Pattern Recognition 48 (2015) 2198–2208

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Geometrically-constrained balloon fitting for multiple connected ellipses

Michael Kemp [a,*], Richard Yi Da Xu [b]

[a] School of Computing and Mathematics, Charles Sturt University, PO Box 883, Orange NSW 2800, Australia
[b] Faculty of Information and Technology, University of Technology Sydney, 2007 Australia

ARTICLE INFO

Article history:
Received 18 February 2014
Received in revised form
24 December 2014
Accepted 27 January 2015
Available online 7 February 2015

Keywords:
Ellipse fitting
Constrained Gaussian mixture model
Expectation maximisation
Geometric constrained fitting

ABSTRACT

This paper presents a framework to fit data to a model consisting of multiple connected ellipses. For each iteration of the fitting algorithm, the representation of the multiple ellipses is mapped to a Gaussian mixture model (GMM) and the connections are mapped to geometric constraints for the GMM. The fitting is a modified constrained expectation maximisation (EM) method on the GMM (maximising with respect to the ellipse parameters rather than Gaussian parameters). A key modification is that the precision of the chosen GMM is increased at each iteration. This is similar to slowly inflating a bunch of connected balloons and so this is called balloon fitting. Extensions of the framework to other constraints and possible pre-processing are also discussed. The superiority of balloon fitting is demonstrated through experiments on several silhouettes with noisy edges which compare other existing methods with balloon fitting and some of the extensions.

1. Introduction

Ellipses are useful for modelling and assessing shape. Although non-linear they have simple equations and so are easy to work with. By coupling several ellipses together, complicated structures with joints can be modelled, for example the human upper body [1], the human hand [2] and fish [3].

In this paper we propose a framework for fitting a silhouette to a predetermined model of multiple connected ellipses. The framework utilises a natural equivalence of multiple ellipses with a Gaussian mixture model (GMM) to determine the fit using a variant of the expectation maximisation (EM) algorithm. The entire silhouette is used to avoid noise along the edges. A possible problem of the iterative methods used in the maximisation is that the algorithm could get stuck in a local maximum. To reduce this problem, we proposed a novel balloon fitting algorithm and apply it iteratively during the E-M procedure. To explain in words, the ellipses are considered as balloons that begin mostly deflated and then are inflated each iteration of the EM algorithm until they become full size (in terms of statistics, the variance of the GMM is increased at the beginning and then slowly reduced each step).

There already exist many ellipse fitting algorithms. Wong et al.'s survey article [4] shows the variety of methods and applications of ellipse fitting. There are two main approaches, least square methods and Hough transforms. The classic work involving least square methods is direct least square fitting in [5], however improvements are still being made [6,7]. Hough transforms use bins to determine the key parameters of ellipses. Although the five parameters could make the searching of all the bins costly, practical algorithms can be achieved using the parallel processing power of GPUs [8] or randomised Hough Transforms (for example [9]). Other tricks are to use Hough transforms to instead detect tangents and then reconstruct the ellipse from its tangents [10]. Both main approaches can readily be used in a scene with multiple ellipses (for example [6,8,9]).

However both of the main approaches to ellipse fitting are susceptible to noise when fitting to silhouettes. All of the cited ellipse fitting methods fit an ellipse curve to some data points. A common scenario (as used in [6,8,9,11,12]) is to fit multiple ellipses to a silhouette. A pre-processing step is used to determine the edge points of this silhouette and then apply the ellipse fitting to these edges. Even if the majority of the silhouette is reasonable, the edges can be difficult to obtain or can be badly affected by noise. In this paper, we overcome this problem by using the whole silhouette in the algorithm.

Ellipse fitting can also be applied where the multiple ellipses are coupled together, for example [1,2,11–13]. These citations all use edge (or boundary) values and so also are susceptible to noisy edges. The coupling of ellipses is achieved through geometric constraints. Least

[*] Corresponding author. Tel.: +61 2 6365 7840.
E-mail addresses: mkemp@csu.edu.au (M. Kemp), yida.xu@uts.edu.au (R.Y.D. Xu).

http://dx.doi.org/10.1016/j.patcog.2015.01.026
0031-3203/

squares methods can include these constraints, but force the least squares to be solved iteratively rather than directly [12]. The downside to numeric iterative methods is that they are sensitive to the initial guess and can get stuck in local optima when performing optimisation. For example, in [2], Jeune et al. fit multiple connected 3D ellipsoids to the edge of silhouette of a hand via an iterative Levenberg–Marquadt (LM) algorithm. The paper acknowledges that "sometimes the algorithm does not find the right solution". A common cause is because the LM algorithm gets "stuck in a local minimum". Weak initial estimates can promote this issue. The paper reduces the issue by using the context of hands to detect the false solution and hence fix it, however their solution is context dependent. In our previous work [11], we reduced the issue by initially segmenting the data and then hierarchically fitting the ellipses according to the joint structure. In this paper, we improved upon our previous work by using balloon fitting to further reduce the issue.

Ellipse fitting can also be done using statistical techniques. Wong et al.'s survey article, [4] classifies these as part of a third category – titled "Other" for short. One example of this framework is [14]. Points of the curve of the ellipse are modelled by Gaussians. Then maximum likelihood estimation (MLE) is used to fit the curve to the noisy data. The maximisation essentially reduces to a least squares method – minimising the reprojection error. A downside to this particular article [14] is that the method applies to all conics and so constraints are required to guarantee an ellipse.

A useful way to get a ellipse specific statistical method is to use the equivalence between ellipses and Gaussians. The articles [1,15] use this equivalence in 3D to model 3D ellipsoids using 3D Gaussians. In our previous work [11], we used this equivalence to cluster the data and improve the initial guess before using an iterative least squares method to refine the process. More precisely, multiple ellipses are equivalent to a GMM. We temporarily ignored the connections of the ellipses then used unconstrained EM to fit the GMM to the silhouette. We then connected the ellipses and did the constrained optimisation in the final step. One improvement given in this paper is to combine these steps by constraining the GMM and doing the constrained optimisation in the M-step of the EM algorithm rather than an entirely separate stage. Phrasing the entire algorithm in terms of a constrained GMM allows the variability to be modified during the optimisation. In a similar way to simulated annealing this can help the algorithm not get stuck in local optimums (which are not global optimums).

The EM algorithm has been used to fit a constrained GMM using maximum likelihood estimation (MLE) in a variety of ways. One simple technique to handle constraints is to run the M-step as normal and then project onto the space of feasible models if the solution does not satisfy a constraint (this approach does not guarantee the full maximum, but it is easy to implement). This is seen for *boundary constraints* where one wishes to constrain the parameters within certain intervals. Alternatively, some constraints merely require a change in the update equations for the M-step. For example with *linear subspace constraints* parameters are restricted to be within a linear subspace. This has been used for the covariance [16] and precision matrices [17]. Another example this applies for is *positive equivalence constraints* [18], where points are labelled as coming from the same class. However, other constraints require an numerical iterative M-step rather than a direct update of the unconstrained GMM. This includes *affine subspace constraints* [19] and *negative equivalence constraints* [18].

Another approach of incorporating constraints with GMM is to use Bayesian priors. Then the EM algorithm is used to find the maximum a posterior (MAP). This is equivalent to adding a penalty term in the M-step. One example of this avoids a singular covariance matrix by adding a prior to make the determinant of the covariance matrix be 0 with a low or zero probability. Another example is with *spatial constraints* [20]. Spatially adjacent points are deemed more probable to be from same mixture. This is achieved by adding a Markov Random Field (MRF) prior. Instead of having a hard constraint, like $\mathbf{c} = \mathbf{0}$, priors allow softer constraints where we only insist that the constraint is close to 0 with high probability. An example of this is the kinematic constraints in [15]. These methods often make the M-step not closed and so numerical optimisation is required.

Similar work to this paper are the kinematically constrained GMMs given in [1,15]. Kinematics is how joints move in a body. For the 2D case, kinematic constraints refer to the position of joints. In [1,15] they have a collection of connected ellipsoids. The size of the ellipsoids is determined manually at the start of the process, whereas the ellipsoids are allowed to be translated and rotated. They have kinematic constraints that join the ellipsoids at fixed locations and then allow movement at the joints (there are several options in a 3D model). The constraints in [15] are achieved by adding a prior so that each constraint equation is not precisely zero, but rather a Gaussian with zero mean. The variance for the constraints is fixed and not modified during the algorithm. Alternatively in [1] the M-step is modified to be a constrained optimisation. For both, the means and covariances (actually just the rotational part of the covariance matrices) of the components are learned using a modified EM algorithm. Although some parts have a closed form, a gradient ascent algorithm is required to handle all the constraints. This paper improves upon the ideas in these articles by getting the EM algorithm to learn the size of the ellipses as well and location and orientation. As with any iterative optimisation method, these previous papers can have issues with getting stuck in local optimums. This paper introduces the balloon fitting modification to reduce this issue. This modification is similar in purpose to simulated annealing where early stages of the algorithm have higher variability so that local optimums can be escaped.

The rest of this article is ordered as follows. In Section 2.1 we introduce our mapping between multiple ellipses and a GMM. Our general framework works regardless of the precise constraints used. However, to make the article concrete we present a reliable way to specify geometric constraints on the ellipses in Section 2.2. We then explain how expectation maximisation (Section 2.3) with the balloon fitting modification (Section 2.4) is used to fit the ellipse structure to an image silhouette. We then complete the method section by looking at some variations to the framework that could be used depending on the context in Sections 2.5 and 2.6. In Section 3 we demonstrate the effectiveness of the framework by comparing balloon fitting against [11,1,15].

2. Our method

2.1. Mapping ellipses to Gaussians

To fit multiple connected ellipses, we need a mapping from an ellipse to a Gaussian distribution. This mapping allows us to convert the ellipse fitting into a maximum likelihood estimate (MLE) for a GMM. We are interested in connected ellipses. Thus we will use the following notation.

We assume that there are M ellipses (or Gaussian mixture components), indexed by l. Each ellipse is parameterised by the following five dimensional set of parameters: $\Theta_E^l = (\mathbf{x}_v^l, \mathbf{x}_c^l, b^l)$. The first parameter, $\mathbf{x}_v^l = (x_v^l, y_v^l)$, corresponds to the location of a vertex of the ellipse. The second parameter, $\mathbf{x}_c^l = (x_c^l, y_c^l)$, corresponds to the centre of the ellipse. The last parameter corresponds to the length of the other semi-axis. For example, if $\mathbf{x}_v^l$ is a vertex on the major axis, b is the length of the semi-minor axis. Of course, this parameterisation depends on the choice of vertex.

Many other parameterisations of ellipses are possible and have been used elsewhere. However, we have selected this one to best suit the constrained optimisation we will perform later. When performing optimisation numerically, all parameters should ideally have a similar scale (so that changes in parameters should be the same order of magnitude). If they do not then the optimisation can perform badly. We experienced this ourselves when we used the angle of rotation as a parameter in an earlier implementation of the algorithm. The different order of magnitude of the change of the angles (a few radians versus distances in tens of pixels) caused the optimiser to not always converge. The five parameters selected here all refer to distances in pixels and so have comparable scales.

Each 2-D Gaussian mixture component is parameterised by $\Theta_G^l = (\boldsymbol{\mu}^l, \Sigma^l)$. The first parameter, $\boldsymbol{\mu}^l = (\mu_x^l, \mu_y^l)$, is the mean. The second parameter, Σ^l, is the covariance matrix. Since the covariance matrix is a symmetric 2×2 matrix, it is three dimensional. Thus Θ_G^l (just like Θ_E^l) is five dimensional.

We can now define a transformation from an ellipse to a 2-D Gaussian. Two dimensional Gaussians are naturally associated with ellipses. Namely, given the Gaussian with parameters $\Theta_G^l = (\boldsymbol{\mu}^l, \Sigma^l)$, then we can define, for positive Z, solid ellipses $E^l(Z)$:

$$(\mathbf{x} - \boldsymbol{\mu}^l)^T (\Sigma^l)^{-1} (\mathbf{x} - \boldsymbol{\mu}^l) \le Z^2.$$

(By taking a double integral, then aligning to the coordinate axes and finally using polar coordinates), the probability that the Gaussian covers the region $E^l(Z)$ is $(1 - \exp(-1/2Z^2))100\%$. Thus Z is linked to the quantile of the Gaussian. For instance, Z=2 gives the 86.5% confidence region.

Broadly speaking the transformation we will use is the opposite of this map for a chosen Z. It will map an ellipse $E^l(Z)$ onto the Gaussian with parameters Θ_G^l. We can give this transformation explicitly as well.

We denote the transformation as

$$\Theta_G = \mathcal{F}_Z(\Theta_E) \tag{1}$$

Firstly, $\boldsymbol{\mu} = \mathbf{x}_c$. Secondly, we determine the directions from the centre of the ellipse to the vertices (which become the unit eigenvectors of the covariance matrix):

$$\mathbf{e}_1 = \left(\frac{x_v - x_c}{a}, \frac{y_v - y_c}{a} \right), \quad \mathbf{e}_2 = \left(\frac{y_v - y_c}{a}, \frac{-(x_v - x_c)}{a} \right) \tag{2}$$

where

$$a = \sqrt{(x_v - x_c)^2 + (y_v - y_c)^2}. \tag{3}$$

The eigenvalues of the covariance matrix are $\lambda_1 = (a/Z)^2$ and $\lambda_2 = (b/Z)^2$. Then using spectral decomposition we have

$$\Sigma_Z(\Theta_E) = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T. \tag{4}$$

The value of Z controls the variability of the Gaussian and hence controls the how flexible the fit is. Notice that $\Sigma_Z(\Theta_E) = (1/Z^2)\Sigma_1(\Theta_E)$ and $\Sigma_Z(\Theta_E)^{-1} = Z^2 \Sigma_1(\Theta_E)^{-1}$. Therefore smaller values of Z give slow decaying Gaussians. This gives room to move and thus allows the fit to be adjusted easily. The higher variability also reduces the chance of getting stuck in undesired local maxima. We will use this case early in our algorithm to quickly improve our initial fitting. On the other hand, larger values of Z give sharp decaying Gaussians. This is useful when the initial fit is already good, and we wish to improve it. We will use this case at the end of our algorithm.

We use Gaussians to model the data because of their simple mathematical properties; their connection to ellipses; and because they handle noise well. However, the data we try to fit may not correspond exactly to a Gaussian, but something closer to a uniform distribution over an ellipse or near ellipse shape with a little noise. This mismatch fortunately matters little. The Gaussians have a great ability to account for the noise. Locations and orientations of the different components are all estimated well. The only issue is that the scale of the covariances can be out. The impact of this is that the ellipses may not match the natural edge of the silhouette. For intermediate steps of the algorithm this is not a problem, but it is critical for the final steps. However, we can fully control the scale of the covariances through the Z parameter. By selecting an appropriate final value for Z and Z_{max}, we can make sure that the final fit of the ellipses corresponds to the boundary of the silhouette. The value of Z_{max} can be learned for a given model, by finding Z_{max} that minimises the distance between the boundary and fitted GMM structure. Alternatively it can be calculated by using theoretical information of the actual distribution of each component (e.g. uniform over an ellipse). The mismatch between the Gaussian and the underlying distribution corresponds to the difference between the covariance matrix corresponding to the ellipse closest to the boundary (that is $\Sigma_1(\Theta_E)$ in the above notation) and the covariance of the distribution (which can be found via integration). The ratio of the components of these two matrices will give Z_{max}^2 (the orientations will match up). For example, if the underlying components are all uniform distributions with ellipses as support, then $Z_{max} = 2$. If the underlying components are rectangles, then choosing the ellipse which minimises the Mahalanobis distance to the boundary gives $Z_{max} = \sqrt{152} = 1.94$. We recommend using a value of $Z_{max} = 2$ as a default.

The transformation readily extends to multiple ellipses:

$$\mathcal{F}_Z(\{\Theta_E^l \mid l = 1 ,..., M\}) = \{\mathcal{F}_Z(\Theta_E^l) \mid l = 1 ,..., M\} \tag{5}$$

Placing constraints on the ellipses automatically implies constraints on the Gaussians. Thus fitting multiple connected ellipses can be achieved by working with a constrained GMM. For short, we write Θ_E to represent all the ellipse parameters – including the constraints placed on them.

2.2. Details of ellipse structure constraints

The framework below works with many types of constraints. The precise constraints used to enforce the connectivity of the multiple ellipses do not affect the general framework. However, to make this paper more concrete we will present two related ways to specify the geometric constraints.

In general the constraints we will consider match a point relative to one ellipse to another point relative to a second ellipse. For example, the points could be vertices of the respective ellipses. In order to specify locations of the points, we will use local coordinates (ϕ, d). Namely, for an ellipse with parameters $\Theta_E = (\mathbf{x}_v, \mathbf{x}_c, b)$, the local coordinates (ϕ, d) define the point:

$$p(\phi, d, \Theta_E) = \mathbf{x}_c + d \cos \phi \ a\mathbf{e}_1 + d \sin \phi b \mathbf{e}_2, \tag{6}$$

where $\mathbf{e}_i$ and a are given by Eqs. (2) and (3). An illustration of the local coordinates appears in Fig. 1.

The first coordinate, ϕ, corresponds to the angle around the ellipse. Multiples of $\pi/2$ give the vertices; multiples of π give vertices on the same axis as $\mathbf{x}_v$. The second coordinate, d, is the Mahalanobis distance: if d=1, then the point is on the boundary of the ellipse; $0 \le d < 1$: the point is inside; $d > 1$: the point is outside. [If one wishes to keep the modelling simple, then use $\phi = 0$ or π and d=1.]

So we can say there is a joint between ellipse i at location ϕ^i, d^i and ellipse j at location ϕ^j, d^j via the following geometric constraint:

$$p(\phi^i, d^i, \Theta_E^i) = p(\phi^j, d^j, \Theta_E^j) \tag{7}$$

We can use these geometric constraints to reduce the total number of parameters. A systematic way to approach this is to specify the connections in terms of a tree. The nodes of the tree are the ellipses. The edges of the tree correspond to the geometric constraints. The root node will have the standard five degrees of freedom. For all other ellipses (indexed by j), we can determine $\mathbf{x}_c^j$ by making that the subject of Eq. (7). This then gives 3 degrees of freedom for all other ellipses. This reduction in the total number of degrees of freedom makes the maximisation step quicker.

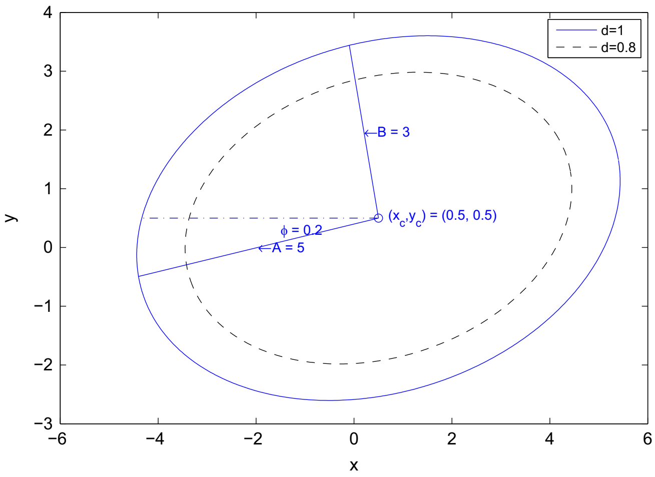


Fig. 1. An illustration of local coordinates of an ellipse. The ellipse (in blue solid line) corresponds to $d=1$. Other values of d represent other ellipses scaled about the centre. For instance, $d=0.8$ corresponds to the black dashed ellipse. Values of ϕ correspond to rays from the centre that make an angle of ϕ from the major axis (of half-length of A in the diagram). The ray $\phi=0.2$ is the dot-dash blue line in the diagram. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

The above approach treats the connections as hard constraints. An alternative is to soften the constraint and instead only insist that $c_{ij}=p(\phi^i,d^i,\Theta_E^i)-p(\phi^j,d^j,\Theta_E^j)$ is close to 0 with high probability. This can be achieved using a Gaussian prior of the distance of c_{ij}. This will be explained further below. The downside to the use of these soft constraints is that there is no reduction of the degrees of freedom. The upside is that ellipses have more freedom to move (which is especially useful for the earlier steps of our algorithm below).

2.3. Maximum likelihood estimation (MLE) using expectation maximisation

By using the mapping of ellipses onto Gaussians, the fitting of the multiple ellipses then becomes that of finding the MLE for the GMM. It is well known that direct maximisation is intractable [21]. Therefore, it has to be resolved using an iterative method. The expectation-maximisation is a standard solver in this context that begins with an initial guess, $\Theta_G^{(t)}$, for the parameters.

The first step of the expectation-maximisation is called the E-step. For a mixture model, we have a hidden random variable which denotes the particular component of the mixture a point comes from. For simplicity, we will denote this variable just by the index, l. Additionally we have matching parameters, α_l, which correspond to the probability of coming from the lth component of the mixture. In the E-step we calculate an expectation over these hidden variables. For a mixture model the expectation is [21]

$$\begin{aligned}Q(\Theta_G,\Theta_G^{(t)}) &= \sum_{l=1}^{M}\sum_{i=1}^{N}\log(\alpha_l p_l(\mathbf{y}_i|\Theta_G^l))p(l|\mathbf{y}_i,\Theta_G^{(t)})\\ &= \sum_{l=1}^{M}\sum_{i=1}^{N}\log(\alpha_l)p(l|\mathbf{y}_i,\Theta_G^{(t)})\\ &\quad+\sum_{l=1}^{M}\sum_{i=1}^{N}\log(p_l(\mathbf{y}_i|\Theta_G^l))p(l|\mathbf{y}_i,\Theta_G^{(t)}) \qquad (8)\end{aligned}$$

where $\mathbf{y}_i$ for $i=1,\ldots,N$ are the 2-D data points, and $p_l(\mathbf{y}_i|\Theta_G^l)$ is the probability density function (pdf) of the lth component. For a mixture model, the E-step calculates the responsibilities [21]:

$$p(l|\mathbf{y}_i,\Theta_G^{(t)})=\frac{\alpha_l^{(t)}p_l(\mathbf{y}_i|\Theta_l^{(t)})}{\sum_{k=1}^{M}\alpha_k^{(t)}p_k(\mathbf{y}_i|\Theta_k^{(t)})} \qquad (9)$$

The second step of the EM algorithm, the M-step, involves the maximisation: $\Theta_G^{(t+1)}=\operatorname{argmax}_{\Theta_G}(Q(\Theta_G,\Theta_G^{(t)}))$. Only the second sum of Eq. (8) contains Θ_G, thus we need to maximise only that sum. After substituting in the expression for the Gaussian pdf (itself in terms of the ellipse parameters), we generate the expression that we wish to maximise

$$\sum_{l=1}^{M}\sum_{i=1}^{N}\left(-\frac{1}{2}\log(|\Sigma_Z(\Theta_E^l)|)-\frac{1}{2}(\mathbf{y}_i-\mathbf{x}_c^l)^T(\Sigma_Z(\Theta_E^l))^{-1}(\mathbf{y}_i-\mathbf{x}_c^l)\right)p(l|\mathbf{y}_i,\Theta_G^{(t)}) \qquad (10)$$

Usually for the M-step for GMM, maximisation is done with respect to the Gaussian parameters, Θ_G. To handle the ellipse connections, we maximise directly in terms of the ellipse parameters Θ_E. The maximisation is applied to the constrained parameters. Adding constraints means that the solution no longer has a closed form and has to be computed using an iterative optimisation process, such as a quasi-Newton algorithm [22].

The other part of the M-step is to update the α_l values. Only the first sum of Eq. (8) contains α_l values. The values which maximise the expression are

$$\alpha_l=\frac{\sum_{i=1}^{N}p(l|\mathbf{y}_i,\Theta_G^{(t)})}{\sum_{k=1}^{M}\sum_{i=1}^{N}p(k|\mathbf{y}_i,\Theta_G^{(t)})}=\frac{\sum_{i=1}^{N}p(l|\mathbf{y}_i,\Theta_G^{(t)})}{N} \qquad (11)$$

It is possible for some of the α_l values to shrink to 0. For unconstrained GMM this corresponds to one component being dropped – which can be desirable in that case. However, in our case, all ellipses are critical to the overall structure. So we do not want α_l shrinking too much. If any α_l value drops below some low value, then just set α_l to that value and then renormalise the other values.

In summary, given the current estimate of the ellipse parameters, $\Theta_E^{(t)}$, the modified E-M algorithm becomes that of

$$\begin{aligned}&\boldsymbol{E-step}\\ &\quad\text{Compute } \Theta_G^{(t)}=\mathcal{F}_Z(\Theta_E^{(t)}) \text{ using Eq. (1).}\\ &\quad\text{Compute responsibilities}: p(l|\mathbf{y}_i,\Theta_G^{(t)})\\ &\boldsymbol{M-step}\\ &\quad\text{Maximise (with constraints) Eq. (10) with respect to } \Theta_E\\ &\quad\text{Update } \alpha_l \text{ using Eq. (11)}\end{aligned} \qquad (12)$$

2.4. Balloon fitting

In the case where the shape silhouette is complex and has a large curvature, the additional constraints can make the fitting become stuck in a local minimum. For an example with only two ellipses, see Fig. 2h. To help explain the problem and see a solution, consider an analogous problem of inserting inflated (and connected) balloons into an enclosure. Suppose we can inflate or deflate the balloons a little as well as move and deform them a little each iteration. If the balloons are fully inflated to begin with, then the balloons can get stuck – especially if there is a bend in the cavity. A more feasible approach is to start with balloons that are deflated, then inflate slowly over time. By starting with deflated balloons, they have more flexibility to begin with and thus can slide around bends. In particular, this approach is much better at determining the locations of the joints. Translating this idea to this work, we initialise the value of Z to correspond to high variability. During the E-M algorithm, the value of Z is increased successively during each step of the fitting so that the Gaussians' variability reduces and precision increases. Once the Z has reached a desired final value, continue E and M steps until convergence is reached.

Therefore, the algorithm becomes that of

```
Initialise $Z, \delta_Z, Z_{max}$.
Loop the following steps
E − step
    Compute $\Theta_G^{(t)} = \mathcal{F}_Z(\Theta_E^{(t)})$ using Eq. (1).
    Compute responsibilities : $p(l|\mathbf{y}_i, \Theta_G^{(t)})$
M − step
    Maximise (with constraints) Eq. (10) with respect to $\Theta_E$
    Update $\alpha_l$ using Eq. (11)
If $Z < Z_{max}$ then increment $Z$ : $Z = Z + \delta_Z$.
Else if the change in log likelihood is below some tolerance,
exit the loop.                                                        (13)
```

Fig. 2 demonstrates how the balloon fitting works in comparison to just using constrained EM. Constrained EM without the balloon fitting adjustment, tries to cover all the silhouette from the first iteration (Fig. 2b). A consequence of this is that joints can be poorly estimated in the early iterations and not recover. Using the balloon fitting adjustment, the first iteration does not try to cover all the silhouette (Fig. 2c). Visually, the ellipses appear deflated. The ellipses are mapped onto Gaussians with high variance though and so their shape and orientation are impacted by the surrounding points. For the second iteration (Fig. 2f), the ellipses are starting to bend. For each subsequent iteration, the ellipses cover more of the silhouette, until they have a tight fit for the final iteration. The greater flexibility for early iterations have stopped the algorithm getting the joint incorrect.

Our process of balloon fitting has similarities to Deterministic Annealing Expectation-Maximisation (DAEM) [23,24]. DAEM likewise has on outer loop that effectively decreases variability for each step. It does this, not by directly changing the covariance matrices, but by altering the responsibilities, $p(l|\mathbf{y}_i, \Theta_G^{(t)})$. For early stages the responsibilities for all points are forced to be similar and so the Gaussians are impacted by most of the data points (and this way the variability is high). This high variability at the early stage reduces the problem of hitting an inferior local maximum. For later stages, the responsibilities take on the standard form and so only points close to that component have an impact on the new estimates, thus allowing fine tuning. For each step of the outer loop, several iterations are required to find the maximum at that stage. For DAEM this is achieved using several E and M steps. (In our method above we use several iterations of a quasi-newton method). This shows that avoiding inferior maximums does require more iterations.

2.5. Other constraints

The balloon fitting we have presented above is a general framework to fit multiple connected ellipses to a silhouette. A particular problem might require further constraints. This section describes some possible variations and how these interact with balloon fitting.

One type of constraint is one that directly reduce the degrees of freedom of the ellipses. An example of this that we have already mentioned is the hard geometric constraints as given by Eq. (7). Other examples are constraining some of the ellipses to be circles (and hence drop the b parameter) or constraining a pair of ellipses to have the same length (and hence reduce the number of parameters for one ellipse). For these types of constraints, the reduction of the number of parameters directly affects the M-step.

Another type of constraint (as mentioned in the introduction) is one that uses a Bayesian prior. The process for using them is slightly modified with balloon fitting. To illustrate the modification needed we will give an example. By definition, the major axis of an ellipse is longer than its minor axis. In the vanilla algorithm, if the vertex $\mathbf{x}_v$ (in Θ_E) starts on the major axis, after a while, it can swap to being on the minor axis. This can be seen in Fig. 2h. For the initial guess (Fig. 2a), the top ellipse is joined at a vertex on the major axis. As the EM algorithm has iterated, the labelling of axes has swapped and the join has finished on the minor axis. If this swapping of axes is not desirable (as in the case of a upper arm), we can use a constraint to keep $\mathbf{x}_v$ on the major axis. A Bayesian exponential prior on the ratio b^2/a^2 (where a is given by Eq. (3)) can achieve this. This amounts to adding the following term to the log-likelihood (Eq. (10)):

$$-\frac{1}{2}w\sum_{l\in S}\frac{(b^l)^2}{(a^l)^2}\alpha_l N. \tag{14}$$

The constant term, which does not impact the optimisation has been removed. The set S would be the subset of $\{1, ..., M\}$ of ellipses that we wish to place the constraint on. For the end part, recall N is the number of point, thus $\alpha_l N$ equals $\sum_{i=1}^{N} p(l|\mathbf{y}_i, \Theta_G^{(t)})$ and so matches the responsibility terms of Eq. (10). The variable w is a shape parameter so that larger values of w correspond to larger values of the precision of the prior (more precisely, $4/w^2$ is the variance). For other constraints using priors, there likewise are extra terms with parameters that increase with increased precision (for a Gaussian prior, the extra parameter is the precision of the Gaussian).

Incorporating constraints with priors has twofold implications with balloon fitting. The first, straightforward, implication is that the equation to optimise in the M-step is changed. Secondly, the idea of balloon fitting is to increase the precision of the covariance matrices for each step. So we modify the precision of the priors (by modifying w in the example above) at each iteration. Start at some low value and increment the parameter whenever Z is incremented. For example, with the prior above, we started with $w=0$ and increased the value at each step until for the final iteration we used $w=1$.

This process for dealing with prior constraints applies also to the soft geometric constraints mentioned at the end of Section 2.2. The extra term added to Eq. (10) is

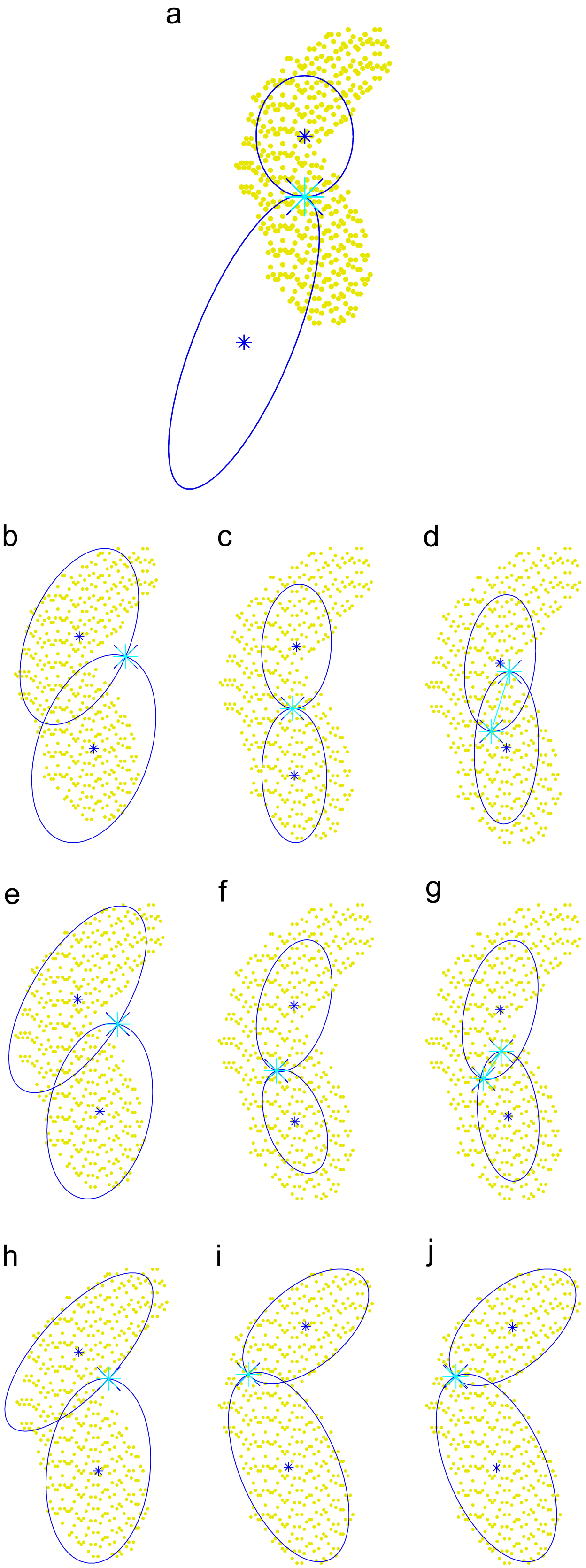
$$-\frac{1}{2}w\|p(\phi^i, d^i, \Theta_E^i) - p(\phi^j, d^j, \Theta_E^j)\|^2 \tag{15}$$

The parameter w (which is the precision) can be increased at each step.

Fig. 2d, g and j shows how joints change when using soft constraints. For the first iteration (2d), the two points on the ellipses which are supposed to be joined (indicated by asterisks) are considerably apart (to allow the ellipses to get the right shape). For the second iteration (2g), the points are getting closer. For the final iteration (2j), the two points are the same making the desired joint. For the two ellipse case in Fig. 2, using soft constraints gave the same final fit as the hard constraints. However, this extra flexibility is useful for more complex shapes as we will see later.

2.6. Context-dependent initial guess

As with any iterative algorithm, a good initial guess can help improving the final fit and rate of convergence. Although balloon fitting tries to reduce the dependency on the initial guess, it nevertheless benefits from better starts. For a particular problem, some context-dependent algorithm may be the most effective at producing the initial guess. For example, for image sequence problems, the fittings of the previous frame can be used as the initial guess of the subsequent frame. For static image application, an alternative strategy that may be used is similar to [11], in which the authors ignored the geometric constraints, mapped the ellipses to Gaussians and then ran the standard EM algorithm to



fit the GMM before mapping the GMM parameters back into ellipses.

3. Experiment

We compared the effectiveness of balloon fitting (with both hard and soft constraints) against three other methods for fitting multiple connected ellipses from the literature (5 methods in total: Section 3.1). We evaluated the methods using the median distance between the fitted ellipse and the boundary of the silhouette (Section 3.2). The main comparisons were performed on synthetic data with known ground-truth locations (Section 3.3). Additionally we showed how balloon-fitting also applies to real world data with complicated models (Section 3.4).

3.1. Methods for comparison

The first method evaluated was our previous work [11] (Subtree for short). The method attempted to fit multiple connected ellipses to an enclosed silhouette contour (as opposed to every points within the silhouette). The connected ellipses were modelled in a hierarchical tree structure, where ellipses joints with more interconnections were placed at a lower depth. During fitting, the tree was converted into a set of mean vectors and covariance matrices that served as the initial parameters for a Gaussian mixture model to be solved by conventional EM algorithm. The converged Gaussian parameters were converted back into the ellipse tree structure identical to the initial configuration. A fine-turning step using non-linear least square method was then applied. The method previously had good performance on silhouette contour with little or no noise.

The other four methods are variations of constrained EM. The second method was balloon fitting using hard constraints (hard for short) as summarised by Eq. (13). The third method was balloon fitting using soft constraints (Soft for short) via Eq. (15).

The fourth method was an adaption of [1] which we called Hunter. It has been adapted in two ways, firstly to apply to ellipses and secondly to allow all parameters of the ellipses to vary (not just location and orientation). The original method applied to 2D projections of ellipsoids, but it easily translates to ellipses. The original method also fixed the size of each ellipse/ellipsoid, then used unconstrained EM to fit the ellipses. To enforce the constraints, the method found the tangent plane to the feasible space at the previous estimate. Then the new estimate, created by the EM method, was projected perpendicularly to the tangent plane back onto the feasible space. When fixing the size of each ellipse, projecting perpendicularly to the tangent space was always able to reach the feasible space. In our adaption, we allow the size of each ellipse to vary. With this adaption projecting perpendicularly to the tangent plane may not reach the feasible space. So instead we

Fig. 2. This figure shows how the various algorithms mentioned in this paper work. The top row (2a) shows the initial fit used for all cases. Then first column of the rest of figures used constrained EM (CEM for short) as summarised by Eq. (12). The second column used balloon fitting (BF for short) using hard constraints as summarised by Eq. (13). The third column used balloon fitting with soft constraints (via Eq. (15)) (SBF for short). The second, third and fourth rows show respectively the first, second and final iterations of the methods. Visually, the coverage of the shape by the two ellipses in the final iteration is good in all cases. However, the joint is only reasonably fitted when using the balloon fitting adjustment. (a) Initial fit, (b) CEM iteration 1, (c) BF iteration 1, (d) SBF iteration 1, (e) CEM iteration 2, (f) BF iteration 2, (g) SBF iteration 2, (h) CEM final fit, (i) BF final fit, and (j) SBF final fit.

have projected to the point in the search direction that is nearest to the feasible space. If the constraints are not met, we forced them to be met by hierarchically adjusting the means. Convergence of the Hunter method cannot be determined by change in log likelihood as it is for the other constrained EM methods because log likelihood is not calculated at the projected point. Instead convergence is judged by the change in the vector of parameters.

The fifth method was an adaption of [15] which we called Cheng. The algorithm in [15] directly applied to fitting ellipsoids to 3D (stereo) images, but the method can readily be applied to ellipses. The method used a constrained EM (as described by Eq. (12)) with soft constraints (using Eq. (15)). The Cheng method was very similar to the Soft method. The only difference was that variability was fixed for the Cheng method but it was decreased gradually for the Soft method. The original method in [15] fixed the size of each ellipse/ellipsoid. We adjusted by allowing the ellipse size to vary. The method can handle that conceptually. The original method's M-step had an explicit update equation for the means and an iterative method to find the maximum for other parameters. By allowing the size to vary, the update equation for the means becomes considerably more difficult to describe, so we just used an iterative method for all parameters.

To make it a fair comparison, we have made the test condition as similar as possible for all these methods: all methods were run for at most 40 iterations. Most cases converged well before this. The few cases which did not converge were stuck in an inferior local optimum. The Hard and Soft methods both started with $Z=1$ and increased this to $Z_{max}=2$ over 20 steps, and then allowed to go for at most 20 more iterations. The Hunter and Cheng methods just kept Z fixed at 2. The final precision for soft constraints (in Cheng and Hunter methods) was $w=1$. For the soft method this started at $w=0$ and was increased over 20 steps. Where possible (namely for the Hard, Soft and Cheng methods) constraints were added to preserve the major/minor axes for the two ellipses (using Eq. (14)). This was not possible for the Subtree and Hunter methods since the method of constraining is not compatible with that type of constraint. The final precision for these constraints was also 1 and for the Hard and Soft methods it started at 0 and was increased over 20 steps.

3.2. Evaluation metrics

The chief metric that was used to evaluate each fit was the root mean square (RMS) distance between the silhouette boundary and the ellipses (since a good fit should match the ellipses with the edge of the silhouette). The shortest Euclidean distance between a point and an ellipse does not have a simple formula. So we used the approximation used in [11,2]. Let $C_l(x,y)=\sqrt{(x-x_c^l)^2+(y-y_c^l)^2}$ be the distance from the centre of an ellipse to a point. Let $M_l(x,y)$ be the Mahalanobis distance between the point and an ellipse. For example: for an ellipse aligned to the coordinate axes, with centre at the origin, $M_l(x,y)=\sqrt{x^2/a_l^2+y^2/b_l^2}$. The distance function we used was thus

$$d_l(x,y)=\begin{cases} C_l(x,y)\left(1-\dfrac{1}{M_l(x,y)}\right), & M_l(x,y)\geq 1 \\ r_l(M_l(x,y)-1), & M_l(x,y)\leq 1 \end{cases} \tag{16}$$

where $r_l=\min(a_l,b_l)$ is the half-length of the minor axis. The distance from a point to the whole ellipse structure is the distance $(d_l(x,y))$ to the ellipse that is closest to it. The units for all the distances are the number of pixels.

The evaluation metric for a given pair was then the root mean square (RMS) of the distances from the boundary points to the ellipse structure (there was about 300 boundary points for each case). For each pair of ellipses, a rank was also given to each of the 5 methods. The Friedman test, and follow up multiple comparisons using 95% confidence intervals, was used to analyse the differences between the methods. For all modes, the p-value of the Friedman test was less than 0.001 indicating significant differences. The results of the multiple comparisons are represented using a compact letter display (cld): methods with a common letter are not significantly different.

3.3. Experiments on synthetic data set

In order to robustly evaluate the effectiveness of our algorithm, we ran two different experiments each consisting of 100 semi-randomly generated noisy images of two joined ellipses with a different way to generate the initial fit. In each experiment, 100 pairs of joined ellipses were randomly generated (just like in Figs. 2 and 3). The ellipses were allowed to vary in size, shape and orientation. However, the sizes of the ellipses were generated so that the joint was always at the major axes. This was done to accommodate the algorithm of the Subtree method and to allow the use of the constraints as given by Eq. (14). We created a noisy version of the test silhouette, where each pixel location has undergone an i.i.d Bernoulli ($p=0.7$) trial to decide if the pixel should be retained. For each sample, an initial set of ellipses was generated. The same initial set of ellipses and sample were inputted for all five methods.

There were two different ways of generating the initial guess. The first we called *random mode*: the initial guess is made relatively close to the ground-truth; in particular the initial ellipses are random perturbations of the ellipses used to generate the image (for example see Fig. 3a where the initial ellipses are slightly translated and modified in shape but the angles are similar). Random mode was designed to simulate situations where sensible initial guesses can be made in certain applications, using typical methods described in Section 2.6, as well as scenarios for which the Hunter and Cheng methods were originally used. The second

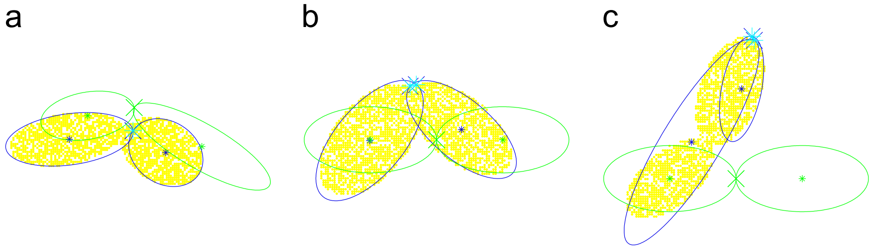


Fig. 3. Examples of ellipses and fitting for the synthetic data set. In all cases, the yellow dots represent the data the ellipses are fitted to. The green ellipses show the ellipses used as the initial fit. The blue ellipses show the final fit. (a) Hunter method with random mode (RMS=1.16) – an example of an adequate fit (other methods were better) (b) Soft method with neutral mode (RMS=0.88) – a good fit . (c) Cheng method with neutral mode (RMS=4.33) – a poor fit with jackknifed ellipses. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

way we called *neutral mode*: where all 100 test cases have the same initial ellipses which form a "neutral" pose: The joint of the neutral pose was placed at the theoretical average place. The size of the neutral pose was slightly larger than the average size (since slightly more variability to begin with is usually a good thing). The sizes of both ellipses were made the same and their orientations were the same. The angle at the join was 180°. Some of the randomly generated ellipse pairs had locations, sizes and orientation considerably different from the neutral pose. For example see Fig. 3b and c: the initial ellipses are the same in both cases, although the ground-truth is considerably different. The neutral mode was designed to test how well the methods perform under scenarios where little prior information is given.

We illustrate the fitting performance of the five algorithms in terms of the RMS distance between the fitted ellipses and the contour points. The medians and spread of the RMS are shown in Figs. 4 and 5 under the random and neutral modes respectively. The mean rank, median RMS and compact letter display (cld) are shown in Tables 1 and 2 under these two modes.

The noisy edges and rounding errors due to using pixels both cause an increase to the RMS. Here is a guide to understanding the RMS values for the synthetic data. See Fig. 3 for some images matching to some RMS values. Values of less than about 1.5 correspond to a fit that visually appears good. Values between approximately 1.5 and 2.5 indicate a gradual degrading of the fit (for example, when the two ground truth ellipses have similar orientations, sometimes the fitted ellipses will have the same orientations, but the joint will slide down a bit). RMS values greater than approximately 2.5 are major misfits (eg jack-knifed ellipses, or joints significantly off). These cutoffs are subjective; fortunately it does not matter, since for the Hard, Soft, Cheng and Subtree methods there is a clear gap in the RMS values between good and bad fits. For real data with more complicated models, higher RMS values should be expected.

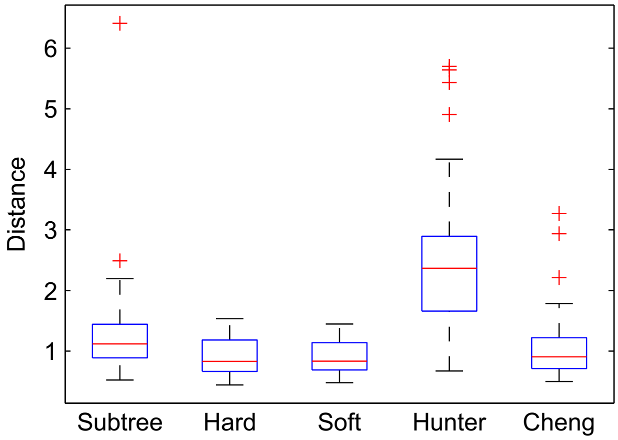


Fig. 4. Box plot of RMS of distances for the five methods in random mode.

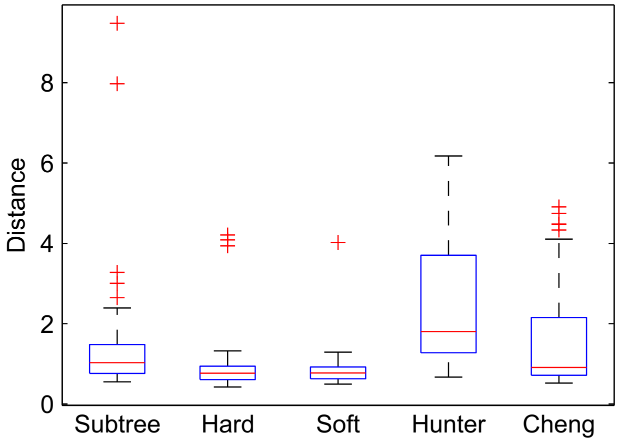


Fig. 5. Box plot of RMS of distances for the five methods in neutral mode.

Table 1
Fitting statistics for random mode.

Method	Mean rank	Median RMS	cld
Soft	2.10	0.83	a
Hard	2.17	0.83	a
Cheng	2.85	0.91	b
Subtree	3.22	1.12	b
Hunter	4.66	2.37	c

Table 2
Fitting statistics for neutral mode. (The last column is the number of the 100 cases where the RMS was greater than 2.50 and measures the number of serious misfits).

Method	Mean rank	Median	cld	> 2.50
Hard	1.92	0.76	a	3
Soft	1.99	0.77	a	1
Subtree	3.02	1.03	b	5
Cheng	3.41	0.91	b	25
Hunter	4.66	1.80	c	44

Random mode: For all ellipses the balloon fitting methods did well and were the best performers (in terms of median RMS and mean rank) and statistically indistinguishable. The Cheng and Subtree methods were next. For most ellipses the Cheng and Subtree methods did well. There were a few cases for Cheng and the Subtree methods (which appear as outliers on the boxplots) which resulted in a poor fit. For instance, the top outlier for the Cheng method had a jack-knife. The next two outliers correspond to ellipses which are almost in a straight line. For these two cases, the Cheng method fitted the orientations of the ellipses well, but was weak on the position of the joint. Hunter's method commonly produced poor results (although about quarter of the time it did produce good fits).

Neutral mode: The results of the neutral mode were similar to the random mode, except that with poorer initial ellipses, there were more misfits. The neutral mode put all methods to the test, and all methods had some ellipses that did not fit properly. All cases where the RMS was greater than 2.5 visibly looked like a misfit. For the Hard, Soft and Cheng methods (and a lesser extent the Subtree method) the distinction between a good fit and a misfit is clear from the RMS value. The box plots for Hard and Soft show the huge jump to the outliers (which are all misfits) from the rest of the box and whiskers (which visually all appear as good fits). For the Cheng method, the 74th (ranked by RMS) ellipse pair had an RMS of 1.17 (and was a good visual fit); the 75th ellipse pair had an RMS of 3.13 (which was jack-knifed). The huge jump is mostly hidden in the box plot because it occurs at the 3rd quartile. The misfits for the Hard, Soft and Cheng methods were all jack-knifed where the angle between the ellipses gets close to 180°. The jack-knifed position corresponds to a local maximum of the likelihood function but not a global maximum. The Hunter method did not have such a clear distinction and the 2.5 cutoff (which marked the first jack-knifed pair of ellipses for the Hunter method) is a bit arbitrary. However, using this cutoff or other values either side showed that by any reckoning the Hunter method has a lot of misfits.

The Hard and Soft balloon fitting methods were statistically inseparable in terms of mean rank and median RMS. However, the Soft method only had only 1% misfits (1 from 100 cases) compared to

3% for the Hard method. Next in terms of mean rank and median RMS were the Cheng and Subtree methods (which were statistically inseparable). Cheng's method works well for lots of cases, and this is reflected in a median RMS of 0.91. However, the fact that 25% of cases Jack-knifed shows it is inappropriate for this mode. The Hunter method was inferior for all the metrics.

3.3.1. *Discussion on synthetic data set*

In both experiments, the balloon-fitting methods performed the best. The Cheng and Soft methods differed only by the fact that the Soft method had the balloon-fitting modification. So by comparing the Cheng method in neutral mode, we saw that the balloon fitting modification greatly reduced the chance of getting stuck in an inferior local maximum. The use of soft constraints as opposed to hard constraints helped reduce the chance a little of being stuck in an inferior local maximum. If the methods avoided the inferior maximum, then the quality of the fit was essentially the same.

The original Hunter method given in [1] which used a projection (or at worst a line search) for each M-step worked quickly and converged for the choice of parameters. Unfortunately, by allowing the size of the ellipses to vary, the projection concept was weakened. Additionally the Hunter method could not handle the constraint to preserve the major/minor axes (using Eq. (14)). Both factors contributed to the poor performance of the Hunter method for many (but not all) cases.

The Subtree method performed reasonably in both modes. However it relies on the boundary and so is more susceptible to noise which is demonstrated by weaker fits (in general) than the balloon-fitting methods. Since the Subtree method uses less points it is quicker than the other approaches and so it could still be considered when there is little noise.

3.4. *Experiments on real-world data set*

The balloon fitting framework is shown to be equally effective for more complicated, real-word images. Three examples are the upper human body (Fig. 6), the back of a guitar (with glare on the neck of the guitar making it less distinct) (Fig. 8) and a fish (using the ellipse

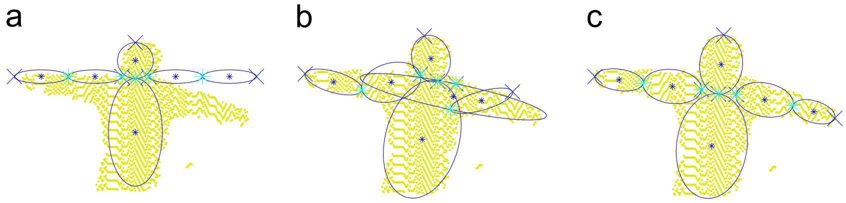


Fig. 6. A comparison for a upper human body image using (a) initial fit and either (b) no balloon fitting (CEM), or (c) balloon fitting (BF).

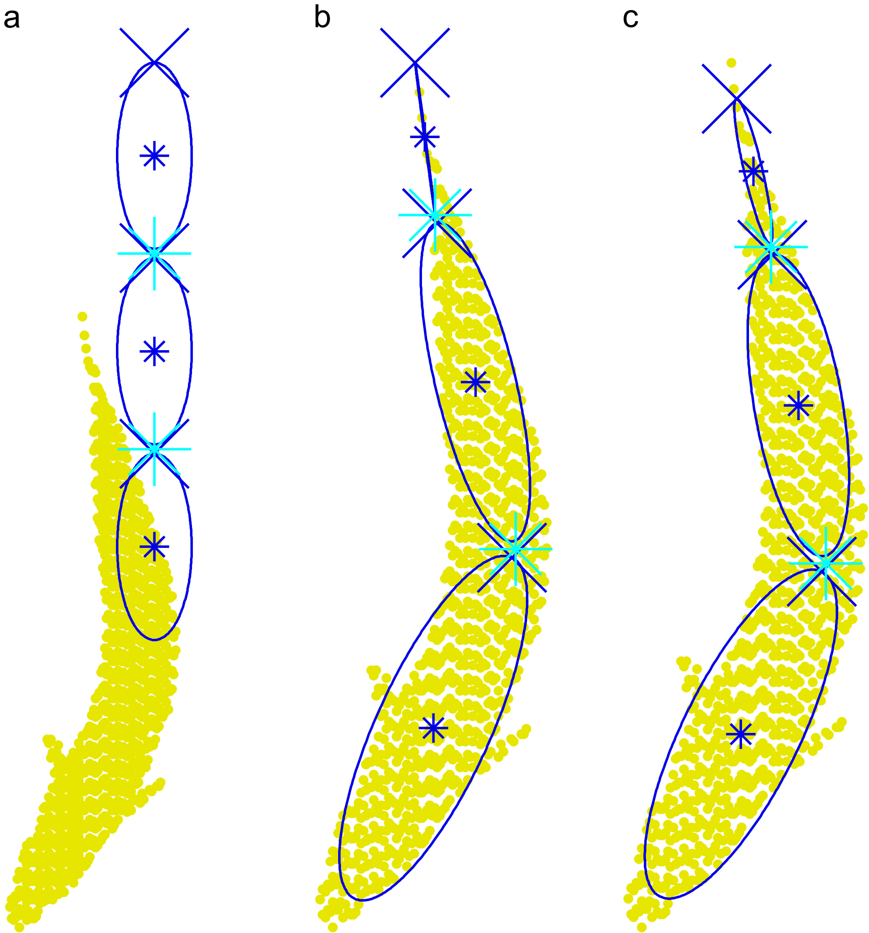


Fig. 7. A comparison for a fish image using (a) initial fit and either (b) no balloon fitting (CEM), or (c) balloon fitting (BF).

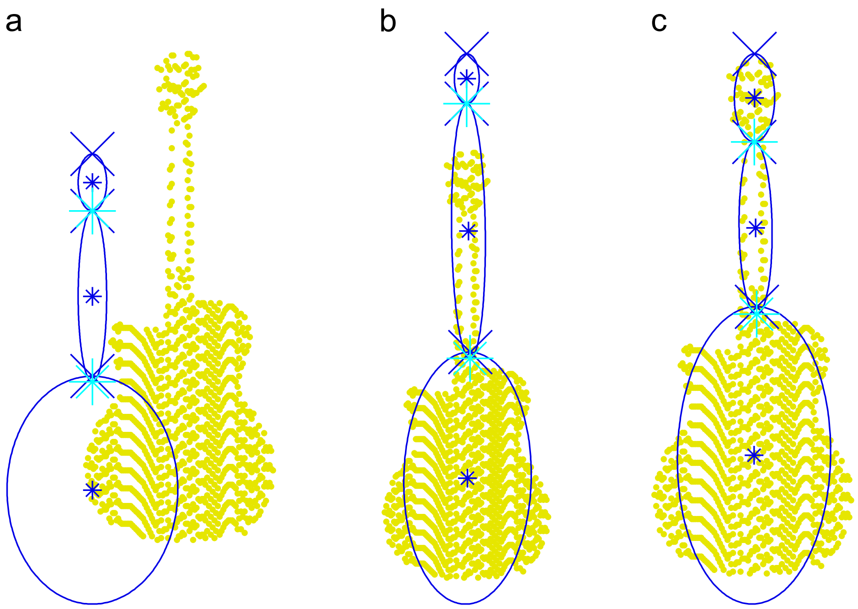


Fig. 8. A comparison for a guitar image using (a) initial fit and either (b) no balloon fitting (CEM), or (c) balloon fitting (BF).

model from [3]) (Fig. 7). All examples used hard constraints with constrained expectation maximisation (CEM) and balloon fitting (BF). The initial fits for the guitar and fish were placed considerably off centre in order to test the CEM and BF methods.

For the upper human body, the CEM method gets the upper right arm segment too big at an early stage and is unable to recover, whereas balloon fitting converges to a visually good fit.

The visual fit for the guitar is not good for the CEM method because the weight (alpha) given to the head of the guitar has vanished to zero and the predicted value of the nut (the junction between head and neck) is therefore way off. The balloon fitting which has more variability early on in the process does not lose the head in the same way and so gets a good fit. Similarly, to the guitar, the CEM method makes the top ellipse too small early on and cannot recover from it, whereas the balloon fitting can (with its changing variability).

4. Conclusion

Balloon fitting is an effective framework to fit connected ellipse models to silhouettes. It is superior to constrained expectation maximisation and other methods from the literature. Additional constraints (as determined by context) can be added to the ellipse model to improve the fit as necessary and these constraints can be handled within the balloon fitting paradigm. Although the scope of this paper is concentrated on the ellipse structure fittings to 2D image silhouettes, the same methodology can easily be change into 3D ellipsoid fittings using Kinect acquired point cloud. Even more, we argue that the methodology may find its usefulness in high dimensional constrained hyper-ellipsoidal fittings in many data-mining applications.

Conflict of interest

None declared.

Acknowledgements

We would like to thank David Nguyen for his assistance with some of the programming. We also would like to thank the referees for their comments to greatly improve the paper.

Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2015.01.026. This includes all the silhouettes referred to in this paper and Matlab source code that implements the balloon fitting algorithm. This code was used in the experiments.

References

[1] E. Hunter, P. Kelly, R. Jain, Estimation of articulated motion using kinematically constrained mixture densities, in: Proceedings of IEEE Nonrigid and Articulated Motion Workshop, Puerto Rico 1997, IEEE, 1997, pp. 10–17.

[2] F.L. Jeune, R. Deriche, R. Keriven, P. Fua, Tracking of Hand's Posture And Gesture, Technical Report, CERTIS, ENPC, 〈http://certis.enpc.fr/publications/papers/04certis02.pdf〉, 2004.

[3] E. Kanso, J. Marsden, C. Rowley, J. Melli-Huber, Locomotion of articulated bodies in a perfect fluid, J. Nonlinear Sci. 15 (2005) 255–289.

[4] C. Wong, S. Lin, T. Ren, N. Kwok, A survey on ellipse detection methods, in: IEEE International Symposium on Industrial Electronics (ISIE), Hangzhou IEEE, 2012, pp. 1105–1110.

[5] A. Fitzgibbon, M. Pilu, R.B. Fisher, Direct least square fitting of ellipses, IEEE Pattern Anal. Mach. Intell. 21 (1999) 476–480.

[6] B. Xiong, J. Chen, G. Kuang, N. Kadowaki, Estimation of the repeat-pass alos palsar interferometric baseline through direct least-square ellipse fitting, IEEE Trans. Geosci. Remote Sens. 50 (2012) 3610–3617.

[7] D.K. Prasad, M.K. Leung, C. Quek, Ellifit: an unconstrained, non-iterative, least squares based geometric ellipse fitting method, Pattern Recognit. 46 (2013) 1449–1465.

[8] Y. Ito, K. Ogawa, K. Nakano, Fast ellipse detection algorithm using hough transform on the GPU, in: The Second International Conference on Networking and Computing (ICNC), Osaka IEEE, 2011, pp. 313–319.

[9] T. Zhou, N. Papanikolopoulos, Enhancing the randomized hough transform with k-means clustering to detect mutually-occluded ellipses, in: The 19th Mediterranean Conference on Control & Automation (MED), Corfu, Greece 2011, IEEE, pp. 327–332.

[10] M. Alemán-Flores, L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra, Camera calibration in sport event scenarios, Pattern Recognit. 47 (2014) 89–95.

[11] R. Xu, M. Kemp, Fitting multiple connected ellipses to an image Silhouette hierarchically, IEEE Trans. Image Process. 19 (2010) 1673–1682.

[12] R.Y.D. Xu, M. Kemp, An iterative approach for fitting multiple connected ellipse structure to silhouette, Pattern Recognit. Lett. 31 (2010) 1860–1867.

[13] A. Fossati, E. Arnaud, R. Horaud, P. Fua, Tracking articulated bodies using generalized expectation maximization, in: CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, IEEE, Anchorage, AK, USA, 2008, pp. 1–6.

[14] K. Kanatani, Y. Sugaya, Compact algorithm for strictly ML ellipse fitting, in: 19th International Conference on Pattern Recognition, 2008. ICPR 2008, Tampa USA IEEE, 2008, pp. 1–4.

[15] S. Cheng, M. Trivedi, Articulated human body pose inference from voxel data using a kinematically constrained Gaussian mixture model, in: CVPR EHuM2: The Second Workshop on Evaluation of Articulated Human Motion and Pose Estimation, vol. 55, 2007.

[16] H. Greenspan, A. Ruf, J. Goldberger, Constrained Gaussian mixture model framework for automatic segmentation of MR brain images, IEEE Trans. Med. Imaging 25 (2006) 1233–1245.

[17] P. Olsen, R. Gopinath, Modeling inverse covariance matrices by basis expansion, IEEE Trans. Speech Audio Process. 12 (2004) 37–46.

[18] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with EM using equivalence constraints, Adv. Neural Inf. Process. Syst. 16 (2004) 465–472.

[19] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, K. Visweswariah, Subspace constrained Gaussian mixture models for speech recognition, IEEE Trans. Speech Audio Process. 13 (2005) 1144–1160.

[20] K. Blekas, A. Likas, N. Galatsanos, I. Lagaris, A spatially constrained mixture model for image segmentation, IEEE Trans. Neural Netw. 16 (2005) 494–498.

[21] J.A. Bilmes, A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, International Computer Science Institute, Technical Report TR-97-021, USA, 1997.

[22] C.G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, IMA J. Appl. Math. 6 (1970) 76–90.

[23] N. Ueda, R. Nakano, Deterministic annealing EM algorithm, Neural Netw. 11 (1998) 271–282.

[24] Y. Itaya, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, T. Kitamura, Deterministic annealing EM algorithm in parameter estimation for acoustic model, in: The Eighth International Conference on Spoken Language Processing, 2004, pp. 433–436.

Michael Kemp received the B.Sc. (Adv) degree and the Ph.D. degree in pure mathematics from the University of Sydney, Australia, in 2000 and 2004 respectively. He is currently a Lecturer in Mathematics at Charles Sturt University, Australia. He has been engaged in theoretical and practical research in the mathematics and statistics area.

Richard Yi Da Xu 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 University of Technology, Sydney (UTS). He has authored more than 30 international journal and conference papers in the area of computer vision algorithms and applications.