Proceedings of 2012 IEEE
International Conference on Mechatronics and Automation
August 5 - 8, Chengdu, China

Object Tracking with Serious Occlusion Based on Occluder Modeling

Peng Wang, Wanyi Li, Wenjun Zhu and Hong Qiao
Research Center of Precision Sensing and Control
Institute of Automation, Chinese Academy of Sciences
Beijing 100190, China
E-mail: peng_wang@ia.ac.cn

***Abstract* – Occlusion is one of the challenging problems in object tracking, and plenty of tracking methods have been proposed to cope with this issue. Most of the methods deal with occlusion relying on observational or prior information of the tracked objects, such as appearance, shapes and motion. However, during occlusion especially serious and long-time occlusion, observations of object are hard to obtain, and prior knowledge, such as motion attributes, changes gradually over time. Therefore, modeling the object motion and then predicting the object's location until the object reappears, is likely to fail to serious and long-time occlusion. To cope with this problem, this paper proposes a novel method for object tracking with serious and long-time occlusion in image sequences based on occluder modeling. Occluder is modeled by detecting and evolving its rough partial contour represented by snake points, through minimizing the proposed energy function in which two novel terms are introduced: the push force and constraint force. Then, we search the tracked object around the neighborhood of the occluder contour until the object reappears. Experimental results demonstrate the effective performance of the proposed method on real sequences with total and long-time occlusions.**

Index Terms* –Object tracking, serious occlusion, occluder modeling, active contour.

I. INTRODUCTION

Over the past several years, object tracking has been one of the most attractive topics in computer vision area. This interest is motivated by numerous applications, such as robot visual navigation, man-machine interfaces, surveillance, safety control, video conferencing, and virtual reality.

However, occlusion is a challenging problem in the object tracking process. It can cause loss of object in many object tracking algorithms, especially when serious and long-time occlusion happens. Occlusion can be classified into three categories: self-occlusion, inter-object occlusion, and occlusion by the background scene structure [1]. In this paper, we mainly aim at resolving the last two kinds of occlusion.

Methods to solve the occlusion problem have been previously presented. One of the most important approaches is merge-split approach [2-5], in which the object blobs are characterized by different operations, such as creating, deleting, merging and splitting. Elgammal [6], Ying [7] and Min [8] handle occlusion by modeling the occlusion process explicitly, such as the object models and the occlusion relation. Also some methods handle occlusions by enforcing shape constraints [9-11]. These methods usually resolve occlusion by using shape priors which are either built ahead of time [10] or built online [11]. H. Tao et al. [12] introduce a complete dynamic motion layer representation in which spatial and temporal constraints are modeled and estimated. Gentile et al. [13] address occlusion by finding a suitable set of parts. They propose a novel segmentation method using a cost function that exhibits a high degree of correlation with the tracking error. Nguyen et al. [14] smooth the appearance features of object temporally when updating the template. The resistance of the template to partial occlusion enables the accurate detection and handling of more severe occlusion. P. Wang and H. Qiao [22] propose an adaptive tracking method based on object model learning and generation, and the method can well deal with partial and short-time serious occlusion. The abovementioned methods can also be divided into two major groups [15]: merge-split approaches [2-5], and straight-through approaches [6-14]. The straight-through approaches simply continue to track the individual blobs throughout the occlusion without merging them.

Most of the methods mentioned above deal with occlusion relying on observational or prior information of the tracked objects, such as appearance, shapes and motion. However, during occlusion especially serious and long time occlusion, observations of object are hard to obtain, and prior information, such as motion attributes, changes gradually over time. Modeling the object motion by linear dynamic models [16] or by nonlinear dynamics [17], and then keeping on predicting the object's location until the object reappears, is likely to fail to serious and long-time occlusion.

In this paper, we propose a novel approach to solve the serious and long-time occlusion problem. We aim at not only modeling the object, but also modeling the occluder which occludes the tracked object using an improved snake algorithm. Once occlusion happens, we model the occluder by detecting and evolving its rough partial contour, which is represented by snake points, through minimizing the proposed energy function in which we introduce two novel terms: the push force and constraint force. Then, we search the object around the neighbourhood of the occluder contour until the object reappears. The object is modeled using colour feature in both RGB and YCbCr colour space, and tracked using mean-shift algorithm.

The paper is organized as follows: In section II, we give the framework of the proposed method. In section III, we

978-1-4673-1278-3/12/$31.00 ©2012 IEEE

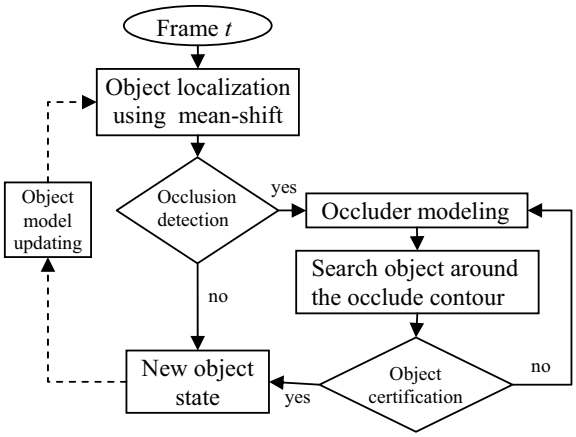


Fig. 1 Framework of the proposed method

describe a snake architecture employed to detecting and evolving the occluder contour, and introduce two novel energy terms. Experimental results and conclusion are sketched in section IV and V, respectively.

II. Framework of the Proposed Method

A. Tracking features

The major idea of the presented approach is to keep track an object appearance model defined as the collection of photometric feature vectors for pixels inside the object region. The components of the feature vector may be the RGB intensity values or other chromaticity values which are independent with intensity [5]. In our approach, we use both of these two kinds of features, in order to make a good tracker which is insensitive to illumination changes and avoid inaccurate detection of occlusion.

The reference object model is presented by its pdf (probability density function) $\bar{q}$ in the RGB colour space, and pdf $\bar{\hat{q}}$ in the CbCr space which is the subspace of the YCbCr colour space, using Epanechnikov kernel. In the subsequence frame, the object candidate at location y is characterized by the pdf $\bar{p}(y)$ and pdf $\bar{\hat{p}}(y)$ also using the same kernel. Then the likelihood between object and its candidate can be denoted by Bhattacharyya coefficient

$$\rho(y) \equiv \rho[\bar{p}(y), \bar{q}] = \sum_{u=1}^{N} \sqrt{p_u(y) q_u} \tag{1}$$

where N is the number of quantized bins in the RGB colour space. Using the same method, we can obtain $\hat{\rho}(y)$ in the CbCr space.

In the YCbCr colour space, Y is the luminance component and Cb and Cr are the chrominance components. Therefore, it is assumed that the significant intensity changes could be caused by lighting changes, if there is no significant chromaticity change.

B. Framework of the proposed tracking method

Fig. 1 shows the framework of the proposed tracking method. We use mean-shift algorithm [18] to track the object, because it is robust to changes in the object shape and rotation, as well as partial occlusion, and its calculation efficiency is higher. For every frame, we first use mean-shift algorithm to compute the object's new location, and then detect the occlusion. If serious occlusion happens, we model the occluder, and search the object around the modeled occlude contour until the object is re-captured and validated. Then the new object state and object model are updated. The details of occlusion detection and handling are presented in the following section.

III. Occlusion Handling Based on Occluder Modeling

During occlusion, especially serious and long-time occlusion, visual features of the occluded objects are not observed and the objects are likely to be lost. Most approaches focus on how to model the tracked objects, or how to predict the object's position and motion parameters in order to resister occlusion. However, they are usually not able to work well especially when serious and long-time occlusion occurs. Therefore, in this paper, unlike the existing occlusion handling approaches, we focus on how to model the occluder, the things that occlude the tracked object, and once the attributes of occlude such as the boundary of occluder are known, serious and long-time occlusion handling becomes easier.

A. Occlusion detection

Occlusion can be detected based on the change of likelihood function given in (1). But because both occlusion and illumination changes can cause low likelihood in image sequences, we use another likelihood function $\hat{\rho}(y)$ in the CbCr space to distinguish them.

Based on the fact that when illumination changes, the pixel values in the whole object appearance usually also changes, but occlusion always happens from one side of the object. Therefore pixel values at occluded side usually changes much more than the other side, especially in the moving direction. Therefore, we divide the object into multiple blocks such as three blocks in moving direction of object. Then the likelihoods of the divided blocks can be defined by $\rho_\alpha, \alpha \in \{A, B, C\}$ in RGB space, and $\hat{\rho}_\alpha$ in CbCr space, where *A, B* and *C* denote the divided three blocks. Then occlusion can be detected by

$$Occ = \begin{cases} 1 & \rho < L\_th \text{ and } \hat{\rho} < L\_th \text{ and } \rho_B < \rho_C \\ 0 & \rho \geq L\_th \text{ or } \hat{\rho} > L\_th \\ -1 & \rho < L\_th \text{ and } \hat{\rho} < L\_th \text{ and } \rho_B < \rho_C \end{cases} \tag{2}$$

where $L\_th \in (0,1)$ is the threshold value of likelihood, and Occ denotes the occlusion state. If $\rho < L\_th$ and $\hat{\rho} < L\_th$,

it means that likelihood becomes lower and it is not caused by illumination variation, and then occlusion happens.

B. Occluder modeling by detecting its contour

1) Snake description. Active contour or snake framework, which was introduced by Kass et al. [19], is introduced to seek the boundary of the occluder. A snake is an ordered set of points $S = [s_1, s_2, \cdots, s_n]$, and $s_i = (x_i, y_i)$, $i \in \{1, n\}$. A tight contour enclosing the occluder can be obtained by minimizing an energy function

$$E = \sum_{i=1}^{n} E_{cont}(s_i) + E_{curv}(s_i) + E_{image}(s_i) \qquad (3)$$

where the first and second terms prevent gaps and rapid bending on the snake, and they are called internal force. E_{image} signifies the energy based on the image observation, and attracts the snake to the salient image features.

E_{cont} is a first-order continuity term which will be with larger values where there is a gap on the snake, and E_{curv} is a second-order term which imposes smoothness constraints to avoid oscillations of the contours. They are defined as follows

$$\begin{aligned} E_{cont}(s_i) &= \left| d - \left| s_i - s_{i-1} \right| \right| \\ &= \left| d - (x_i - x_{i-1})^2 - (y_i - y_{i-1})^2 \right| \end{aligned} \qquad (4)$$

$$\begin{aligned} E_{curv}(s_i) &= \frac{\vec{\mu}_i \cdot \vec{\mu}_{i+1}}{|\vec{\mu}_i| \cdot |\vec{\mu}_{i+1}|} \\ &= \frac{(x_i - x_{i-1})(x_{i+1} - x_i) + (y_i - y_{i-1})(y_{i+1} - y_i)}{([(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2][(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2])^{1/2}} \end{aligned} \qquad (5)$$

where $\left| s_i - s_{i-1} \right|$ represents the distance between s_i and s_{i-1}, $d = \frac{1}{n} \sum_{i=1}^{n} \left| s_i - s_{i-1} \right|$ is the mean distance of all snake points, and $\vec{\mu}$ is the direction vector between two adjacent points.

In practice, E_{image} is commonly defined in terms of the image gradient [19]. If we set $E_{image} = -\left| \nabla I(x, y)^2 \right|$, then the snake is attracted to contours with large image gradients. Here, E_{image} is approximated as

$$\begin{aligned} E_{image}(s_i) &= -(\left| grad_x(s_i) \right|^2 + \left| grad_y(s_i) \right|^2) \\ &= -(\left| I(x_{i-1}, y_i) - I(x_{i+1}, y_i) \right|^2 + \left| I(x_i, y_{i-1}) - I(x_i, y_{i+1}) \right|^2) \end{aligned} \qquad (6)$$

where $grad_x(s_i)$ and $grad_y(s_i)$ denote the gradients in x and y directions respectively, and $I(x_{i-1}, y_i)$ denotes the pixel intensity in point (x_{i-1}, y_i).

Because of the existing of image noises and the possibility of convergence to a local minimum, we approximate $grad_x$ or $grad_y$ as 0, if their absolute values are smaller than a threshold L_grad.

2) Proposed energy function. When the snake points defined in (3) are not initialized close enough to contours, they cannot be attracted to the real contour. To solve this problem, we define a new snake model by adding an external force like the balloon force in [20], and the snake behaves like being pushed by an external force. When the snake passes by edges, it will be attracted on the edges if the edges are salient, otherwise it will pass through the edges due to the edges are too weak. This avoids the snake converging to a local minimum, and makes the result much more insensitive to the initial condition. If the occlusion happens in the X-coordinate direction, then the proposed external force can be defined as

$$E_{push}(s_i) = \text{-}Occ \cdot (x_i - x_{i0})^2 \qquad (7)$$

where Occ is the occlusion state defined in (2), which controls the direction of the push force, and x_{i0} is the initial X-coordinate of s_i.

Snakes usually tightly enclose the whole contours of the objects in most literatures. Here, snake is used just for seeking the partial boundary of the occluder where the occluded object will reappear, so we only need some open contours formed by snakes. We impose constraints to the first and last snake points, and let them move along the required trajectories. Then the proposed energy term can be defined as

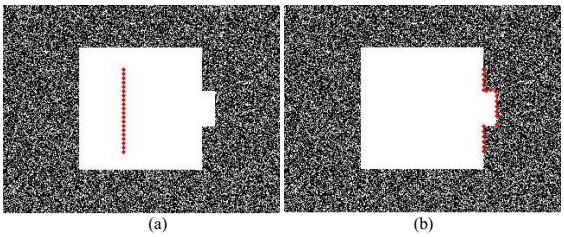
$$E_{con}(s_i) = \begin{cases} +\infty & \text{if } f(x_i, y_i) \neq 0, \text{and } i = 0 \text{ or } n \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

where $i \in \{1, n\}$, $f(x, y) = 0$ is the required trajectory function of the first and last snake points, such as $x_i = x_{i0}$, i.e., $E_{con}(s_i)$ is designed for the first and last snake points, and for the other snake points, $E_{con}(s_i) = 0$.

Based on equations (3), (7) and (8), the proposed energy function can be defined as

$$\begin{aligned} E = \sum_{i=1}^{n} & \alpha(s_i) E_{cont}(s_i) + \beta(s_i) E_{curv}(s_i) + \\ & \gamma(s_i) E_{image}(s_i) + \eta(s_i) E_{push}(s_i) + \xi(s_i) E_{con}(s_i) \end{aligned} \qquad (9)$$

where $\alpha(s_i), \beta(s_i), \gamma(s_i), \eta(s_i)$ and $\xi(s_i)$ control the relative influence of the corresponding terms. In experiments, we simplify them as constants by experience.



(a) (b)

Fig. 2 (a) The initial snake points (b) contour detecting result in one direction

3) Snake initialization and energy minimization. When occlusion is detected, let (X_0, Y_0) denotes the centre of the tracked object, and $S = [(x_{10}, y_{10}), \cdots, (x_{no}, y_{no})]$ denote the initial positions of the snake points. Then the snake can be initialized by

$$\begin{cases} x_{i0} = X_0 + Occ \cdot w / 2 \\ y_{i0} = Y_0 - h / 2 + h \cdot i / n \end{cases} \tag{10}$$

where Occ is defined in (2); w and h are the width and the height of the tracked object bounding box respectively, $i \in \{1, n\}$.

For each snake point, m neighbourhood points are used, and then the minimum energy of the snake points can obtained iteratively using greedy algorithm introduced by [21], which is fast, having computational complexity $O(nm)$. We normalize each energy term in (9) firstly by

$$\hat{E}_\phi(s) = (\min - E_\phi(s)) / (\max - \min) \tag{11}$$

where $\phi \in \{cont, curv, image, push, con\}$; max and min are the maximum and minimum values of $E_\phi(s)$ in each neighbourhood of point s respectively.

For each snake, we search its new location iteratively until the number of moving points is smaller than a threshold. In order to reduce the computational complexity, snake points which form the rough contour of occluder are as few as possible. Fig. 2 shows the initial snake points and contour detecting result in one direction in a synthetic image with random noise.

C. Evolving the contour

After detecting the rough contour of occluder, for stationary camera and motionless occluder, the position of the contour is unaltered during occlusion. When the camera or the occluder are moving over time, the occluder contour is likely to change over time during occlusion, so we need to update the contour from frame to frame. The contour will not change drastically between two continuous frames, so we evolve the contour by minimizing the energy function (9), and set the push force term E_{push} to zero. Then the snake will converge to local minimum in each point neighbourhood iteratively, and the new contour is obtained. Fig. 3 shows the detecting and evolving results of the occluder contour in one of our experiments (the snake points are denoted as white points).

(a) (b) (c)

Fig. 3 (a) Occluder contour detecting result when occlusion happens; (b) (c) Occluder contour evolving results during occlusion.

D. Object searching and certification

During occlusion, for every frame, we first detect or evolve the occluder contour, then search the tracked object around the contour neighbourhood. It is assumed that the appearance of tracked object would not change drastically during occlusion. When the likelihood of the object $\rho > L\_th$ or $\hat{\rho} > L\_th$, we think the object reappears, and then the object model is updated and the object is tracked sequentially.

IV. Experiments

We have tested our algorithm on various sequences acquired from both stationary and mobile cameras. The object model is initialized manually in the first frame as a bounding box. In all experiments, the following parameters are used.

1. The object is modelled using colour histogram in the RGB colour space with $8 \times 8 \times 8$ bins, and 8×8 bins in the CbCr space.
2. the likelihood threshold used to detect occlusion $L\_th = 0.6$, and
3. the parameters of the corresponding terms in function (9) are defined as, $\alpha = \gamma = \xi = 1$, $\beta = 0.6, \eta = 0.4$. And the gradient threshold used in (6) is defined as $L\_grad = 15$. The snake is minimized in 3×3 neighbourhood of each point.

Some experimental results are shown in Fig. 4, 5 and 6 to provide insights to the tracking performance. Fig. 4 illustrates the tracking performance with stationary camera and motionless occluder. From frame 24 to frame 39 the object is occluded totally with long time. When the object reappears in frame 40, the system tracks it immediately. Fig. 5 and 6 illustrate the tracking performance with mobile camera and moving occluder. In Fig. 5 two persons move across with complete and long-time occlusion. In Fig. 6, the face is tracked with complete and long-time occlusion, and it is easy to be confused by the hand which has the similar colour feature with the face.

Fig. 7 shows the likelihood function defined in (1) of the tracking sequences in Fig. 5. From frame 22 to frame 27, the object is temporarily "lost", the likelihood is very low, and from frame 28 it reappears and is recaptured by the proposed method.

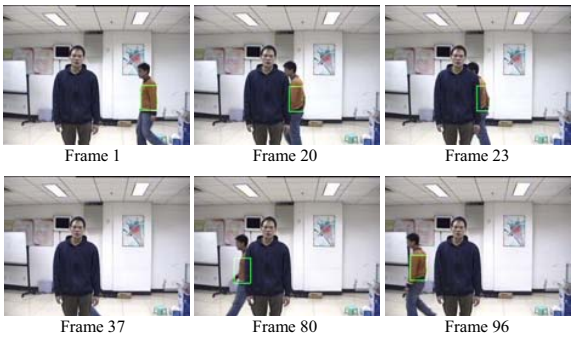


Fig. 4 Person tracking results with stationary camera and motionless occluder under total and long-time occlusion.

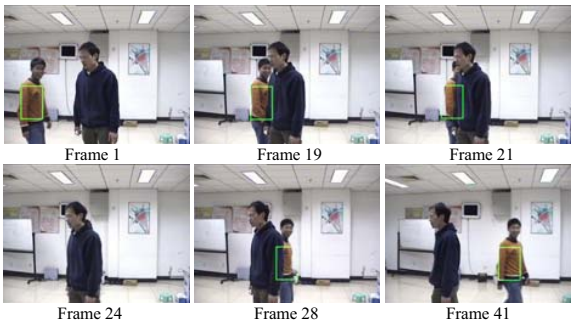


Fig. 5 Person tracking results with mobile camera under moving across condition under total and long-time occlusion.

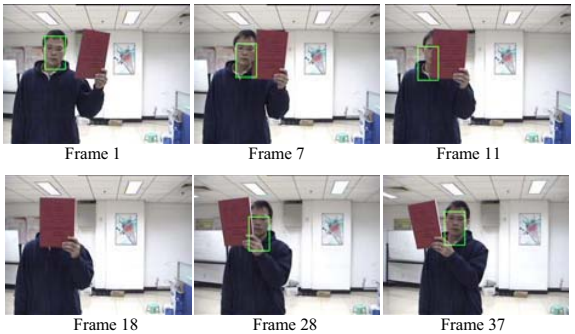


Fig. 6 Face tracking results with mobile camera and moving occluder under total and long-time occlusion.

V. CONCLUSION

This paper proposes a method for tracking object with serious and long-time occlusion in image sequences based on occluder modeling. We aim at not only modeling the object, but also modeling the occluder using improved snake algorithm, when occlusion happens. Occluder is modeled by detecting and evolving its rough partial contour which is represented by snake points, through minimizing the proposed energy function in which we introduce two novel terms $E_{push}(s)$ and $E_{con}(s)$. Then, we search the object in the neighbourhood of the occluder contour until the object reappears. The experimental results demonstrate the robust performance of the proposed method under total and long-time occlusion in image sequences acquired from both stationary and moving cameras.

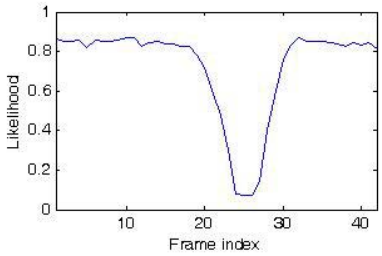


Fig.7 Likelihood function of the experiment shown in Fig. 5.

The quantitative evaluation and comparison with other methods will be done in the future work.

ACKNOWLEDGMENT

This work was partly supported by the NNSF (National Natural Science Foundation) of China under the grant 61100098.

REFERENCES

[1] A. Yilmaz, O. Javed and M. Shah, "Object Tracking: A Survey", *ACM Journal of Computing Surveys*, 2006, Vol.38, No. 4.

[2] A. Senior, et al. "Appearance Models for Occlusion Handling", In *Second International workshop on Performance Evaluation of Tracking and Surveillance systems*, 2001.

[3] Y. Te Tsai, H. Chia Shih, and C. Lin Huang, "multiple human objects tracking in crowded scenes", In *the 18th International Conference on Pattern Recognition*, 2006.

[4] T. Yang, S. Z. Li, Q. Pan, J. Li, "real-time multiple objects tracking with occlusion handling in dynamic scenes", In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] S. J. McKenna, S. Jabri, Z. Duric and A. Rosenfeld. "Tracking groups of people". *Computer Vision and Image Understanding*, 2000, (80):42–56.

[6] E. A, Davis L S. "Probabilistic Framework for Segmenting People under Occlusion", In *Proc. of IEEE 8th International Conference on Computer Vision*, 2001.

[7] Y. Wu, T. Yu and G. Hua, "Tracking appearances with occlusions", In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2003. Vol: 1, pp. 789-795.

[8] M. Hu, W. Hu, T. Tan, "tracking people through occlusion", In *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[9] J. MacCormick and A. Blake. "A Probabilistic Exclusion Principle for Tracking Multiple Objects." *International Journal of Computer Vision*, 2000, 39(1): 57-71.

[10] D. Cremers, T. Kohlberger and C. Schnorr, "Nonlinear Shape Statistics in Mumford-Shah Based Segmentation", In *European Conference on Computer Vision*, 2002 .

[11] A. Yilmaz and M. Shah, "Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November, 2004, Vol. 26, No. 11.

[12] H. Tao, H. S. Sawhney and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jan., 2002, vol. 24, no. 1, pp. 75-89.

[13] C. Gentile, O. I. Camps and M. Sznaier, "Segmentation for robust tracking in the presence of severe occlusion", *IEEE Transactions on Image Processing*, 2004, 13(2): 166-178.

[14] H. T. Nguyen and A. W.M. Smeulders, “Fast Occluded Object Tracking by a Robust Appearance Filter”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2004, Vol.26, No.8, pp. 1099-1104.

[15] P. F. Gabriel, J. G. Verly, J. H. Piater, and A. Genon. “The State of Art in Multiple Object Tracking Under Occlusion in Video Sequences”, In *Proceedings of Advanced Concepts for Intelligent Vision Systems*, September, 2003.

[16] D. Beymer, and K. Konolige, “Real-time tracking of multiple people using continuous detection”, In *IEEE International Conference on Computer Vision (ICCV) Frame-Rate Workshop*, 1999.

[17] M. ISARD, and J. Maccormick, “Bramble: A bayesian multiple-blob tracker”, In *IEEE International Conference on Computer Vision*, 2001, 34–41.

[18] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 564-575.

[19] M. Kass, A. Witkin and D. Terzopoulos, “Snakes: active contour models,” *International Journal of Computer vision*, 1988, 321-331.

[20] L. D. Cohen, “On active contour models and balloons”, *CVGIP: Image Understanding*, 1991, 53(2):211-218.

[21] D. J. Williams and M. Shah, “a fast algorithm for active contour”, In *Proceedings of the third International Conference on computer vision*, 1990, 592-595.

[22] P. Wang and H. Qiao, “Online appearance model learning and generation for adaptive visual tracking”, *IEEE Transaction on Circus and System for Video Technology*, 21(2), Feb. 2011, 156-169.