Jeongho Shin, Sangjin Kim, Sangkyu Kang, Seong-Won Lee, Joonki Paik, Besma Abidi, Mongi Abidi, "Optical flow-based real-time object tracking using non-prior training active feature mode," ELSEVIER Real-Time Imaging, Vol. 11, pp. 204-218, USA, June 2005.

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Real-Time Imaging 11 (2005) 204–218

www.elsevier.com/locate/rti

Optical flow-based real-time object tracking using non-prior training active feature model

Jeongho Shin[a], Sangjin Kim[a], Sangkyu Kang[b], Seong-Won Lee[c], Joonki Paik[a,*], Besma Abidi[d], Mongi Abidi[d]

[a]*Department of Image Engineering, Chung-Ang University, Seoul, Korea*
[b]*Digital Media Research Lab., LG Electronics, Seoul, Korea*
[c]*Department of Computer Engineering, Kwangwoon University, Seoul, Korea*
[d]*Department of Electrical and Computer Engineering, University of Tennessee, Knoxville, USA*

Available online 6 June 2005

Abstract

This paper presents a feature-based object tracking algorithm using optical flow under the non-prior training (NPT) active feature model (AFM) framework. The proposed tracking procedure can be divided into three steps: (i) localization of an object-of-interest, (ii) prediction and correction of the object's position by utilizing spatio-temporal information, and (iii) restoration of occlusion using NPT-AFM. The proposed algorithm can track both rigid and deformable objects, and is robust against the object's sudden motion because both a feature point and the corresponding motion direction are tracked at the same time. Tracking performance is not degraded even with complicated background because feature points inside an object are completely separated from background. Finally, the AFM enables stable tracking of occluded objects with maximum 60% occlusion. NPT-AFM, which is one of the major contributions of this paper, removes the off-line, preprocessing step for generating a priori training set. The training set used for model fitting can be updated at each frame to make more robust object's features under occluded situation. The proposed AFM can track deformable, partially occluded objects by using the greatly reduced number of feature points rather than taking entire shapes in the existing shape-based methods. The on-line updating of the training set and reducing the number of feature points can realize a real-time, robust tracking system. Experiments have been performed using several in-house video clips of a static camera including objects such as a robot moving on a floor and people walking both indoor and outdoor. In order to show the performance of the proposed tracking algorithm, some experiments have been performed under noisy and low-contrast environment. For more objective comparison, PETS 2001 and PETS 2002 datasets were also used.

1. Introduction

Object tracking by analyzing motion and shape in video gains more attractions in many computer vision applications including video surveillance, motion analysis and extraction for computer animation, human–computer interface (HCI), and object-based video compression. Tracking a deformable object in consecutive frames is of particular concern in video surveillance systems.

There have been various researches for video-based object extraction and tracking. One of the simplest methods is to track regions of difference between a pair of consecutive frames [1], and its performance can be improved by using adaptive background generation and subtraction. Although the simple difference-based tracking method is efficient in tracking an object under noise-free circumstances, it often fails under noisy, complicated background. The tracking performance is further degraded if a camera moves either intentionally

*Corresponding author. Tel.: +82 2 820 5300.
E-mail addresses: shinj@ms.cau.ac.kr (J. Shin), layered372@wm.cau.ac.kr (S. Kim), sangkyukang@lge.com (S. Kang), swlee@kw.ac.kr (S.-W. Lee), paikj@cau.ac.kr (J. Paik), besma@utk.edu (B. Abidi), abidi@utk.edu (M. Abidi).

1077-2014/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rti.2005.03.006

Table 1
Various object tracking algorithms and their properties

Algorithm by	Camera	Tracking entity	Occlusion handling by	Specific task
W4 [1]	Single grayscale,	Silhouettes of people	Appearance model	Surveillance of people
Amer [2]	Single	Shape, Size, Motion	Non-linear voting	Surveillance, retrieval
Wren [3]	Single, color	Human body model	Multi-class statistical model	Tracking human
Comaniciu [4,5]	Color	Elliptical kernel without shape model	Appearance histogram	Tracking non-rigid objects
Baumberg [6]	Single	Eigenshapes	Shape space training	Analyzing human motion
Isard [9]	Single	Shape space known	State-space sampling	Tracking objects in the shape space
ASM-based [10]	Single, color	Boundary of object	Active shape model	Tracking human
Yilmaz [11]	Color, IR, moving	Contour	Shape prior	Tracking level sets
Erdem [12]	Single, color	Contour	Prediction-correction	On/off-line tracking
Proposed NPT-AFM algorithm	Single, color	Feature points	Non prior training active feature model	Tracking both rigid and non-rigid objects

or unintentionally. For tracking objects in the presence of shadows, noise, and occlusion, a non-linear object feature voting scheme has been proposed in [2].

As an alternative method of the difference-based tracking, a blob tracking method using simple geometric models, e.g., ellipse or rectangle, can track the centroid of an object. Based on the assumption of stationary background, Wren et al. proposed a real-time blob tracking algorithm [3]. Another blob tracking method that uses object's histogram and mean shift approach has been proposed in [4,5].

For more robust analysis of an object, shape-based object tracking algorithms have been developed, which utilize a priori shape information of an object-of-interest, and project a trained shape onto the closest shape in a certain frame. This type of methods includes active contour model (ACM) [6,7], active shape model (ASM) [8], and the Condensation algorithm [9].

Although the existing shape-based tracking algorithms can commonly deal with partial occlusion, they exhibit two major problems in the practical applications, such as: a priori training of the shape of a target object and iterative modeling procedure for convergence. The first problem hinders the original shape-based method from being applied to tracking objects of unpredictable shapes. The second problem becomes a major bottleneck for a real-time implementation. Selected object tracking algorithms of significant interests are summarized in Table 1.

This paper presents a feature-based object tracking algorithm using optical flow under the non-prior training active feature model (NPT-AFM) framework which generates training shapes in real-time without preprocessing. The proposed tracking algorithm extracts moving objects by using motion between frames, and determines feature points inside the object. Selected feature points in the next frame are predicted by a spatio-temporal prediction algorithm. If a feature point is missing or failed in tracking, a feature correction process restores it. The proposed AFM can track deformable, partially occluded objects by using the greatly reduced number of feature points rather than taking entire shapes in the existing shape-based methods. Therefore, objects can be tracked without any a priori information or any constraint with respect to camera position or object motion.

In summary major contribution of the proposed object tracking algorithm using NPT-AFM is twofold: (i) a real-time implementation framework obtained by removing a prior training process and (ii) ASM-based restoration of occlusion using a significantly reduced number of feature points.

The paper is organized as follows. In Section 2 we briefly introduce the overview of the proposed feature-based tracking algorithm. Optical flow-based tracking of feature points is presented in Section 3, and the restoration of occlusion using NPT-AFM is proposed in Section 4. Section 5 summarizes experimental results and Section 6 concludes the paper.

2. Feature-based tracking: an overview

The proposed feature-based tracking algorithm is shown as a form of flowchart in Fig. 1.

The upper, dotted box represents the algorithm of initial feature extraction. The remaining part of the flowchart relates two consecutive frames, named as the tth and the $t+1$st frames. In the motion-based object detection block, we extract motion by the simple Lucas–Kanade optical flow [13] and classify object's movement into four directions such as left, right, up, and down. As a result, the region of moving objects can be extracted and the corresponding region is then labeled based on motion direction.

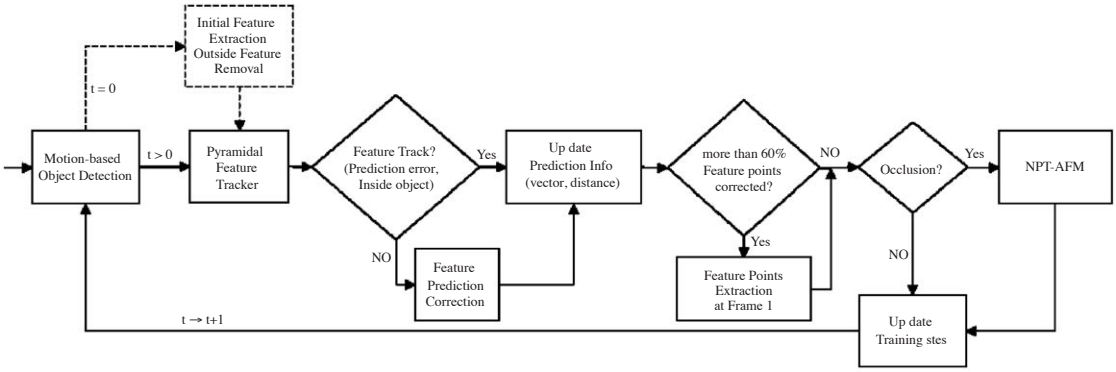

Fig. 1. Flow chart of the proposed feature-based tracking algorithm.

After detecting moving objects from background, we extract a set of feature points inside the object and predict the corresponding feature points in the next frame. We keep checking and restoring any missing feature points during the tracking process. If over 60% of feature points are restored, we decided the set of feature points are not proper for tracking, and redefine new set of points. If occlusion occurs, the NPT-AFM process, which updates training sets at each frame to restore entire shapes from the occluded input, is performed to estimate the position of occluded feature points.

The advantages of the proposed algorithm can be summarized as: (i) it can track both rigid and deformable objects because a general feature-based tracking algorithm is applied, (ii) it is robust against object's sudden motion because motion direction and feature points are tracked at the same time, (iii) its tracking performance is not degraded even with complicated background because feature points are assigned inside an object rather than near boundary, and (iv) it contains the NPT-AFM procedure that can restore partial occlusion in real-time.

3. Optical flow-based tracking of feature points

The proposed algorithm tracks a set of feature points based on optical flow. A missing feature point during the tracking is restored by using both temporal and spatial information inside the predicted region. One important contribution of this work is to provide a restoration process for missing feature points, which occurs at almost every frame under realistic, noisy environment.

3.1. Motion-based object detection

For the detection of an object from background, we use optical flow as an initial clue. The fundamental condition of optical flow is that intensity of a point on the object does not change during the sufficiently small duration. Let $S_C(x,y,t)$ represent the distribution of intensity in a continuous frame, the optical flow condition can be defined as [13]

$$\frac{\mathrm{d}S_C(x,y,t)}{\mathrm{d}t}=0. \tag{1}$$

By applying the chain rule to (1), we have that [13]

$$\frac{\partial S_C(x,y,t)}{\partial x}\,v_x(x,y,t)+\frac{\partial S_C(x,y,t)}{\partial y}\,v_y(x,y,t)+\frac{\partial S_C(x,y,t)}{\partial t}=0, \tag{2}$$

where $v_x(x,y,t)=\mathrm{d}x/\mathrm{d}t$, and $v_y(x,y,t)=\mathrm{d}y/\mathrm{d}t$. From (2), we can evaluate optical flow as

$$\varepsilon_{OF}(v(x,y,t))=\langle\nabla S_C(x,y,t),v(x,y,t)\rangle+\frac{\partial S_C(x,y,t)}{\partial t}=0, \tag{3}$$

where $\nabla S_C(x,y,t)=[\partial S_C/\partial x\ \partial S_C/\partial y]^{\mathrm{T}}$, $v(x,y,t)=[v_1(x,y,t)\ v_2(x,y,t)]^{\mathrm{T}}$, and $\langle\cdot,\cdot\rangle$ denotes vector inner product [13]. Optical flow that satisfies the constraint in (3) is prone to noise because the difference approximation between adjacent pixels is used for evaluating derivatives. To reduce noise amplification, Horn–Schunk's and Lukas–Kanade's methods are widely used in the literature [14,15]. Error from the optical flow constraint can be measured as

$$E=\sum_{(x,y)\in R}\varepsilon_{OF}^2, \tag{4}$$

where R represents a neighboring region [13]. In minimizing (4), we ignore trivial motion vectors so as to reduce error due to noise. The object detection algorithm is summarized in Fig. 2.

For the real-time implementation of optical flow, the simple Lucas–Kanade method based on block motion model [13] is used, which does not use iterative procedure in estimating optical flow. By using the simple Lucas–Kanade optical flow, we extract motion from a video sequence and segment regions based on the

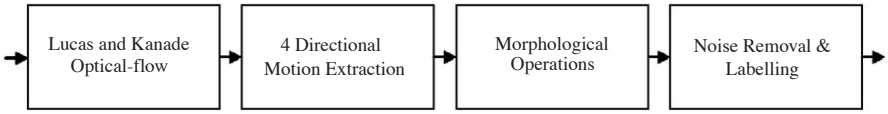

Fig. 2. Motion-based object detection algorithm.

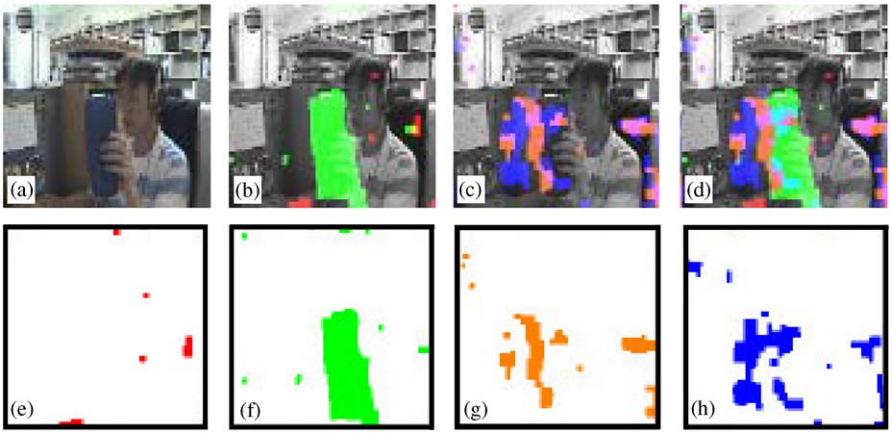

Fig. 3. Results of motion detection in four directions: (a) an input frame, (b) horizontal motion, (c) vertical motion, (d) combined horizontal and vertical motions, (e) motion from left to right, (f) motion from right to left, (g) upward motion, and (h) downward motion.

direction of motion as shown in Fig. 3. As shown in Fig. 3(d), extracted results labeled by different color include noise.

Due to the nature of optical flow, an extracted region is subject to noise, and has many holes, which are removed by morphological operations that consist of twice opening after twice closing with the 5×5 diamond-shaped structuring element.

In order to separate a moving object and noise in low-gradient conditions, we used the following measure, which is called the normalized difference:

$$D_i = \frac{1}{N_i} \sum_{(x,y)\in R_i} |I_t(x,y) - I_{t-1}(x,y)|, \tag{5}$$

where R_i represents the ith segment, and N_i the total number of pixels in R_i. If the normalized difference is smaller than a pre-specified threshold denoted by T_D, the corresponding region is considered to be noise. As a result, the region of moving objects can be extracted, and the corresponding region is then labeled based on motion direction.

Although a user can control the threshold, T_D, the value of 5.5 was used for all test sequences. In general, the range of 4.5 to 16.9 is reasonable for object segmentation. While the value below 4.5 cannot effectively reduce noise, the value over 16.9 cannot detect the object. When the threshold value is equal to 5.5, the result of noise removal is shown in Fig. 4. As known from the result in the lower row of Fig. 4(a), the original frame includes an object with motion from right to left in Fig. 4(f).

3.2. Feature point extraction

After detection of an object from background, we extract a set of feature points inside the object by using the Bouguet tracking algorithm [9,16]. Due to the nature of motion estimation, motion-based object detection algorithms usually extract the object slightly larger than the real size of the object, which results in false extraction of feature points outside the object. Let the position of a feature point at frame t be v_i^t, where i represents the index of feature points and $v_i^t = [x_i^t, y_i^t]^{\mathrm{T}}$. These outside feature points are removed by considering the distance between feature points given as

$$v_i^t = \begin{cases} v_i^t, & d_i \geqslant T \\ 0, & d_i < T \end{cases}, \tag{6}$$

where $d_i = \sum_{t=0}^{K-1} \sqrt{(x_i^{t+1} - x_i^t)^2 + (y_i^{t+1} - y_i^t)^2}$, t represents the index of frames, and i the index of feature points. Here d_i represents the sum of distance between tth frame and $t+1$st frame with respect to ith feature point. In general, the moving distance of a feature point in the background (outside object) is further less than that of a feature point in the tracked object. Although the value of T is equal to 3.5 or 7 was used for all test sequences, users can control the value depending on the environmental factors such as illumination, noise, and the complexity of background. The results of outside feature point removal are shown in Fig. 5. Three feature points outside the 'Tang' indicated by circle in Fig. 5(a) are efficiently removed in Fig. 5(b).

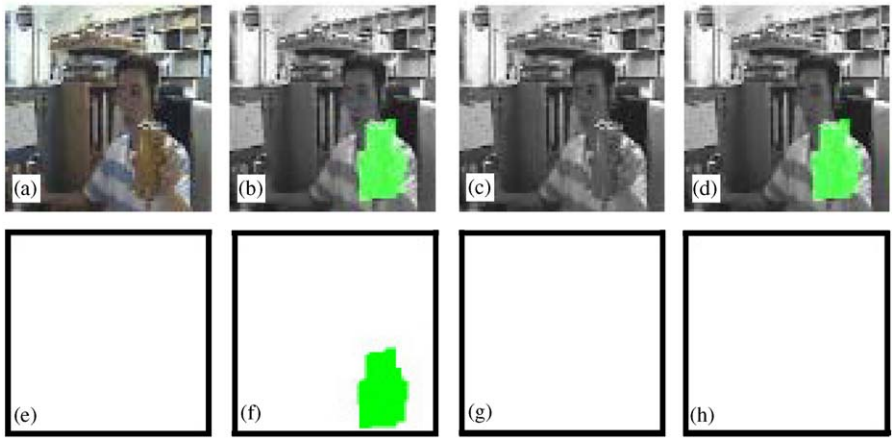

Fig. 4. Results of noise removal using the normalized difference given in (5): (a) an input frame, (b) horizontal motion, (c) vertical motion, (d) combined horizontal and vertical motions, (e) motion from left to right, (f) motion from right to left, (g) upward motion, and (h) downward motion.

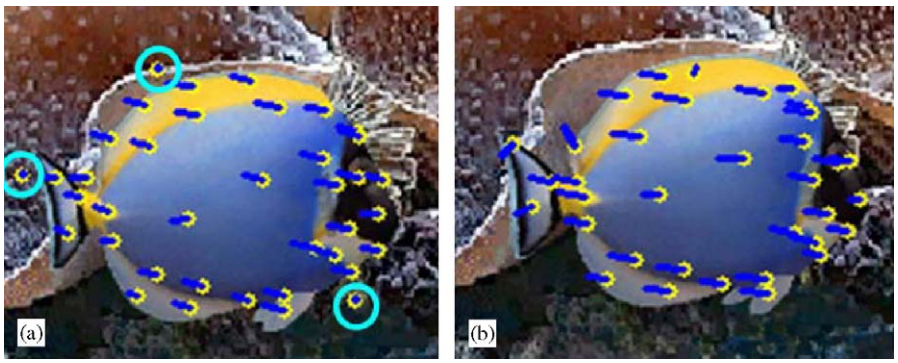

Fig. 5. Results of outside feature point removal: (a) the 2nd frame with three outside feature points highlighted by circles and (b) the 7th frame without outside feature points.

3.3. Feature point prediction and correction

Sometimes, a tracking algorithm may fail to track a proper feature point in the next frame. A feature point is defined as untracked when an error value within small window is over a pre-defined threshold. Specifically, a threshold value is determined by distance between average vectors predicted by a spatio-temporal prediction. After the spatio-temporal prediction, re-investigation is performed. Then, both tracked and untracked feature points are updated in a list.

In many real-time, continuous video tracking applications, a feature-based tracking algorithm fails due to the following reasons: (i) self or partial occlusions of an object and (ii) feature points on or outside the boundary of the object, which are affected by changing background.

In order to deal with the tracking failure, we should correct the erroneously predicted feature points by using the location of the previous feature points and inter-pixel relationship between the predicted points. This algorithm is summarized in Table 2.

A temporal prediction is suitable for deformable objects while a spatial prediction is good for non-deformable objects. Both temporal and spatial prediction results can also be combined with proper weights. Although users can control the value of the number of frames, K, the value of 7 was used for all test sequences. The larger the value of K is, the better the performance of the algorithm is. Because of trade-off between processing time and accuracy, the value around 7 was found to be reasonable for temporal prediction.

4. Restoration of occlusion using NPT-AFM

A popular approach to tracking two-dimensional (2D) deformable objects is to use an object's boundary, and ASM-based tracking falls into this category. ASM can robustly analyze and synthesize a priori trained shape of an object even if an input is noisy and/or occluded [17]. On the other hand, pre-generation of training sets and iterative convergence prevent the ASM from being used for real-time tracking.

Table 2
Spatio-temporal algorithm for correction of predicted feature points

Step 1	*Temporal prediction*:
	If the ith feature point at the tth frame is lost in tracking, it is re-predicted as [12] $\hat{v}_i^{t+1} = v_i^t + \frac{1}{K}\sum_{k=0}^{K-1} m_i^{t-k}$, (7)
	where $m_i^t = v_i^t - v_i^{t-1}$, and K represents the number of frames for computing the average motion vector. In (7), the location of the missing feature point can be predicted using the average of its motion vectors in the last frames.
Step 2	*Spatial prediction*:
	We can correct the erroneous prediction by replacing with the average motion vector of successfully predicted feature points. The temporal and spatial prediction results of Step 1 and Step 2 can be combined to estimate the position of feature points.
Step 3	*Re-investigation of the predicted feature point*:
	Assign the region including the predicted and corrected feature points in temporal and spatial prediction steps. If a feature point is extracted within a certain extent in the following frame, it is updated as a new feature point. While the re-predicted feature points are more than 60% of the entire feature points, feature points keeps being estimated.

We propose a real-time updating method for the training set instead of off-line preprocessing, and also modify the ASM by using a reduced number of feature points instead of the entire set of landmark points. NPT-AFM refers to the proposed real-time, efficient feature-based modeling structure to reconstruct feature points.

4.1. Landmark point assignment using feature point

The existing ASM algorithm manually assigns landmark points on an object's boundary to make a training set. A good landmark point has balanced distance between adjacent landmark points and resides on either high-curvature or "T" junction position. A good feature point, however, has a different requirement from that of a good landmark point. In other words, a feature point is recommended to locate inside the object because a feature point on the boundary of the object easily fails in optical flow or block matching-based tracking due to the effect of changing, complicated background.

Consider n feature points form an element shape in the training set. We update this training set at each frame of input video, and at the same time align the shape onto the image coordinate using Procrustes analysis [18]. In this work, the training set has 70 element shapes.

4.2. Principal component analysis

A set of n feature points, which is a member of the training set, represent the shape of an object as a 2D outline. Instead of using all feature points in a member of the training set, the principal component analysis (PCA) technique helps to model the shape of the object with fewer parameters. Given a set of feature points the input shape can be modeled by using PCA as summarized in [9].

4.3. Feature model fitting

The best set of parameters that represents the optimal location and shape of an object can be obtained by matching the shape of models in the training set to the real object in the image. The matching is performed by minimizing the error function as

$$E = (y - Mx)^{\mathrm{T}} W (y - Mx), \tag{8}$$

where x represents the coordinate of the model, y represents the coordinate of the real object, W represents a diagonal matrix whose diagonal element is the weight to each landmark points, and M is a matrix for the geometrical transform which consists of rotation θ, transition t, and scaling factor s. The weight decides the distance between previous and new feature points. By minimizing (8), the best pose denoted by M, can be computed to match the features in the model coordinate x, to the features in the real image coordinate y.

The geometrical transformation matrix for a point $(x_0, y_0)^{\mathrm{T}}$ can be represented as

$$M\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = s\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{9}$$

Once the set of geometrical parameters $\{\theta, t, s\}$ is determined, the projection of y onto the frame of model parameters is given as

$$x_p = M^{-1} y. \tag{10}$$

The set of updated feature points generate a new feature model at y in the image coordinate, and can be used to find the best fitting feature model by (8). After finding the best location using M, a new feature model is projected onto the eigenvectors, and finally updates model parameter, which is described in [9]. As a result, only principal components can affect the feature model.

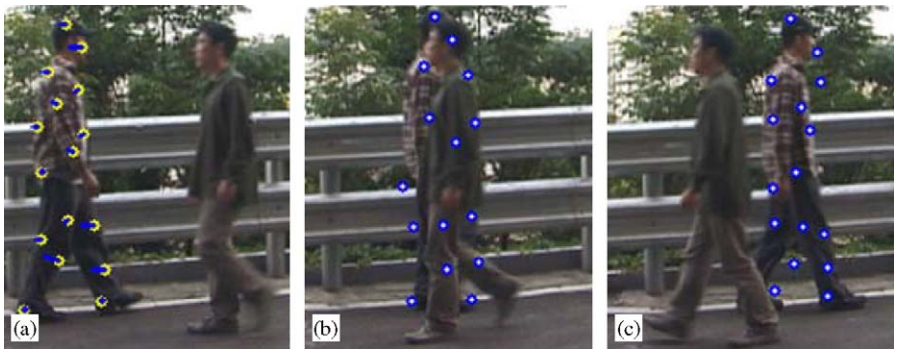

Fig. 6. Model fitting procedure of NPT-AFM: (a) optical flow-based feature tracking at the 40th frame, (b) model fitting at the 74th frame, and (c) model fitting at the 92nd frame.

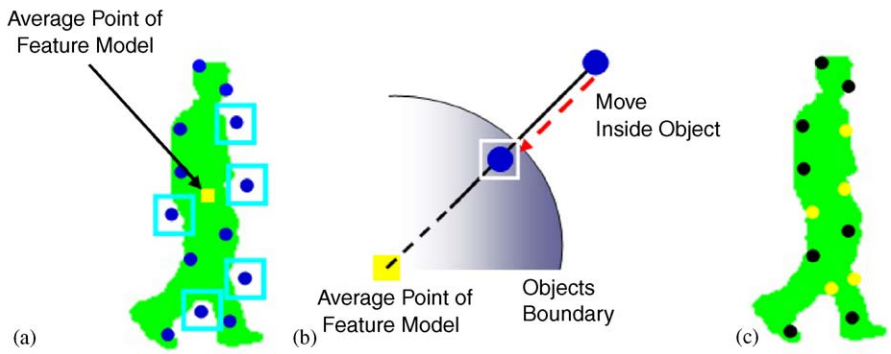

Fig. 7. Reconstruction of the feature model: (a) feature model fitting result, (b) relocation of an outside feature point for feature reconstruction, and (c) result of feature reconstruction.

Fig. 6 shows the result of optical flow-based model fitting with 51 training shapes. As shown in Fig. 6(b) and (c), a few mismatches between the feature model and the real object can be found after occlusion. Therefore, only the model fitting cannot guarantee successful tracking.

4.4. Reconstruction of feature model and occlusion handling

The proposed real-time tracking framework can model object's feature distribution and solve the occlusion problem by using the following steps: feature point assignment, PCA, and feature model fitting.

In spite of theoretical completeness, a feature model obtained from the model fitting step does not always match a real object because it has been constructed using a training set of features in the previous frame as shown in Fig. 6. Practically, reconstruction of the feature model is necessary after the initial model fitting to solve the feature mismatching problem. While the ASM-based tracking algorithms have advantages of robustness to occlusions, it is hard to implement them in real-time. For example, it requires an iterative searching procedure for finding the best landmark or feature position along the orthogonal direction to the object's contour, a prior training set should be built for each object, and a number of feature points should be estimated to track the entire shape of an object.

In order to reduce the computational load, a feature reconstruction algorithm is proposed as shown in Fig. 7. The feature reconstruction algorithm moves an outside feature point toward the average position of all feasible feature points as shown in Fig. 7(b). In other words, the feature reconstruction algorithm makes feature points stay inside the object as shown in Fig. 7(c). While moving the outside feature point, we search the best path among three directions toward the average position. If the number of outside feature points is more than 60% of the total feature points, the feature extraction process is repeated.

In addition to reconstructing the feature model, occlusion handling is another important requirement in a realistic tracking algorithm. The proposed NPT-AFM-based occlusion handling algorithm first detects occlusion if the size of a labeled region becomes larger than that of the original region by a certain ratio. In addition, the decision is made with additional information such as the direction and magnitude of motion vector in the correspondingly labeled object region and

the directional motion information extracted in Section 3.1. For example, if the size of labeled region is 1.6 times larger, the average of the motion vector abruptly changes, and the size of each directionally labeled region is similar to that of the region in the previous frame, the occlusion results from objects with different motion direction as shown in Fig. 8.

On the other hand, if the size of the labeled region is 1.6 times larger, the same directional motion vector abruptly increases, and the average of the motion vector changes, the occlusion results from objects with the same motion direction. In other words, an object is considered to be occluded when the size of the labeled region is 1.6 times larger and the magnitude of motion vector in the labeled region is changed more than 60%. Although a user can control the rates, the values of 1.6 and 60% were used for all test sequences. In the paper, the specific rates were manually determined by experiments.

If an occlusion is detected, untracked feature points can be roughly estimated by utilizing the spatio-temporal motion prediction algorithm described in Table 2. However, it is not easy to track non-rigid objects because the spatio-temporal motion prediction algorithm considers the location of the previous feature points and inter-pixel relationship between the predicted points. By performing the feature model fitting algorithm described in Section 4.3, therefore, feasible feature points can be estimated. In order to track objects during occlusion, we separately preserve the previous labeling information to keep multiple objects' feature models.

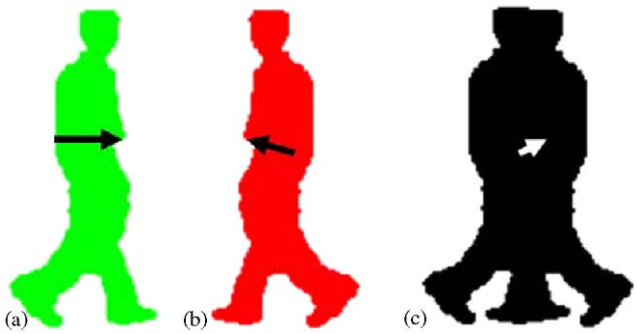

Fig. 8. Occlusion detection and handling: (a) an object moving right, (b) the other object moving left, and (c) the occluded objects.

If the size of labeled region retains between $0.8L$ and $1.2L$, where L represents the original size of the labeled region after occlusion, we consider that the occlusion disappears. By reconstructing the feature model after occlusion, all of feature points at each frame become fit for tracking.

4.5. Selection of parameters for NPT-AFM

For the proposed NPT-AFM-based tracking algorithm, several threshold values should be properly determined for the proposed tracking algorithm to provide good performance. The parameters used in Sections 3 and 4 are analyzed in Table 3. For the best performance of the proposed tracking algorithm, all parameters were experimentally determined in various simulations.

5. Experimental results

We used 320×240, both indoor and outdoor video sequences to test tracking performance for rigid and deformable objects, some of which are standard test sequences downloaded from the website of PETS 2001/2002 (PETS 2001 Dataset 1 and PETS 2002 People—Training: Dataset 1) as shown in Fig. 9. In most experimental images, bright circles represent successfully tracked feature points while dark ones represent corrected, reconstructed feature points.

5.1. Tracking rigid objects

For the experiment of a rigid object tracking, we captured an indoor robot image sequence using a hand-held camera. Because the image sequence is captured by a hand-held camera, the video includes both global motion by hand shaking and object motion. The results of tracking are shown in Fig. 10. Fig. 10(b) shows the region detected by motion-based object segmentation. Tracking results in the 3rd, 10th, and 34th frames are shown in Fig. 10(c), (e), and (g), respectively. Fig. 10(d), (f), and (h), show the corresponding trajectories.

The additional results using rigid object of PETS 2001 Dataset are shown in Fig. 11, where we notice that all feature points in the white van are accurately tracked.

Table 3
Analysis of parameters for the proposed tracking algorithm

T_D in (5)		T in (6)		K in (7)		Rate of outside feature to all features in Section 4.4		Region size after occlusion in Section 4.4	
3.0–4.0	Not good	1.5–3.5	Not good	3	Not good	50%	Good	0.8L–1.2L	Good
4.0–6.0	Good	3.5–7	Good	7	Good	60%	Good	0.7L–1.3L	Not good
6.0–9.0	Not good	7–10	Not good	10	Not good	75%	Not good	0.6L–1.6L	Not good

Fig. 9. Tracking data sets: (a) Robot, (b) PETS 2001 Dataset 1, (c) moving person 1, (d) moving person 2, (e) moving person 3, (f) PETS 2002 People—Training: Dataset 1, and (g) moving person 4.

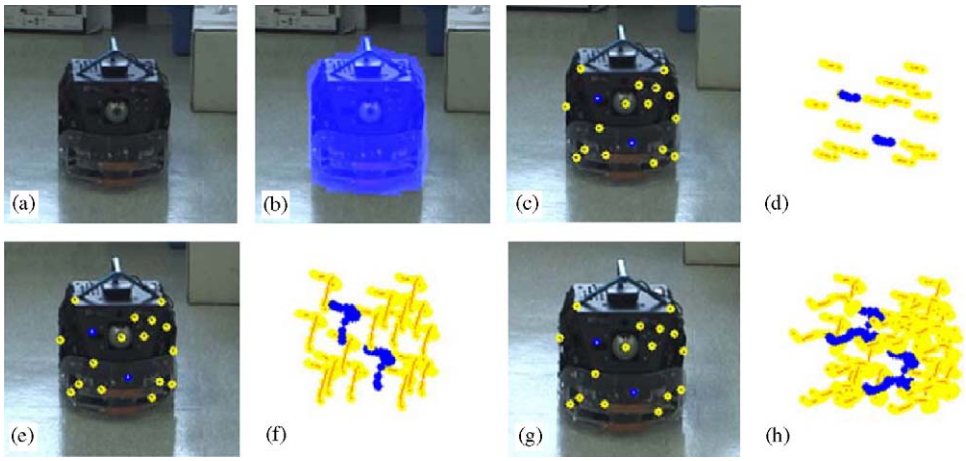

Fig. 10. Feature tracking of a rigid object and the resulting trajectory: (a) the original indoor, robot frame, (b) motion-based object detection result (only motion in the 270° direction is displayed), (c) the 3rd frame and (d) its trajectory, (e) the 10th frame and (f) its trajectory, (g) the 34th frame and (h) its trajectory.

Fig. 11. Rigid object tracking results in PETS 2001 Dataset 1: 175th, 189th, 202nd, and 214th frames.

Fig. 12. Feature tracking of deformable object: 5th, 13th, 32nd, and 39th frames.

Fig. 13. Feature tracking of deformable object: 27th, 58th, 75th, and 113th frames.

5.2. Tracking deformable objects

For the various experiments of tracking a non-rigid object, we captured three image sequences using SONY 3CCD DC-393 color video camera with an auto-iris function. Figs. 12 and 13 respectively include a moving person wearing different patterned shirt in the simple background, while Fig. 14 includes a moving person with a complicated background. Tracking results using the proposed algorithm are shown in Figs. 12–14. Predicted feature points are classified into two classes: successful and reconstructed, which are separately

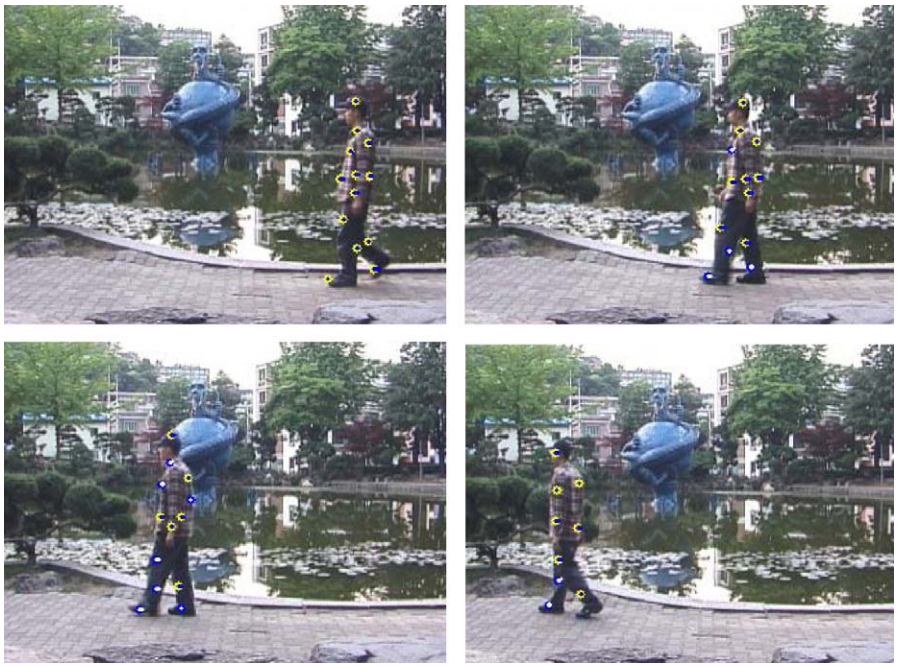

Fig. 14. Feature tracking of a deformable object in an outdoor sequence: 12th, 46th, 98th, and 133rd frames.

displayed in Fig. 13. Tracking results in an outdoor sequence are shown in Fig. 14, where we notice that all feature points are accurately tracked even under complicated background. The additional results using PETS 2002 Dataset are shown in Fig. 15, where we also notice that all feature points are accurately tracked.

5.3. NPT-AFM-based occlusion handling

The restoration results of occlusion by using the proposed NPT-AFM are shown in Fig. 16. Fig. 16 shows the model fitting results using the NPT-AFM before and after occlusion. By reconstructing feature points in the presence of occlusion, the NPT-AFM can track occluded object as shown in Fig. 16. While each feature point in the 2nd and 3rd columns of Fig. 16 is reconstructed by the NPT-AFM, each feature point in the 1st and 4th columns of Fig. 16 is well-tracked. Although the proposed algorithm tracks two occluded objects as shown in Fig. 16, the proposed tracking algorithm can track more than three occluded objects depending on object motion. If there are multiple occluded objects with similar motion in a frame, it is not easy to track the multiple objects.

5.4. Comparison between the proposed NPT-AFM-based tracking and the shape-based tracking

In Fig. 17, the proposed tracking algorithm was compared with shape-based tracking algorithms such as ACM-based and ASM-based algorithms.

In both ACM and ASM, initial contour is chosen as the mean shape of a priori training set. Fig. 17(a) shows the iteratively optimized tracking result using the ACM. Fig. 17(b) shows the similar result but using an ASM-based tracker. Fig. 17(c) shows the result of the proposed NPT-AFM algorithm. In the sense of the processing speed, the ACM- and ASM-based tracking respectively run at 0.95 frames per second (fps) and 1.04 fps for the iteration to convergence. On the other hand, the proposed NPT-AFM tracking algorithm runs at 16.1 fps, so it is suitable for real-time implementation.

5.5. Performance analysis of the proposed algorithm under noisy or low contrast environment

In order to show the performance of the proposed tracking algorithm under noisy or low contrast environments, two experiments were performed as shown in Fig. 18.

One is for tracking performance under noisy environments; the other is for tracking performance under low contrast environment. For tracking in the noisy sequence, 10 and 5 dB Gaussian noise were respectively added to the original PETS 2001 sequence, where the noise performance is measured by means of the well-known signal-to-noise ratio (SNR). Nevertheless, the object is perfectly tracked as shown in the upper row of Fig. 18.

For the test of tracking low contrast environment, the number of gray level in a test image sequence was decreased as 200, 175, and 120. As shown in the lower row of Fig. 18, the proposed NPT-AFM algorithm

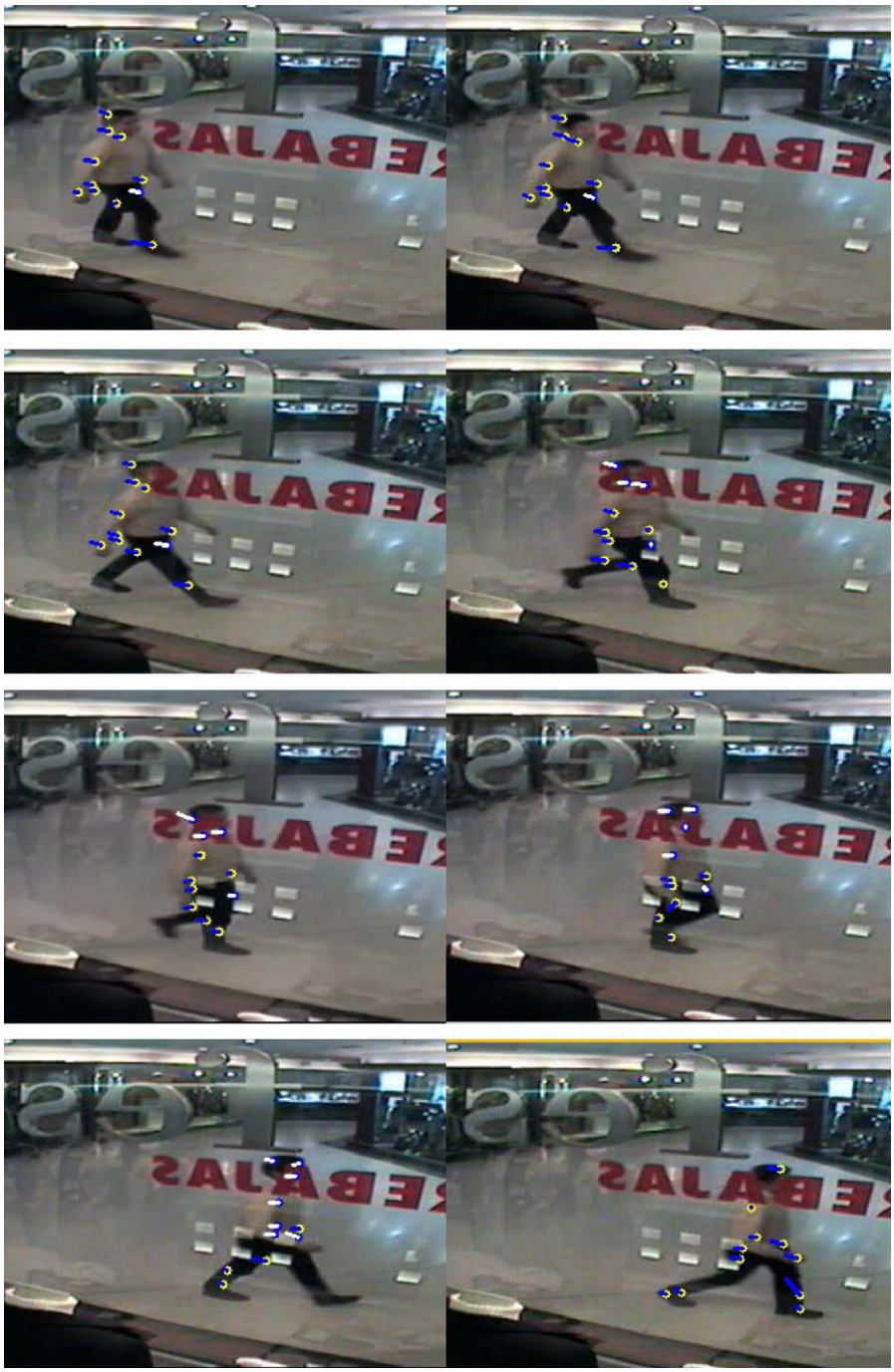

Fig. 15. Deformable object tracking results in PETS 2002 People—Training: Dataset 1: 97th, 100th, 103rd, 106th, 109th, 111th, 114th, and 117th frames.

performs well for reduced contrast level up to 175. When the contrast level is further reduced the proposed tracking algorithm fails to track objects.

5.6. Discussion

The proposed algorithm contains a number of limitations. First, although it can track multiple occluded objects, the performance of tracking depends on the object's motion. In the case of dealing with multiple objects with similar motion, the tracking algorithm tends to fail. Second, the proposed algorithm can track objects on static background as shown in Fig. 10. However, the proposed tracking algorithm was partially tested to track rigid objects under global motion.

Fig. 16. Occlusion handling results using the proposed NPT-AFM algorithm: 106th, 125th, 165th, and 177th frames (left: tracking the person with textured shirt, right: tracking the person with uniform shirt).

Fig. 17. Tracking results using (a) active contour-based tracker, (b) ASM-based tracker, and (c) the proposed NPT-AFM-based tracker for the 251st frame in PETS 2001 Dataset 1.

Fig. 18. Tracking results using (a) the 503rd original frame in PETS 2001 Dataset 1, (b) frame with 10 dB Gaussian noise, (c) frame with 5 dB Gaussian noise, (d) frame with contrast range [0–200], (e) frame with contrast range [0–175], and (f) frame with contrast range [0–120].

6. Conclusions

We presented a novel method for tracking both rigid and deformable objects in video sequences. The proposed tracking algorithm segments object's region based on motion, extracts feature points, predicts the corresponding feature points in the next frame using optical flow, corrects and reconstructs incorrectly predicted feature points, and finally applies NPT-AFM to handle the occlusion problem.

NPT-AFM, which is one of the major contributions of this paper, removes the off-line, preprocessing step for generating a priori training set. The training set used for model fitting can be updated at each frame to make more robust object's shape under occluded situation. The on-line updating of the training set can realize a real-time, robust tracking system.

We also presented a complete set of experimental results including both indoor and outdoor sequences with rigid and deformable objects. Experimental results prove that the proposed algorithm can track both rigid and deformable objects under various conditions such as noisy and low-contrast condition, and it can also track the object-of-interest with partial occlusion and complicated background.

Acknowledgment

This work was supported by Korean Ministry of Science and Technology under the National Research Laboratory Project, by Korean Ministry of Information and Communication under the Chung-Ang University HNRC-ITRC program, by the Korea Research Foundation Grant funded by Korean Government (MOEHRD)(R08-2004-000-10626-0), by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, and by the DOD/TACOM/NAC/ARC Program, R01-1344-18.

References

[1] Haritaoğlu I, Harwood D, Davis L. W-4: Real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22(8):809–30.
[2] Amer A. Voting-based simultaneous tracking of multiple video objects. Proceedings of the SPIE Visual Communication Image Processing 2003;5022:500–11.
[3] Wren C, Azerbayejani A, Darrel T, Pentland A. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19(7):780–5.
[4] Comaniciu D, Ramesh V, Meer P. Real-time tracking of nonrigid objects using mean shift. Proceedings of the IEEE International Conference on Computer Vision, Pattern Recognition 2000;2:142–9.
[5] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 2003;25(4):1–14.
[6] Baumberg AM. Learning Deformable Models for Tracking Human Motion. PhD dissertation, School of Computer Studies, University of Leeds, UK, October 1995.
[7] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. International Journal of Computer Vision 1987;1(4):321–31.
[8] Cootes T, Taylor C, Cooper D, Graham J. Training models of shape from sets of examples. Proceedings of the British Machine Vision Conference, September 1992. p. 9–18.
[9] Isard M, Blake A. Condensation-conditional density propagation for visual tracking. International Journal of Computer Vision 1998;29(1):5–28.
[10] Koschan A, Kang S, Paik J, Abidi B, Abidi M. Color active shape models for tracking non-rigid objects. Pattern Recognition Letters 2003;24(11):1751–65.
[11] Yilmaz A, Shah M. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26(11):1531–6.

[12] Erdem C, Tekalp A, Sankur B. Video object tracking with feedback of performance measures. IEEE Transactions on Circuits and Systems for Video Technology 2003;13(4):310–24.
[13] Tekalp AM. Digital video processing. Englewood Cliffs, NJ: Prentice-Hall; 1995.
[14] Shi J, Tomasi C. Good features to track. Proceedings of the IEEE Computer Vision, Pattern Recognition, June 1994. p. 593–600.
[15] Horn B, Schunck B. Determining optical flow. Artificial Intelligence 1981;17(1–3):185–203.
[16] Bouguet J. Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithms. OpenCV Documentation, Micro-Processor Research Labs, Intel Corporation, 1999.
[17] Cootes T, Taylor C, Lanitis A. Active shape models: evaluation of a multi-resolution method for improving image search. 1994 Proceedings of the British Machine Vision Conference, 1994. p. 327–36.
[18] Goodall C. Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society, Part B 1991;53(2):285–339.