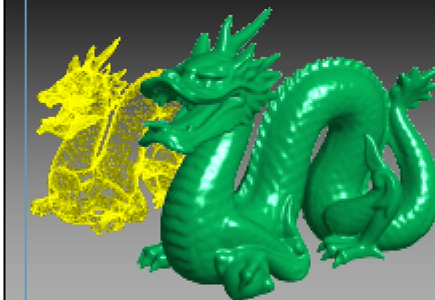
Reconstruction and Representation of 3D Objects with Radial Basis Functions

J. C. Carr[1,2], R. K. Beatson[2], J. B. Cherrie[1,2], T. J. Mitchell[1,2], W. R. Fright[1], B. C. McCallum[1] & T. R. Evans[1]

[1]Applied Research Associates NZ Ltd,
[2]Dept. Mathematics & Statistics, University of Canterbury, NZ.

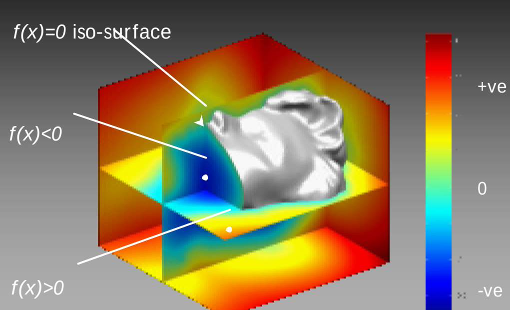


472,000 point-cloud → Implicit function $f(x)$ (32,000 term RBF)

Laser scanning & mesh repair

Mesh repair

Implicit surface modeling



RBF surface modeling

The problem

To find an interpolant s such that

$s(x_i) = 0$, $i = 1,\ldots,n$ (known surface points)

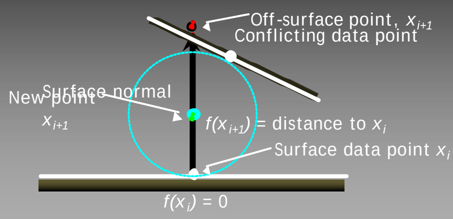
$s(x_i) = d_i \neq 0$, $i = n+1,\ldots,N$ (*off*-surface points)

Our method

- Form a signed-distance distribution
- Interpolate distance field (fit an RBF)
- Iso-surface RBF

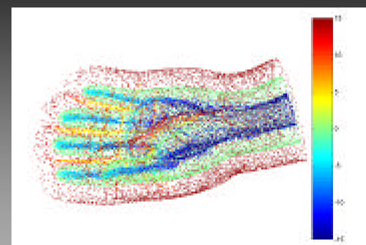
Generating off-surface data

Ensure a consistent distance-to-surface field



- Validate normal lengths

Forming a signed-distance function



Outward normal points

On-surface points

Inward normal points

- Off-surface points are projected along surface normals

Minimal energy interpolants

We want to find the *smoothest* function which fits our distance-surface data.

minimize
$$\int_{\mathbb{R}^3} \left(\frac{\partial^2 s(x)}{\partial x^2}\right)^2 + \left(\frac{\partial^2 s(x)}{\partial y^2}\right)^2 + \left(\frac{\partial^2 s(x)}{\partial z^2}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial x \partial y}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial x \partial z}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial y \partial z}\right)^2 dx$$

i.e., minimize the 2nd derivative

Thin-plate spline in 3D

The minimizing interpolant has the form :

Linear polynomial Scalar weight

$$s(x) = p(x) + \sum_{i=1}^{N} \lambda_i |x - x_i|,$$

How far is $\boldsymbol{x}$ from $\boldsymbol{x}_i$

Radial Basis Functions

This is a specific example of an RBF

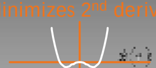
$$s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi(|x - x_i|),$$

where $\phi(r) = |r|$

Choices for ϕ :

$\phi(r) = |r|$

Minimizes 2nd derivative in 3D



$\phi(r) = r^2 \log(r)$

$\phi(r) = r^3$

Minimizes 3rd derivative in 3D / Minimizes 2nd derivative in 2D

How do we find the weights λ_i?

Form & solve the linear system :

N+4 x N+4

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

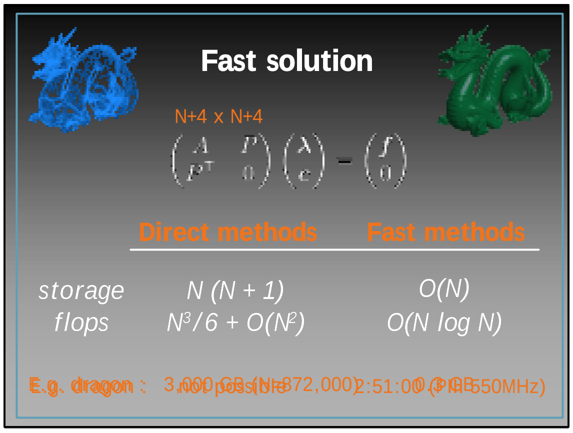
Matrix dependent on the locations of the data points

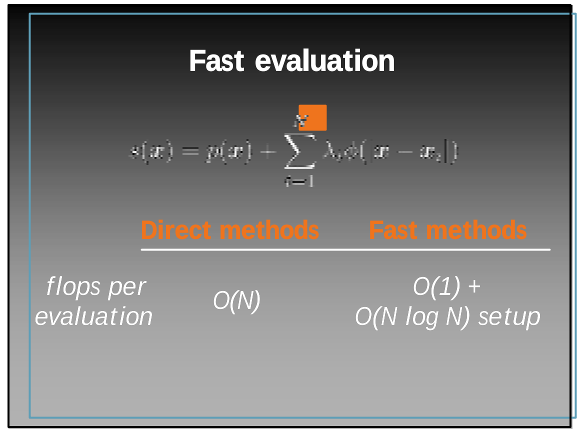
Unknowns to find

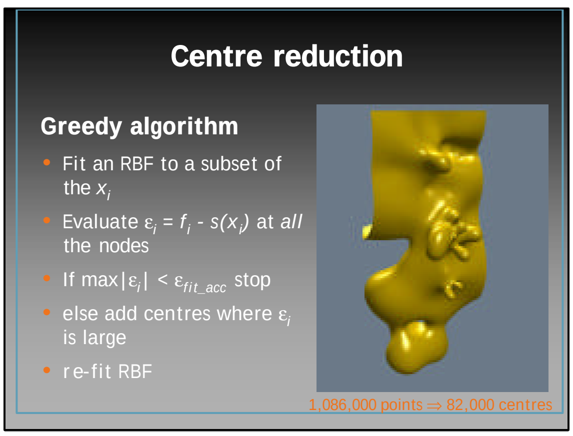
Known distance values at x_i

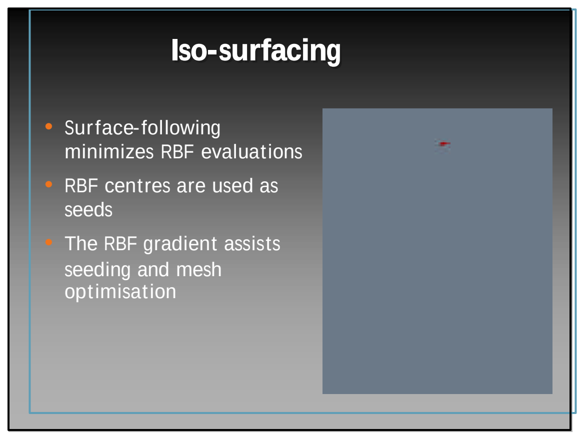
where $A_{i,j} = \phi(|x_i - x_j|), \quad i, j = 1, \ldots, N,$

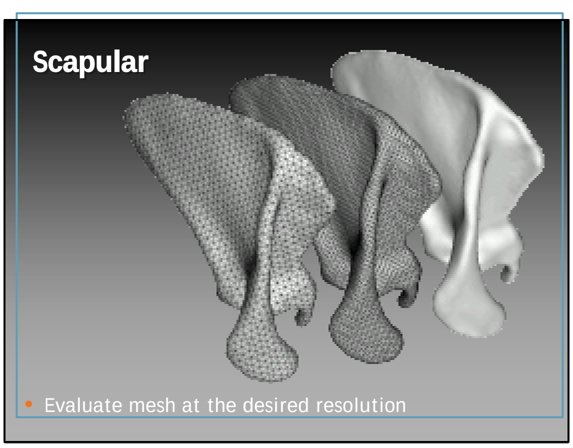
$P_{i,j} = p_j(x_i), \quad i = 1, \ldots, N, \quad j = 1, \ldots, 4.$

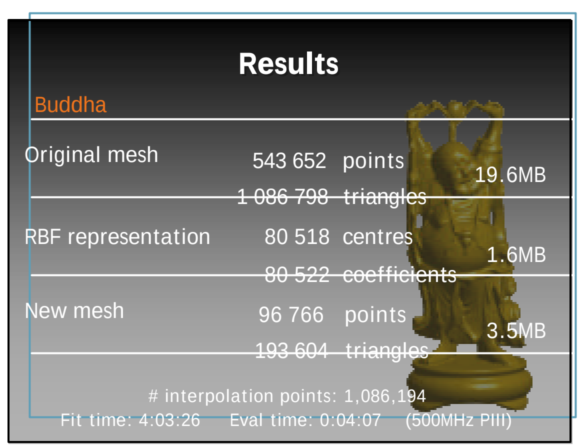

Fast solution
N+4 x N+4
Direct methods
Fast methods
storage
flops
N (N + 1)
N³/6 + O(N²)
O(N)
O(N log N)
E.g. dragon :


Fast evaluation
Direct methods
Fast methods
flops per evaluation
O(N)
O(1) + O(N log N) setup

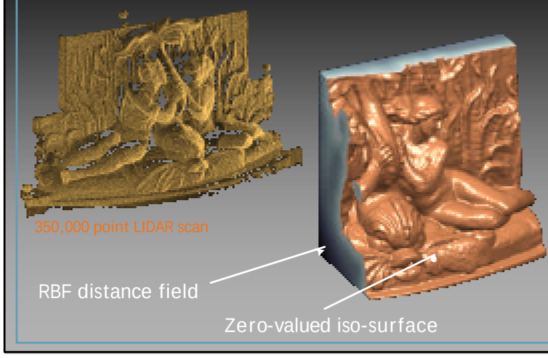

Centre reduction
Greedy algorithm
Fit an RBF to a subset of the x_i
Evaluate ε_i = f_i - s(x_i) at all the nodes
If max|ε_i| < ε_fit_acc stop
else add centres where ε_i is large
re-fit RBF
1,086,000 points ⇒ 82,000 centres


Iso-surfacing
Surface-following minimizes RBF evaluations
RBF centres are used as seeds
The RBF gradient assists seeding and mesh optimisation


Scapular
Evaluate mesh at the desired resolution


Results
Buddha
Original mesh
543 652 points
1 086 798 triangles
19.6MB
RBF representation
80 518 centres
80 522 coefficients
1.6MB
New mesh
96 766 points
193 604 triangles
3.5MB
interpolation points: 1,086,194
Fit time: 4:03:26 Eval time: 0:04:07 (500MHz PIII)

Interpolating noisy data



RBF smoothing

Look for the function $s^\star$ that minimizes

$$\rho\|s\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left(s(x_i) - f_i\right)^2$$

2nd derivative energy term (smoothness)

Closeness of fit

$s^\star$ is also an RBF with the coefficients given by

$$\|s\|^2 = \int_{\mathbb{R}^3} \left(\frac{\partial^2 s(x)}{\partial x^2}\right)^2 + \left(\frac{\partial^2 s(x)}{\partial y^2}\right)^2 + \left(\frac{\partial^2 s(x)}{\partial z^2}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial x \partial y}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial x \partial z}\right)^2 + 2\left(\frac{\partial^2 s(x)}{\partial y \partial z}\right)^2 dx.$$

RBF smoothing

Look for the function $s^\star$ that minimizes

$$\rho\|s\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left(s(x_i) - f_i\right)^2$$

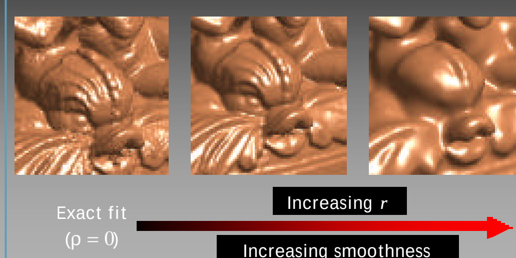
Rearranging :

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix}\begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f - \boxed{8N\pi\rho\lambda} \\ 0 \end{pmatrix}$$

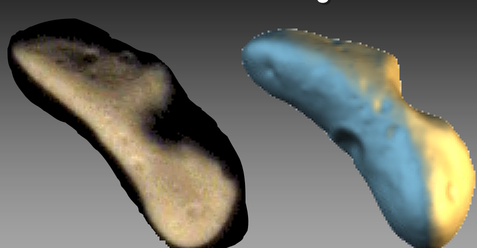
Deviation at each data point

r determines amount of smoothing

Spline smoothing with RBFs



Reconstructing Eros

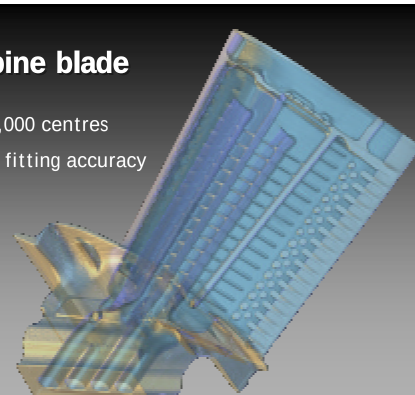


Interpolating irregular, non-uniformly sampled range data from NASA's NEAR spacecraft

http://near.jhuapl.edu/iod/20000728/index.html

Turbine blade

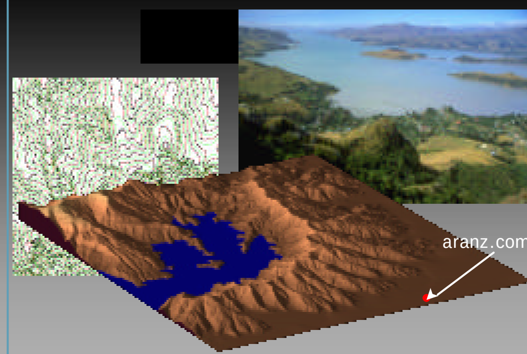
- 594,000 centres
- 10^{-4} fitting accuracy



Conclusions

- A functional representation of a complex object is possible i.e. $f(x)$
- Smooth RBF interpolation is ideal for mesh repair
- The smoothest surface, most consistent with the input data, is produced
- Gradients are determined analytically, i.e. $\nabla f(x)$
- Fast evaluation is essential

Email : j.carr@aranz.com

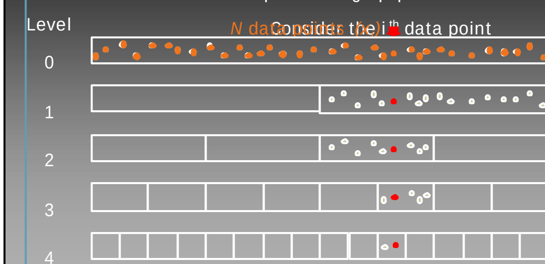


RBF model of Christchurch & Lyttelton $\phi(r) = r^2 \log(r)$

Fast multipole methods

Fast multipole methods

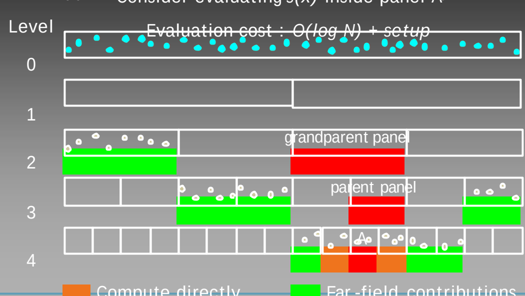
1. Hierarchical partition of space



Fast multipole methods

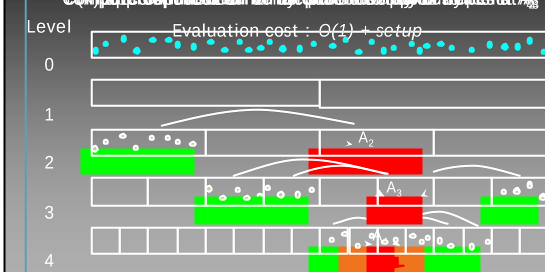
2. Approximate far panel contributions with Laurent series

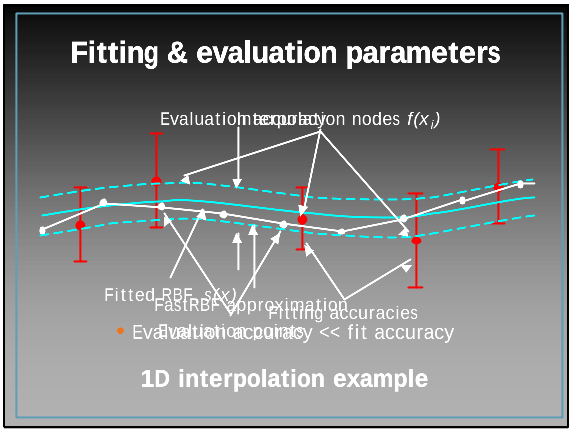
Evaluation cost : $O(\log N)$ + setup

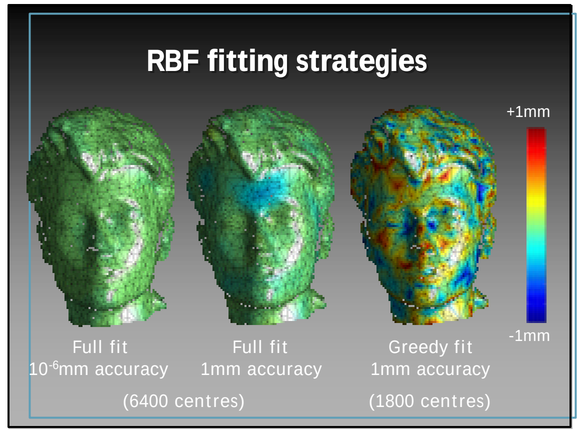


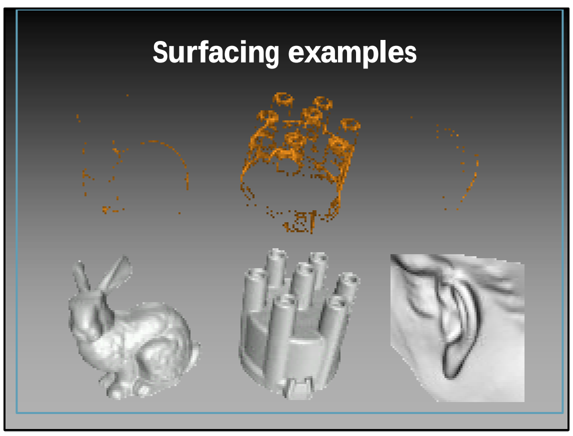
Fast multipole methods

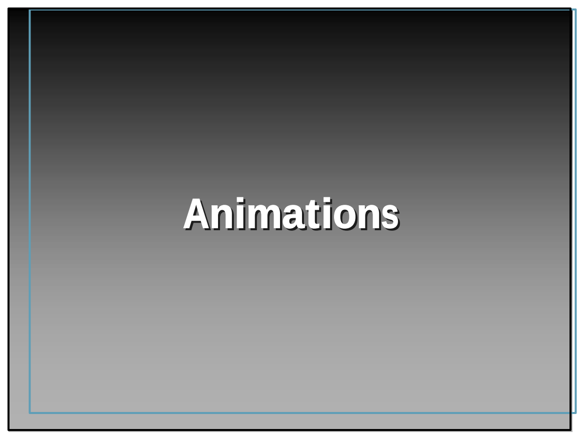
Evaluation cost : $O(1)$ + setup



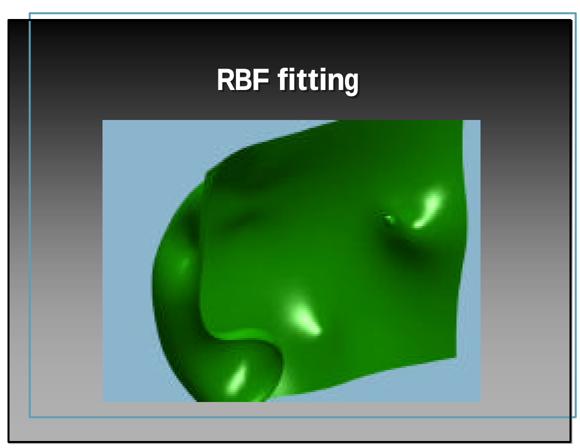

Fitting & evaluation parameters
Evaluation accuracy
Interpolation nodes f(x_i)
Fitted RBF s(x)
FastRBF approximation
Fitting accuracies
Evaluation points
Evaluation accuracy << fit accuracy
1D interpolation example


RBF fitting strategies
+1mm
-1mm
Full fit
10^-6mm accuracy
Full fit
1mm accuracy
Greedy fit
1mm accuracy
(6400 centres)
(1800 centres)


Surfacing examples

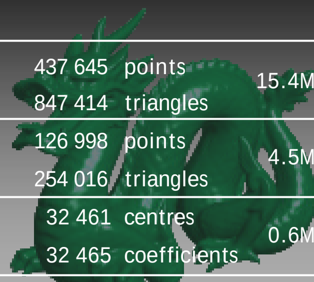

Animations


RBF smoothing


RBF fitting

Results

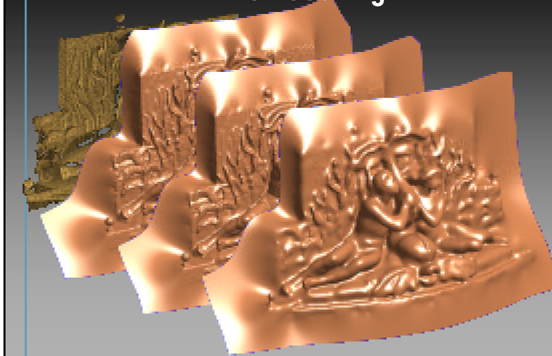
Dragon



Original mesh	437 645 points 847 414 triangles	15.4MB
New mesh	126 998 points 254 016 triangles	4.5MB
RBF representation	32 461 centres 32 465 coefficients	0.6MB

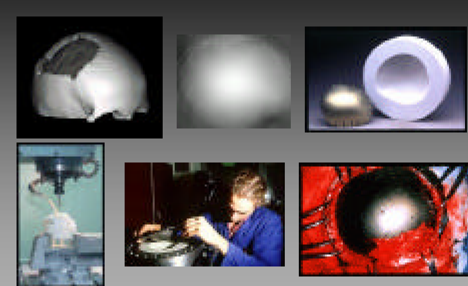
interpolation points: 872,487 Fit time: 2:51:09
Eval time: 0:04:40

RBF smoothing



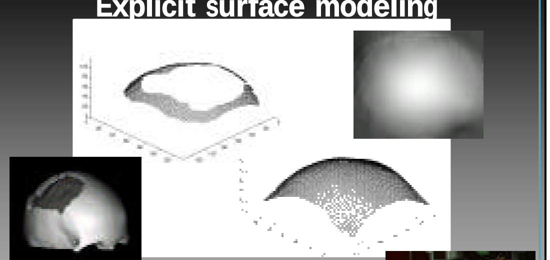
RBF Examples

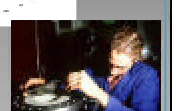
Clinical CAD-CAM example



Construction of a cranial prosthesis using an RBF model

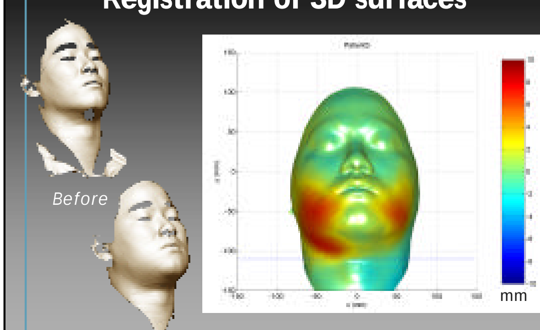
Explicit surface modeling





An RBF interpolates the surface of a skull to form a CAD model for the manufacture of a precision-fit titanium prosthesis.

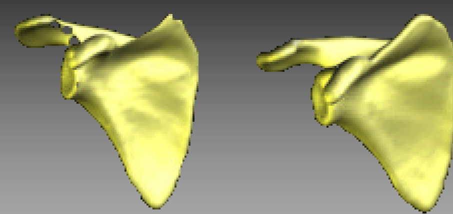
Registration of 3D surfaces



RBF difference – tissue swelling

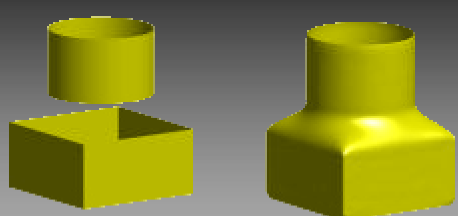
Surfacing examples

Mesh repair (hole-filling)



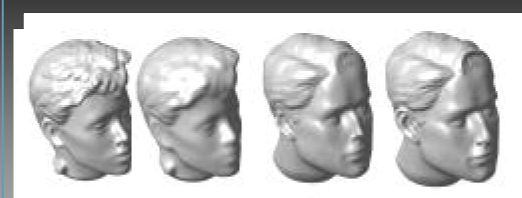
Water -tight surface fitted to an incomplete laser scan. No user -interaction was required.

Filleting



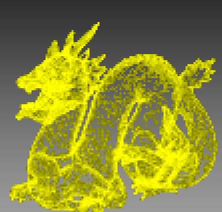
Fitting a smooth surface between two objects

Further applications



Blending & morphing

Point cloud reconstruction

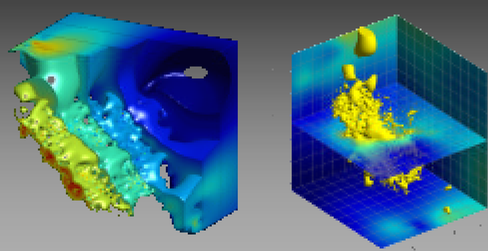


472,000 point cloud

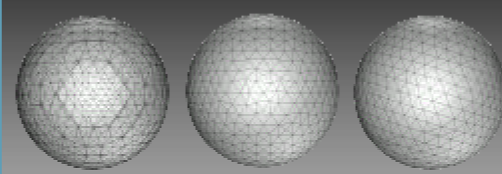
32,000 RBF terms

Density examples

3D density data visualisation



Mesh optimisation



Grid-constrained mesh

Optimised mesh FastSurf™

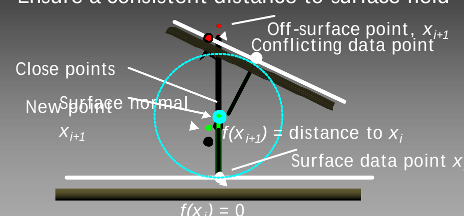
Mesh vertices constrained to lie on parallel planes

Acknowledgements

- Hand, statue & mannequinn data courtesy of Polhemus corporation
- LIDAR data courtesy of Allen Instruments & Supplies, 1474 Theresa St, Carpinteria, CA93013, USA
- Eros data courtesy NASA & Cornell university
- Buddha & dragon data courtesy of Stanford Computer Graphics laboratory
- All other data courtesy Georgia Institute of Technology

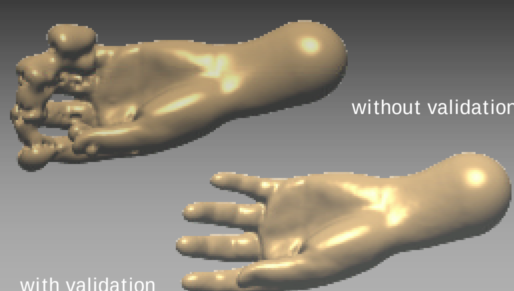
Generating off-surface data

Ensure a consistent distance-to-surface field



- Validate normal lengths
- Remove "duplicate" points

Normal length validation



Finding the weights l_i

The coefficients λ are uniquely determined by the interpolation conditions :

$$s(x_i) = f_i, \qquad i = 1, \dots, N, \tag{1}$$

and the orthogonality conditions :

$$\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0. \tag{2}$$

where $x = (x, y, z)$

i.e., all linear polynomials must be zero at the $\boldsymbol{x}_i$

Radial Basis Functions

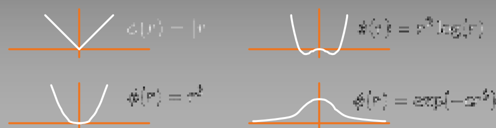
This is a specific example of an RBF

$$s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi(|x - x_i|),$$

where $\phi(r) = |r|$

Choices for f :

Minimizes 2nd derivative in 3D



$\phi(r) = |r|$

$\phi(r) = r^2 \log(r)$

$\phi(r) = r^3$

$\phi(r) = \exp(-cr^2)$

Gaussian – “compact” support

How do we find the weights λ_i?

Form & solve the linear system :

interpolation conditions

N+4 x N+4

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

Orthogonality conditions to ensure $|s|^2$ finite

Matrix dependent on the distance locations of the data points

Unknowns to find

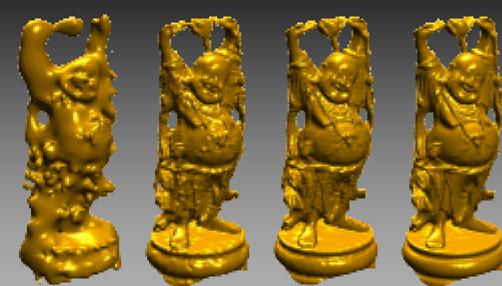
Known values at x_i

where $A_{i,j} = \phi(|x_i - x_j|)$, $i, j = 1, \ldots, N$,

$P_{i,j} = p_j(x_i)$, $i = 1, \ldots, N$, $j = 1, \ldots, \ell$.

Animation - static images

Iterative fitting of an RBF

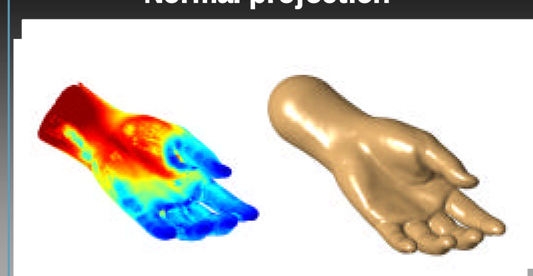


iteration 1 iteration 10 iteration 30 iteration 50

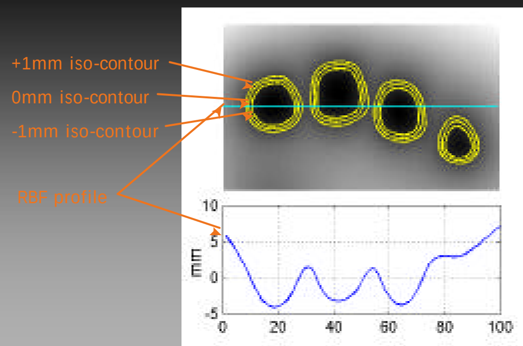
*Data courtesy of Stanford Computer Graphics Laboratory

Normal validation

Normal projection

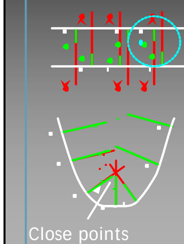


Normal lengths (left) and fitted surface (right)

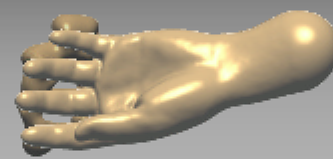


Normal length validation

Ensure a consistent distance-to-surface field

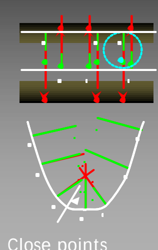


- Validate normal lengths
- Remove "duplicate" points

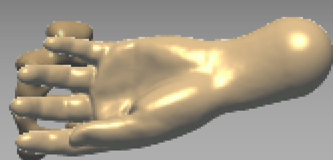


Normal length validation

Ensure a consistent distance-to-surface field



- Validate normal lengths
- Remove "duplicate" points



Talk outline

- Implicit surface representation
- Radial Basis Function (RBF) interpolation
- Fast algorithms for computing RBFs
- Results & applications

Why use RBFs?

RBFs are ideally suited to scattered data interpolation

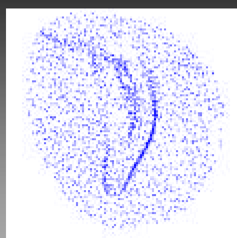
- Data are not required to lie on a regular grid
- Associated system guaranteed to be invertible under very mild conditions
- Poly harmonic splines minimize certain energy semi-norms and are therefore "smoothest" interpolators
- Results & applications

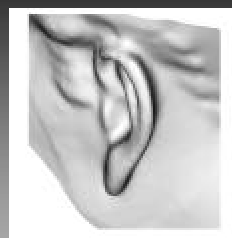
How to fit an implicit function

RBFs are ideally suited to scattered data interpolation

- Construct a distance-surface distribution
- Fit an RBF to the distance distribution
- Polygonize the object's boundary by iso-surfacing the RBF

Surfacing examples





Surface constructed from non-uniform point cloud data

Thin plate spline in 3D

If $s^* = \arg\min_{s \in S} \|s\|,$

Then

Linear polynomial Scalar weight

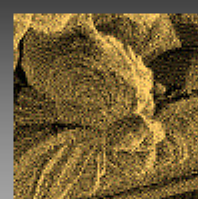
$$s^*(x) = p(x) + \sum_{i=1}^{N} \lambda_i |x - x_i|,$$

How far is ***x*** from ***x****i*

Fast algorithms

	Direct methods	Fast methods
Fitting		
storage	*N (N + 1)*	*O(N)*
flops	$N^3/6 + O(N^2)$	*O(N log N)*
Evaluation $s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi(\lvert x - x_i \rvert)$		
flops per evaluation	*O(N)*	*O(1) + O(N log N) setup*

Spline smoothing with RBFs

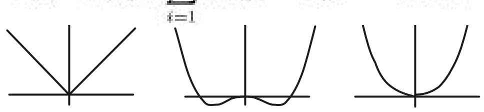


Detail of data from 350,000 point Cyra LIDAR scan

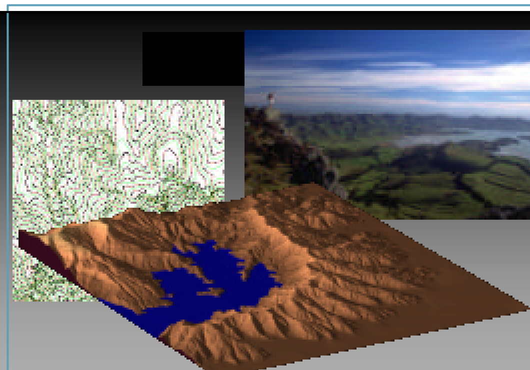
What is an RBF?

An RBF is a weighted sum of translates of a radially symmetric basic function augmented by a polynomial.

$$s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi(|x - x_i|), \qquad x \in \mathbb{R}^d,$$



$\phi(r) = r$ $\phi(r) = r^2 \log(r)$ $\phi(r) = r^3$



RBF model of Christchurch & Lyttelton $\phi(r) = r^2 \log(r)$