a person in the video stream (**step 1**). (This paper does not focus on this human detection part and so we choose to use the state-of-the-art Viola and Jones object detection algorithm.) Then, the foreground and background models are computed through Expectation-Maximisation (EM) clustering (**step2**). The initial contour of the body is then estimated through the level sets method (**step 3**), the energy function being determined using the previously estimated foreground and background models.



(a) left image (b) disparity map (c) segmentation

Figure 2: The disparity map and its segmented result.

Second, **step 4**, the tracking is performed by a combined mean shift / Kalman filter tracker where the level set output is used as an adaptive kernel for the mean shift.

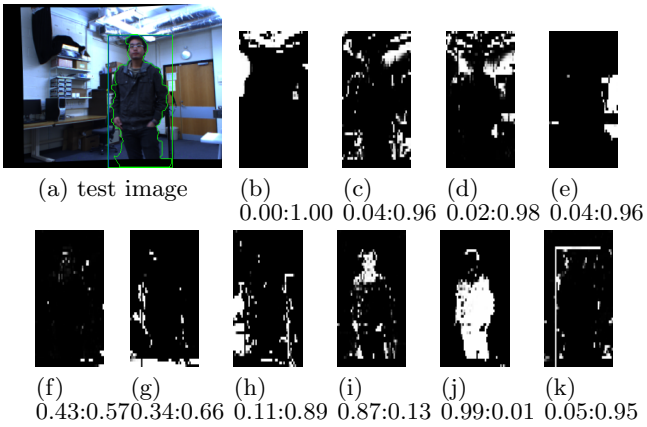
Third, **steps 5 and 6**, the contour of the object is refined using level sets only if no strong occlusion has been detected. The presence of an occlusion is determined by computing the similarity between the kernel estimated in the previous frame and the rectangular kernel found by the mean shift algorithm in the current frame. If the similarity is low, an occlusion is detected. The level set step is skipped and the position of the object is given by the Kalman prediction. The velocity of the object is assumed to be constant and added to the velocity induced by the robotic head movement. Else, if the similarity is high, there is no occlusion and the level set is performed (**step 5**). The result is discarded if the colour histograms of the previous and the new foreground regions are too different, meaning a failure in the level sets process. Otherwise, the foreground and background models are actualised (**step 6**) and the result of the active contour feeds back as the kernel to the mean shift for the next frame.

2.1 Body detection

Step 1 is the initial detection of a human body. This task is carried out to define an initial position and ROI for the tracker. The OpenCV implementation of Viola and Jones is used without any modification which provides both the implementation of the algorithm and the trained body classifier.

2.2 Foreground and background model computation

The aim of **steps 2 and 6** is to determine the probabilistic models of the foreground and the background respectively. Many previous works dealing with this exist [18, 15, 2]. In this paper, our approach is first to segment the previous frame using EM clustering. The features used are three



(a) test image (b) 0.00:1.00 (c) 0.04:0.96 (d) 0.02:0.98 (e) 0.04:0.96

(f) 0.43:0.57 (g) 0.34:0.66 (h) 0.11:0.89 (i) 0.87:0.13 (j) 0.99:0.01 (k) 0.05:0.95

Figure 3: Result of the EM clustering: (a) image segmented (b-k) individual clusters represented in white with their associated scores (foreground : background).

channels of colour (YCbCr) and one channel of disparity map. The disparity map is a virtual depth information estimated by the difference between the left and right images of the same scene. The disparity map function is implemented using the semi-global block matching algorithm [7]. An example of disparity map is shown in Figure 2.

The result of the EM clustering is a set of regions of homogeneous features described by a 4-D gaussian. The probabilistic models of the foreground and background are then defined as a linear combination of those primary gaussians. In the foreground (resp background) model, the Gaussians are weighted using the score defined as the ratio of the area of the foreground (resp background) that belongs to that cluster as shown in Figure 3.

Steps 2 and 8 differ in the way that foreground and background are defined in the previous image. In **step 2** there is no previous result; the result of the body detection (**step 1**) is used. The ellipse region inside the detection result is assumed to be the foreground region, and the surrounding area is assumed to be the background region. The reason of using the ellipse region is that the ellipse better fits the human shape than the rectangle. In **step 6**, the foreground and background regions are known in the previous image from the tracking result.

2.3 Active contour computation

The energy function used in the level sets evolution (**step 3 and 5**) is adopted from [5] and expressed as

$$\begin{aligned} E(C) \quad = \quad & \int\int_{\Omega} \mu \cdot \delta_0(\phi(x,y)) \cdot |\nabla\phi(x,y)| \\ & +\lambda_{fg} F_{fg} H(\phi(x,y)) \\ & +\lambda_{bg} F_{bg} (1 - H(\phi(x,y)) dxdy. \end{aligned} \tag{1}$$

where C is a point on the boundary, F_{fg} and F_{bg} are the models of a foreground and a background computed by EM clustering (see pararaph 2.2). F_{fg} should have a lower value for a foreground pixel and a higher value for a background pixel, and vice-versa for F_{bg}. μ ,λ_{fg}, λ_{bg} are normalizing terms. As in the guideline [12], the Euler-Lagrange equation is used to minimize the energy of the velocity function with respect to ϕ [5]:

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \cdot \nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_{fg} F_{fg} + \lambda_{bg} F_{bg} \right] \tag{2}$$

The Dirac delta function δ_ϵ controls the movement only for the pixels around the contours, $\frac{\nabla\phi}{|\nabla\phi|}$ is the direction normal to the contour, so $\nabla\frac{\nabla\phi}{|\nabla\phi|}$ is the divergence of the normal (i.e. the curvature) and controls the smoothness of the curve. Equation 2 is similar to the velocity function used in active contours based on colour (as in [4]) due to the similar approach taken to solve the Euler-Lagrange equation.

The active contour's tasks are 1) an input kernel of the mean shift, 2) a tracker itself. The similarity of the histograms between the tracking kernel in the previous frame and the result of the active contour in the current frame is checked in order to exclude false segmentations. The result is discarded in the case of low similarity. Otherwise, the ROI's window size is adjusted according to the boundary of the contours. The position of the ROI is also moved to the centre of the contours. Finally, the histogram of the tracking object is updated by using the newly obtained kernel.

2.4 Mean shift and Kalman filtering

In **step 4**, the mean shift tracker acts as the backbone of the application while the active contour is optional (its result can be discarded when a strong occlusion occurs). [6] proposed a mean shift tracker using the colour histogram as a representation of the target. The most similar target's candidate in the next image sequence has the smallest Bhattacharyya coefficient. To estimate the optimum position of the target, the Taylor expansion of Bhattacharyya coefficient is computed and the term involving y_k minimized. Therefore, in each iteration, the new target's position from y_0 to y_1 is estimated as

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i^* w_i g(||\frac{y_0 - x_i^*}{h}||^2)}{\sum_{i=1}^{n_h} w_i g(||\frac{y_0 - x_i^*}{h}||^2)}, \tag{3}$$

where $g(x) = k'(x)$,

$$k = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1 - ||x||^2) & , \text{if} ||x|| < 1 \\ 0 & , \text{otherwise} \end{cases}, \tag{4}$$

$$w_i = \sum_{u=1}^{m} \delta(b_i(x) - u) \sqrt{\frac{q_u}{p_u(y_0)}}., \tag{5}$$

where q_u, p_u are the tracked object and the target's histogram. The algorithm iterates until convergence

$$||y_1 - y_0|| < \epsilon. \tag{6}$$

The OpenCV mean shift function is used in this implementation where the back-projection has to be computed first before feeding as the input of mean shift function. The back-projection is basically the likelihood of each pixel to belong to the object according to the probability distribution given by the histogram. The back-projection acts as the weight (Eq. 5) in the mean shift algorithm. In addition, the back-projection can be masked in any shape, which limits the search area for the mean shift. This search area is estimated using the prediction step of the Kalman filter. Then, the position of the object is determined by the mean shift tracker. Afterwards, this found position is used as measurement in the Kalman filter method. The robot head is also moved to keep the center of the object in the center of the image.

Note that the test for occlusion (between **step 4** and **step 5**) is done thanks to the similarity measurement in the mean shift. To combine with level sets, the similarity should be superior to a quite high threshold (e.g. ≥ 0.5). In case the similarity is lower than that threshold, an occlusion is assumed to occur.

There are many approaches to define components in Kalman filter. For example, [11, 19, 13] defined the state as the position and the velocity (x, y, v_x, v_y), or [16] also added the acceleration $(x, y, v_x, v_y, a_x, a_y)$. This paper follows the idea of [8] where the state X contains only the position (x, y):

$$X_n = [x_n, y_n, 1]^T. \tag{7}$$

The adaptive velocity is dependent on the filter state but still affects the predicted position by embedding it in the transition matrix D defined as

$$D_{n+1,n} = \begin{bmatrix} 1 & 0 & dx_{n+1,n} \\ 0 & 1 & dy_{n+1,n} \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where $dx_{n+1,n}$ and $dy_{n+1,n}$ are the velocities or translations in x and y directions. The velocities are adaptive. The learning rate depends on the similarity measurement and can be any decreasing function. For this implementation, the learning rate is $e^{-5 \cdot \rho}$, where ρ is the Bhattacharyya distance. The measurement Z obtained from the mean shift tracker is a position (x,y), therefore, the measurement matrix is simply written as

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{9}$$

Both prediction noise covariance and measurement noise covariance are

$$Q = R = \begin{bmatrix} h_x & 0 & 0 \\ 0 & h_y & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{10}$$

where h_x and h_y are the width and the height of the kernel. With the help of the Kalman filter, the tracker can deal with some occlusion cases. Because Kalman filter is a single-hypothesis method [1], the tracker might track the occluding object if it is similar to the tracked object.

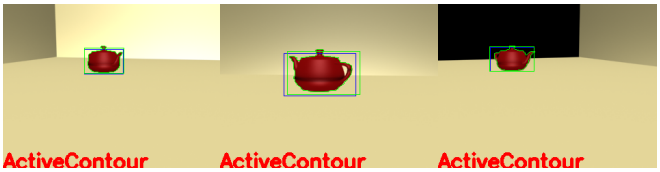


Figure 4: Synthetic sequence: no occlusion, static camera.

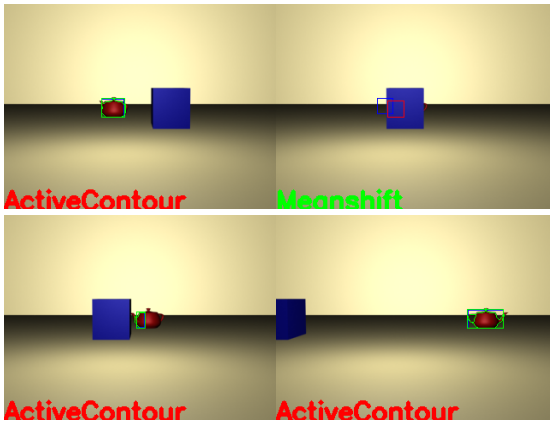


Figure 5: Synthetic sequence: occlusion, static camera.

3 Results

We first discuss the speed performance of the algorithm then present evaluation of accuracy. The algorithm was implemented on a computer with Intel Core 2 Extreme QX9650 3000 MHz and 3.25GB RAM. The robot head is composed of twos Point Grey Flea2 cameras with FUJINON DF6HA-1B external lens and attached on a Biclops PT-M Pan-Tilt head. The input camera resolution is 640 by 480 pixels. The resulting frame rate is 5 fps with the image size reduced to 50 x 50 in the active contour step. This scale was chosen empirically by measurements on synthetic sequences. The resulting contours were scaled back to the original size. The algorithm performs in real-time: the PT head is successfully instructed to move so that the person of interest keeps being in the centre of the cameras images at all time.

3.1 Evaluation on synthetic sequences

Three synthesised sequences are presented to illustrate the detail of the proposed algorithm. These sequences, with their associated depth map and ground truth, have been created in 3Ds Max.

In the first sequence (Fig. 4), there is no occlusion. The active contour is re-computed each time and the similarity model is consistent in time. In the second sequence (Fig. 5), an occlusion occurs. In this case, the result of the combined mean shift / Kalman filter tracker is used to predict the position of the object but no active contours is computed as no information is available. The same is illustrated in the third sequence (Fig. 6). Here, a camera motion occurs and is compensated in the mean shift step.

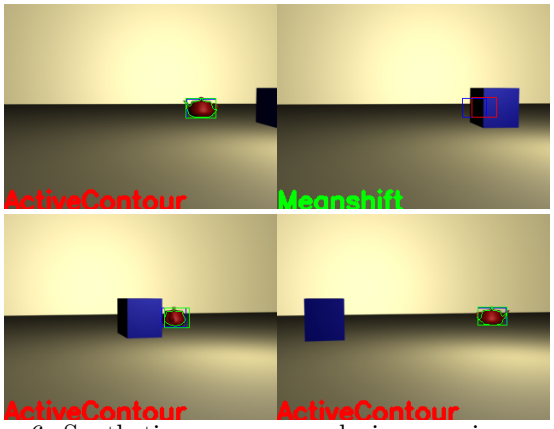


Figure 6: Synthetic sequence: occlusion, moving camera.

Sequence	Mean Sensitivity	Mean Square Error
1. Fig. 4	97.84%	0.17%
2. Fig. 5	97.81%	0.15%
3. Fig. 6	94.53%	0.12%
4. Fig. 7	96.85%	3.15%

Table 1: Average sensitivity and mean square error by sequences

In Figures 4 to 6, the caption on each image gives the end state of the algorithm which normally is the active contour. However, during occlusion, the active contour step is skipped and the tracked position relies on the meanshift estimation. On each image is displayed: (*Green rectangle*) the estimated position from Kalman filter with measurement, (*Red rectangle)* the predicted position from Kalman filter without the correction step (i.e. during an occlusion), (*Blue rectangle*) the result of the mean shift, (*Green contours*) the accepted contour, and (*Red contours*) the discarded contour.

Table 1 gives the average of sensitivity and mean square error on every frames of a sequence.

3.2 Evaluation on real data

In addition, the results on the real sequence showed in Fig. 7 was evaluated against a manually created ground truth. The average sensitivity and mean square error is also given in table 1.

The performance of the mean shift is improved by using the active contour's result as the histogram of the tracking object excludes any background region. As a result, the similarity measurement and the histogram comparison can be computed efficiently. The mean shift with Kalman filter cloud deals with occlusions, while the active contour improves the occlusion detection by improving the similarity measurement accuracy. However, the Kalman filter tracker might not work in every cases of occlusion, especially when the occluded object is similar to the tracked object. This algorithm seems to work only in


Figure 7: Test sequence in real indoor environement

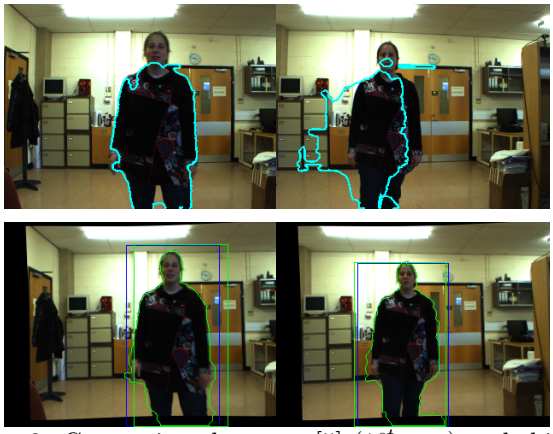
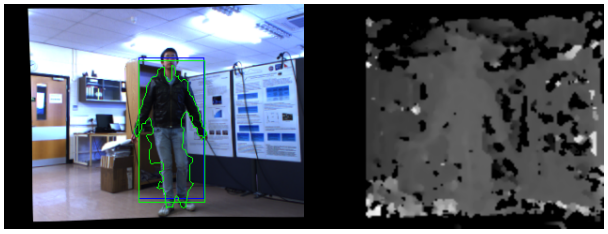


Figure 8: Comparison between [5] (1^{st} row) and this paper's result (2^{nd} row).

the case when the duration of an occlusion is relatively short and the robot displacement is not too large because the real object velocity is not constant like in synthesis images. As the velocity is assumed to be constant during occlusions, the algorithm may fail if the object changes direction or speed.

Comparing to [5], the segmentation in an in-door environment is now possible by adding a disparity map (Fig. 8). However, it also introduces more noise around the boundaries of the object because the disparity map depends on the matching between left and right images and the matching algorithm is still far from perfect as shown in figure 9. This problem could be solved by using sensors which measure the depth directly (e.g. PMD).

This algorithm has to be improved to support the apparition of a new colour on the tracked object. This new colour will be classified to be an object or a background region depending on its similarity to either region. For example, this problem occurs when the diffusion light projects a bright colour on the object surface (Fig. 10) which becomes a part of background because the background is composed of a bright object (the white wall).



(a) The noisy result. (b) The disparity map.

Figure 9: Inaccurate boundary due to the disparity map

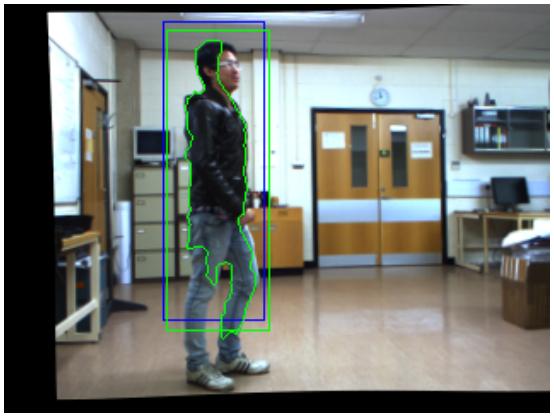


Figure 10: New colour appears on the object.

4 Conclusions and Future Work

This paper presents a new dynamic tracker that combines mean shift, the Kalman filter and active contours. The aim is to track a person from a pan-tilt head and extract the person boundary (silhouette) that may further be used in a gesture recognition system. A feature, the disparity map, is added to the colour space to work in a 4D space. The use of this feature improves the segmenation between foreground and background in a cluttered in-door environment. However, it also introduces noise on the boundaries of the object in the segmentation result. For future work, more suitable features should be searched in order to improve the accuracy on the border of the object tracked. For the tracker part, the Kalman filter has been added in order to be tolerant to occlusions. Further optimization will aim to reduce computation time to allow more smooth operation of the PT head. The present work presents results for tracking people, however, no *a priori* restriction on the tracked object is made by the method: other rigid or deformable objects could easily be tracked, given a suitable detection algorithm.

References

[1] A.Cavallaro and E.Maggio. Video tracking theory and practice. 2010.

[2] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Colour, texture, and motion in level set based segmentation and tracking. *Image and Vision Computing*, 28(3):376–390, 2010.

[3] K. Cannons. A review of visual tracking. Technical report, Technical Report CSE-2008-07, York University, Department of Computer Science and Engineering, 2008.

[4] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[5] Z. Chen and A.M. Wallace. Active segmentation and adaptive tracking using level sets. In *Proc. of British Machine Vision Conference*, pages 920–929, 2007.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *cvpr*, page 2142. Published by the IEEE Computer Society, 2000.

[7] H. Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2386–2393. IEEE, 2006.

[8] V. Karavasilis, C. Nikou, and A. Likas. Visual Tracking by Adaptive Kalman Filtering and Mean Shift. *Artificial Intelligence: Theories, Models and Applications*, pages 153–162, 2010.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[10] S. Lankton, D. Nain, A. Yezzi, and A. Tannenbaum. Hybrid geodesic region-based curve evolutions for image segmentation. In *Proc. of SPIE Vol*, volume 6510, pages 65104U–1. Citeseer.

[11] H. Lu, R. Zhang, and Y.W. Chen. Head detection and tracking by mean-shift and kalman filter. *icicic*, page 357, 1899.

[12] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer Verlag, 2003.

[13] N.S. Peng, J. Yang, and Z. Liu. Mean shift blob tracking with kernel histogram filtering and hypothesis testing. *Pattern Recognition Letters*, 26(5):605–614, 2005.

[14] M. Rousson and N. Paragios. Prior knowledge, level set representations & visual grouping. *International Journal of Computer Vision*, 76(3):231–243, 2008.

[15] M. Unger, T. Mauthner, T. Pock, and H. Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted TV. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 193–206. Springer, 2009.

[16] X. Yang, H. Li, and X. Zhou. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(11):2405–2414, 2006.

[17] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm Computing Surveys (CSUR)*, 38(4):13, 2006.

[18] A. Yilmaz, X. Li, and M. Shah. Object contour tracking using level sets. In *Asian Conference on Computer Vision*. Citeseer, 2004.

[19] Z. Zhu, Q. Ji, K. Fujimura, and K. Lee. Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. *Pattern Recognition*, 4:40318, 2002.