Multimed Tools Appl (2011) 51:913–933
DOI 10.1007/s11042-009-0423-4

Real-time multiple people tracking for automatic group-behavior evaluation in delivery simulation training

Jungong Han · Peter H. N. de With

Published online: 11 November 2009

Abstract This paper aims at generating an automated way to evaluate the team-behavior of trainees in a delivery simulation course using video-processing techniques with emphasis on multiple people tracking. The paper is composed of two interacting, but clearly separated stages: moving people segmentation and multiple people tracking. At people segmentation stage, the combination of the Gaussian Mixture Model (GMM) and the Dynamic Markov Random Fields (DMRF) technique helps to extract the foreground pixels. For a better extraction of the human silhouettes, the energy function of DMRF is extended with texture information. At multiple people tracking stage, we concentrate on solving human-occlusion problem caused by interacting persons based on silhouette data and a non-linear regression model. Our model effectively transfers the person location problem during the occlusion into the finding of the local maximum points on a smooth curve, so that visual persons in the partial or complete occlusion can still be precisely captured. We have compared our algorithm with two other popular tracking algorithms: mean-shift and particle-filter. Experimental results reveal that the correctness of our method is much higher than the mean-shift algorithm and slightly lower than a particle-filter, however, with the major benefit of being a factor of 10–15 faster in computing.

Keywords Multiple people tracking · MRF · Automatic assessment · Group behavior analysis · Real time

J. Han (✉) · P. H. N. de With
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: jungonghan@hotmail.com

P. H. N. de With
CycloMedia Technology, Waardenburg, The Netherlands

1 Introduction

During the last 10 to 15 years, simulation-based training has emerged as a strategy that can be used to promote patient safety in maternity and newborn settings. After being exposed to both low- and high- risk scenarios in the realistic setting, trainees are better able to make nursing care decisions. Feedback from their instructors allows them to learn from their mistakes and further develop their clinical reasoning skills, ultimately leading to safer delivery of care. Performance assessment of trainees is an important and essential step in entire simulation training procedure. Basically, there are two types of performance assessment: on-line assessment and off-line assessment. On-line assessment is that the instructor immediately gives remarks once he finds mistakes, while off-line assessment is to provide feedbacks after analyzing the videotape of the training course. Currently, both assessments are conducted manually, which is apparently low efficiency with the increasing number of trainees and also training course. Consequently, it highly demands for an automatic performance evaluation system for delivery simulation training. To this end, the motion and activity of trainees need to be detected and analyzed.

Video analysis system has proved ability for human behavior analysis [15], and such systems have been widely applied in healthcare domain during the recent years. Koile et al. [14] uses a computer-vision system to monitor the location of a person in a room in order to determine which activities are being completed, and with which objects a user is interacting. In [21], a prototype system is designed to monitor activities of elderly persons at home to assist the independent living. In this system, authors construct an advanced silhouette extraction, human detection and tracking algorithm for indoor environments and an adaptive learning method is used to estimate the physical location and moving speed of a person from a single camera view without calibration. Our previous work [11] also intends to monitor the daily activity of elderly person, but focuses on detecting fall incidents. The new idea proposed by this paper is to determine the fall event by analyzing the changes of the main-axis of human. The work reported in [3] aims at detecting social interactions of the elderly in a skilled nursing facility using audio/visual records, because authors believe that changes in interaction patterns can reflect changes in the mental or physical status of a patient. Although most of above systems are dealing with human behavior analysis in healthcare domain, straightly mapping their techniques to automatic evaluation of delivery simulation may not succeed due to two problems. First, most of the above systems only need to track one or two people in the scene. However, in our scenario it is required to track small groups of people (up to 4 persons), and they are always moving together or interacting with each other. In these cases, individual people are not visually isolated, but are partially or totally occluded by other people. Second, the computational cost of most above systems is far from satisfactory, so that they cannot be directly used in our application, where a (near) real-time system is expected.

This paper proposes a *real-time* system to detect, track and analyze a group of people, which is intended to be the basic system of an automatic performance assessment of trainees in a delivery simulation course. Our system is original in three aspects:

- Unlike the existing object-segmentation algorithms that only consider the spatial and temporal information of the object, our Spatial-texture (StMRF) model seamlessly incorporates texture information as a pixel feature into the MRF framework. Additionally, we design a dynamic texture detector to indicate the edges caused by foreground objects, in which both contrasts of the original image and background image are employed. With our method, the segmented silhouettes of the human are more complete and accurate than the silhouettes extracted by conventional algorithms, such as Spatial MRF (SMRF) model and Spatial-Temporal MRF (STMRF) model.

- We propose an occlusion-handling approach, which models the horizontal projection histograms of the human silhouettes, using a nonlinear regression algorithm. Our model effectively changes the person locating during the occlusion into the finding of the relative maximum points on a smooth curve (function), so that visual persons in the partial or complete occlusion can still be precisely captured. We also improve the template-matching algorithm used by our previous work [8], where the key idea is that we adaptively assign different weights to non-overlapped regions and overlapped regions when generating the appearance-model for an object during the occlusion. This treatment will improve the accuracy of template matching in object tracking, especially for the occlusion case.
- A camera calibration algorithm is embedded into the system, thereby we can compute the real speed of each person by considering the physical displacement of him/her in two successive frames. This visual feature in the real-world domain is invariant to the camera position. Different with the existing camera-calibration technique applied in the human behavior analysis system that manually labels the initial feature points, such as [2], our algorithm is robust and fully automatic.

2 Prior work in moving object segmentation and tracking

2.1 Moving object segmentation

Frame differencing [12] is a straightforward approach, which thresholds the difference between two frames, and large changes are considered to be the foreground. Another approach is to build a representation of the background that is used to compare against new images. Pixel-wise median filter [5] is a widely used background modeling technique, where the background is defined to be the median at each pixel location of all the frames in the buffer. Instead of using the median value of a group of pixels, a more reasonable assumption is that the pixel value follows a Gaussian distribution in the temporal direction, and a model is used to compute the likelihood of background and foreground for a particular pixel. When single Gaussian is not able to adequately account for the variance, a Mixture of Gaussians [22] is used to improve the accuracy of the estimation. An alternative model is proposed by Elgammal et al. [6], which estimates the probability of observing pixel intensity values based on a sample of intensity values of each pixel. The model is supposed to adapt quickly to changes in the scene, thereby enabling sensitive detection of moving targets. A completely different idea, proposed by Oliver et al. [17], investigates global statistics rather than the local one. Similar to eigenfaces, a eigenbackground is created to capture the dominant variability of the background. In [1, 18], authors employ an Markov Random Field (MRF) model to extract the moving object, in which the idea is to determine the segmentation mask as a Maximum A-Posteriori (MAP) approximation. In this MRF model, both spacial and temporal coherency of the object are considered, so that the state of a pixel is also affected by its neighbors. Generally speaking, MRF model takes a good balance between complexity and accuracy of the algorithm.

2.2 Moving object tracking

We can classify existing human tracking systems, according to their efficiency (real-time or non-real-time) and their functionality (track single person, multiple persons, handle occlusion). Pfinder [19] solves the problem of real-time tracking of people in complex scenes in which there is a single unoccluded person and fixed camera. The system called W^4 [9] is a real-time visual surveillance system for detecting and tracking people and their body parts, and monitoring their activities in an outdoor environment. However, this system has a limited capability to handle occlusions. Mean-shift [4] is a real-time non-parametric technique that searches along density gradients to find the peak of probability distributions. This approach is computationally effective, but it is susceptible to converge to a local maximum and the occlusion remains problematic. The particle-filter technique [16] performs a random search guided by a stochastic motion model to obtain an estimate of the posterior distribution describing the object's configuration. This method is robust in the sense that it allows to handle clutters in the background, and it recovers from temporal loss of tracking. Unfortunately, there are high computational demands in the approach, and this is the bottleneck to apply it in real-time systems. In the layer tracking approaches, such as [20], shape, motion and appearance models of the different layers corresponding to foreground objects and the background are estimated along with the ordering of the layers. Foreground layers behind background layers are labeled as occlusion. The drawback of these methods is the high computational cost and maintaining the layer visibility. The work of [10] is highly related to our research, in which a silhouettes-based occlusion-handling algorithm is proposed. Both the shape of the silhouette boundary and the projection histograms of the silhouette are used to locate the people during the occlusion. This method assumes that the complete body silhouette of each person is detected and the size of the people in the picture should be sufficiently large (at least 75 × 50 pixels). Unfortunately, such assumptions do not always hold in real applications. Summarizing, none of the approaches solves simultaneously the two primary problems mentioned at introduction section.

3 System overview

Figure 1 shows the flowchart of the proposed system, which consists of five components.

1. **Object segmentation.** We perform an adaptive background subtraction [22] to produce the initial masks for the moving persons. These masks are input to a MRF-based segmentation algorithm that incorporates the spatial coherence for robust foreground extraction. Here, the dynamic concept [13] helps to realize a real-time algorithm.

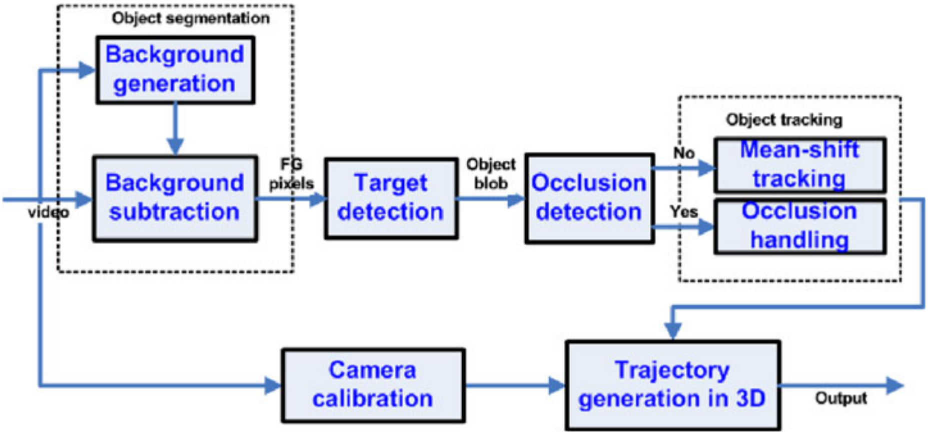


Fig. 1 Human detection, tracking and analysis system

2. **Camera calibration.** This component is to find the relationship between the image coordinate system and world coordinate system. By doing so, the visual features detected in the picture, such as positions of moving objects, can be transferred to the real-world domain. These 3D visual features will significantly facilitate to the semantic-level analysis due to its invariance to the location of camera.
3. **Occlusion detection.** This component distinguishes the occlusion and the non-occlusion cases, since we use different methods to treat these two cases. In our method, if the distances in both x and y directions between several persons are smaller than a predefined threshold in the current frame, we consider that the occlusion may happen in the next frame. Otherwise, we keep the object descriptions separated. Based on this scheme, we can even deduce the number of persons involved in the occlusion.
4. **Object tracking.** The mean-shift algorithm [4] is explored to track moving persons in the non-occlusion case, and our nonlinear regression algorithm is used to deal with occlusion.
5. **Trajectory generation.** We base the trajectory generation on our previous work [7], which adopts a Double Exponential Smoothing (DES) operator to model the position of people collected from the tracking component. Using this method, we can obtain more accurate and smooth moving trajectories.

In this paper, the emphasises are on our contributions: camera calibration, texture-controlled object segmentation and human occlusion handling. More details about other algorithms, e.g., adaptive background subtraction, mean-shift tracking and DES-based trajectory generation, can be found in [4, 7, 22].

4 Camera calibration based on homography mapping

The task of the camera calibration is to provide a geometric transformation that maps the points in the image domain to the real-world coordinates. In our system, we base the human-behavior analysis on the trajectory of person on the ground, so that the height information of the human is not required. Since both the ground and the displayed image are planar, the mapping between them is a homography, which can be written as a 3×3 transformation matrix $\mathbf{H}$, transforming a point $\mathbf{p} = (x, y, w)^\top$ in image coordinates to the real-world coordinates $\mathbf{p}' = (x', y', w')^\top$ with $\mathbf{p} = \mathbf{H}\mathbf{p}'$, which is equivalent to

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix}. \tag{1}$$

The transformation matrix $\mathbf{H}$ can be calculated from four points whose positions are both known in the real-world and in the image. In our previous work [7], we have developed an automatic algorithm to establish the homography mapping for analyzing the tennis video, where the court lines and their interaction points are identified in the picture. Such lines and points are related to the lines and points in a standard tennis court. By finding the correspondences, homography mapping described in (1) can be established. In our current system, we still want to apply this method, though there are no court lines on the ground. Our basic idea is to manually put four white lines forming a rectangle on the ground, whose function is as same as the court lines in the tennis game. We then measure the length of each line in the real world, therefore generating a physical model in the real world domain. This setup is accepted and will be adopted in the new training center of our hospital partner. The complete system comprises three steps, whose basic ideas are described below:

1. *White Pixel Detection.* This step identifies the pixels that belong to white lines. Since the lines that we put on the ground are usually white, this step is essentially a white pixel detector. The mandatory feature of this step is that white pixels that do not belong to white lines (white table, white wall) should not be marked.
2. *Line Parameter Estimation.* Initializing with the detected white pixels, line parameters can be extracted using a RANSAC-based line detector, which hypothesizes a line using two randomly selected points. If the hypothesis is verified, all points along the line are removed and the algorithm is repeated to extract the remaining dominant lines in the image.
3. *Model Fitting.* After a set of lines have been extracted from the image, we need to know which line in the image corresponds to which line in the physical model. It may also be the case that lines are detected other than those present in the model or that some of the lines were not detected. This assignment is obtained with a combinatorial optimization, in which different configurations of lines are tired and evaluated. Finally, the configuration with the minimal cost is selected as the best one. Once these correspondences are known, the homography between real-world coordinates and the image coordinates can be computed (Fig. 2).

Our camera calibration is only executed in the first several frames, because the camera in our scenario is supposed to be fixed. For more details concerning the camera calibration algorithm, we refer to an earlier publication [7].

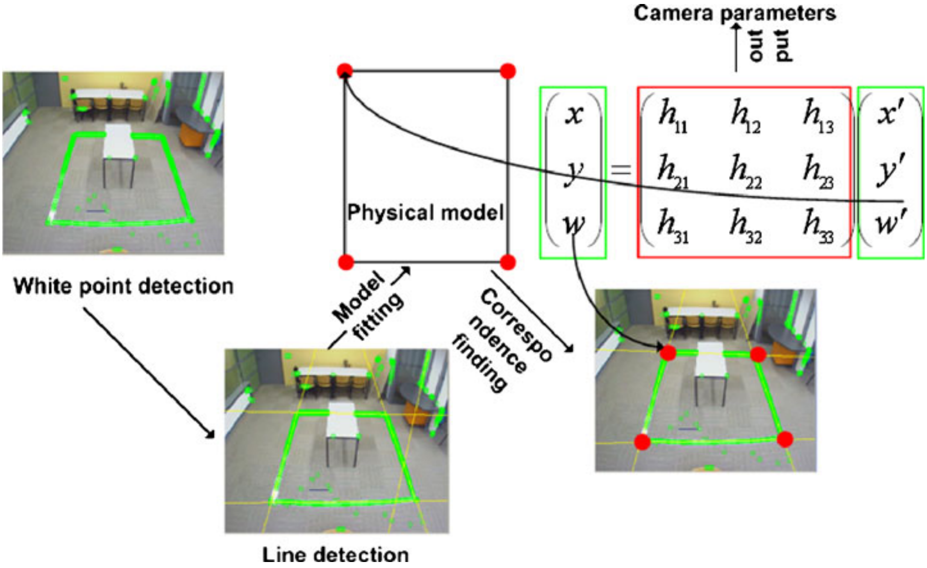


Fig. 2 Illustration of camera calibration, which consists of white point detection, line detection, model fitting and parameter computation

5 Texture-controlled object segmentation

5.1 Notation and basic model

Object segmentation can be treated by a Markovian-based Maximum A-Posterior (MAP) approximation. Given the observation d and the configuration of labels f, the posterior probability of f is

$$P(f|d) = \frac{P(d|f)P(f)}{P(d)}. \tag{2}$$

Maximizing $P(f|d)$ equals to maximizing the product of the class conditional probability $P(d|f)$ and the priori probability $P(f)$. Normally, we assume that $P(d|f)$ and $P(f)$ take the form of a Gibbs distribution, then the MAP problem can be solved by minimizing a Gibbs energy $E(f|d)$. For moving object segmentation, only the foreground label F and the background label B are considered. Energy $E(f|d)$ is composed of a node energy E_n and a smoothness energy E_s [1], such that

$$E(f|d) = E_n + E_s = \sum_i V_i(f, d) + \sum_{(i,j)\in C_i} V_{i,j}(f, d). \tag{3}$$

The node energy is simply the sum of a set of per-pixel node costs $V_i(f, d)$, and the smoothness energy E_s is the sum of spatially varying horizontal and vertical nearest neighbor smoothness costs. Here, C_i is the clique of location i. A key issue in the MRF model is the definition of the energy function. There are two definitions used in the segmentation algorithm, each being briefly explained below.

- **Spatial MRF model (SMRF).** SMRF only considers the spatial coherency of the object, where the smoothness energy E_s in (3) is developed in a spatial network, which consists of nodes connected by edges that indicate conditional dependency.
- **Spatial-Temporal MRF Model (STMRF).** In [18], a temporal continuity term has been added into the energy function, thereby allowing to maintain the coherency of the segmentation through time. Basically, if a region has been classified as foreground several times in the past, it is assumed to be classified as foreground again in the current frame. According to our experiments, the gain of this model is very limited in contrast to SMRF model, because the above assumption may not keep valid in some situations, especially for nonrigid object, such as human in our scenario.

5.2 Proposed texture-controlled MRF model

MRF-based algorithm achieves better object-segmentation performance by taking the object coherency in both spatial and temporal dimensions into account, in contrast to the methods that consider the pixels independently [7, 22]. In the presently existing algorithms, color/intensity and the relation of the pixel locations are important features for object representation, though they are insufficient to represent all types of objects. For this reason, in many cases such approaches fundamentally fail to precisely approximate the boundary of an object, as they do not directly consider texture, which is one of the most important perceptual attributes of any object. To address this problem, we propose a new MRF-based object-segmentation algorithm that seamlessly incorporates texture information as a pixel feature in the MRF framework. Our new system sequentially consists of four stages, which are addressed below.

5.2.1 GMM-based initialization

In our system, an adaptive Gaussian Mixture Model (GMM) for background subtraction [22] is employed to produce the initial foreground masks. This algorithm maintains a *Gaussian*-mixture probability function for each pixel separately, where the parameters for each *Gaussian* distribution are updated in a recursive way. This algorithm also involves a shadow-detection algorithm to remove shadow pixels.

5.2.2 Dynamic edge detection

Edge information is an indication of being an object boundary. Normally, there are two kinds of edge: static edge and dynamic edge. The former refers to the edge in the background part of the image, and the dynamic edge is caused by the moving object. Apparently, we are only interested in edges that caused by the moving object, so that dynamic edge is required to be detected. Since the background is known, a straightforward idea is to subtract the edge in the background image from the edge of the current image. However, the method based on this idea must accidently remove some dynamic edge pixels that belong to the static edge as well. This situation happens when the dynamic edge and static edge are overlapping somewhere due to the moving of the object. To address this problem, we define a dynamic edge pixel detector, which can attenuate the edge in the background while preserving the edge across foreground/background boundaries by considering the background model of the image as a reference. This detector is formulated as:

$$Dtexture(i) = tex(i) \cdot \frac{1}{1 + tex^B(i) \cdot exp\left(-\frac{|I(i)-I^B(i)|}{c}\right)}, \tag{4}$$

where i is the index of the pixel, and $tex(i)$ is the texture value of the pixel i in the current image by performing *Sobel* operator. Similarly, $tex^B(i)$ refers to the texture value of the pixel i in the background image. Parameters $I(i)$ and $I^B(i)$ represent the intensity value of the pixel i in the current image and in the background image, respectively. If $\left|I(i) - I^B(i)\right| \to 0$, the pixel i has a high probability to be a background pixel, and $exp\left(-\frac{|I(i)-I^B(i)|}{c}\right) \to 1$. Otherwise, it may belong to the edge caused by the foreground boundary, and $exp\left(-\frac{|I(i)-I^B(i)|}{c}\right) \to 0$. The parameter c is a constant, which is set to 50 in all our experiments. Figure 3 shows the comparison results by using simple subtraction method and our method. Obviously, most dynamic edges caused by foreground object are well preserved, but most static edges are greatly attenuated by our method. Differently, though simple subtraction method can remove the static edges successfully, it unfortunately deleted some dynamic edges.

5.2.3 Spatial-textural MRF model (StMRF)

We have found that the extracted object has no a sharp/clear boundary (either underestimating or over-estimating) if we employ the MRF models introduced above. The major reason is that they consider a boundary pixel as a normal pixel, whose energy is always affected by its neighboring pixels in the MRF framework. To solve this problem, we have involved texture information when designing the energy function, since texture is a good clue to detect an object boundary. More specifically, if a pixel is labeled as the boundary pixel by our texture analyzer, the relation energy covered by an edge energy term, between this pixel and its neighborhood is assumed to be smaller. This is because we assume that the boundary pixel more relies on its node energy, but has a reduced influence from its neighborhood. Conversely, we can reduce the influence of the boundary pixel on its neighborhood as well, thereby

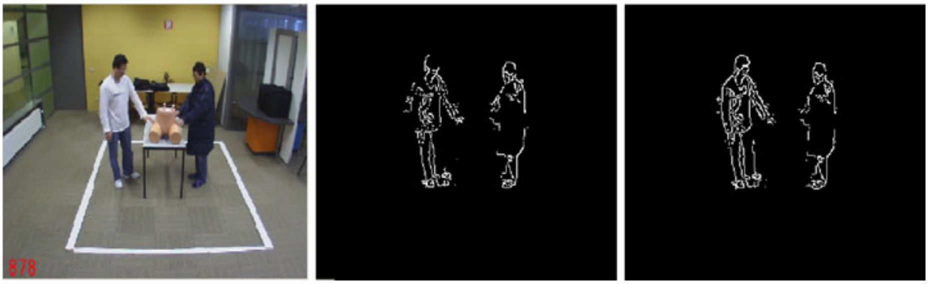


Fig. 3 Dynamic edge detector. *Left*: original image. *Middle*: dynamic edge by subtracting the edge in the background image from the edge of the current image. *Right*: our method

avoiding over-estimating boundary situations. Mathematically, the energy function is defined as:

$$E(f|d) = E_n^s + E_n^t + E_s^s + E_s^t, \tag{5}$$

where parameters E_n^s, E_n^t refer to the node energy in the spatial dimension and the node energy taking texture information into account, respectively. Since the temporal continuity term cannot bring a big gain, we do not consider it in our energy function design. Parameter E_n^s is defined as:

$$E_n^s = \sum_i V_i^s(f, d), \quad \text{and} \quad V_i^s(f, d) = \delta(f_i, F). \tag{6}$$

Here, $\delta(\cdot, \cdot)$ denotes the Kronecker delta function. The new parameter E_n^t refers to the node energy based on texture information, which is specified as:

$$E_n^t = \sum_i V_i^t(f, d), \quad \text{and} \quad V_i^t(f, d) = Dtexture(i)/255. \tag{7}$$

The smoothness energy E_s^s in (5) is developed in a spatial network, which consists of nodes connected by edges that indicate conditional dependency. Normally, E_s^s is modeled by a Generalized Potts distribution, which is specified by:

$$E_s^s = \sum_{(i,j)\in C_i} V_{i,j}^s(f, d), \quad \text{and} \quad V_{i,j}^s(f, d) = \delta(f_i, f_j). \tag{8}$$

The parameter E_s^t in (5) is the smoothness energy based on texture information, defined by

$$E_s^t = \sum_{(i,j)\in C_i} V_{i,j}^t(f, d), \quad \text{and} \quad V_{i,j}^t(f, d) = -0.5 \times \delta(I_i, I_j) \times Dtexture(i)/255. \tag{9}$$

5.2.4 Energy function minimization by dynamic graph cuts

Dynamic graph cuts [13] is proposed to solve any *similar* energy function. Consider two MRFs M_a and M_b, whose corresponding energy functions E_a and E_b differ by a few terms. Suppose the MAP solution of M_a can be available by solving the max-flow problem on the graph G_a and the energy E_a, and now we want to find the solution of M_b. The core idea of this dynamic concept is that we only find the MAP solution for the different regions between G_a and G_b in the residual graph, but reuse max-flow solution of G_a for the unchanged regions. Since G_b has only slight difference with G_a, the whole optimization procedure of MRF is significantly enhanced in execution, thereby resulting in a real-time algorithm.

6 Occlusion handling using nonlinear regression

In this paper, we also employ the silhouette-related image features to locate people during the occlusion. However, our technique is different from the method in [10] in two aspects: (1) we do not use natural vertices at the silhouette boundary of the people, since we have found that this image feature is too sensitive to the noise and the shape of the silhouettes; (2) unlike the method [10] that directly treats the projection histograms of the silhouettes, we firstly *model* the contour of the silhouette by a nonlinear regression algorithm. Afterwards, we search relative maximum points on a *smooth* curve which is automatically generated by the regression technique. In our model, each relative maximum point corresponds to one visual person in the occlusion. Our method is robust against the case where the silhouette of the person is fragmented, e.g., only a part of the head is segmented.

6.1 Locating the people during the occlusion

Once acquiring the segmented binary map of humans, we can obtain the contour of the upper part of the human by using:

$$C(x) = \max\{x \mid (x, y) \in Q\}, \tag{10}$$

where Q is the collection of the foreground pixels (x, y) on the binary map. After obtaining the contour map, the next step is to find the relative maximum points on it. Instead of directly searching on the map, we search relative maximum points on a *smooth* curve which is automatically generated by the regression technique. Therefore, our method is robust against uncompleted body silhouettes and the noises. Figure 4 illustrates the basic components of our complete occlusion-handling scheme, where two major components are people locating and people recognizing.

Given the contour $C(x)$, we intend to find a smooth curve (function) to model it. This problem can be solved by minimizing χ^2, which is the sum of the squared residuals over m points for the model $F(\mathbf{c}, x)$, hence

$$\chi^2 = \sum_{i=1}^{m} (C(x_i) - F(\mathbf{c}, x_i))^2 . \tag{11}$$

The parameters of the model are in the vector $\mathbf{c} = \{c_0, c_1, \ldots\}$. The model $F(\mathbf{c}, x)$ used in this algorithm is a *Gaussian* model, whose term number equals to the number of persons in the occlusion. The number of the person in the occlusion can be easily determined by the occlusion-detection component. For instance, if we know that there are n persons involved in the occlusion, the applied Gaussian model would have n terms, and be written as:

$$F(\mathbf{c}, x) = \sum_{j=1}^{n} c_{j1} exp\left(-\frac{(x - c_{j2})^2}{2c_{j3}^2}\right). \tag{12}$$

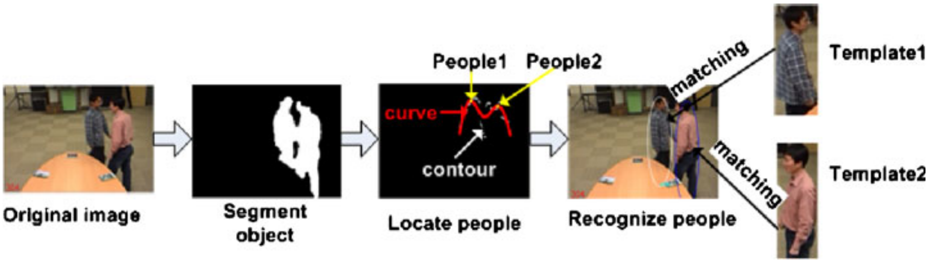


Fig. 4 The illustration of the complete occlusion handing scheme, where two major components are people locating and people recognizing

The Levenberg-Marquardt optimization algorithm helps to solve the minimization problem mentioned above, returning the best-fit vector $\mathbf{c}$ and their covariance matrix of the individual components. Once we obtain all the parameters of $F(\mathbf{c}, x)$, the next step is to find the relative maximum (peak) points on this curve, each corresponding visually to the head of one person. Our model is fully automatic, e.g., it can output two peak points when two persons are partially occluded, but provide only one peak point when they are completely occluded.

6.2 Recognizing the people during the occlusion

Apart from locating the person during the occlusion, it is also required to recognize the person. People recognition is actually a procedure that finds the correspondences between detected candidates and known templates. In our previous work [8], we have designed a template matching scheme based on mean-shift algorithm [4]. Assuming that there are N templates for N persons, and all the templates are generated and maintained by the mean-shift algorithm before the occlusion occurs. The probability of the feature $\{u_i\}_{i=1\ldots m}$ in the n^{th} template is $\hat{T}_n(u_i)$. Here, u_i represents the color histogram distribution and i denotes the bin number of the histogram. Furthermore, we use the same method to model the appearance of the blobs obtained by the people-locating module, and represent them by $\hat{W}_j(u_i)$. The aim of template matching is to find the best match to the template. Mathematically, finding the best match means maximizing the correspondence between the templates and the blobs in the occlusion, so that we compute

$$C_j = \arg\max_n \rho\left(\hat{W}_j(u_i), \hat{T}_n(u_i)\right). \tag{13}$$

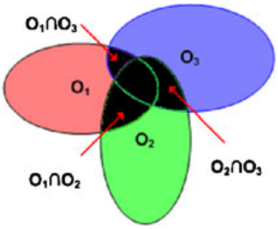
The term $\rho\big(\hat{W}_j(u_i), \hat{T}_n(u_i)\big)$ is a metric to measure the matching between the templates and the target candidate. For metric ρ, the Bhattacharyya coefficient [4] is used because of its performance, which is defined by:

$$\rho\left(\hat{W}_j(u_i), \hat{T}_n(u_i)\right) = \sum_{i=1}^{m} \sqrt{\hat{W}_j(u_i) \cdot \hat{T}_n(u_i)}. \tag{14}$$

If we look at our color distribution model more carefully, we can find that each pixel within the blob equally contributes to construct the model. This is absolutely correct for non-occlusion case. However, it is not optimal when having occlusion among objects, because some pixels belonging to one object must be overlapped by some other pixels of another object in this case. Apparently, if those overlapped pixels are still counted in the color model of this object, a large error would be generated. For this reason, we need a specific color model for occlusion case, which takes the object interaction into account. Our basic idea is that we trust non-overlapped area more, but attenuate the contributions from overlapped area when constructing color distribution model. More specifically, if a pixel belongs to non-overlapped region, it is expected to have bigger weight in the template-matching procedure. Otherwise, we assign a small weight to it. This idea can be formulated as:

$$\rho(\cdot,\cdot) = \sum_{i=1}^{m} w_i \sqrt{\hat{W}_j(u_i) \cdot \hat{T}_n(u_i)},\ and\ w_i = \frac{\hat{T}_n(u_i)}{\sum_{j=1}^{N} \hat{T}_j(u_i)}. \tag{15}$$

Fig. 5 Illustration of object interaction. We assign small weights to *black areas*



Here, w_i is a weight factor assigned to each pixel within the blob. If the pixel is not overlapped by others and its color does not appear in other objects, $w_i \to 1$, which degenerates to (14). On the other hand, if the pixel is a overlapped pixel and its color must appear in other objects, $w_i \to 0$, which means this pixel would give zero contribution to the matching metric. Our weight-based template-matching algorithm adequately considers the interactions between objects and adaptively assigns the weight to a pixel according to its possibility to be overlapped. A geometrical explanation is illustrated in Fig. 5, in which three persons are interacted with each other. In this example, we assign small weights to pixels in the black areas, which can be computed by (15).

7 Experimental results

Our proposed system is implemented using C++ on the Windows XP platform, and the computation time is measured on a 3 GHz Pentium 4 computer with 1 G RAM. We have tested the presented algorithms on a database, containing 15 videos. These videos are selected based on the suggestions from our hospital partner, which can represent most of the typical situations in a delivery training course. The length of each video varies from 2 min to 25 min. Two test videos and their ground truth data are found from public databases, which are used to evaluate our object segmentation and tracking algorithms. The rest videos are captured by a TRV30E video camera from a leading manufacturer.

7.1 Evaluation of camera calibration algorithm

We have tested our camera calibration algorithm using two video sequences, which were captured at our office (slightly different lighting conditions and view angles). Our algorithm is correct for almost 100% of the sequences, but only fails at several image samples, where one of the white lines is largely occluded by moving persons ($>$ 70% pixels are invisible). Fortunately, this false calibration can be easily detected by measuring the difference of camera parameters between two consecutive frames, because camera parameters in these two frames should be similar when the camera is fixed in the scene, which is indeed our case. Note that we only compute and refine the camera parameters using the first 10 frames of a sequence in our system, since the camera in our application is supposed to be fixed. Figure 6 shows example pictures, where the results of white-pixel detection, white-line detection and model fitting are illustrated. It is observed from the second example that our algorithm is robust against the situations, in which the moving person wares a cloth with white color or

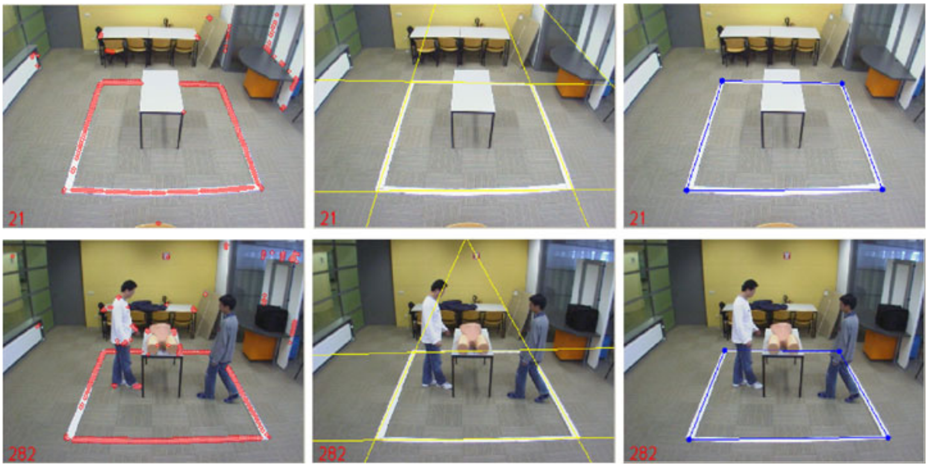


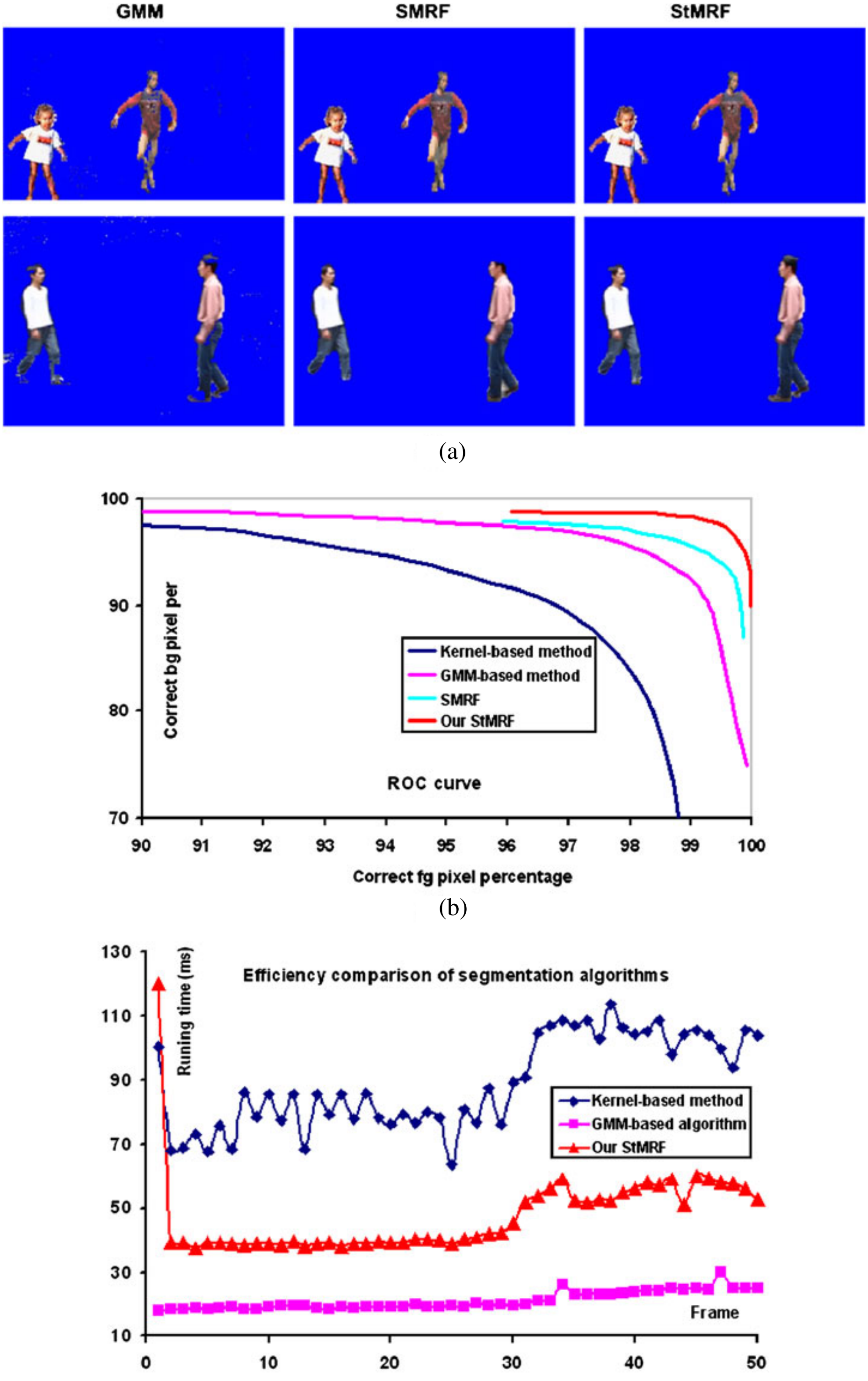
Fig. 6 Line detection-based camera calibration. *Left*: white point detection. *Middle*: line detection. *Right*: model fitting

the white lines are partially occluded by moving persons. As we mentioned before, this camera calibration algorithm reuses and simplifies the idea we proposed in the previous work. For more intensively experimental results, we refer to our earlier publication [7].

7.2 Evaluation of StMRF-based object segmentation algorithm

Our proposed object segmentation algorithm has been tested on two video sequences, where the first sequence with ground truth is downloaded from VSSN'05 (foreground detection competition) and the second one is captured in our office. Both sequences demonstrate the indoor environment with stable lighting condition, which is similar to the scenario of delivery simulation training course in the hospital. We have also compared our StMRF algorithm with Kernel-based algorithm [6], GMM-based algorithm [22], and the SMRF-based algorithm. The results are reported in Fig. 7, where the subfigure (a) gives several image samples of different methods, in which the results achieved by the GMM algorithm, SMRF model, and StMRF model are shown, respectively. It can be observed that foreground segmented by the GMM algorithm includes noises, which are not intended foreground objects. Though SMRF model filter noisy areas out, the boundary of the moving person is considerably less or over estimated by this model. Conversely, our proposed StMRF model effectively reduces the influence from the noise, while keeping a sharp boundary at object of interest (see foot and head areas of the image). The subfigure (b) draws the ROC curves of GMM algorithm, kernel-based algorithm, SMRF algorithm and also our StMRF algorithm. Obviously, StMRF is much better than the first two methods and slightly better than SMRF algorithm, which also reflects that

Fig. 7 Object segmentation results. **a** Image samples achieved by three algorithms. **b** ROC curves of four algorithms. **c** Efficiency comparison of three algorithms ▶


GMM
SMRF
StMRF
100
90
80
70
Correct bg pixel per
Kernel-based method
GMM-based method
SMRF
Our StMRF
ROC curve
90
91
92
93
94
95
96
97
98
99
100
Correct fg pixel percentage
130
110
90
70
50
30
10
Runing time (ms)
Efficiency comparison of segmentation algorithms
Kernel-based method
GMM-based algorithm
Our StMRF
Frame
0
10
20
30
40
50

(a)

(b)

(c)

our method is less sensitive to the thresholds used to determine the foreground pixels. For the ideal indoor scenario (fixed camera + stable lighting condition), our method can achieve $> 97\%$ accuracy. Finally, the subfigure (c) compares the execution time of each algorithm tested on the second video (320×240), where we can find that GMM-based algorithm is the fastest one. Both our algorithm and kernel-based algorithm require an *initialization* step in the first frame. Kernel-based algorithm uses this initialization step to estimate the background model, but our algorithm uses it to initialize the graph-cut optimization. It can be viewed that our method spends 120.4 ms on initialization and 45.1 ms per frame for the rest frames. The graph in Fig. 7c indicates that our dynamic concept-based system is close to a real-time system with small fluctuations between successive frames.

7.3 Evaluation of regression-based occlusion handling algorithm

The proposed occlusion handling algorithm has been tested on 11 video clips (each one has more than 1, 000 frames), where 8 videos are captured at our office including 2–3 persons, 2 videos are captured at hospital when a real delivery training course happening, and 1 video is downloaded from a public database (PETS2004). In our dataset, we have divided occlusion scenarios into two categories. The first category describes the situation that persons walk towards each other and immediately split after occlusion. The second category is that several persons join, and then walk circlewise after they meet. Obviously, the second one is more challenging, in which occlusion occupies more than 70% frames.

We compared our system with the mean-shift algorithm [4], the particle-filter algorithm (200 particles per object) [16] and also the algorithm proposed in our previous work [8], where we did not consider the interactions among objects when generating the appearance model for each object. Table 1 shows the system capability of treating the occlusion, in which NP is a simplified form of number of person. The ground-truth data was manually marked, whose evaluation criterion is that at least 70% of the human body is included in the detection window. To sum up, the mean-shift algorithm achieves an averaged 76.4% tracking accuracy during the occlusion events. The particle-filter algorithm achieves an averaged 92.6% accurate rate, our previous algorithm is correct for 88.7% frames when the occlusion occurs, and our current algorithm is correct for 89.9% frames in the same condition. The result has

Table 1 System performance comparison

	Occlusion type	NP	MS	PF	Han et al. [8]	Our algorithm
Clip1	Walk towards	2	100%	93.9%	100%	100%
Clip2	Walk towards	2	60.8%	85.9%	84.4%	85.1%
Clip3	Walk towards	2	100%	100%	100%	100%
Clip4	Walk towards	2	86.7%	97.2%	94.3%	95.1%
Clip5	Walk circlewise	2	80.6%	99.5%	95.6%	96.1%
Clip6	Walk circlewise	2	54.8%	99.2%	79.6%	82.2%
Clip7	Walk circlewise	2	100%	100%	100%	100%
Clip8	Walk circlewise	3	90.1%	81.1%	90%	90.5%
Clip9	Walk circlewise	3	70.2%	85.5%	93.1%	94.8%
Clip10	Walk circlewise	3	40.3%	89.8%	71.3%	74.7%
Clip11	Walk circlewise	3	56.7%	85.4%	70.1%	73.8%

proved that our weight-based template-matching algorithm has better performance when tracking people during the occlusion. Note that the most common failure was caused by the situation, where our people locating algorithm mistakenly indicates the position of each person. Figure 8 portrays four examples, where we show the tracking results before the occlusion, during the occlusion and after the occlusion. The first and third videos were captured at our office, the second one is from PETS2004 database and the last video demonstrates the real delivery training course at a hospital.

In addition to comparing the performance of these four algorithms, we have also evaluated their efficiency. Figure 9 gives the time consumed for each tracking algorithm during the *occlusion*, and also the execution times of the whole system. In the occlusion situation, the average execution times per frame for the tracking component of mean-shift, particle-filter, our previous work and our current technique are

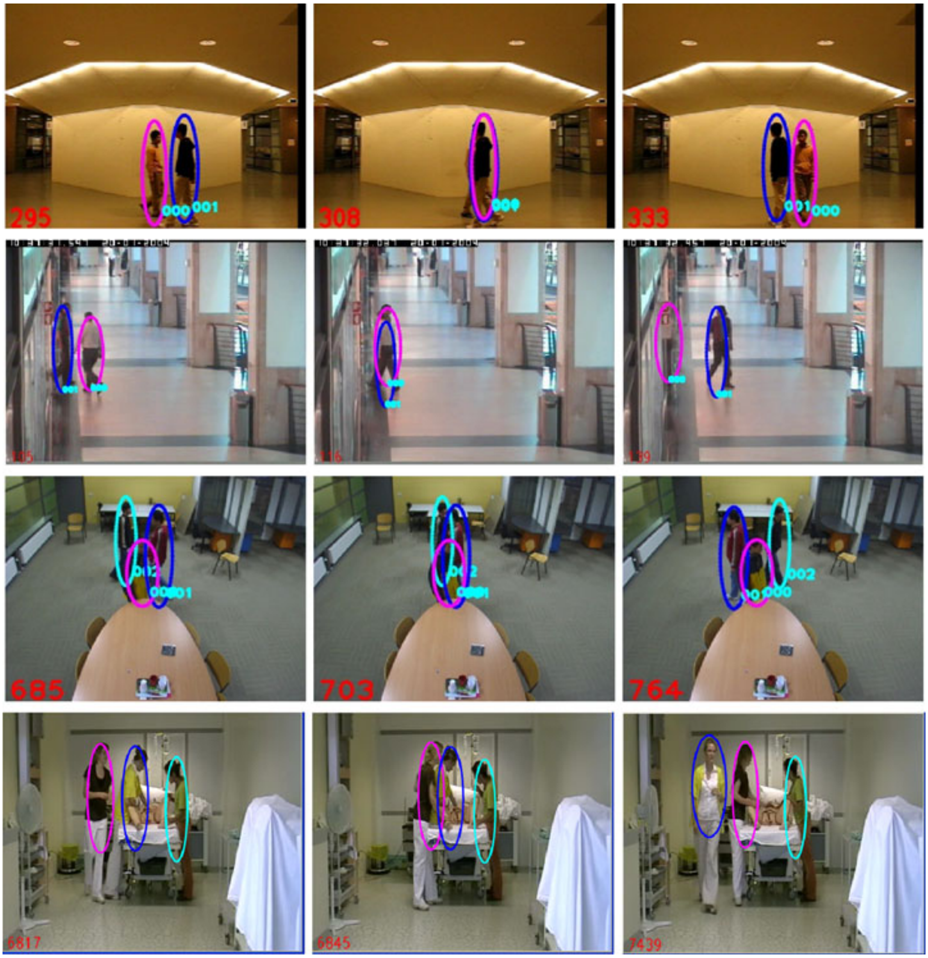
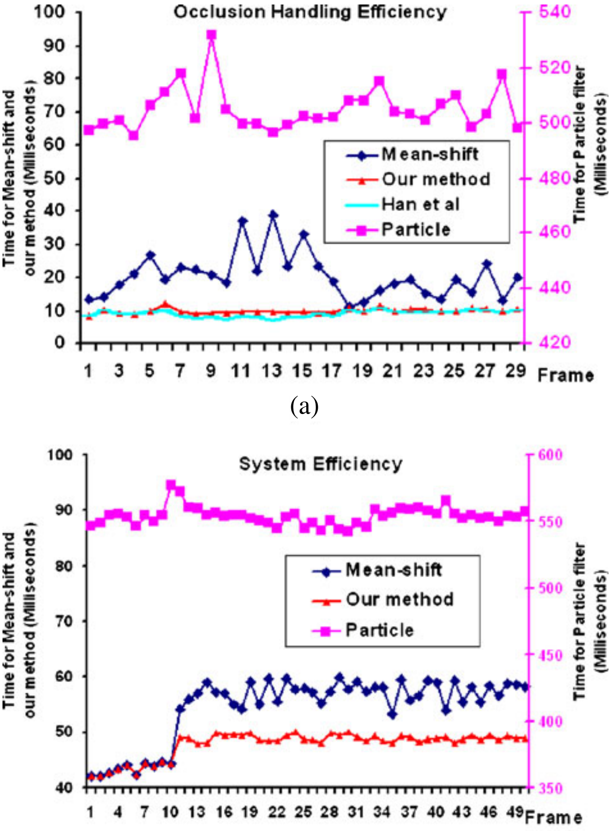


Fig. 8 Occlusion handling. *Left*: prior to occlusion; *middle*: during occlusion; *right*: after occlusion

Fig. 9 System efficiency. **a** Running time of the algorithm during the occlusion. **b** Running time of the whole system



(a)

(b)

20.45, 504.92, 9.12 and 9.87 ms, respectively. Apparently, our previous technique [8] is the most efficient algorithm. And, our current technique is slightly expensive than our previous work, because it requires to compute the wights for some overlapped pixels. Moreover, the execution times per frame for the complete system (mean-shift, particle-filter, and our current method) including object segmentation, object tracking and trajectory generation, are 54.52, 553.78 and 49.86 ms, respectively. The graph in Fig. 9 reveals that our proposal is a near real-time system with small fluctuations between successive frames.

8 Conclusion

In this paper, a new system for automatic analysis of the behavior of a small group of persons based on video signal has been introduced, thereby concentrating on three primary contributions. First, we take the texture information into account when designing the energy function for DMRF-based object segmentation, so that the silhouette of the object can be extracted accurately. Second, to address the occlusion problem, we have proposed an approach, where we model the silhouette-related image features into a feature vector which is optimized using a nonlinear regression algorithm. Therefore, the object position can be located through the finding of the peak points of a smooth curve when object occlusion occurs. The last contribution can be revealed that we embedded a camera calibration algorithm into our framework. By doing so, the real-world speed of each person can be achieved, which is invariant to the changes of camera position. It has been shown that our algorithm can operate at near *real-time* speed with around 90% tracking accuracy during the occlusion.

The future work will be the creation of the behavior-analysis phase based on the input visual features, such as real-world speed and trajectory of each person, which have been provided by the introduced system. Such a complete and automatic evaluation system will benefit the delivery training in two aspects. Firstly, it facilitates to find the mistakes made by trainees during the training course, allowing them to further develop their clinical skills. Secondly, automatic system is highly efficient with the increasing number of trainees.

Acknowledgements The authors would like to thank Mr. Minwei Feng for undertaking a part of algorithm implementation and evaluation work. We would also like to thank our hospital partner (Maxima Medisch Centrum of Eindhoven) for the kind cooperation.

References

1. Aach T, Kaup A (1995) Bayesian algorithms for adaptive change detection in image sequences using markov random fields. Signal Process Image Commun 7:147–160
2. Antonakaki P, Kosmopoulos D, Perantonis S (2009) Detecting abnormal human behavirour using multiple cameras. Signal Process 89(9):1723–1738
3. Chen D, Yang J, Malkin R, Wactlar H (2007) Detecting social interactions of the elderly in a nursing home environment. ACM Trans Multi Comp Comm Appl 3:1–22
4. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–577
5. Cutler R, Davis L (1998) View-based detection. In: Proc. Int. Conf. Patt. Reco., vol 1, pp 495–500
6. Elgammal A, Harwood D, Davis L (2002) Backgournd and foreground modeling using mon-parametric kernel density estimation for visual surveillance. Proc IEEE 90(7):1151–1163
7. Han J, Farin D, de With PHN (2008) Broadcast court-net sports video analysis using fast 3-D camera modeling. IEEE Trans Circuits Syst Video Technol 18(11):1628–1638
8. Han J, Feng M, de With PHN (2008) A real-time video surveillance system with human occlusion handling using nonlinear regression. In: Proc. Conf. ICME, vol 1, pp 305–308
9. Haritaoglu I, Harwood D, Davis L (1998) W4: who, when where, what: a real time system for detecting and tracking people. In: Proc. Conf. Face and Gesture Recognition, pp 222–227
10. Haritaoglu I, Harwood D, Davis L (2000) Real-time surveillance of people and their activities. IEEE Trans Pattern Anal Mach Intell 22(8):809–830
11. Hazelhoff L, Han J, de With PHN (2008) Video-based fall detection in the home using principal component analysis. In: Proc. Int. Conf. on Adva. Concep. for Intel. Visi. Syst, vol 1, pp 298–309
12. Jain R, Nagel H (1979) On the analysis of accumulative difference pictures from image sequences of real world scenes. IEEE Trans Pattern Anal Mach Intell 1:206–214

13. Kohli P, Torr P (2007) Dynamic graph cuts for efficient inference in markov random fields. IEEE Trans Pattern Anal Mach Intell 29(12):2079–2088
14. Koile K, Tollmar K, Demirdjian D, Shrobe H, Darrell T (2003) Activity zones for context-aware computing. In: Proc. Int. Conf. on Ubiquitous Computing, pp 90–106
15. Lao W, Han J, de With PHN (2007) A matching-based approach for human motion analysis. In: Proc. SPIE. Int. Conf. on Multimedia Modeling, vol 2, pp 405–414
16. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. Image Vis Comput 21(1):99–110
17. Oliver M, Rosario B, Pentland A (2000) A bayesian computer vision system for modeling human interactions. IEEE Trans Pattern Anal Mach Intell 22(8):831–843
18. Tsaig Y, Averbuch A (2000) A region-based MRF model for unsupervised segmentation of moving objects in image sequences. In: Proc. Int. Conf. on Compu. Visi. and Patt. Reco.(CVPR), pp 889–896
19. Wren C, Azarbayejani A, Darrell T, Pentland A (1997) Pfinder: real-time tracking of the human body. IEEE Trans Pattern Anal Mach Intell 19(7):780–785
20. Zhou Y, Tao H (2003) A background layer model for object tracking through occlusion. In: Proc. Conf. ICCV, pp 1079–1085
21. Zhou Z, Chen X, Chung Y, He Z, Han T, Keller J (2008) Activity analysis, summarization and visualization for indoor human activity monitoring. IEEE Trans Circuits Syst Video Technol 18(11):1489–1498
22. Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. In: Proc. Int. Conf. Pattern Recognition (ICPR), pp. 28–31

Jungong Han received the B.S. degree (with honor) in control and measurement engineering from Xidian University, China, in 1999. In 2004, he received his Ph.D. degree in communication and information engineering from Xidian University. In 2003, he has been a visiting scholar at Internet Media group of Microsoft Research Asia, China, with the topic on scalable video coding. Since 2005, he joined the department of signal processing systems at the Technical University of Eindhoven, The Netherlands, where he is leading the research on video content analysis. His research interests are content-based video analysis, video compression and scalable video coding. He is now involved in creating research programs to actively exploit content analysis in a few hospitals in the Netherlands for improving healthcare.

Peter H. N. de With (IEEE Fellow) graduated in electrical engineering from the University of Technology in Eindhoven and received his Ph.D. degree from the University of Technology Delft, The Netherlands in 1992. He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993, he was involved in several European projects on SDTV and HDTV recording. In this period, he contributed as a principal coding expert to the DV standardization for digital camcording. In 1994, he became a member of the TV Systems group at Philips Research Eindhoven, where he was leading the design of advanced programmable video architectures. In 1996, he became senior TV systems architect and in 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty Computer Engineering. In Mannheim he was heading the chair on Digital Circuitry and Simulation with the emphasis on video systems. Between 2000 and 2007, he was with LogicaCMG (now Logica) in Eindhoven as a principal consultant. Early 2008, he joined CycloMedia Technology, The Netherlands, as vice-president for video technology. Since 2000, he is professor at the University of Technology Eindhoven, at the faculty of Electrical Engineering and leading a chair on Video Coding and Architectures. He has written and co-authored over 200 papers on video coding, architectures and their realization. Regularly, he is a teacher of the Philips Technical Training Centre and for other post-academic courses. In 1995 and 2000, he co-authored papers that received the IEEE CES Transactions Paper Award, in 2004 the VCIP Best Paper Award and in 2006 the ISCE paper award. In 1996, he obtained a company Invention Award. Mr. de With is a Fellow of the IEEE, program committee member of the IEEE ICCE, ICIP and VCIP, board member of the IEEE Benelux Chapters for Information Theory and Consumer Electronics, scientific advisor to Philips divisions, and other companies, the Dutch Imaging school ASCII.