

Reconstruction with blobby shapes

W. E. Ong¹R. K. Beatson²C. J. Price³

(Received 18 January 2011; revised 2 August 2011)

Abstract

We investigate fitting a surface to an object using blobby models as a coarse level approximation. The aim is to achieve a given quality of approximation with relatively few parameters. This process involves an optimisation procedure where a number of blobs (ellipses or ellipsoids) are separately fitted to a cloud of points. Then the optimised blobs are combined to yield an implicit surface approximating the cloud of points. The results for our test cases in two and three dimensions are very encouraging. For many applications, the coarse level blobby model itself will be sufficient. For example, adding texture on top of the blobby surface can give a surprisingly realistic image.

Contents

1 Introduction

C597

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3885> gives this article, © Austral. Mathematical Soc. 2011. Published August 8, 2011. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

<i>1 Introduction</i>	C597
2 Previous work	C598
3 Fitting ellipses in 2D	C599
3.1 2D results and discussion	C603
4 Fitting ellipsoids in 3D	C605
4.1 3D results and discussion	C607
5 Conclusions	C609
References	C610

1 Introduction

The wide spread adoption of 3D scanning devices has enabled the acquisition of many large ‘cloud of points’ data sets. These represent a variety of objects, from sculptures, body parts, and machine parts, through to the interiors of mines. For various applications, it is required to convert the point clouds into a ‘water tight’ surface.

The motivation of this work is to identify the basic structure of an object by a blobby model representation. We present an automatic method to approximate the shape of a given cloud of points with a low parameter model:

- using ellipses in two dimensional space (2D); and
- using ellipsoids in three dimensional space (3D).

Our aim is to find a very efficient coarse level approximation for hierarchical implicit curve (2D), or surface (3D), models. By efficient we mean a coarse level approximation that reproduces the overall shape of the underlying object while being very fast to evaluate. The speed of evaluation of the coarse level approximation is critical as it is used for every function evaluation when rendering the surface. Implicit radial basis function models [1] and hierarchical

radial basis function models (unpublished), indicate that the current method easily outperforms radial basis function models for this application.

The two stage process involves automatic initialisation and optimisation of blobs (ellipses or ellipsoids) approximating parts of a point cloud. The parts approximated by the different blobs are not identified explicitly and usually overlap. Next, an implicit blobby model is fitted by combining all the optimised blobs through least squares fitting to all the data.

Suitable objective functions and heuristics have been developed which work well on our test cases, particularly in fitting the outline of a hand in 2D and 3D. We review some related work in the following section. Shape reconstruction in 2D using ellipses is discussed in section 3. The method of fitting ellipsoids to points in 3D is presented in section 4.

2 Previous work

Söderkvist [6] gave a good general overview of various methods used in surface reconstruction prior to 1999. Since we focus on blobby model shape reconstruction, we only review related research; particularly that reported in the last decade.

Bischoff and Kobbelt [3] presented a technique for fitting ellipsoids to the volume bounded in a given mesh. The robust transmission of geometric objects in the ellipsoidal decomposition guarantees a good approximation even if some parts of the data are lost during transmission.

Simari and Singh [5] restructured and re-meshed polygon meshes using ellipsoidal based segmentation. Their algorithm allows fitting the ellipsoids to a given mesh by surface or volume enclosed orientation. They also take ‘negative ellipsoids’ as primitives to capture bowl shaped concavities. The shape of the model is then obtained by smoothing segmentation boundaries.

Jin et al. [7] proposed an automatic algorithm for approximating polygonal

meshes with blobby objects. They constructed a medial axis sphere-tree of the given mesh and used the spheres as initial blobs to approximate the object's shape. Once the sphere representation is achieved, an isosurface of the object is generated employing field functions in the blobby model. An implicit representation of the object is constructed through minimising the distance between the isosurface of blobs and the given mesh.

Recently, Liu et al. [4] extended the work of Jin et al. [7] and reconstructed a blobby model using ellipsoidal blobs. Their work employed a modified scheme of Bischoff and Kobbelt [3] in the ellipsoidal decomposition process. Their main contributions are applying the ellipsoidal blobby model in geometry data reduction and giving two geometric based schemes for cloud animation.

3 Fitting ellipses in 2D

Given a set of scattered data points in the plane, (x_i, y_i) where $i = 1, 2, \dots, N$, we approximate part of the shape of the scattered points by an ellipse. Generally, an arbitrarily oriented ellipse is defined by

$$(\mathbf{x} - \mathbf{c})^T \mathbf{A} (\mathbf{x} - \mathbf{c}) = 1, \quad (1)$$

where $\mathbf{x} = (x, y)$ are points on the ellipse, $\mathbf{c} = (x_c, y_c)$ is the centre of the ellipse and \mathbf{A} is a symmetric positive definite matrix. The matrix \mathbf{A} can be decomposed as $\mathbf{A} = \mathbf{Q} \mathbf{D} \mathbf{Q}^T$ where the rotation matrix

$$\mathbf{Q} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (2)$$

The eigenvalues of \mathbf{A} are given by $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2)$ and the columns of \mathbf{Q} are corresponding eigenvectors. These eigenvalues are the inverse squares of the lengths of the major and minor semi axes of the ellipse, \mathbf{a} and \mathbf{b} .

In this work, ellipses are fitted inside the object independently of each other. Thus, in the first stage, we set up the fitting problem as a constrained

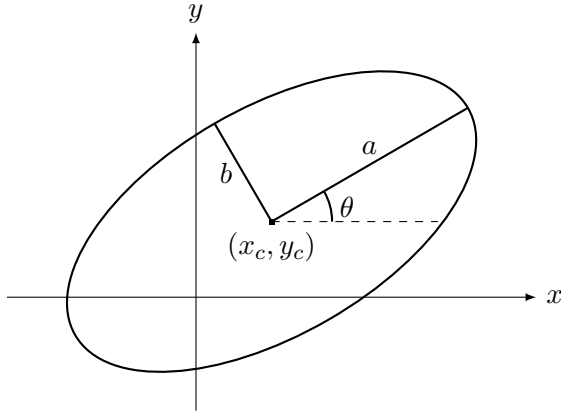


FIGURE 1: Parameters of an ellipse used in the optimisation algorithm.

optimisation problem of the form

$$\text{minimise } f(\mathbf{z}) \text{ such that } \mathbf{l}_b \leq \mathbf{z} \leq \mathbf{u}_b \quad (3)$$

where $\mathbf{z} = [x_c \ y_c \ a \ b \ \theta]^T$.

Here, we set x_c , y_c , a , b and θ as the five parameters to be optimised, see Figure 1. Parameter θ represents the angle of rotation of the ellipse counter clockwise from the x-axis. For each point $\mathbf{x}_i = (x_i, y_i)$,

$$\rho_i = (\mathbf{x}_i - \mathbf{c})^T \mathbf{A} (\mathbf{x}_i - \mathbf{c}) - 1 \quad (4)$$

is the offset of \mathbf{x}_i from the ellipse. The offset ρ_i is positive if \mathbf{x}_i lies outside the ellipse, equals zero if \mathbf{x}_i is on the ellipse, and is negative if \mathbf{x}_i lies inside the ellipse. Thus, to encourage points outside to be on the ellipse, we set our objective function as

$$f(\mathbf{z}) = \sum_{i=1}^N \frac{[\rho_i]_+^2}{K + [\rho_i]_+^2} \quad (5)$$

for some $K > 0$ and defining $[\rho_i]_+ = \max(\rho_i, 0)$. The constraints consist of simple bounds on the decision variables

$$\begin{aligned} x_{\min} &\leq x_c \leq x_{\max}, & y_{\min} &\leq y_c \leq y_{\max}, \\ \ell_{\min} &\leq \mathbf{a}, \mathbf{b} \leq \ell_{\max}, & 0 &\leq \theta \leq \pi, \end{aligned}$$

together with the nonlinear constraints $\mathbf{c}_i = -\rho_i \leq 0$, which are designed to prevent any data point from lying inside the optimised ellipse.

Once we obtain all optimised ellipses, we need a method to blend these ellipses to describe the shape of the scattered data points. Each optimised ellipse has the form of equation (1) for some symmetric matrix \mathbf{A} and centre \mathbf{c} . The left hand side of equation (1) is a positive definite quadratic form. The square root of a quadratic form,

$$s = \sqrt{(\mathbf{x} - \mathbf{c})^T \mathbf{A} (\mathbf{x} - \mathbf{c})}, \tag{6}$$

is suitable for representing a single ellipse via the equation $s(\mathbf{x}) = 1$, but not for combining several ellipses. The difficulty with the root quadratic form in equation (6), is that it becomes larger as one moves away from the ellipse's centre. In a linear combination of such functions the distant ellipses' root quadratic forms can swamp those of closer ellipses. To avoid this effect we need a function which is negligible (and preferably zero) far from an ellipse.

To this end we introduce the sigmoidal function

$$\phi(s) = \begin{cases} 2, & \text{if } t \leq 0, \\ 2 - t^3/(3h^3), & \text{if } 0 < t \leq h, \\ 2 - \left(\frac{1}{3} + \frac{t-h}{h} + \frac{(t-h)^2}{h^2} - \frac{2(t-h)^3}{3h^3} \right), & \text{if } h < t \leq 2h, \\ (3h - t)^3/(3h^3), & \text{if } 2h < t \leq 3h, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $h > 0$ is the skirt width parameter, $s = \sqrt{(\mathbf{x} - \mathbf{c})^T \mathbf{A} (\mathbf{x} - \mathbf{c})}$, and $t = s - (1 - \frac{3}{2}h)$.

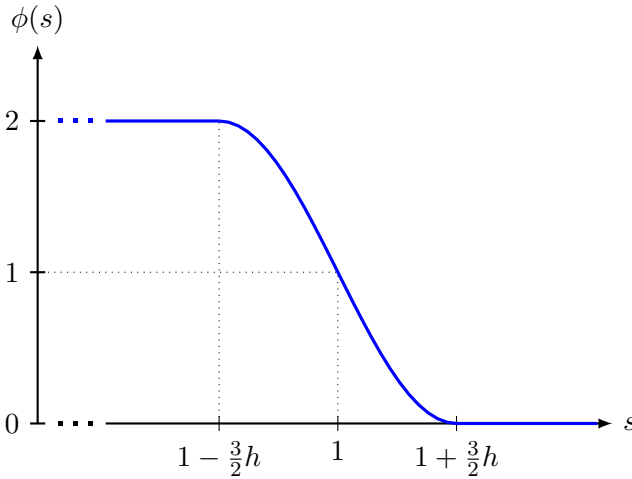


FIGURE 2: The sigmoidal function $\phi(s)$.

The sigmoidal function $\phi(s)$ equals two for small s and decreases monotonically as s increases through the interval $[1 - \frac{3}{2}h, 1 + \frac{3}{2}h]$, as is shown in Figure 2. For large s , $\phi(s) = 0$ so this particular ϕ vanishes when far from its ellipse. A single ellipse is recovered from $\phi(s)$ via the equation $\phi(s) = 1$. Different ellipses give rise to different distance functions s_j . Applying the sigmoidal function we define

$$\psi(\mathbf{x}) = \sum_j \lambda_j \phi(s_j(\mathbf{x})). \quad (8)$$

The curve corresponding to the combined ellipses is given by $\psi(\mathbf{x}) = 1$. The λ_j 's are chosen to optimise the fit of $\psi(\mathbf{x}) = 1$ to the data points. This is done by performing a least squares solution of the system

$$\psi(\mathbf{x}_i) = \sum_j \lambda_j \phi(s_j(\mathbf{x}_i)) \approx 1, \quad \text{for all } i, \quad (9)$$

where \mathbf{x}_i is the i th data point. This system is usually overdetermined as

the interesting case is when the number of ellipses is small compared to the number of data points.

3.1 2D results and discussion

Our optimisation algorithm requires an initial estimate of the parameters. Therefore we set up an automatic algorithm to identify potential centres of ellipses to be optimised one by one. We first optimise the largest inscribed ellipse. This is done by generating the midpoints of some randomly chosen pairs of data points. Then, choose the midpoint that has the largest distance from its closest data point as the initial centre of an inscribed ellipse. A kd-tree structure is constructed to find the nearest point in the given data set, to any specified point.

Once we have identified the first ellipse, we use the implicit function $\psi(\mathbf{x}) = 1$ to determine potential centres. Note that any data point on the implicit curve $\psi(\mathbf{x})$ returns the value one. Therefore we can locate data points that are far away from the first ellipse. To insert an ellipse for optimisation, we identify one far away data point \mathbf{x}_{far} and its neighbouring data points. We fit a best line through this cluster of points and return the inward normal \mathbf{n}_{in} of the line. We define

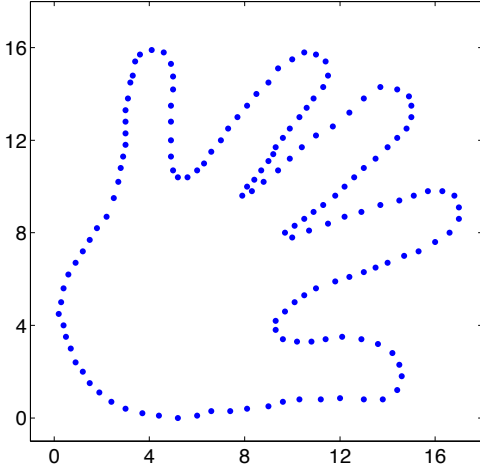
$$\mathbf{x}^* = \mathbf{x}_{\text{far}} + \mathbf{k} \mathbf{d} \mathbf{n}_{\text{in}}, \quad (10)$$

where \mathbf{x}^* is the estimated centre of the new ellipse, \mathbf{d} is the distance from \mathbf{x}_{far} to the closest data point and \mathbf{k} is some parameter with $1 < \mathbf{k} < 2$.

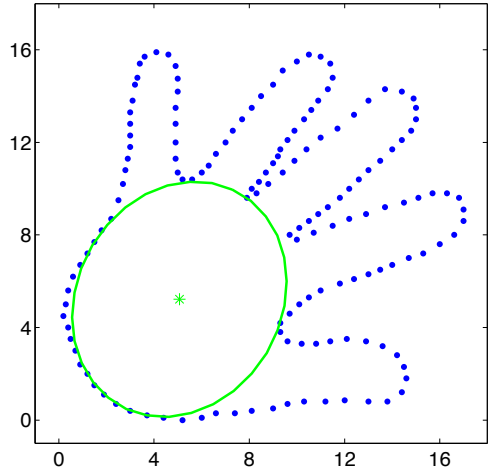
After each optimisation step, we calculate the residual sum of squares (RSS),

$$\text{RSS} = \sum_{i=1}^N (\psi(\mathbf{x}_i) - 1)^2, \quad (11)$$

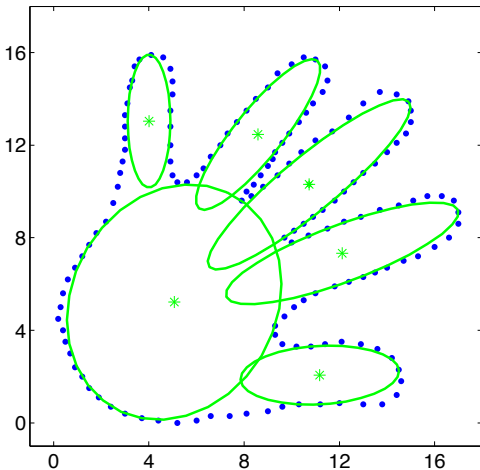
for all data points. The process of updating the implicit curve $\psi(\mathbf{x}) = 1$ and inserting an ellipse is repeated until the change in RSS due to adding a new ellipse is less than some constant, in this case 1.



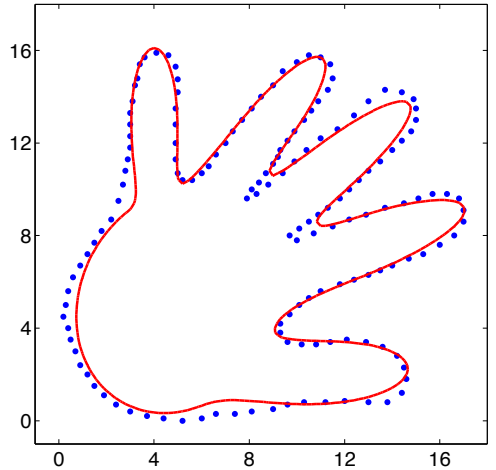
(a) The given data points of a hand.



(b) The first optimised largest inscribed ellipse.



(c) All the six optimised ellipses.



(d) The final shape of the hand.

FIGURE 3: Reconstructing the coarse shape of a hand using ellipses.

Figure 3 shows the process of constructing the coarse shape of a hand using ellipses. A prototype implementation was written in MATLAB. The 151 points of a hand are shown in Figure 3(a). Figure 3(b) shows the largest inscribed ellipse. A total of six optimised ellipses are identified for gross features using $K = 1$ in equation (5) and $k = 1.7$ in equation (10). To blend the ellipses, $h = 0.6$ is used in equation (7). Figure 3(d) shows the extracted contour $\psi(\mathbf{x}) = 1$ of the implicit function. The shape of the hand is clearly reconstructed.

4 Fitting ellipsoids in 3D

We extend the previous method and algorithm to obtain a coarse approximation to a shape in three dimensional space. Given a set of scattered data points in 3D, (x_i, y_i, z_i) where $i = 1, 2, \dots, N$, we fit the shape of the scattered points by some ellipsoids. The equation of an ellipsoid is

$$(\mathbf{x} - \mathbf{c})^T \mathbf{A} (\mathbf{x} - \mathbf{c}) = 1$$

where $\mathbf{x} = (x, y, z)$ is point on the ellipsoid, $\mathbf{c} = (x_c, y_c, z_c)$ is the centre of the ellipsoid and \mathbf{A} is a 3×3 symmetric positive definite matrix. Matrix \mathbf{A} has spectral decomposition $\mathbf{A} = \mathbf{Q} \mathbf{D} \mathbf{Q}^T$, where $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$ are the eigenvalues of \mathbf{A} and \mathbf{Q} is a rotation matrix. The eigenvalues give information about the semi axes of the ellipsoid,

$$\mathbf{a} = \frac{1}{\sqrt{\sigma_1}}, \quad \mathbf{b} = \frac{1}{\sqrt{\sigma_2}} \quad \text{and} \quad \mathbf{c} = \frac{1}{\sqrt{\sigma_3}}, \quad (12)$$

shown in Figure 4. The semi axes lengths \mathbf{a} , \mathbf{b} and \mathbf{c} are all positive real numbers that determine the shape of the ellipsoid.

To represent the ellipsoid's orientation, we use Rodrigues' formula to write the rotation matrix in exponential representation [2]. Other choices such as Euler angles, or a quaternion representation have disadvantages in the optimisation

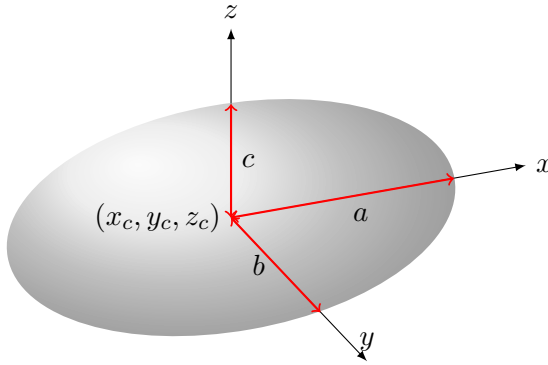


FIGURE 4: Parameters of an ellipsoid used in the optimisation algorithm.

procedure. Euler angle representations include bounds that must be dealt with, and quaternions use one additional parameter.

In the exponential representation, $\mathbf{Q} = e^{\hat{\mathbf{w}}}$ is a rotation about axis \mathbf{w} through $\|\mathbf{w}\|$ radians where the skew symmetric matrix

$$\hat{\mathbf{w}} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \quad (13)$$

Given $\mathbf{w} \in \mathbb{R}^3$, Rodrigues formula gives the matrix exponential

$$e^{\hat{\mathbf{w}}} = \mathbf{I} + \frac{\hat{\mathbf{w}}}{\|\mathbf{w}\|} \sin \|\mathbf{w}\| + \frac{\hat{\mathbf{w}}^2}{\|\mathbf{w}\|^2} (1 - \cos \|\mathbf{w}\|). \quad (14)$$

To avoid loss of significance when $\|\mathbf{w}\|$ is small, we rewrite equation (14) by applying a half angle formula obtaining

$$e^{\hat{\mathbf{w}}} = \mathbf{I} + \frac{\hat{\mathbf{w}}}{\|\mathbf{w}\|} \sin \|\mathbf{w}\| + 2 \sin^2 \left(\frac{\|\mathbf{w}\|}{2} \right) \frac{\hat{\mathbf{w}}^2}{\|\mathbf{w}\|^2}. \quad (15)$$

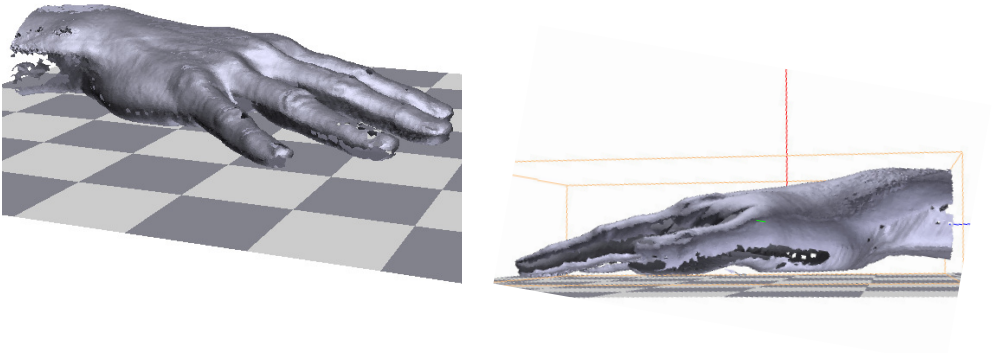


FIGURE 5: Original data set of a hand with 57,126 laser scan points.

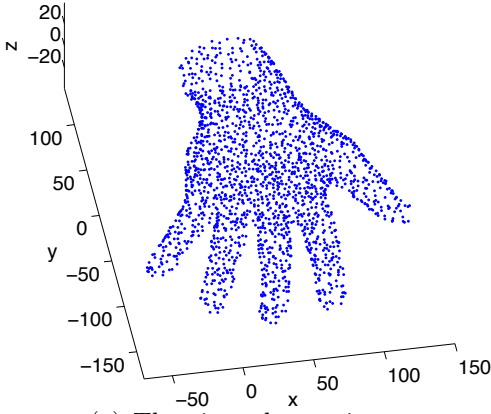
From equation (15), the matrix exponential $e^{\hat{w}} \rightarrow I$ as $\|w\| \rightarrow 0$. The rotation matrix Q is therefore given by (15) when $w \neq 0$, and is I when $w = 0$. Using this definition, the optimiser can adjust w without constraint.

In fitting an ellipsoid, the same optimisation algorithm as described in section 3 is used, with minor modifications. We now have nine parameters to be optimised, three to indicate the ellipsoid's centre, three to represent the lengths of the semi axes, and three to represent its orientation, that is,

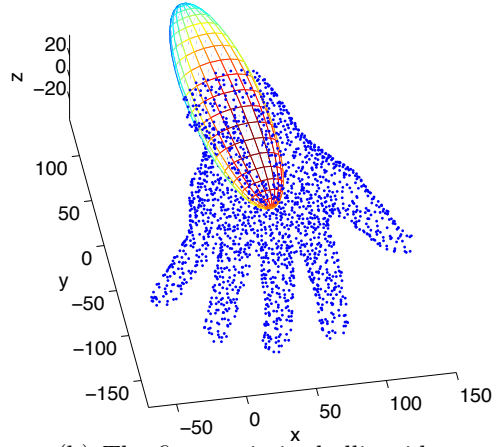
$$z = [x_c \quad y_c \quad z_c \quad a \quad b \quad c \quad w_1 \quad w_2 \quad w_3]^T.$$

4.1 3D results and discussion

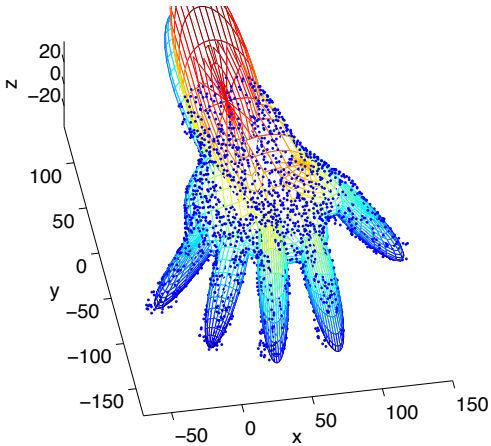
For the 3D case, we also developed an automatic method for identifying the initial ellipsoids for the coarse level approximation. For a given set of 3D 'on surface' points and their corresponding point normals, we form an initial centre of an ellipsoid using equation (10). In this case, the first ellipsoid is selected by choosing the largest ellipsoid from a small number of ellipsoids generated from random starting data points. After that, a procedure similar to the 2D case is used, that is, update the implicit function ψ , identify a



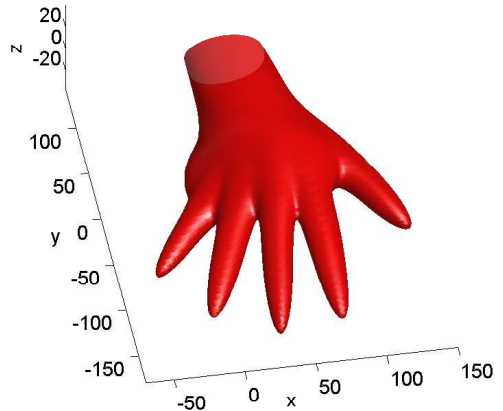
(a) The given data points.



(b) The first optimised ellipsoid.



(c) The original data points with 15 optimised ellipsoids.



(d) The final isosurface generated by combining the ellipsoids.

FIGURE 6: Reconstructing the coarse shape of a hand using ellipsoids.

poorly fitted region and insert subsequent ellipsoids for optimisation in that region.

The 3D hand model used in our test case contains 1905 data points. It is a heavily down-sampled version of the 57,126 point laser scan shown in Figure 5 and supplied to us by Brent Price of Applied Research Associates New Zealand Limited. Even the original data set contains significant holes, particularly on sides of the fingers and the thumb, and at the wrist. The down-sampled dataset also has holes at the fingertips. Since our optimisation algorithm is designed such that any ellipsoid will extend between the data points, ellipsoids may extend through the holes. In our experience ellipsoids that are completely outside the data do not improve the model. These were therefore detected and removed, increasing the simplicity of the model. Detecting such ellipsoids is easily done by comparing normals to the ellipsoids and data normals at data points lying on the ellipsoid.

The first optimised ellipsoid, shown in Figure 6(b), partly extends outside the data range. The final fifteen optimised ellipsoids are shown in Figure 6(c) using parameters $K = 1$ in equation (5) and $k = 1.3$ in equation (10). However, we are only interested in recovering the coarse surface of the approximation within the data range. In regards to that, our final implicit surface is not affected by ellipsoids that partly extend outside the data range. Figure 6(d) shows the isosurface of ψ corresponding to $\psi(\mathbf{x}) = 1$ with $h = 0.6$.

5 Conclusions

We demonstrated a method to approximate a given point cloud with an implicit blobby representation. Our work focused on a coarse level approximation with relatively few parameters. The results show that the reconstructed curves and shapes can describe the original point clouds well in both 2D and 3D.

References

- [1] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T. R. Evans, *Reconstruction and Representation of 3D Objects With Radial Basis Functions*, pp.67–76 in SIGGRAPH 2001, ACM Press/ACM SIGGRAPH, 2001. [C597](#)
- [2] Y. Ma, S. Soatto, J. Kosecka and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, 2004, Chapter 2. [C605](#)
- [3] S. Bischoff and L. Kobbelt, *Ellipsoid decomposition of 3D-models*, 3DPVT Proceedings, 2002, pp.480–488. [C598](#), [C599](#)
- [4] S. Liu, X. Jin, C. C. L. Wang and K. Hui, *Ellipsoidal-blob approximation of 3D models and its applications*, Computers & Graphics, 31(2), 2007, pp.243–251. [doi:10.1016/j.cag.2006.12.004](https://doi.org/10.1016/j.cag.2006.12.004) [C599](#)
- [5] P. D. Simari and K. Singh, *Extraction and remeshing of ellipsoidal representations from mesh data*, Proceedings of Graphics Interface 2005, 2005, pp.168. [C598](#)
- [6] I. Söderkvist, *Introductory overview of surface reconstruction methods*, Research Report 10, Department of Mathematics, Luleå Univeristy, Sweden, 1999. [C598](#)
- [7] X. Jin, S. Liu, C. C. L. Wang, J. Feng and H. Sun, *Blob-based liquid morphing*, Computer Animation and Virtual Worlds, 16 (3–4), 2005, pp.391–403. [doi:10.1002/cav.84](https://doi.org/10.1002/cav.84) [C598](#), [C599](#)

Author addresses

1. **W. E. Ong**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.

2. **R. K. Beatson**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.
3. **C. J. Price**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.