Field Functions, Height Surfaces, Depth Images

Topics

- Scalar field (2D), height surfaces, depth images
- Fields, Curves, and Surfaces
- Introduction to (height) surfaces

- Representations (polynomial surface patches, tensor-product cubic splines)
- Reconstruction
  - Interpolation
  - Approximation (Surface fitting)
- Segmentation (of point measurements into surface patches)
- Registration (of surfaces with point measurements)

Topics

- Basic methods for converting point measurements from (data sources)
  - Binocular stereo,
  - Active triangulation, and
  - Range cameras,

  into simple surface representations.

- Basic methods include
  - Converting point measurements into a mesh of triangular facets,
  - Segmenting range measurements into simple surface patches,
  - Fitting a smooth surface to the point measurements, and
  - Matching a surface model to the point measurements.

Fields (with coordinates & measurements)

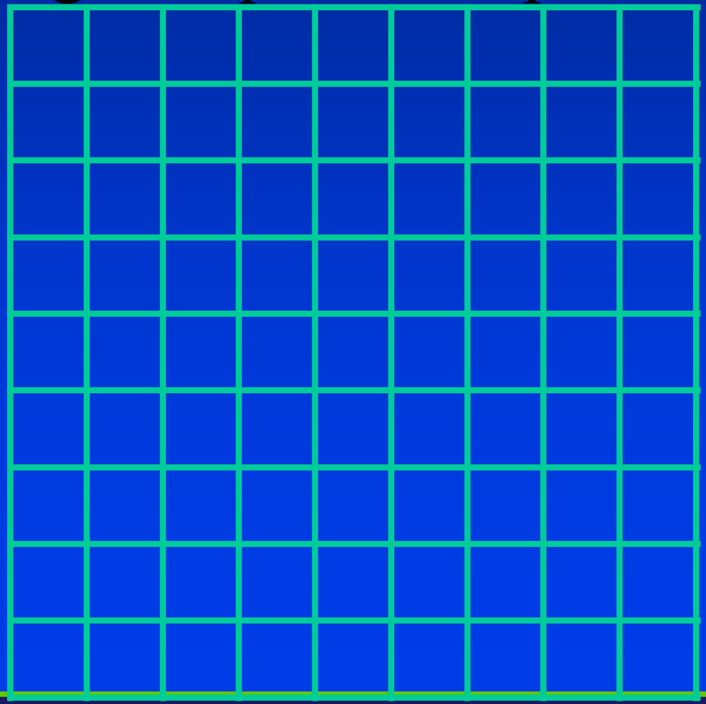
- Used to represent *measurements*, which are a mapping from the coordinate space to the data space.
- Image-related measurements include intensity and depth.
- Measurements are a mapping from the coordinate space to the data space.
- Coordinate space:
  - specifies the locations at which measurements were made;
- Data space:
  - specifies the measurement values.
- Data values are scalar measurements if the data space has only 1D; else, vector measurements.
- Examples - weather data, image.

Fields

- Coordinate systems
- Used to represent *measurements*, which are a mapping from the coordinate space to the data space
- Image-related measurements include intensity and depth
- Three types of fields are *uniform, rectilinear*, and *irregular*

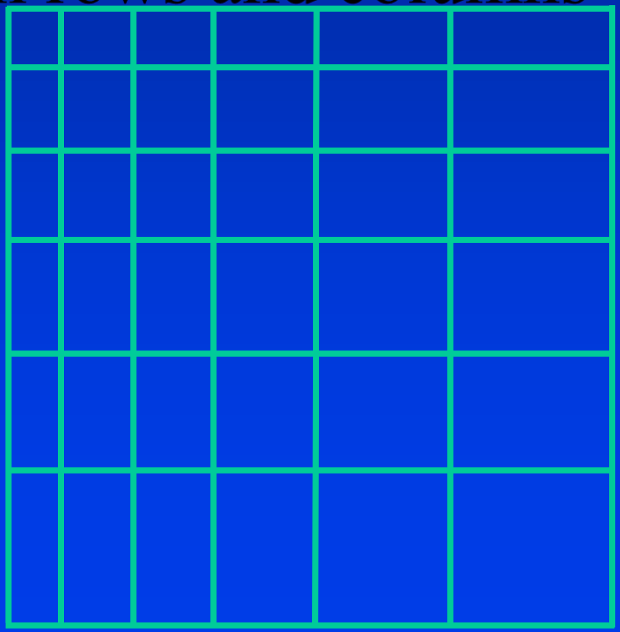
Uniform Fields

- Measurements stored in a rectangular grid
- Equal spacing between rows and columns
- Images – each grid square is a pixel



Rectilinear Fields

- Data samples not equally spaced along the coordinate axes
- Data samples organized on rectangular grid with varying distances between rows and columns



Irregular Fields

- Contain scattered (randomly-located) measurements or any pattern of measurements not corresponding to a rectilinear structure
- No overall organizational structure
- Similar to coordinate systems used in standard mathematics

Importance of Fields

- Allow flexibility in the representation of measurements.
- Example:
  - Depth measurements can be represented as displacement measurements in the uniform field of an image
  - Depth measurements can also be represented as points in the irregular field of three-dimensional space

The use of fields should become clearer later in the presentation

Introduction to Height Surfaces

- Typically used to provide a model for depth measurements.
- Surfaces are interpolated or approximated from depth measurements.
- Surfaces are then segmented into regions with similar characteristics (curvature).

Importance of Height Functions

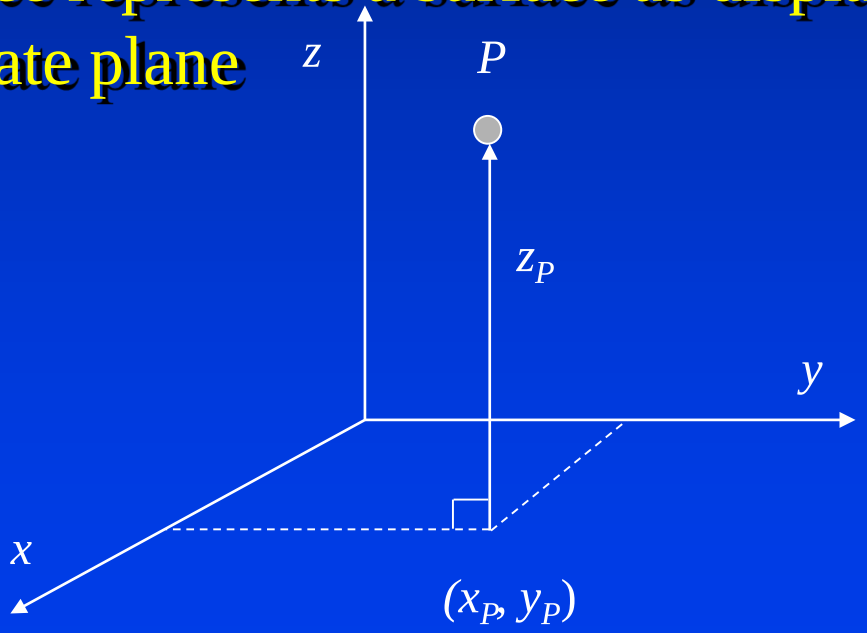
- Representation of depth measurements
- Analysis of depth measurements
- Data visualization
- Object recognition

Surfaces – Height Functions

- Typically used to provide a model for *depth measurements*
- Surfaces are interpolated or approximated from depth measurements
- Surfaces are then segmented into regions with similar characteristics (curvature)
- Discussion is similar to contours, which are 2D.
- Surfaces are 3D.

Geometry of Surfaces - Heights

- The explicit form is good for *graph surfaces*, which are surfaces represented as displacements from a coordinate plane
- A graph surface represents a surface as displacements normal to the coordinate plane



Surface Interpolation

- The surface representations can be used to interpolate samples of a graph surface, like depth measurements, obtained with binocular stereo or active triangulation.
- May be necessary when depth measurements do not conform to the uniform grid format required for image processing.
- May be necessary to interpolate depth measurements onto a uniform grid before using image processing algorithms, such as edge detection and segmentation.

- Types:
  - Triangular mesh interpolation
  - Bilinear interpolation
  - Robust interpolation

Planes

- Three points, $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$, define a plane in space
- The normal vector to the plane, $\mathbf{n}$, is defined as

$$\mathbf{n} = \mathbf{e}_1 \times \mathbf{e}_2$$

$$\text{where} \quad \mathbf{e}_1 = \mathbf{p}_1 - \mathbf{p}_0 \;\; \& \;\; \mathbf{e}_2 = \mathbf{p}_2 - \mathbf{p}_0$$

- The implicit equation for a plane is

$$(\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{n} = 0$$

where, $\mathbf{p}$ is a point that lies in the plane

- a, b, and c are the elements of $\mathbf{n}$

$$\text{or, equivalently} \quad ax + by + cz + d = 0$$

Triangular Mesh Interpolation

- Suppose, we have samples of a graph surface, $z = f(x, y)$, at scattered points (irregular field) using binocular stereo or active triangulation.
- We need to interpolate the depth measurements at grid locations [i,j] in the image plane, i.e., the z value at each point.
- Create a triangular mesh using the scattered point coordinates (x_i, y_i) and depth values (z).
- Connect the points in space to form a mesh of triangles.
- Since the depth measurements are from a graph surface, each triangle defines a plane explicitly:

$$z = a_0 + a_1 x + a_2 y$$

- Imagine overlapping the triangular mesh with the uniform image plane. Each pixel has coordinates (x_i, y_i).
- For each grid location, find the triangle that encloses point (x_i, y_i) and use the equation corresponding to this triangle to calculate the z at the grid location:

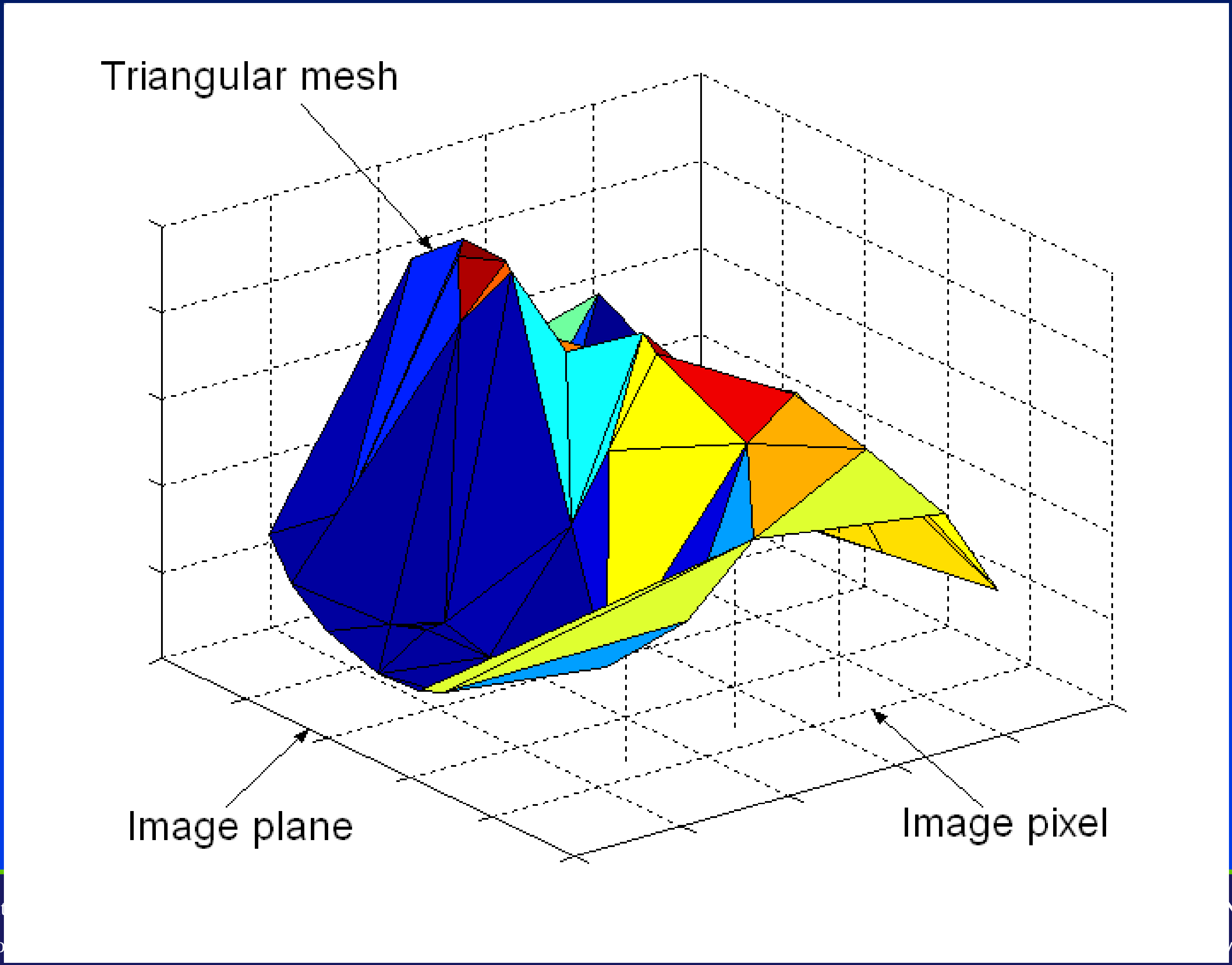
$$z_{ij} = a_0 + a_1 x_j + a_2 y_i$$

Triangular Mesh Interpolation

- Have samples of a graph surface, $z = f(x, y)$, at scattered points (irregular field)
- Want to interpolate the depth measurements at grid locations in the image plane
- Create a triangular mesh using the scattered point coordinates (x and y) and depth values (z)
- Connect the points in space to form a mesh of triangles
- Each triangle defines a plane with equation:

$$\boxed{z = a_0 + a_1 x + a_2 y}$$

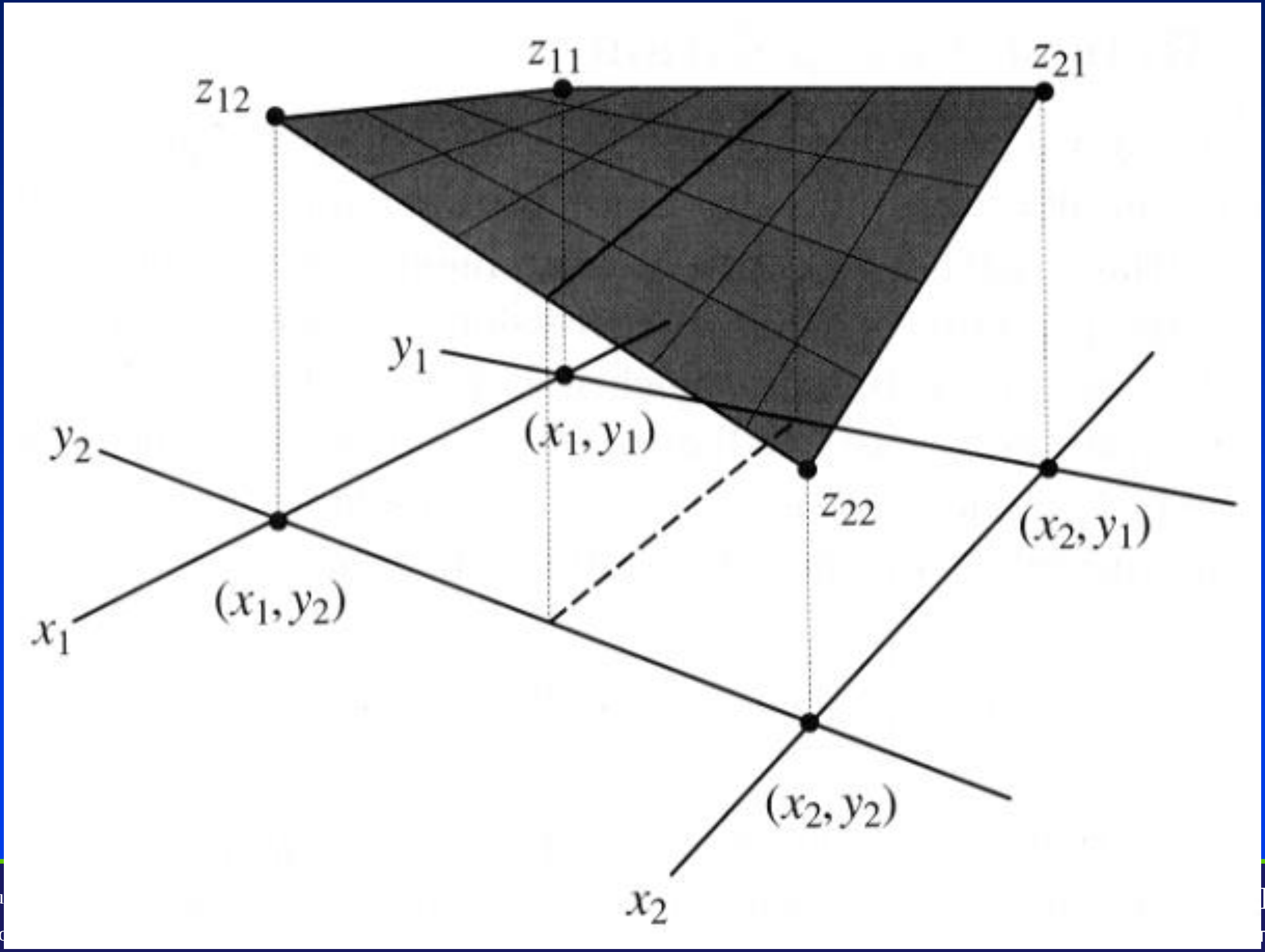
Triangular Mesh Interpolation - Example



Bilinear Interpolation

- Interpolates values on a rectilinear grid
- Can be used to interpolate a measurement, (x, y), between grid coordinates using the measurements at the four nearest grid locations.
- The four grid locations $(x_1, y_1), (x_1, y_2), (x_2, y_1)$, and (x_2, y_2) with measurements z_{11}, z_{12}, z_{21}, and z_{22} define the corners of a rectangle with sides parallel to the x and y axes containing (x, y).
- The aim is to find a bilinear surface patch that interpolates the four corners, then use this patch to interpolate the measurement at (x, y).

Bilinear Interpolation



Bilinear Interpolation

- A surface is bilinear if each cross section parallel to a coordinate axis is a line segment:

$$f(x,y)=a_1+a_2x+a_3y+a_4xy$$

- To find the bilinear surface patch, plug the 4 corner coordinates into the above equation. Solve the system of equations for
- $a_1, a_2, a_3, \text{ and } a_4.$

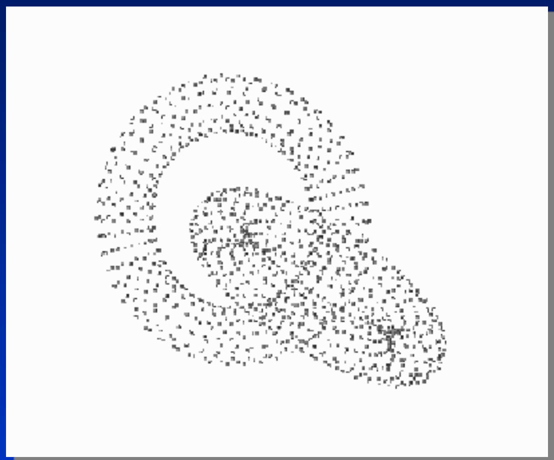
Use $f(x,y)$ to interpolate the measurement for (x,y).

- Special case for uniform grid (image):

$$f(\delta x,\delta y)=z_{11}+\delta x(z_{21}-z_{11})+\delta y(z_{12}-z_{11})+\delta x\delta y(z_{11}-z_{12}-z_{21}+z_{22})$$

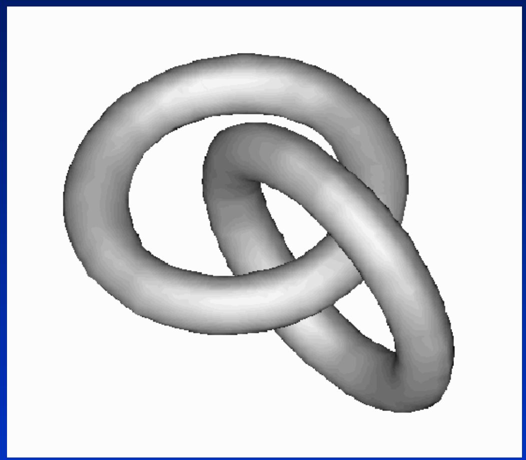
where $(\delta x,\delta y)$ is the offset from the upper left corner of a square

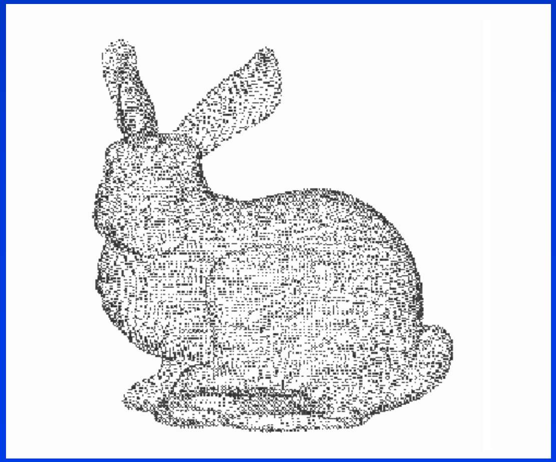
From Points to Surfaces



1200 points

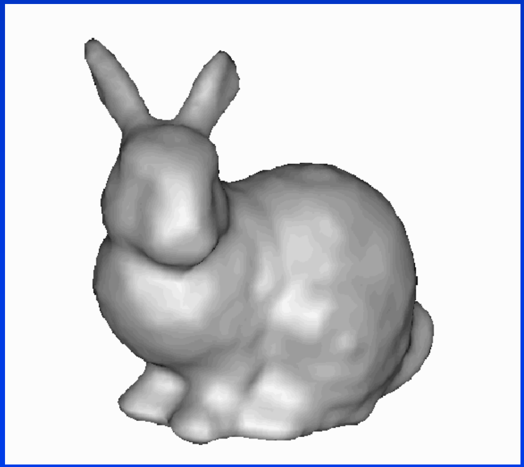
2 tori





35947 points

Bunny



Importance of Surfaces

- Representation of depth measurements
- Analysis of depth measurements
- Data visualization
- Object recognition

Surface Patches - Types

- Bilinear patches- any c/s // a coordinate axis is a line,

$$z = a_0 + a_1 x + a_2 y + a_3 xy$$

- Biquadratic patches-

$$z = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

- Bicubic patches-

$$\begin{aligned} z = {} & a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \\ & + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3 \end{aligned}$$

- Biquartic patches-

$$\begin{aligned} z = {} & a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \\ & + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3 \\ & + a_{10} x^4 + a_{11} x^3 y + a_{12} x^2 y^2 + a_{13} xy^3 + a_{14} y^4 \end{aligned}$$

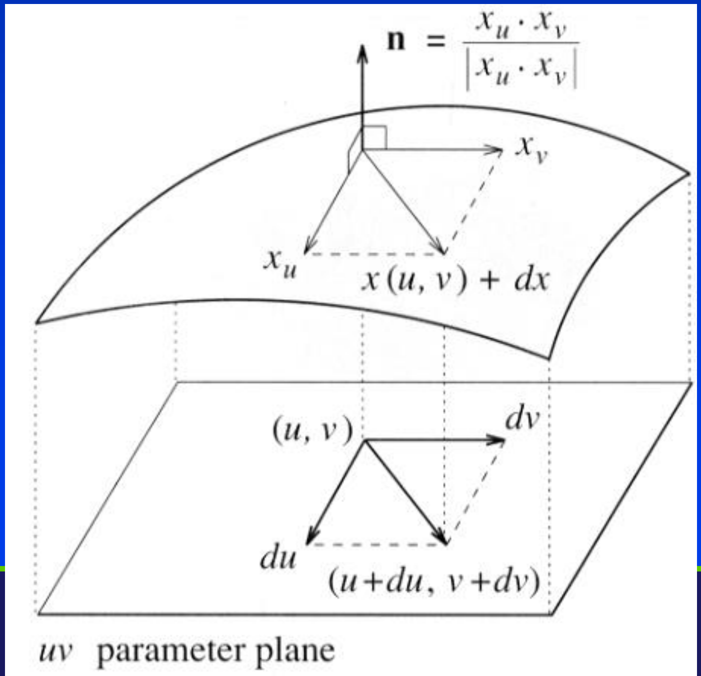
Biquadratic, bicubic and biquartic patches are bivariate polynomials that are frequently used to represent surface patches.

Utility of Surface Patches

- Good for modeling portions of a surface, such as the neighborhood around a point.
- Not convenient for modeling an entire surface.
- Can only be used to model graph surfaces.
- More complex surfaces can be modeled using cubic splines.

Differential Geometry

- Local analysis of how small changes in position (u,v) in the planar domain affect the position on the surface **p**(u,v), the first derivatives, $p_u(u,v)$ & $p_v(u,v)$ and the surface normal, **n**(u,v).
- The first derivatives at a point are two orthogonal vectors that span the tangent plane
- The surface normal **n** at this point **p** is the unit vector orthogonal (normal) to the tangent plane



Differential Geometry

- Slicing the surface with a plane containing the normal vector produces an infinite number of normal curves, depending on the orientation of the slicing plane.
- The minimum and maximum curvatures (principal curvatures) can be used to calculate the *Gaussian curvature* and *mean curvature*
- *Umbilic points* are locations on the surface where all normal curvatures are equal (end of an egg)

Robust Interpolation

- Select the n depth measurements closest to the grid point
- Fit a surface patch to all possible combinations of m data points selected from the n points. The number of subsets will be:

$$k = \binom{n}{m}$$

- Compute the median of the squared residuals for this patch:

$$\chi_k^2 = \operatorname*{med}_i [(z_i - f(x_i, y_i; a_k))^2]$$

- Once all subsets have been considered, select the parameter $\boldsymbol{a_k}$ with the smallest median of squared residuals.

Robust Interpolation

- Beneficial when depth measurements have outliers.
- Uses least-median-squares regression to fit surface patches
- Tolerates up to 50% outliers.
- Finds parameters (*a*) that minimize the median of the squared residuals (difference between depth measurement and model):

$$\min_{a}\left\{\operatorname*{med}_{(x_i,y_i)\in N}\left[\left(z_i - f(x_i,y_i;a)\right)^2\right]\right\}$$

- To do this, surface patches are fit to a grid point based on the neighborhood of depth measurements.

Robust Interpolation Properties

- Computationally expensive due to the large number of subsets of points for which surface patches must be fit
- Each surface fit is independent, which allows for parallelization

Surface Patches

Algorithm – Follow the Edges Clockwise Around a face

Inputs- A pointer to the record for the face to traverse and a procedure to invoke for each edge that is visited.

1. Get the 1st edge from the face record and make it the current edge.
2. Process the current edge: perform whatever operations must be done as each edge is visited. E.g., compile a list of vertices clockwise around the face, record the vertex at the end of the edge in the direction of traversal.
3. If the west face of the current edge is being circumnavigated, then the next edge is the SW wing.
4. If the east face of the current edge is being circumnavigated, then the next edge is the NE wing.
5. If the current edge is the 1st edge, then the traversal is finished.
6. Otherwise, go to step 2 to process the new edge.

Surface Approximation

- Types:
  - Variational methods
  - Regression splines
  - Weighted spline approximation

Surface Approximation

- Fits surfaces to the data points where the surfaces do not necessarily include the data points
- Sometimes easier to approximate the data instead of forcing interpolation of points
- In general, find $z = f(x,y)$ that minimizes:

$$\chi^2 = \sum_{i=1}^{n} (z_i - f(x_i, y_i))^2$$

Variational Methods

- A good approximation scheme leads to a single, good, clear solution
- Choosing a function that approximates the data and is a smooth surface leads to a good solution choice

Variational Methods – Regularization

- Select the function $z = f(x,y)$ that minimizes the norm:

$$\chi^2 = \underbrace{\sum_{i=1}^{n} (z_i - f(x_i, y_i))^2}_{\text{Problem constraint (error)}} + \alpha^2 \underbrace{\iint \frac{\partial^2 f}{\partial x^2} + 2\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial y^2}\, dx\,dy}_{\text{Smoothness term (stabilizer)}}$$

- α is the *regularizing parameter* and defines the tradeoff between a good approximation (small) and a smooth surface (large)
- To find the solution, variational calculus and numerical methods are used

Regression in Splines

- Substitutes a generic surface representation for the approximating function and solves the regression problem for the parameters of the generic surface
- Tensor-product splines are often used for the generic surfaces:

$$\chi^2 = \sum_{i=1}^{n} (z_i - \underbrace{f(x_i, y_i)}_{\text{Substitute with tensor-product splines}})^2$$

- Tensor-product splines are composed of a linear combination of *basis functions*:

$$f(x, y; a_0, a_1, \ldots, a_m) = \sum_{i=0}^{m} a_i B_i(x, y)$$

Basis Functions

- A basis function in 2D is composed of several polynomial segments
- Cubic (4^{th}-order) basis functions have 4 segments
- Each segment is a cubic polynomial curve defined only on the segment's integer interval
- The polynomial curves are joined at locations known as *knots*

Regression and the B-Spline Curve

- Need to solve for the coefficients a_i of the basis functions so the B-spline curve can be used to solve the regression problem
- There are m + 1 coefficients, so we will need m + 1 equations that constrain the coefficients
- Select m + 1 data points to create m + 1 equations
- The data points must be chosen so that each coefficient is constrained by at least one equation
- Each equation has the form:

$$a_{i-3}b_3(x_i)+a_{i-2}b_2(x_i)+a_{i-1}b_1(x_i)+a_ib_0(x_i)=z_i$$

where z_i is the value of the data point at x_i

- For regression, minimize over different choices of coefficients:

$$\chi^2=\sum_{k=1}^{N}\left[z(k)-\left(\sum_{i=0}^{m}a_iB_i(x)\right)\right]^2$$

Extending Regression Splines to 3D

- B-spline curve becomes a B-spline surface where each basis function is a tensor product of basis functions:

$$f(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij} B_{ij}(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij} B_j(x) B_i(y)$$

- Each basis function covers 16 grid rectangles, and each rectangle is covered by 16 basis functions
- Each basis function is composed of 16 bicubic polynomial patches
- Each patch is defined over a single rectangle
- A patch surface is formed by the product of 2 cubic polynomial curves, one in *x* and one in *y*

Extending Regression Splines to 3D

- Each basis function, $B_j(x)$, is composed of the same 4 polynomials: $b_0(x)$, $b_1(x)$, $b_2(x)$, and $b_3(x)$
- Each basis function, $B_i(y)$, is composed of the same 4 polynomials: $b_0(y)$, $b_1(y)$, $b_2(y)$, and $b_3(y)$
- The formula for the B-spline surface is then:

$$\sum_{k=i-3}^{i} \sum_{l=j-3}^{j} a_{kl} b_{i-k}(x) b_{j-l}(y)$$

- As in the 2D case, only the coefficients a_{kl} depend on the grid rectangle $[i,j]$ containing (x,y)

Extending Regression Splines to 3D

- Need to solve for the coefficients a_{kl} of the basis functions so the B-spline surface can be used to solve the regression problem
- There are $(m + 1)(n + 1)$ coefficients, so we will need $(m + 1)(n + 1)$ equations that constrain the coefficients
- Select $(m + 1)(n + 1)$ data points to create $(m + 1)(n + 1)$ equations
- The data points must be chosen so that each coefficient is constrained by at least one equation
- For regression, minimize over different choices of coefficients:

$$\chi^2 = \sum_{k=1}^{N} \left[z(k) - \left(\sum_{i=0}^{n} \sum_{j=0}^{m} a_{ij} B_{ij}(x, y) \right) \right]^2$$

Regression Splines vs Patches

- Even though B-spline surfaces are composed of surface patches, they differ in some ways
- The patches in a B-spline surface are continuous even where the patches join
- This is not required of normal surface patches
- Because the B-spline surface is smooth overall, it can be used to model objects such as human organs, vehicles, and aircraft

Weighted Spline Approximation

- Previous methods produce smooth surfaces regardless of the discontinuity in the data that may be surface boundaries
- One solution is to reduce smoothing in the discontinuous areas
- *Weighted regularization* accomplishes this with a weight function that is small at discontinuities, and are large elsewhere

Weighted Spline Approximation

- Weighted spline surface approximation:

$$\chi^2 = \sum_{i=1}^{n} (z_i - f(x_i, y_i))^2 + \iint w(x, y)\left[\frac{\partial^2 f}{\partial x^2} + 2\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial y^2}\right] dx dy$$

- Weight function:

$$w(x, y) = \frac{\alpha^2}{1 + \|\rho(x, y)\|^2}$$

where $\rho(x, y)$ is the gradient of the surface

Surface Reconstruction Applications

- Fetal ultrasound images



Surface Reconstruction Applications

- Lidar data

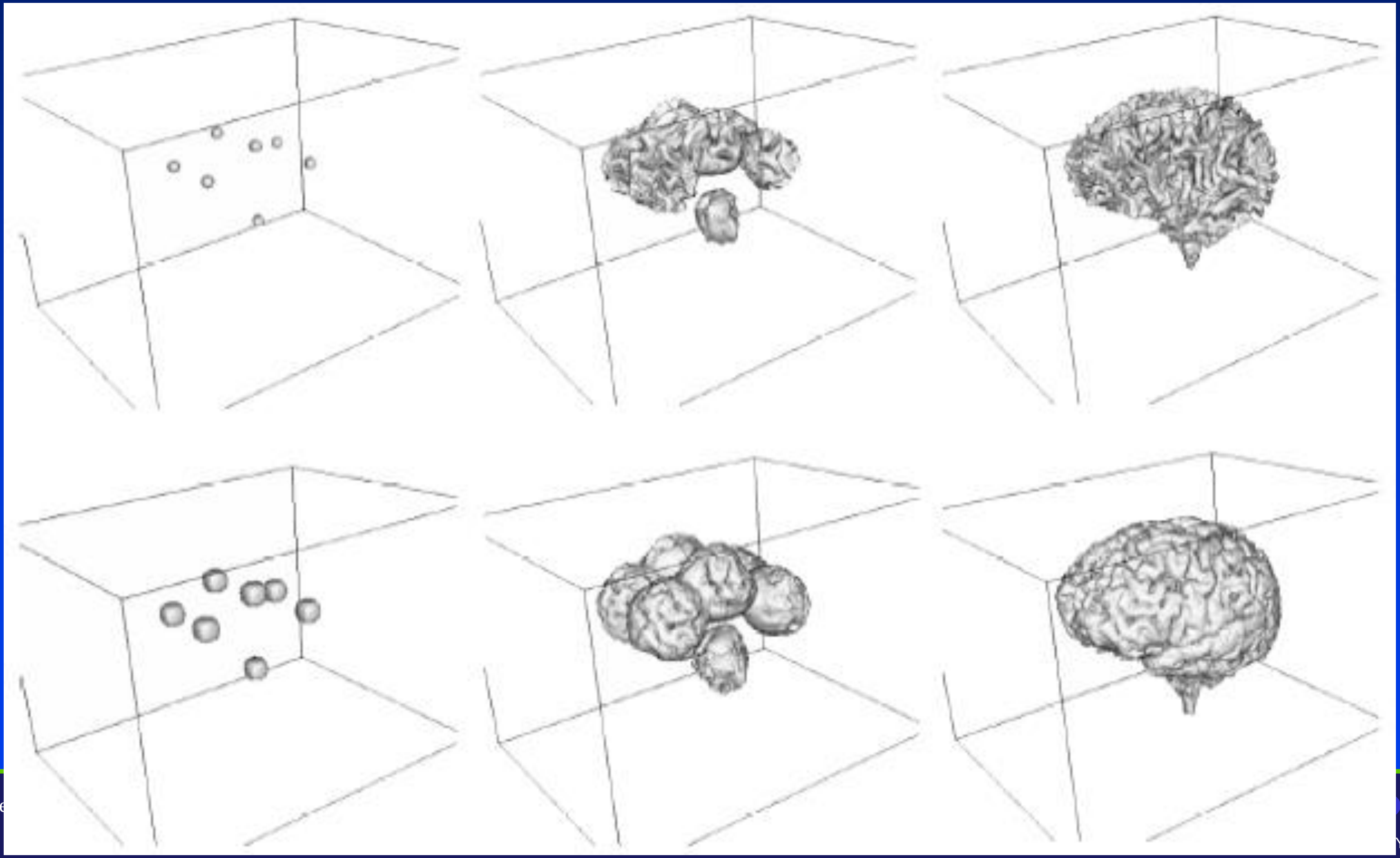
Surface Segmentation

- Segment depth measurements on a uniform grid into regions
- The results can be used for object recognition
- Each region has similar curvature and can be approximated with low-order bivariate polynomials
- One approach is the *variable-order surface segmentation algorithm*

Variable-order Surface Segmentation

- Core regions are estimated from surface curvature properties
- Regions are modeled by bivariate surface patches
- The regions are grown to cover additional measurements
- Surface patches are extended to cover neighboring measurements if the error between these measurements and the surface patch is low

Example



Variable-order Surface Segmentation

- Final surface will consist of a piecewise smooth graph surface that can be partitioned into smooth surface primitives:

$$z = f(x, y) = \sum_{l=1}^{n} f_l(x, y)\xi(x, y, l)$$

- Each l represents a region, R_l
- $\xi(x, y, l)$ represents the segmentation of the surface:

$$\xi(x, y, l) = \begin{cases} 1 & \text{if } (x, y) \in R_l \\ 0 & \text{otherwise} \end{cases}$$

- Each region is approximated by a polynomial patch:

$$f_l(x, y) = \sum_{i+j \le m} a_{ij} x^i y^j$$

Algorithm

1. Estimate the first and second partial derivatives (f_x, f_y, f_{xx}, f_{xy}, and f_{yy}) by convolving the range image with separable filters.

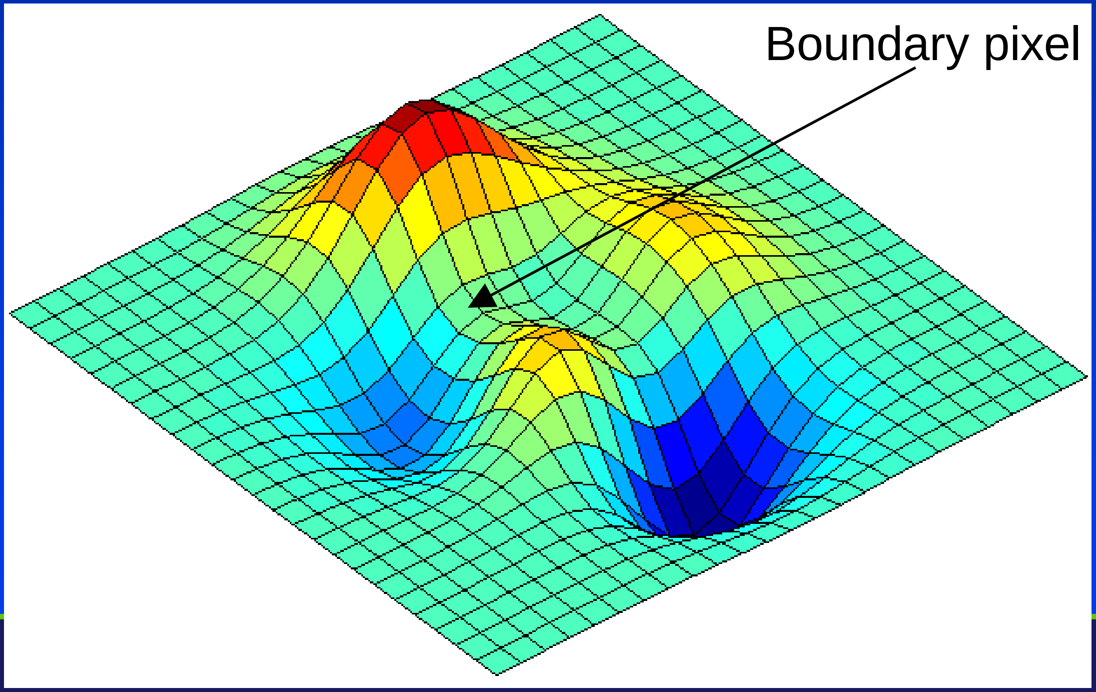
2. Use the derivatives from step 1 to compute the mean and Gaussian curvatures ($H[i,j]$ and $K[i,j]$) at each image grid location. The signs of the curvatures (+, -, or 0) determine the surface type.

3. Label each range pixel with the surface type:

$$T[i, j] = 1 + 3(1 + \operatorname{sgn}(H[i, j])) + (1 + \operatorname{sgn}(K[i, j]))$$

Algorithm

4. Shrink the labeled regions to eliminate false labels near the boundaries. It can be difficult to determine what region a pixel on a boundary belongs to when only curvature is considered.



Algorithm

5. Use the sequential connected components algorithm to find seed regions:
   a) Group range samples with identical labels into connected regions
   b) Use shrink or erosion operations to reduce the regions to include a small core of samples
   c) Discard regions smaller than a certain threshold
6. Fit a bivariate patch to each region. Start with a planar patch and increase the order of the patch until a good fit is obtained. A region should be discarded if no good fit can be achieved such that the root-mean-square error is below some threshold.

Algorithm

7. For each region, find a set of neighboring pixels with values close to the surface patch. These pixels will be considered for inclusion in the region.

8. Refit the surface patch with the pixels selected in step 7 added to the original pixels. The order of the patch may need to be increased to get a good fit. If the fitting error is below a threshold, add the pixels to the region; otherwise, discard them.

9. Repeat steps 7 and 8 until no region is changed.

Surface Registration

- Aligns range samples with an object model or another set of range samples

- Often necessary to piece together two partial sets of an object
- Two methods:
  - Iterative closest point
  - Trimmed iterative closest point

Iterative Closest Point

- Useful when the correspondence between the two sets of points is not known
- Approximates sets of conjugate pairs by mapping the closest points between the two samples
- The following distance measure is used to determine the closest points:

$$d(p, M) = \min_{q \in M} \|q - p\|$$

- This selects the point *q* in sample *M* with minimum distance to point *p* in the other sample
- The two samples are then realigned based on the closest points
- The process is repeated until the views are approximately aligned and the sum of the squared distances between closest points (registration) is below a threshold

ICP Algorithm

1. Compute the set of closest points for every point in the surface to align.
2. Compute the registration between the point sets. This is essentially the error in the alignment.
3. Apply a rigid body absolute orientation transform to register the point sets.
4. Return to step 1 if the registration error is above a tolerance threshold. Otherwise, the algorithm is finished.

Trimmed Iterative Closest Point

- When computing the registration, only consider a subset of point mappings with the least squared distances between each pair of points. These distances are the least trimmed squares (LTS).
- Provides better handling of:
  - Outliers
  - Shape defects
  - Partial overlap

TrICP Algorithm

1. Compute the set of closest points for every point in the surface to align. For each pair, compute the squared distance between them.
2. Sort the squared distances in ascending order and calculate the sum S'_{LTS} of the desired number of least distances, N_0.
3. Stop if any of the following conditions exist:
   a) The maximum number of iterations has been reached
   b) The trimmed MSE $e = S'_{LTS} / N_0$ is sufficiently small.
   c) The relative change in the MSE $\left| e - e' \right| / e$ is sufficiently small.

TrICP Algorithm

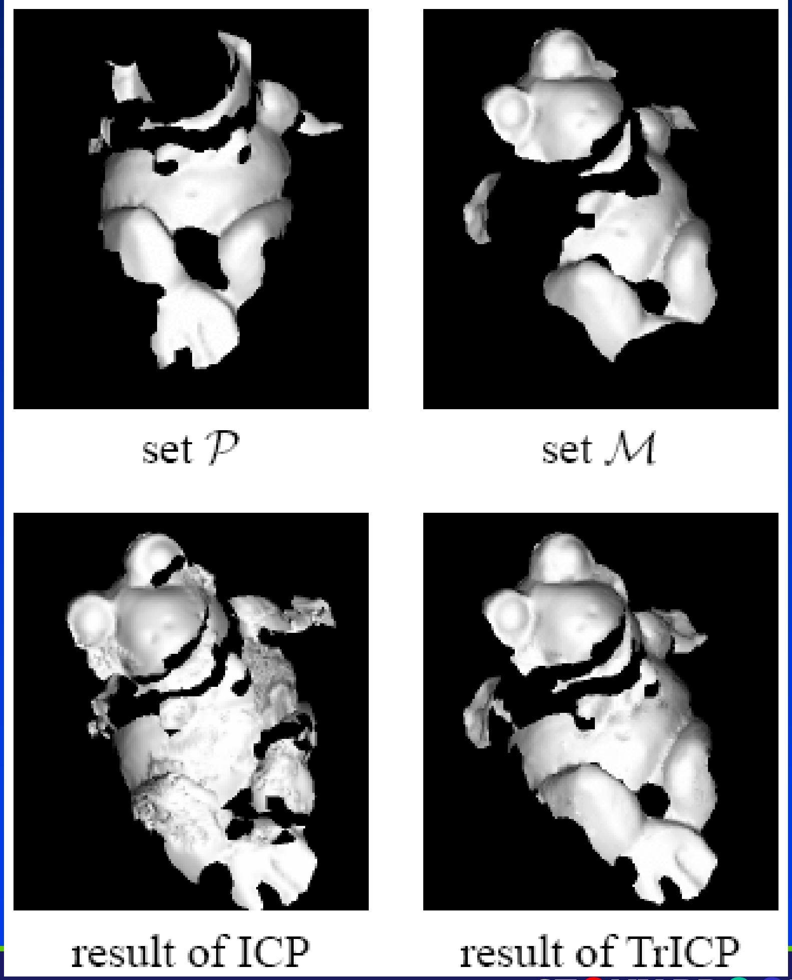
4. Compute the registration that minimizes

$$S'_{LTS}$$

5. Apply a rigid body absolute orientation transform to register the point sets and go back to step 1.

Comparison of ICP and TrICP

- Goal is to align set P with set M
- The results show that the ICP method considers all points and the TrICP method considers 70% of the points
- The TrICP method executed faster and produced less error than the standard ICP method.



Experimental Results

Method	N_{iter}	MSE	Exec.time
ICP (100%)	45	5.83	7 sec
TrICP 70%	88	0.10	2 sec

Summary

- Introduction to surfaces
- Representations
- Reconstruction
- Segmentation
- Registration

References

- Chetverikov, Dmitry, et al. "The Trimmed Iterative Closest Point Algorithm." Proceedings of the 16th International Conference on Pattern Recognition 3 (2002): 545-548.
- Goldenberg, Roman, et al. "Cortex Segmentation: A Fast Variational Geometric Approach." IEEE Transactions On Medical Imaging 21.2 (2002): 1548.
- http://ego.psych.mcgill.ca/misc/fda/files/CRM-BasisBasics.ppt.
- http://www.farfieldtechnology.com/casestudies/lidar.
- http://www.obgyn.net/us/gallery/gallery.htm.
- http://www.olympus.net/personal/mortenson/preview/definitionss/surfacepatch.html.
- Jain, Ramesh, Rangachar Kasturi, and Brian G. Schunck. Machine Vision. Boston: McGraw-Hill, 1995.
- Johnson, A.E, and M. Herbert. "Surface Registration by Matching Oriented Points." Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ont. Canada, 12-15 May 1997. 1997. 121-128.