Fitting Multiple Connected Ellipses to an Image Silhouette Hierarchically

Richard Yi Da Xu, *Member, IEEE*, and Michael Kemp

***Abstract*—In this paper, we seek to fit a model, specified in terms of connected ellipses, to an image silhouette. Some algorithms that have attempted this problem are sensitive to initial guesses and also may converge to a wrong solution when they attempt to minimize the objective function for the entire ellipse structure in one step. We present an algorithm that overcomes these issues. Our first step is to temporarily ignore the connections, and refine the initial guess using unconstrained Expectation-Maximization (EM) for mixture Gaussian densities. Then the ellipses are reconnected linearly. Lastly, we apply the Levenberg-Marquardt algorithm to fine-tune the ellipse shapes to best align with the contour. The fitting is achieved in a hierarchical manner based upon the joints of the model. Experiments show that our algorithm can robustly fit a complex ellipse structure to a corresponding shape for several applications.**

***Index Terms*—Curve fitting, image edge analysis, image shape analysis.**

I. Introduction

ELLIPSE fitting is used in many computer vision and pattern recognition applications, ranging from modeling the human body parts [1], [2] to contour profiling [3]. The advantage of using ellipses in modeling is that ellipses can be easily parameterized and the parameters convey information including the location, orientation and variation of data. Furthermore, connecting ellipses can constrain their movements to reflect real-world problems, such as body gestures, figures and other non-rigid complex shapes.

In this paper, we seek to fit a model specified in terms of multiple connected ellipses to an image silhouette. The desire is to especially match the contour of the silhouette to the edges of the ellipses. The ellipses are required to be connected according to a prespecified structure after the fitting process is completed. In this paper, we choose the connections to occur at vertices of the ellipses. These requirements are useful in many practical applications. For example, if we were to represent the human body silhouette using multiple ellipses, then the connectivity of the ellipse's major axis can ensure the limbs and other body parts are joined appropriately. Furthermore, the connection between vertices allow us to extract extra information, such as angles between the ellipses, which is useful in many applications [4].

Fitting noisy data points to a single ellipse is a well-studied problem, and can be achieved using various methods including Direct Least Square Fitting [5] and Hough Transform alike [6] techniques. Fitting a connected set of ellipses to data points is not as trivial. One would have to ensure both the connectivity between the individual ellipses and alignment of the ellipses to the data points. One may also need to deal with membership association between points and individual ellipses.

A method which bears a close resemblance to our work is found in [2]: the authors use a k-means style algorithm to cluster data into two ellipses, head and body. Then individual ellipses are fitted to each cluster using the direct least square fitting method [5]. The clusters are recalculated and then the process repeated until the change in the ellipses/clusters drops below a threshold. The author's intention is only to extract the position of the head. Therefore, in order to simplify the algorithm the connectivity requirement between the two ellipses is not enforced (i.e., the orientation between the head and body at the joint is not relevant).

Another method of a similar nature is found in [7], where a generalized expectation maximization (GEM) approach is used to assign edge pixels to the body parts. In the M-step of the algorithm, the likelihood is increased by selecting a new (nearby) pose from a database of predefined poses. The authors avoid the explicit computation of the GMM structure, as well as the computation to ensure joints connections. This is because the databases contain only valid human figures, where the corresponding limbs are already joined. However, by limiting the solution space to only a discrete number of classes, misalignments can occur between the image data and the model, especially when the size of the database is small. We desire a method which allows us to compute an optimal parameter for any possible image silhouette given. A recent attempt is to use an iterative closest point (ICP) style method [15] for fitting ellipses, which involves two iterative steps: 1) segment the contour into their closest ellipse and 2) minimize the fit between the ellipse structure and the segmented points. Good results were reported, however, at a computational expense of using two nested iterations.

It is natural to present the problem of fitting as finding a solution for an ellipse structure (parameterized by $\mathbf{p}$) that minimizes the sum of squares of distances between the data points $\{\mathbf{x}_i\}$ and

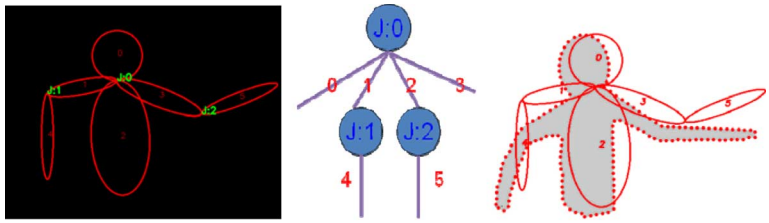
Manuscript received June 11, 2009; revised January 26, 2010. First published March 08, 2010; current version published June 16, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark Liao.

R. Y. D. Xu is with the School of Computing and Communications, University of Technology, Sydney, NSW, 2007, Australia (e-mail: yida.xu@uts.edu.au).

M. Kemp is with the School of Computing and Mathematics, Charles Sturt University, NSW, 2800, Australia (e-mail: mkemp@csu.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2045071



(a) (b) (c)

Fig. 1. (a) Six connected ellipses representation of a human upper-body, the green number indexes the connecting joints; (b) corresponding tree representation of (a); and (c) input image silhouette is superimposed with the initial ellipse model.

the ellipse structure. If we were to define a function, $fd\,()$, to compute the distance between a data point and ellipse structure, the problem can then be written as

$$\arg\min_{\mathbf{p}} \left\{ \sum_{i=1}^{N_{\text{point}}} (fd(\mathbf{p}, \mathbf{x}_i))^2 \right\}. \tag{1}$$

The authors in [4] use this approach, when they fit multiple ellipsoids to a cloud of noisy 3D hand points. The algorithm compares several nonlinear optimization methods to compute (1), including the geometrically constrained Levenberg-Marquardt (L-M) algorithm, as fingers can only swing around the joints within an angular threshold.

However, there are two problems associated with this general framework: The first one is the sensitivity to an accurate initial guess, which is a problem that commonly occurs for most numerical approaches to solve nonlinear least square systems. In [4], the paper acknowledges that without an accurate initial guess, "sometimes the algorithm does not find the right solution." The L-M algorithm with weak initial estimates can promote the issue of having the solution getting "stuck in a local minimum." The second challenge is that the objective function becomes complex which involves many parameters even with a few ellipses and joints. When minimizing the objective function, the solution often aligns with only a few of the ellipses in the structure and misaligns the others due to the axis-joining constraints. Consider the example illustrated in Fig. 1(a), which is a structure having six ellipses and three joints; it has 22 degrees of freedom. Therefore, becoming stuck in an inaccurate local minimum is a real concern. Our experiments show that when a connected ellipses structure involves four or more ellipses, this problem becomes particularly noticeable.

In this paper, we aim to deal with both challenges simultaneously: we propose to insert a preliminary step to improve the initial guess which reduces the effect of convergence into a false local minimum. This step needs to be efficient and much less sensitive to the initial guess, where the entire data points within the silhouette are used for modeling the shape to be "globally" aligned with the data. For fittings in general, we model the pixels within the contour with Gaussian mixture model (GMM) using the classical Expectation-Maximization (EM) approach and its efficiency is achieved by dropping the constraint of ellipse connectivity. Another reason for choosing GMM is that the mean and covariance matrix of an individual Gaussian Model has a direct correspondence with an ellipse's parameters. In general, GMM provides us with the desired improved ellipse model. For more specific image objects (such as a human body), heuristic approaches which exploit other features of the object, can be used as an alterative method to ellipse model's initial refinement. An example of heuristic-based model refinement is found in [13], where the contour is smoothed to various levels and the curvature is calculated. Points of maximum and minimum curvature then provide estimates of key points on the contour (fingertips, neck etc.). These key points are then used to obtain a set of unconnected ellipses. The details of this work can be found in [13]. As shown in Section V, within the domain of human upper-body pose fitting, the heuristic method provided us with an improved initial guess (and, hence, the final fitting) than using the general-purpose GMM method.

Second, in order to avoid using a single complex objective function involving the entire ellipse parameters, we choose to optimize only a subset of ellipses at a time. When trying to find the best fitting location for a joint that connects ellipses, only the neighboring ellipses and nearby points will have a strong effect on the fit. Hence, we optimize by considering each joint in turn and optimizing the objective function just for neighboring points. At each step we then have fewer degrees of freedom, therefore, the objective function is simpler and the chance to align all the ellipses is greater. The order of optimization of the joints is determined from an elliptical tree representation of the joint structure. In our work, the most interconnected joint is chosen to be represented as the ellipse tree's root node, and joints with the least numbers of connections with the rest of structure are represented as the leaf nodes. Although hierarchical optimization methods have already been used to solve other fitting problems in computer vision, its application to multiple connected ellipses is new. The hierarchical optimization is designed to divide complex problems into subproblems either in a bottom-up [8] or in the case of our work, a top-down approach. Another example of a top-down hierarchical optimization approach is [9]. The authors aim to find a transformation from some preknown model to a surface. They have a tree structure for specifying the type of transformation (e.g., affine). A top-down traversal is performed through the tree and optimization is based upon L-M at each node.

The rest of this paper is organized as follows: In Section II, we will describe our connected ellipse data structure, where joints and ellipses are represented by tree nodes and tree branches, respectively, to indicate the optimization hierarchy. In Sections III

and IV, we will describe the fitting algorithm in general, which is comprised of three stages: unconstrained Expectation Maximization of GMM followed by the linear connections of axis points and lastly fine-tuning the ellipse parameters by traversing the tree and for each node applies the L-M algorithm by minimizing the sum of squares of (approximate) distances between the contours to the corresponding subtree. This is illustrated in Section V. Finally, we compare our results with several existing approaches in Section VI.

II. Connected Ellipses Representations

The models we consider in this paper are multiple ellipses that are connected at the vertices on either the major or the minor axes. We assume that ellipses have at most two joints and are connected acyclically, which allows us to model the connections using a tree structure. Furthermore, to make the descriptions in this paper simpler, we will assume that if there are two joints on an ellipse, they will be on opposite vertices. Each node corresponds to a vertex of a connecting axis of an ellipse. Each branch connecting two nodes corresponds to an ellipse. A point that connects two or more ellipses is called a joint and corresponds to a node with children. For example, in Fig. 1(a), we show that a human upper body is modeled using six ellipses and three joints. The optimization is performed via a top-down traversal of the tree. Therefore, the more important joints, i.e., where there are more interconnections present are placed at a lower depth. In Fig. 1(a) and (b), for instance, the joint at the neck is placed at the 0th depth (the root-node) as it connects four ellipses together, whereas the elbow joints are placed at 1st depth.

Note that the tree structure is manually specified before image analysis. The problem of automatically determining the tree structure, i.e., model selection, is still open.

Each ellipse requires five parameters, which can be modeled in various ways. In our work, The first pair is the pointer to the joint of the parent node, x_j, y_j. The third and fourth parameters are the pointer of the joint of its child node: x'_j, y'_j. We will label the center of the ellipse to be x_c, y_c which can be calculated as the midpoint of the joints. The last parameter is the length of the other nonconnecting axis B. Note it is possible to modify the structure to allow vertices on adjacent rather than opposite vertices. However, this complicates the description of the algorithm for little gain in modeling power. Therefore, we will not consider this modification.

III. Ellipse Estimation Using E-M and Vertex Realignments

Prior to the application of our algorithm, we need both the image silhouette and a rough estimate for the ellipse model, as shown in Fig. 1(c). In many computer vision applications, the silhouette typically associates with the foreground object and can be obtained using the background subtraction [10] or alike methods if the camera is fixed. The method to obtain the rough initial ellipse model is application-dependent. Templates or heuristics are both reasonable choices. For example, for modeling a human pose, the PCA-based template matching [7] can be used.

Fig. 1(c) illustrates an example of a silhouette used to explain our proposed algorithm in this paper: the contour points are sampled at regular intervals and the initial configuration of the ellipse model is far from the accurate fitting. As mentioned in the introduction, using a minimization method at this stage leads to a poor fit [see also Fig. 11(a)]. Thus, as a preliminary step we use a Gaussian mixture model (GMM) and Expectation Maximization (EM) to improve the initial guess so that a minimization method is appropriate.

A. E-M Clustering on Silhouette Image

The goal of this stage is to focus on fitting the ellipses broadly and leaving precise curve matching to later. Thus, we use clustering based upon GMM, where the probability of a pixel location $\mathbf{x}$ *inside* the silhouette are thought to be

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{1}^{M} w_i \, G_i\left(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right). \tag{2}$$

The parameters, $\boldsymbol{\theta} = \{ (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, w_i) \}$, are the mean vectors and covariance matrices, $\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$, of an individual Gaussian probability density function $G_i(.)$ and w_i is the weight of each cluster. The number of clusters is M, which equates to the number of ellipses in the structure. We used a classical EM algorithm [11] to find the maximum likelihood estimates of all the parameters given the silhouette image.

Although the GMM produces a density estimate, there is a correspondence between 2D Gaussians and ellipses that we can exploit by using confidence regions. To illustrate this correspondence, consider a single Gaussian with mean, $\boldsymbol{\mu}_i$, covariance matrix $\boldsymbol{\Sigma}_i$ and density function $G_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Let $E_i(z)$ be a region of a solid ellipse given by

$$(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \leq z. \tag{3}$$

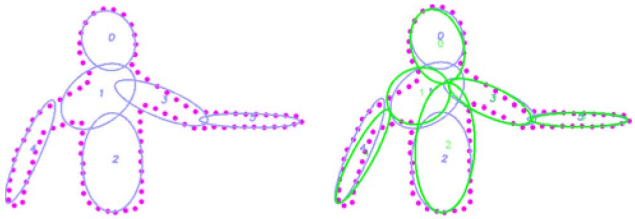
By using an orthogonal transformation to map this ellipse onto the coordinate axes and then using polar coordinates, we have

$$\int\int_{E_i(z)} G_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) d\mathbf{x} = 1 - \exp\left(-\frac{1}{2} z^2\right). \tag{4}$$

Therefore, $E_i(z)$ is a $\left(1 - \exp\left(-1/2z^2\right)\right) 100\%$ confidence region for the Gaussian. If the cluster that is used to generate the Gaussian is close to an ellipse and has many pixels then the level of confidence will correspond closely to the proportion of the cluster that will be in the confidence region. In order to avoid noise, we have chosen to use 99.5% confidence (other high levels would also be sufficient), which corresponds to letting $z = 3.26$. If the cluster is not close to an ellipse then correspondence naturally breaks down, but this is sufficient for a preliminary step. For a mixture density, the union of the confidence regions for each component gives the overall confidence region. Thus, a confidence region of the GMM is a collection of multiple ellipses.

B. Initializing the EM Algorithm

We initialize the EM algorithm by using the confidence region correspondence between ellipses and Gaussians. Explicitly, the initial GMM parameter vector $\boldsymbol{\theta}^{(0)} = \left\{ \left(\boldsymbol{\mu}_i^{(0)}, \boldsymbol{\Sigma}_i^{(0)}, w_i^{(0)} \right) \right\}$ is



(a) (b)

Fig. 2. (a) Showing the ellipse sets derived from final GMM parameters $(\boldsymbol{\mu}_i^{(f)}, \boldsymbol{\Sigma}_i^{(f)})$ obtained from the EM results. (b) Green ellipses set are the newly reconnected ellipses produced from the disjoint blue ellipses set of (a).

determined from the initial configuration of the ellipses, where each mean is computed from the corresponding ellipse center from its parent joint and joint to its children

$$\boldsymbol{\mu}_i^{(0)} = [x_c, y_c]^T. \tag{5}$$

Each initial covariance matrix $\boldsymbol{\Sigma}_i^{(0)}$ is computed using its two eigenvalues and associated eigenvectors. The two unit eigenvectors are determined from the joint and the center:

$$\mathbf{e}_1 = \left[\begin{array}{cc} \frac{x_j - x_c}{\lambda} & \frac{y_j - y_c}{\lambda} \end{array}\right]^T \quad \mathbf{e}_2 = \left[\begin{array}{cc} \frac{y_j - y_c}{\lambda} & \frac{-(x_j - x_c)}{\lambda} \end{array}\right]^T$$
$$\text{where}$$
$$\lambda = \sqrt{(x_j - x_c)^2 + (y_j - y_c)^2}. \tag{6}$$

The eigenvalues are:$\lambda_1 = (\lambda/z)^2$ and $\lambda_2 = (B/z)^2$ where $z = 3.26$ was explained previously. Each covariance matrix $\boldsymbol{\Sigma}_i^{(0)}$ is constructed using the spectral decomposition of a symmetric matrix

$$\boldsymbol{\Sigma}_i^{(0)} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T. \tag{7}$$

Finally, we set each initial weight $w_i^{(0)}$ to be the ratio of the area of the i^{th} ellipse to the sum of areas of all ellipses.

C. Reconstructing the Connected Ellipse Structure

The EM algorithm then iteratively calculates the most likely values for the parameters of the GMM by using all points of the contour interiors

$$\boldsymbol{\theta}^{(f)} = \left\{ \left(\boldsymbol{\mu}_i^{(f)}, \boldsymbol{\Sigma}_i^{(f)}, w_i^{(f)} \right) \right\}. \tag{8}$$

We then reverse the steps described in this section to convert the Gaussian parameters into ellipse parameters. Namely, each ellipse center is obtained from the 2D mean vector $\boldsymbol{\mu}_i^{(f)}$ and the ellipse's vertices are obtained from the eigenvalues (λ_1 and λ_2) and unit eigenvectors ($\mathbf{e}_1$ and $\mathbf{e}_2$) of the corresponding covariance matrix $\boldsymbol{\Sigma}_i^{(0)}$

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \pm z\sqrt{\lambda_1}\mathbf{e}_1, \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \pm z\sqrt{\lambda_2}\mathbf{e}_2. \tag{9}$$

In Fig. 2(a), we demonstrate the result of the EM step:

However, the ellipse sets obtained are not connected at this stage. We use a simple and effective method to solve this problem. We set the joint corresponding to the root node to be the average value of all the nearest vertices for all ellipses that should connect at that joint. We then traverse the ellipse tree top-down. For the other joints, we consider the vertex of its parent ellipse that is opposite the parent joint and all the nearest vertices for all its children ellipses. We set the value of the joint to be the average value of these vertices. During this traversal, we also set the value of B for each ellipse to be the length of the nonconnecting axis. Although simple, this is sufficient to seed the next part of the algorithm. The result of this step is shown in Fig. 2(b).

IV. Hierarchical Precise Fitting Using Levenberg-Marquardt Algorithm

The result obtained in step III (C) [Fig. 2(b)] shows the connected ellipses are fitted significantly closer to the edges of the enclosed silhouette than the initial guess shown in Fig. 1(c). This improved estimate provides the seed for the hierarchical distance minimization step which aims to move the edges of the ellipses even closer to the contour. We use the Levenberg-Marquardt (LM) algorithm to minimize the sum of squares of the approximated distances between the sampled points on the contour and the ellipse structure. The objective function used in L-M sums over a set of N points on the silhouette contour

$$\mathbf{p}_{(sub)L-M} = \arg\min_{\mathbf{p}} \left(\sum_{i=1}^{N} d^2(\text{tree}(\mathbf{p}), (x_i, y_i)) \right). \tag{10}$$

We will explain the distance function referred to here in Sections IV-A and IV-B.

A. Shortest Distance to Individual Ellipse Edge

The shortest Euclidean distance of a point from an ellipse does not have a simple formula. The next most popular distance, the algebraic distance has a high curvature bias [12]. Thus, we use the formula from [4] for the distance function which is a compromise between a simple formula and a reasonable approximation to the real distance. First, we define the distance between a point (x_j, y_j) to the closest edge of an ellipse (which is centered at (0, 0), with semiaxes lengths of a and b) to be

$$d\left(\text{ellipse}, (x_i, y_i)\right) = \begin{cases} d_1, & \text{if } (x_i, y_i) \text{ is inside of ellipse} \\ d_2, & \text{if } (x_i, y_i) \text{ is outside of ellipse} \end{cases}$$
$$d_2(x_i, y_i) = \sqrt{x_i^2 + y_i^2}\left(1 - \frac{1}{\sqrt{\left(\frac{x_i}{a}\right)^2 + \left(\frac{y_i}{b}\right)^2}}\right)$$
$$d_1(x_i, y_i) = r\left(1 - \sqrt{\left(\frac{x_i}{a}\right)^2 + \left(\frac{y_i}{b}\right)^2}\right) \tag{11}$$

where r is the half-length of the minor axis.

To obtain the distance for a general ellipse, we use an orthogonal transformation to center it at (0,0) and align the axes with the coordinate axes then use the previous formula. In Fig. 3, we show the plotting of $d\left(\text{ellipse}, (x_i, y_i)\right)$ of three ellipses for every point in the image plane.

B. Shortest Distance to an Ellipse Structure Set

We define a pseudo-distance function $d\left(tree(\mathbf{p}), (x_i, y_i)\right)$ that approximates the distance from a contour point (x_i, y_i) to

Fig. 3. Plot of $d\left(\text{ellipse}, (x_i, y_i)\right)$ over an image plane.

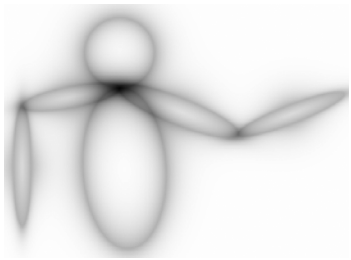


Fig. 4. Distances function according to the ellipse tree in Fig. 1(a).

PROCEDURE L-M Sub-Tree Optimization
INPUT: The current node of the ellipse tree

1. Determine the sub-tree consisting of the current node and all nodes one edge away.
2. Segment the contour points and extract only the ones close to the current sub-tree.
3. Perform L-M procedure on the sub-tree using only the chosen points.
4. Update the ellipse structure.
5. Fix the parameters in the current node.

FOR Every child the current node has

CALL **PROCEDURE** L-M Sub-Tree Optimization
INPUT: a child node

END

Fig. 5. Recursive algorithm for hierarchical ellipse tree fitting.

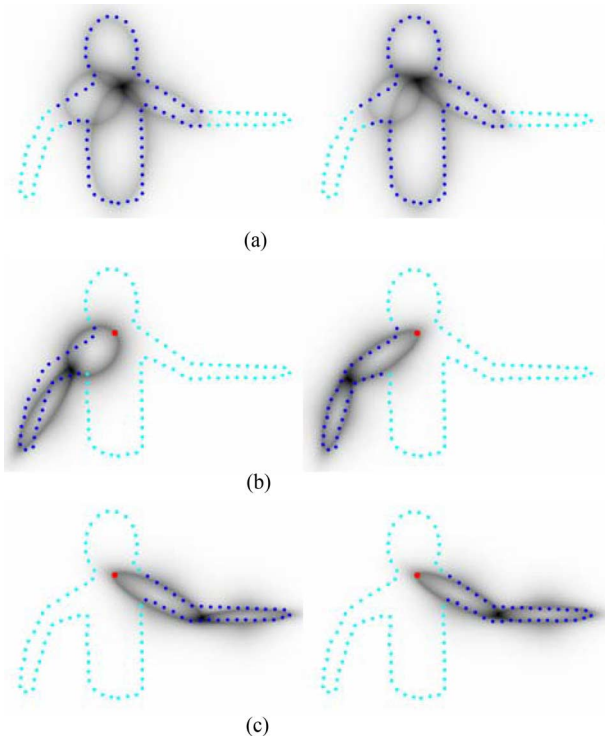


Fig. 6. Progressive result of hierarchical ellipse fitting using L-M algorithm. (a) Optimization result before and after at subtree: J-0 (root node); (b) optimization result before and after at subtree: J-1 (nonroot node); (c) optimization result before and after at subtree: J-2 (nonroot node).

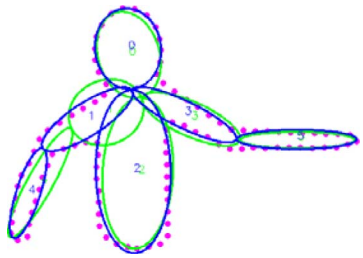


Fig. 7. Result of fitting the ellipse structure produced by the L-M subtree algorithm (blue) to fitting the contour (purple). For comparison, the ellipse structure produced from Section III-B is shown in green.

a set of ellipses in the joint tree (or subtree) structure $tree(\mathbf{p})$ to be

$$(tree(\mathbf{p}), (x_i, y_i)) = \frac{\sum_{l=1}^{M} 1 - \exp\left(-\eta d\left(\text{ellipse}(\mathbf{p})_l, (x_i, y_i)\right)\right)}{M.} \tag{12}$$

The rationale for this function is that

$$\exp\left(-\eta d\left(\text{ellipse}(\mathbf{p})_l, (x_i, y_i)\right)\right)$$

represents the proximity of a point to the ellipse which is highest when the point is on the ellipse and falls away sharply as the point moves away from the ellipse. The proximity is then added. If this total proximity is high then the point is close to at least one of the ellipses. We minus it from 1 in order to make the function increase as a point moves away from the ellipses (as is needed for the LM algorithm). We also normalize by M, the number of ellipse involved. The parameter η is called the decreasing factor and is discussed in Section IV-D.

The function $d(tree(\mathbf{p}), (x_i, y_i))$ is plotted in Fig. 4 using the entire ellipse structure in Fig. 1(a) over every (x_i, y_i) on the image plane. This function is not a true distance function since it is not zero for all points that lie on the ellipse structure. Nevertheless, it works in the LM algorithm. As Fig. 4 demonstrates, a point further away from any ellipse edges (both outside and inside of the ellipse) has a larger value indicated by the higher image intensity. Those points close to any of the ellipse edges have a smaller value.

C. Hierarchical L-M Fitting

As explained in the introduction: if we use the entire ellipse sets into the objective function in (12), then the solution is very likely to satisfy only a subset of the ellipses, but not all. More discussions on this topic will be presented in Section V.

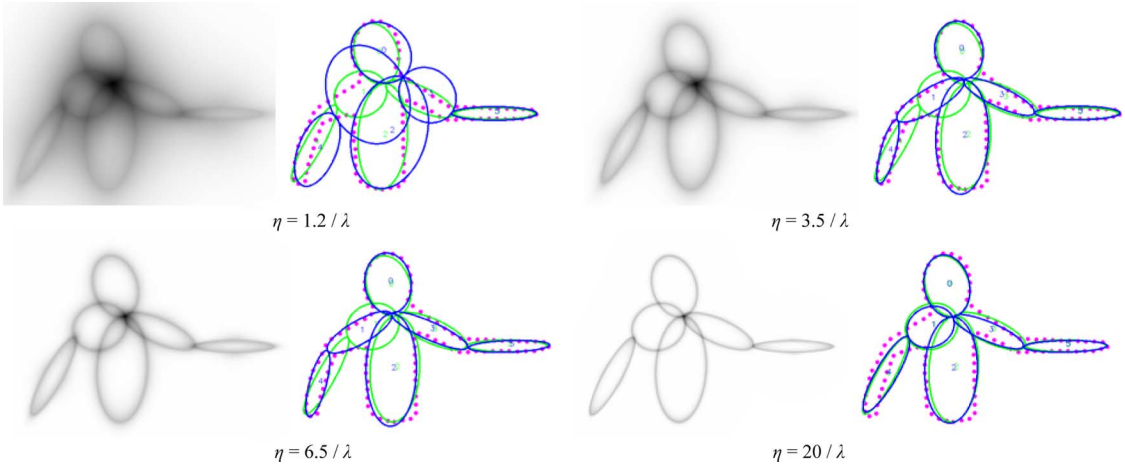


Fig. 8. Effect of different values of η. Plots of $d(tree(\mathbf{p}), (x_i, y_i))$ are on the left and the effect on the fitting result is on the right (the result of the L-M step is shown in blue; the result from Section III-B is shown in green).

Therefore, we aim to divide the optimization process into several L-M procedures, each using only a subset of ellipses at a time. The optimization occurs according to a top-down traversal of the joint tree structure. In order to ensure only local changes are made to the ellipses sets (since GMM provides us with a close approximation already), we only use in the L-M algorithm the contour points close to the current subset of ellipses. Using fewer points and parameters speeds up the process, as demonstrated in Table II. We determine these points by segmenting the contour points according to which ellipse they are closer to (by (11) and then including only the contour points associated with the current subset of ellipses. The algorithm is illustrated as follows:

At each stage of this algorithm, the L-M procedure is applied to the subtree consisting of the current node and all edge ellipses and their other joints. For the root node, the parameter vector includes the root's joint position plus the other three remaining parameters for each of its children ellipses. Therefore, the number of parameters used in the objective function is: $2 + 3M$, where M is the number of children ellipses the root node has. In the example given, the root node has 14 parameters. For a nonroot node, the L-M optimization also involves its parent edge ellipse as well. However, we fix the two parameters of the parent ellipse that correspond to the joint position to the parent's parent. The parent ellipse has 3 remaining parameters (including 2 for the current node and B which is stored with the edge). As before, each child ellipse has three remaining parameters. The total number of parameters, hence, is $3(M + 1)$, where M is the number of children ellipses the current node has. The algorithm iterates until reaching nodes with no children. The algorithm is explained using the ellipse structure example illustrated in Fig. 1(b). In this case, there are three joints and, hence, the optimization has three steps:

1) Subtree at J-0: the root node contains ellipses (1, 2, 3, and 4) without a parent node;
2) Subtree at J-1: the tree contains ellipse 4) and with also its parent ellipse 1);
3) Subtree at J-2: the tree contains ellipse 5) and with also its parent ellipse 2).

The progressive result at each subtree recursion is illustrated in Fig. 6.

The left image in Fig. 6 is the plot of $d(tree(\mathbf{p}), (x_i, y_i))$ using the current level ellipse subsets before L-M procedure starts. It is superimposed with the segmented contour points where this subset is closer to. The right hand side image shows the plot of $d(tree(\mathbf{p}), (x_i, y_i))$ when L-M procedure converges. As illustrated, the resultant ellipse sets align with the contour points significantly closer. In the case of nonroot node [inFig. 6(b) and (c)], we also plot the parent' parent's node in red. This indicates that although the parent ellipse participates in the L-M for a nonroot node, however, its other joint position is fixed during L-M. The final fitting result is illustrated in Fig. 7, where it shows a significant closer fit than produced by the GMM shown in Fig. 2(b).

D. Decreasing Factor η

Within the distance function, there is an expression that corresponds to the proximity of a point (x_i, y_i) to an ellipse $\text{ellipse}(\mathbf{p})_l$: $\exp(-\eta d(\text{ellipse}(\mathbf{p})_l, (x_i, y_i)))$. Within this, η is used to adjust how quickly the proximity decreases as a point moves away from the ellipse and, hence, how quickly $d(tree(\mathbf{p}), (x_i, y_i))$ increases. We have made $\eta = k/\lambda$ where k is a constant and λ is the average length of all the ellipse's semimajor axes computed at step III.C. We divide by λ so that the proximity is scale-invariant.

Fig. 8 shows the effect of varying k values on $d(tree(\mathbf{p}), (x_i, y_i))$. When smaller values are used, such as $\eta = 1.2/\lambda$, the distance function poorly discriminates between ellipses and the L-M algorithm is likely to converge to a poor fit. When a high k value is used, the distance function creates a sharp-falling valley around the ellipse edges, which means only the contour points already close to the ellipse structure will impact the L-M fine tuning. Hence, for $\eta = 20/\lambda$ the result from Section III-B (the green line) and the result after L-M fine-tuning is almost identical. We found the range between $\eta = 3.0/\lambda$ to $\eta = 8.0/\lambda$ produced the most ideal results. We have used $\eta = 5.0/\lambda$ for the rest of this paper.

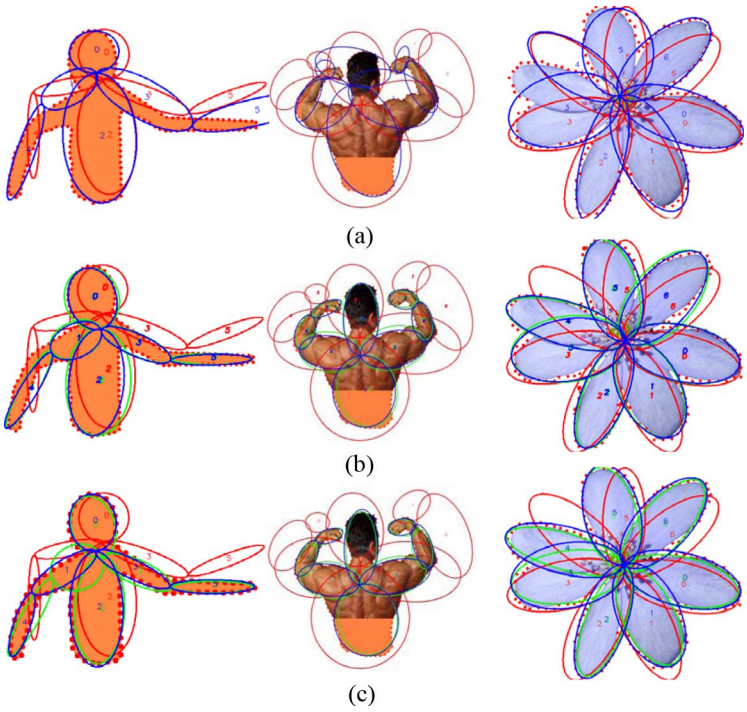


Fig. 9. Comparison of ellipse fitting methods used. (a) We attempted to apply L-M directly without GMM modeling (b) using single objective function for L-M (c) fitting ellipses hierarchically.

V. Results and Discussions

A. Measure for the Quality of the Fitting

In order to quantitatively measure the result of fitting, we define QoF to be the root mean of square (RMS) of distances between each point x_i and an ellipse indexed by $e(\mathbf{x}_i, \mathbf{p})$ (using (11) that enjoys the closest distance from the point $\mathbf{x}_i$.

$$QoF\left(\mathbf{p}, \{\mathbf{x}_i\}\right) = \sqrt{\frac{\sum_{i=1}^{N} d^2\left(e(\mathbf{x}_i, \mathbf{p}), \mathbf{x}_i\right)}{N}}. \tag{13}$$

For the ellipse structure and associated contour points given in the previous section, QoF of the rough initial structure is: 31.54. After using the GMM clustering and linear joining, the QoF is reduced to 8.99. After L-M fine-tuning, the QoF is further reduced to 4.94.

TABLE I
QoF for the Three Test Images and the Associated Ellipse Structures

Direct application of L-M:

	Rough Initial setting	Alignment after GMM	After L-M
Upper-body	31.54	N/A	10.53
Strong man	37.47	N/A	16.23
Flower	23.90	N/A	9.90

Single L-M Fitting:

	Rough Initial setting	Alignment after GMM	After L-M
Upper-body	31.54	9.56	8.83
Strong man	37.47	11.54	9.75
Flower	23.90	6.25	5.06

Hierarchical L-M Fitting:

	Rough Initial setting	Alignment after GMM	After L-M
Upper-body	31.54	8.99	4.94
Strong man	37.47	11.25	8.65
Flower	23.90	7.65	4.40

B. Results and Comparisons

As stated in the introduction, our proposed method addresses two issues: sensitivity to an inaccurate initial guess, and over complexity in the objective function making the optimization process unable to satisfy all ellipse parameters simultaneously. Our approach of reducing the sensitivity to poor initial estimates was to use GMM modeling as discussed in Section III. In order to demonstrate the effectiveness of this step, we omitted the GMM modeling and applied the hierarchical L-M algorithm directly. The results are shown in Fig. 9(a). Visually, the results obtained were far from satisfactory, except the flower image, where its initial guess was close.

To reduce the over-complexity of the objective function we used a subtree hierarchical approach which reduces the number of parameters used at each step. To show the effectiveness of this approach, we applied the L-M algorithm (12) on the entire elliptical tree structure at once. The result of such implementation is shown in Fig. 9(b). The result was better from skipping the E-M, but still has room for further improvements.

Finally, the result of our proposed work is shown in Fig. 9(c), which shows better alignments between the contour points and its ellipse edges compared with both previous approaches. In Fig. 9, the prespecified rough initial ellipse configurations are shown in red color, and the final fitting is shown in blue. The green ellipses in Fig. 9(b) and (c) are the results of the intermediate step, i.e., alignment after GMM described in Part III of this paper. Note that Fig. 9(b) is from our earlier work where the order which the ellipses are made connected is not according to the hierarchical tree structure as in the case of this

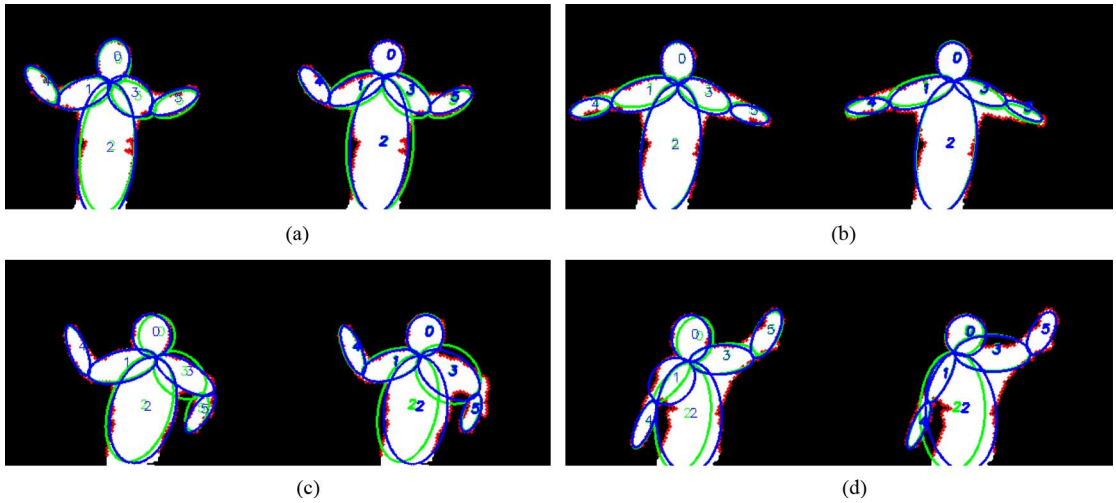


Fig. 10. Result of ellipse fitting on human upper-body images using proposed approach (left) and single L-M (right) for the four pose cases. The green ellipses are the initial estimate provided by curvature analysis and the blue ellipses are the final fit.

paper (Section III-B). Therefore the intermediate alignment results (green ellipses) vary marginally between Fig. 9(b) and (c) (hence, their corresponding QoF values also vary marginally). Please also note that although color images are used, they are simply treated as binary silhouette images. The corresponding QoF values for these images are shown in Table I.

The Quality of Fitting for the three approaches is shown in Table I. For each image, 150 points were sampled along the contour at a regular interval. The results demonstrated that the proposed method outperforms both earlier cases.

C. Computation Time

We also measured the complexity of our proposed algorithm in terms of its execution time. We implemented the algorithm in Visual Studio C++ (Debug Mode), using OpenCV's GMM implementation and LEVMAR's L-M implementation. The algorithm was executed on a DELL dual-core laptop running at 2.62 GHZ. We set the number of iterations for L-M algorithm to be 100. The execution times in milliseconds, for both the GMM and Hierarchical L-M components of the program are shown in Table II. For each test case, we also record the L-M execution times required for each subtree optimization. For example, for the "strong man" image, its ellipse structure has five joints and, therefore, we record the execution times for all of the five subtrees.

As Table II shows, the GMM's execution time is lower than that of the L-M algorithm, making it an ideal preprocessing step. The execution time for the L-M algorithm increases significantly when the number of ellipses associated with each subtree increases. For example, for the "flower" image, there is only one joint and, hence, only one L-M iteration. However this iteration needs to vary all 23 parameters of the structure and use all the contour points. In comparison, the "strong man" image has more ellipses, but its joint structure divides the L-M fitting into five steps. Each step has fewer parameters and fewer contour points. The reduction of the number of parameters at each step greatly reduces the execution time. This result is expected,

TABLE II
COMPUTATION TIME FOR THE THREE TEST IMAGES AND THE ASSOCIATED ELLIPSE STRUCTURES

	GMM	Hierarchical L-M, total execution time				
Upper-body	31	141	Sub-trees #	# of L-M Parameters	# of contour points	Execution time:
			1	14	92	110
			2	6	40	16
			3	6	50	15
Strong man	140	171	1	14	93	109
			2	6	27	16
			3	6	27	15
			4	6	34	16
			5	6	31	15
Flower	125	343	1	23	150	343

as the dimension the Jacobian matrix used in the L-M algorithm is determined by the number of parameters multiplied by the number of contour points. Therefore, our hierarchical approach is more computationally efficient than using all parameters of an ellipse structure during a single L-M optimization.

D. Application to Human Poses Detection

In addition to the previously shown test images, we also tested our algorithm on a mini-database containing twenty real human upper body foreground images. Simple preprocessing was used to remove noise pixels to ensure the foreground contained a single "clean" contour. Instead of using the general purpose GMM to improve upon the initial ellipse modeling parameters, for this experiment, we used a heuristic, curvature-based approach [13]. This approach exploits some of the features of the human upper body.

The same hierarchical fitting algorithm described in Section IV was then applied. We demonstrate the results on the left of Fig. 10. For the purpose of comparison, we also attempted fitting to this database by applying the L-M

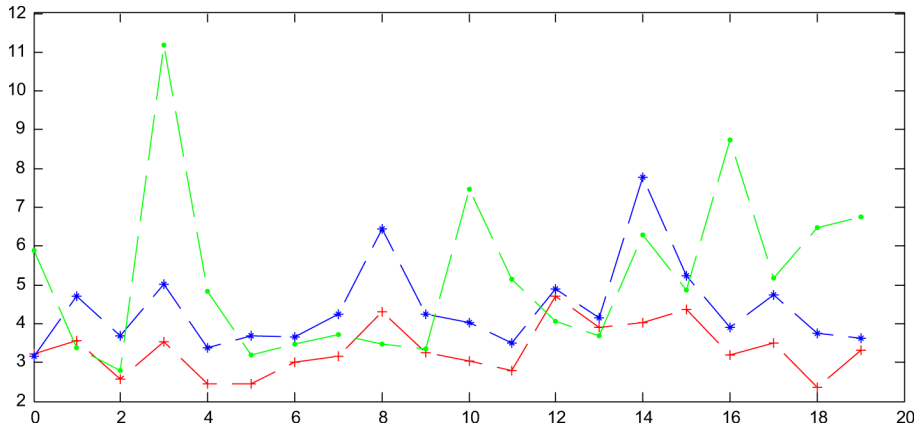


Fig. 11. Comparisons between QoFs: the red "+" and the blue "∗" represents the hierarchical approach and the single L-M algorithm approach, respectively, for the 20 poses in the database, where the heuristic initial refinement was used. The green "." represents the GMM initial refinement followed by the hierarchical fitting.

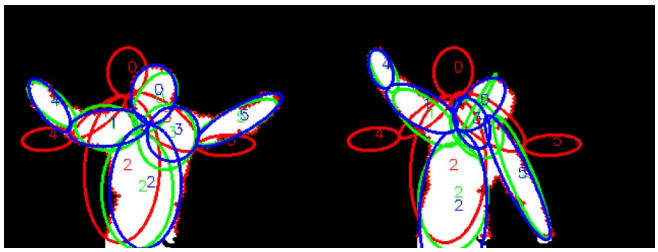


Fig. 12. Result of ellipse fitting on human upper-body images using the Hierarchical approach with GMM as the initial step. The red ellipse is a rough initial guess and set to be the same for all pose images. The green ellipses are the initial estimate provided by GMM. The blue ellipses are the final fit.

algorithm (12) on the entire elliptical tree structure at once. The corresponding images are shown on the right hand side of Fig. 10. The results of all 20 images can be found at: http://robotic-camera.com/tip_2010/

The initial estimates provided by curvature analysis are shown in green. Note that in [13], the order which the ellipses are made connected is not according to the hierarchical structure specified by the tree structure as in the case of this paper, therefore, the intermediate alignment results (green ellipses) vary marginally between the left and right figures.

In order to verify also the effectiveness of the heuristic-based initial ellipse refinement [13], we also performed the experiment where we used the general-purpose, GMM-based modeling as the initial refinement step, followed by the Hierarchical Fitting. In this case, all twenty test images use the same rough initial guess, shown in red color in Fig. 12. The green ellipses are the initial refinement and the blue ellipses are the final fitting result.

We computed the difference in QoF for each of 20 images in the database (as before), and plotted them in Fig. 11. Both visually and quantitatively in all cases, we notice that when the heuristic, curvature-based initial refinement was used, the hierarchical subtree approach (red line with "+") clearly outperforms the single L-M algorithm approach (blue line with "∗") except pose 0. In that case, the QoF is virtually the same.

We also plotted the QoF for the approach where GMM refinement was used followed by the Hierarchical fitting (green line with "."). The result in general is worse than the heuristic initial refinement followed by the Hierarchical fitting in almost all cases. This is because, for the GMM refinement, the same rough initial estimate was used in all cases. Thus, the GMM case is biased slightly towards poses similar to this generic pose. On the other hand, the heuristic initial refinement has no bias. The result is mixed when it is compared with the heuristic initial refinement followed by the single L-M approach:

VI. Conclusion and Areas for Further Research

In this paper, we presented a set of methods which we used to fit multiple connected ellipses to an enclosed silhouette contour, where only a rough initial model guess is given. We model the connected ellipses in a hierarchical tree structure, where ellipses joints with more interconnections are placed at a lower depth. We then convert the ellipse tree to a set of mean vectors and covariance matrices that serve as the initial parameters for a Gaussian mixture model which is optimized using the Expectation-Maximization algorithm. The final parameters produced by E-M procedure are converted back into the ellipse tree structure then their joints are linearly joined to ensure connectivity. As a final tuning step, we apply the Levenberg-Marquardt algorithm hierarchically to adjust the ellipse tree and ensure a closer match between the ellipse's outer edges with the points on the silhouette's contour.

We demonstrated the improvement of our results which overcomes both the challenges of weak initialization and complexity. The problem of weak initialization is addressed by using the Gaussian mixture model to refine the initial estimates. The issue of complexity is addressed by dividing the fittings recursively using less parameters and less points at each iteration.

Our future work is to concentrate on several areas. First, in this paper, we assume there is only one elliptical tree structure specified for image analysis. The problem of simultaneously selecting a tree structure (with varying number of ellipses and the way joints are specified) from a database and fitting to an image silhouette optimally is still open. Second, this paper concentrates on solving the minimization of distances in the image plane domain, in the future, we may also try to solve the problem by maximization in a probability domain. A preliminary work along this line of thought has already started using a variant of

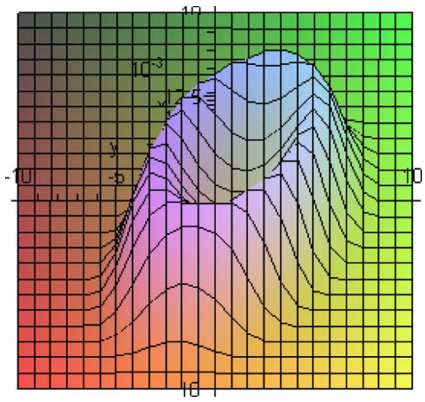


Fig. 13. Example of f, given the five parameters of ellipse to be, $\mu_{(1)x} = 2.0$, $\mu_{(1)y} = 1.0$, $\alpha_{(1)} = \pi/4$, $A_{(1)} = 5.0$, $B_{(1)} = 3.0$.

distance function where the function has peaks around the ellipse edges, and its plot is found in Fig. 13

$$f\left(x, y|\mu_{(l)x}, \mu_{(l)y}, \alpha_{(l)}, A_{(l)}, B_{(l)}\right) = \frac{1}{A_{(l)}B_{(l)}} \exp\left(-\left(\xi_{(l)} - 1\right)^2\right) \quad (14)$$

where

$$\xi_{(l)} = \frac{B_{(l)}^2\left(\cos\alpha_{(l)}(x - \mu_{(l)x}) - \sin\alpha_{(l)}(y - \mu_{(l)y})\right)^2}{A_{(l)}^2 B_{(l)}^2} + \frac{A_{(l)}^2\left(\sin\alpha_{(l)}(x - \mu_{(l)x}) + \cos\alpha_{(l)}(y - \mu_{(l)y})\right)^2}{A_{(l)}^2 B_{(l)}^2}.$$

Using appropriate normalization, we can ensure $f\left(x, y|\mu_{(l)x}, \mu_{(l)y}, \alpha_{(l)}, A_{(l)}, B_{(l)}\right)$ is altered into a probability density function (*pdf*). The connection requirement is thought to be solved using prior probability in a Bayesian framework.

From an application point of view, our current work is image feature (ellipse structure) detection on a single image. In the future, we also seek to explore further to apply this work to model a sequence of images. In such case, the prior knowledge from the previous ellipses fittings is also needed to be taken into consideration when fitting the current image. This is similar to many estimation problems using Kalman and Particle [14] Filters.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for helpful comments.

REFERENCES

[1] B. Kwolek, "Stereovision-based head tracking using color and ellipse fitting in a particle filter," in *Proc. Eur. Conf. on Computer Vision*, 2004, pp. 192–204.
[2] N. Grammalidis and M. G. Strintzis, "Head detection and tracking by 2-D and 3-D ellipsoid fitting," in *Proc. Int. Conf. Computer Graphics*, 2000, p. 221.
[3] A. Ciobanu, H. Shahbazkia, and H. d. Buf, "Contour profiling by dynamic ellipse fitting," in *Proc. 15th Int. Conf. Pattern Recognition (ICPR)*, 2000, vol. 3, p. 3758.
[4] F. L. Jeune, R. Deriche, R. Keriven, and P. Fua, Tracking of Hand's Posture and Gesture, CERTIS, ENPC [Online]. Available: http://certis.enpc.fr/publications/papers/04certis02.pdf2004
[5] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.
[6] L. Xu and E. Oja, "Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities," *CVGIP: Image Understanding*, vol. 57, pp. 131–154, 1993.
[7] A. Fossati, E. Arnaud, R. Horaud, and P. Fua, "Tracking articulated bodies using generalized expectation maximization," in *Proc. CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, Anchorage, AK, 2008, pp. 1–6.
[8] R. P. Würtz, "Organic computing methods for face recognition," *Inf. Technol.*, vol. 47, pp. 207–211, 2005.
[9] R. Szeliski and S. Lavallée, "Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines," *Int. J. Comput. Vis.*, vol. 18, pp. 171–186, 1996.
[10] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999, pp. 246–252.
[11] J. A. Bilmes, *A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Berkley, CA: International Computer Science Institute, 1997.
[12] P. L. Rosin, "A note on the least squares fitting of ellipses," *Pattern Recogni. Lett.*, vol. 14, pp. 799–780, 1993.
[13] R. Y. D. Xu and M. Kemp, "Multiple curvature based approach to human upper body parts detection with connected ellipse model fine-tuning," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Cairo, Egypt, 2009, pp. 2577–2580.
[14] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," *Comput. Vis. Pattern Recognit.*, pp. 126–133, 2000.
[15] R. Y. D. Xu and M. Kemp, "An iterative approach for fitting multiple connected ellipse structure to silhouette," *Pattern Recognit. Lett.* 2010, 10.1016/j.patrec.2010.01.017.

Richard Yi Da Xu (M'09) received the B.Eng degree in computer engineering from the University of New South Wales, Sydney, Australia, in 2000 and the Ph.D. degree in computer sciences from the University of Technology, Sydney, in 2005.

He is currently a senior lecturer at the School of Computing and Communications at University of Technology, Sydney (UTS). He has authored more than thirty international journal and conference papers in the area of computer vision algorithms and applications.

Michael Kemp received the B.S. degree in advanced science and the Ph.D. degree in pure mathematics from the University of Sydney, Australia, in 2000 and 2004, respectively.

He is a currently a Lecturer in Mathematics at the School of Computing and Mathematics at Charles Sturt University, Australia. He has been engaged in theoretical and practical research in the mathematics and statistics area.