3D Non-rigid Registration for MPU Implicit Surfaces

Tung-Ying Lee and Shang-Hong Lai
{tylee, lai}@cs.nthu.edu.tw
National Tsing Hua University, Hsinchu, Taiwan

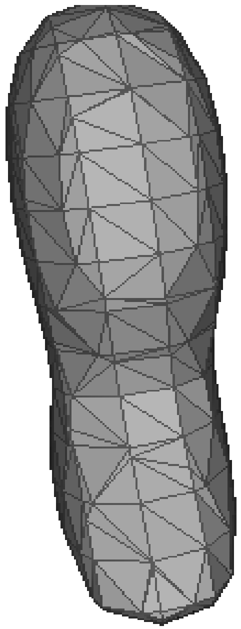
CVPR Workshop on NORDIA

Outline

- Rigid and non-rigid registration
- Implicit Surface and MPU Implicits
- Previous work
- Our proposed method
- Experimental results
- Conclusion and Future work

Rigid registration

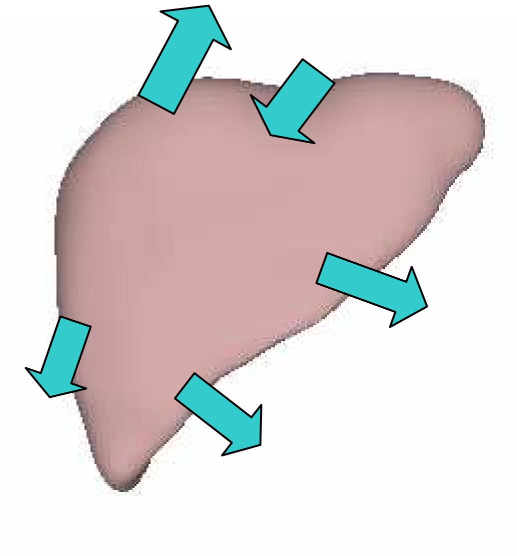
- Rigid registration
  - To find the transformation (rotation and translation) between two very similar models or different poses of the same model

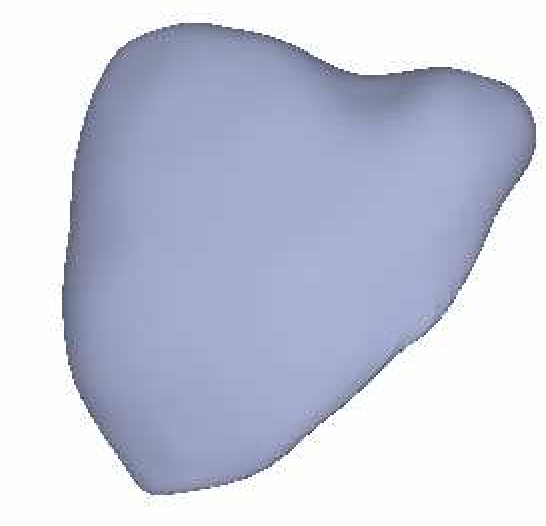


A cartoon bone

Non-rigid registration

- For two models which are deformed versions of the other
  - More degree of freedom is required to represent this deformation.

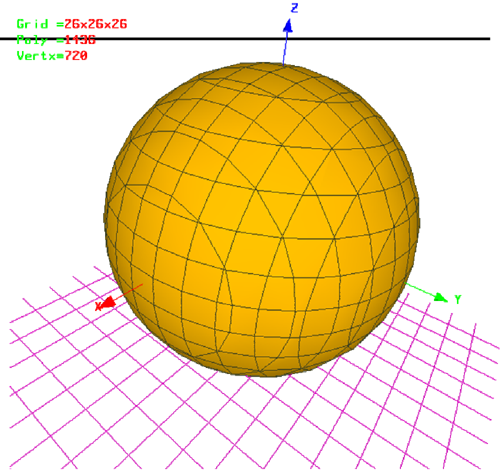




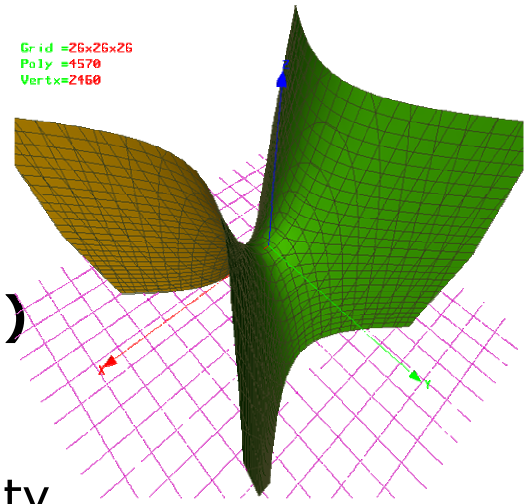
3D Object representation

- Many sources of 3D models
  - Range Scanner
  - Existing object databases
  - Mechanical parts
  - Medical images (volume data)
- Different kinds of object representations
  - Point sets/point clouds
  - Triangular meshes
  - **Implicit surfaces**
  - …

Implicit Surface



- A function F
  - A surface *S* can be defined by "F(x, y, z)=c
- For example,
  - Sphere:
    - $F(x, y, z)=x^2+y^2+z^2-4=0$
  - Hyperbolic paraboloid:
    - $F(x, y, z)=x^2-y^2-z=0$
- Many classes of implicit surfaces
  - metaball surfaces (Blobby)
  - Variational implicit surfaces
  - **Multi-level partition of unity implicits (MPU implicit surface)**

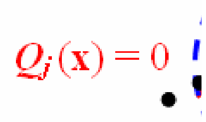


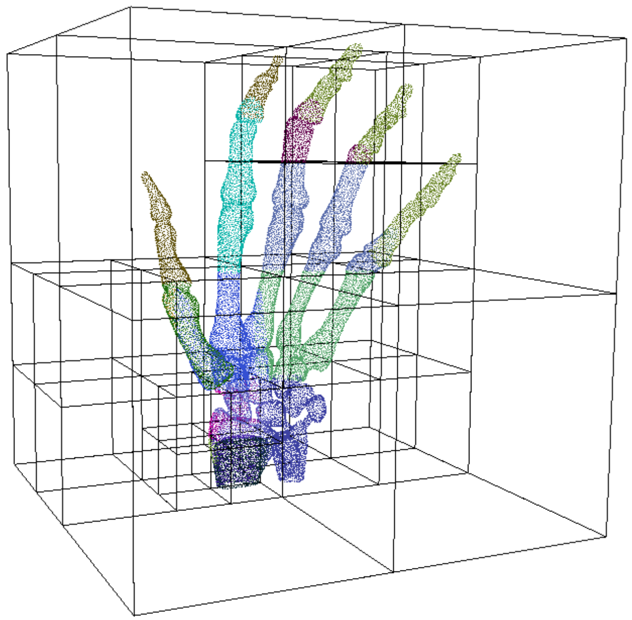
(MPU) Y. Ohtake, et al. Multi-level partition of unity implicits. In SIGGRAPH 2003.

MPU Implicits

- MPU is an error controlled mechanism of creating implicit surfaces. (Ohtake et al.)
  - Octree subdivision
  - Fitting of local shap
  - Smooth blending.

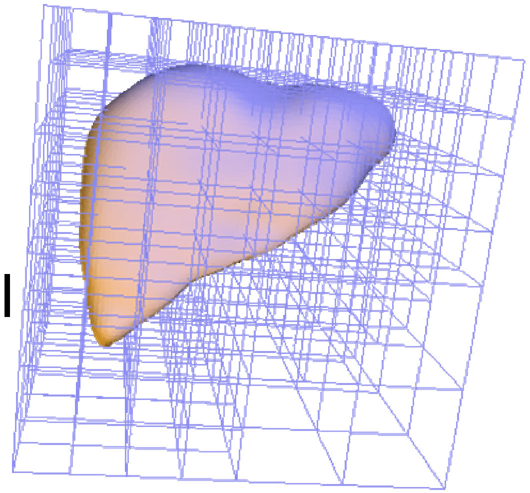
$$f(\mathbf{x}) = \frac{\sum_i \omega_i(\mathbf{x}) f_i(\mathbf{x})}{\sum_i \omega_i(\mathbf{x})}$$

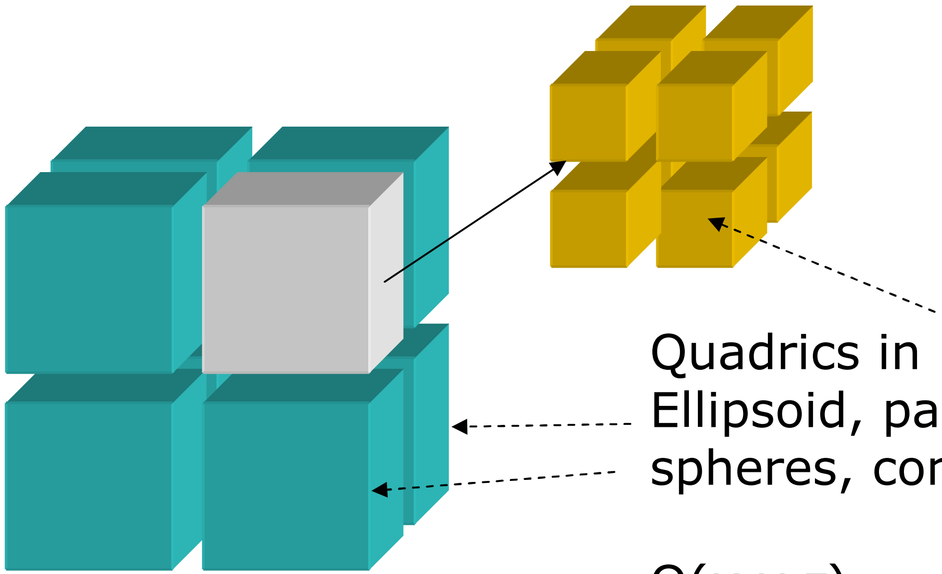




MPU Implicits

- Local shape function
  - **A quadric**
  - A quadratic polynomial
  - A piecewise quadratic polynomial





Octree

Quadrics in 3D:
Ellipsoid, paraboloid, ...
spheres, cones, cylinders, ...

$Q(x,y,z)=$
$Ax^2+By^2+Cz^2+Dxy+Eyz+Fxz+Gx+Hy+Iz+J$

Implicit surfaces and non-rigid registration

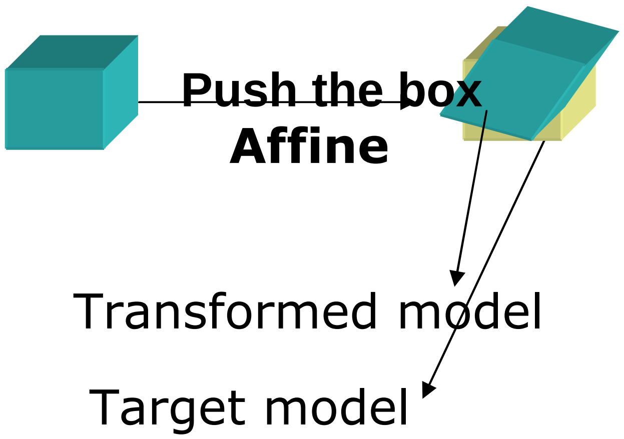
- Advantages of IS
  - It can reconstruct surface from range scans and incomplete point data.
  - The functional operations, like offsetting and twisting, can be applied to the implicit function easily.
- Additional advantages of MPU
  - Error-controlled framework
  - The value of the signed distance function at a point can be evaluated rapidly.
  - Some applications: Surface recovery [14], shape completion [15]
- Applications of Registration
  - Reconstructing a complex model from 3D point sets of different views
  - Building a statistical model requires dense correspondence.

Previous work

- Non-rigid registration on two mesh models (Allen et al. [3], Amberg et al. [4])
  - Apply an affine transformation to each point
- Non-rigid registration in implicit spaces (Paragios et al. [5], Huang et al. [6])
  - Sampling points on the space to achieve free-form deformation
  - Cannot be directly applied to implicit surface representation
- Registration and integration of variational implicit surfaces (Claes et al. [7])
  - This method only handles rigid registration

Our proposed approach

- Our approach
  - Based on MPU framework
  - Apply an affine transformation for each box
  - Combine affine transformations by partition of unity
  - Solve the affine parameters by energy-minimization
- Features
  - Non-rigid registration between two implicit surface models
  - Directly applied to implicit surfaces

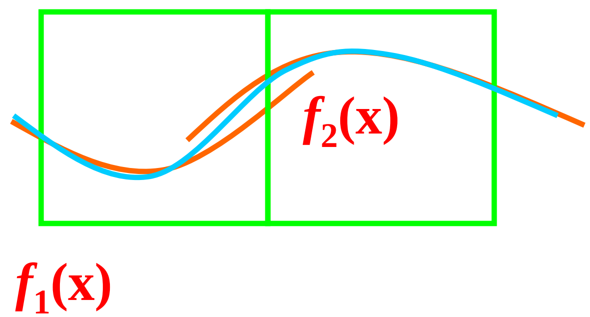


Correspondence issue in our approach

- Correspondence between boxes
  - The two models are assumed registered roughly.
  - One cell ***C*** in source implicit surface corresponds to a cell **corr(*C*)** in target implicit surface
  - Split the cell in the octree if needed

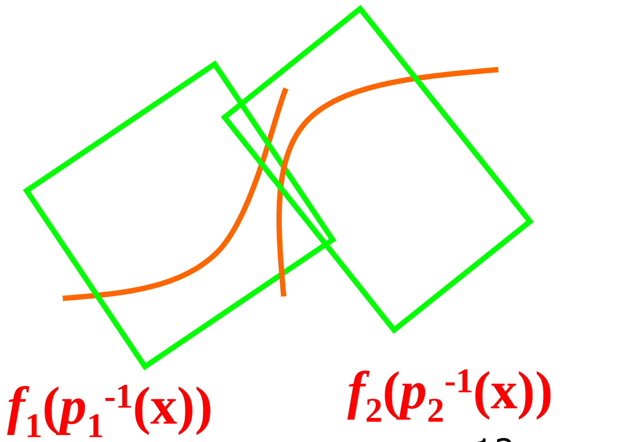
Blending affine transformations

- Continuity of transformed models
  - MPU: Blend all implicits f_i with the weighting function w_i



$$f(\mathbf{x}) = \frac{\sum_i \omega_i(\mathbf{x}) f_i(\mathbf{x})}{\sum_i \omega_i(\mathbf{x})}$$

  - Our method: Blend all transformed implicits with the corresponding affine p_i



$$doform(\mathbf{x}) = \frac{\sum_i w_i(p_i^{-1}(\mathbf{x})) f_i(p_i^{-1}(\mathbf{x}))}{\sum_i w_i(p_i^{-1}(\mathbf{x}))}$$

Energy-Minimization Formulation

- The energy consists of two parts

$$E = \alpha E_{data} + E_s$$

  - Data energy
  - Smoothness energy

α : constants

Data Energy

- Data energy
  - The difference of signed distance function
  - integration over the range of the cell in the source model

$$E_{data} = \sum_i \int \left\| Q^T_{corr(i)}\left(p(\mathbf{x})\right) - Q^S_i\left(\mathbf{x}\right) \right\|^2 d\mathbf{x}$$

Quadrics

$$Q(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

Affine transformation

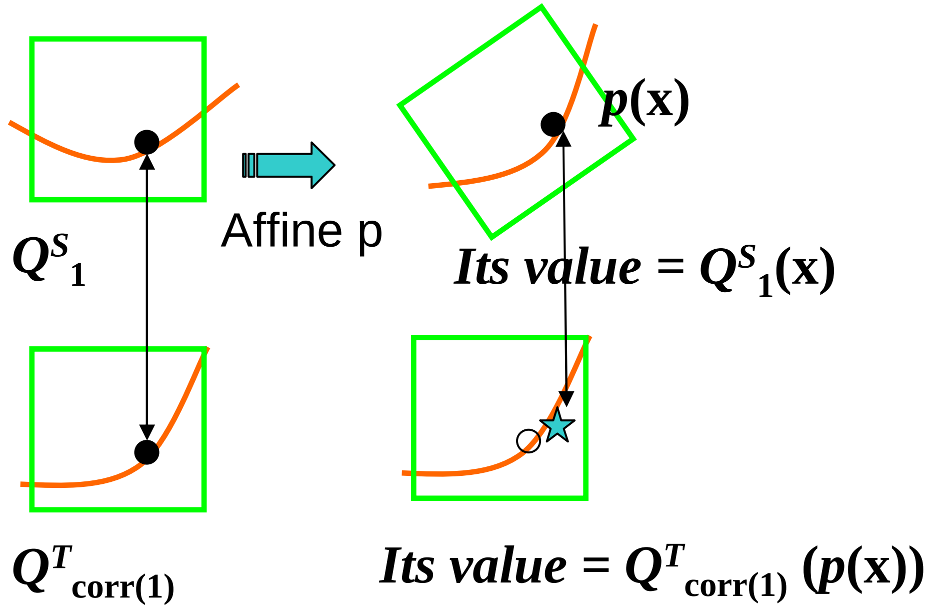
$$p(\mathbf{x}) = \mathbf{K}\mathbf{x} + \mathbf{h}$$

A is a symmetric 3x3 matrix, **b** is a 3x1 matrix, and *c* is a constant
K is a 3x3 matrix and **h** is a 3x1 vector

Data Energy (cont'd)

$$E_{data} = \sum_i \int \left\| Q^T_{corr(i)}\left(p(\mathbf{x})\right) - Q^S_i\left(\mathbf{x}\right) \right\|^2 d\mathbf{x}$$

- S: source IS
- T: target IS
- corr: correspondence between source and target IS



Point ***x*** in S → Point ***p***(***x***) in T

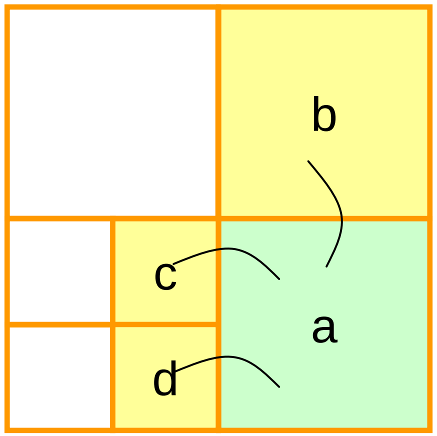
Smoothness energy

- Smoothness energy
  - Make the deformation smooth

$$E_s = \sum_{(i,j)\in AC} \gamma \left\|\mathbf{K}_i - \mathbf{K}_j\right\|_F^2 + \delta \left\|\mathbf{h}_i - \mathbf{h}_j\right\|_F^2$$

AC is the set of adjacent pairs.
(a, b), (a, c), and (a, d) are in the set AC.

The adjacent cells of a are b, c, and d.



Optimization

- Minimize the energy

$$E = \alpha E_{data} + E_s$$

  - Gradient descent method
    - The gradient w.r.t $\{\mathbf{K}_1, \mathbf{h}_1, \mathbf{K}_2, \ldots\}$ must be evaluated.
    - The gradient of this energy can be derived in a closed form, because the local shape function is a quadric.

Data energy gradient

$$E = \alpha E_{data} + E_s$$

$$\begin{cases} Diff = \left\| Q_{corr(i)}^T \left(p(\mathbf{x})\right) - Q_i^S(\mathbf{x}) \right\|^2 \\ \dfrac{\partial}{\partial K_i} E_{data} = \displaystyle\int_{Celli} \dfrac{\partial Diff}{\partial \mathbf{K}_i} d\mathbf{x} \\ \dfrac{\partial Diff}{\partial \mathbf{K}_i} = 2\left(\dfrac{\partial Q_{corr(i)}^T}{\partial \mathbf{K}_i}\right)\left(Q_{corr(i)}^T - Q_i^S(\mathbf{x})\right) \\ \dfrac{\partial Q_{corr(i)}^T}{\partial \mathbf{K}_i} = \left(2\mathbf{A}_{corr(i)}^T \mathbf{K}_i \mathbf{x}\mathbf{x}^T\right) + \left(\mathbf{A}_{corr(i)}^T \mathbf{h}_i + \mathbf{b}_{corr(i)}^T\right)\mathbf{x}^T \end{cases}$$

A, b: the parameters of quadrics

Smoothness energy gradient

$$\begin{cases} \dfrac{\partial}{\partial \mathbf{K}_i} E_s(\gamma, \delta) = \sum_{(i,j)\in AC} 2\gamma\left(\mathbf{K}_i - \mathbf{K}_j\right) \\ \dfrac{\partial}{\partial \mathbf{h}_i} E_s(\gamma, \delta) = \sum_{(i,j)\in AC} 2\delta\left(\mathbf{h}_i - \mathbf{h}_j\right) \end{cases}$$

Approximation for integration

- The partial derivatives of K_i involve integration over the *i*th cell.

$$\frac{\partial}{\partial K_i} E_{data} = \int_{Celli} \frac{\partial Diff}{\partial \mathbf{K}_i} d\mathbf{x}$$

- We can approximate the integration by discrete method, but we have the formula for the partial derivatives.

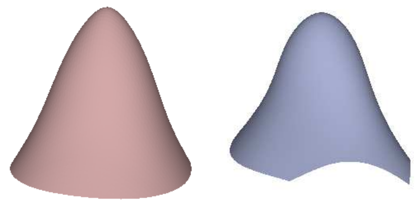
$$\frac{\partial Diff}{\partial \mathbf{K}_i} = 2\left(\frac{\partial Q^T_{corr(i)}}{\partial \mathbf{K}_i}\right)\left(Q^T_{corr(i)} - Q_i^S(\mathbf{x})\right)$$

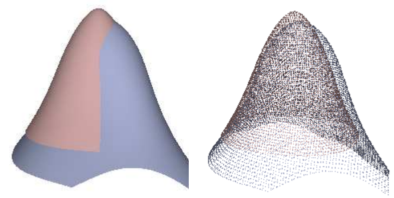
$$\frac{\partial Q^T_{corr(i)}}{\partial \mathbf{K}_i} = \left(2\mathbf{A}^T_{corr(i)}\mathbf{K}_i\mathbf{x}\mathbf{x}^T\right) + \left(\mathbf{A}^T_{corr(i)}\mathbf{h}_i + \mathbf{b}^T_{corr(i)}\right)\mathbf{x}^T$$

Experimental results (1)

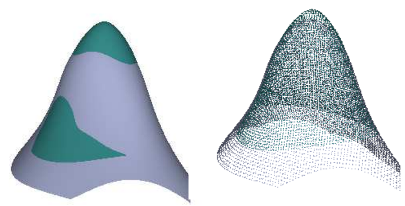
- Geometric objects
  - Open surfaces
- (Initial)
- Before registration
- (Transformed: green)
- After registration

Source & Target





Average distance error: 0.0895

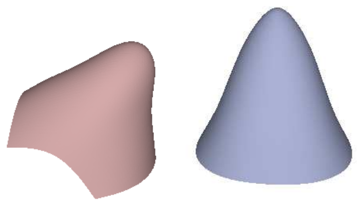


Average distance error: 0.0297

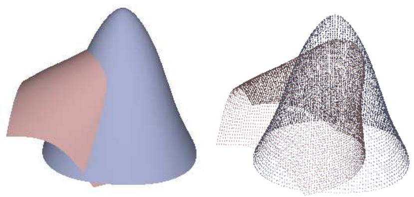
Experimental results (2)

- Geometric objects
  - Open surfaces

Source & Target

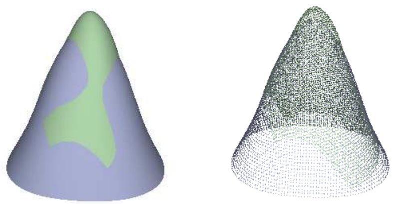


- (Initial)
- Before registration



Average distance error: 0.1398

- (Transformed: green)
- After registration

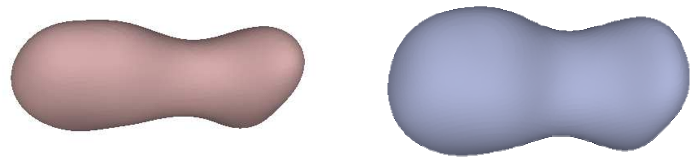


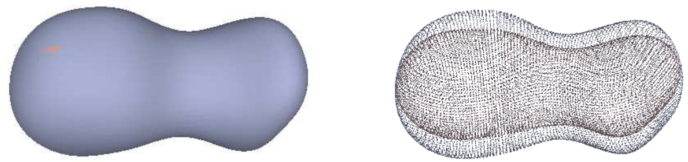
Average distance error: 0.0304

Experimental results (3)

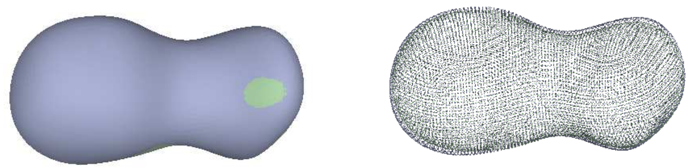
- Geometric objects
  - Closed surfaces
- (Initial)
- Before registration
- (Transformed: green)
- After registration

Source & Target





Average distance error: 0.049

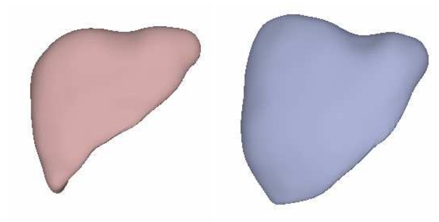


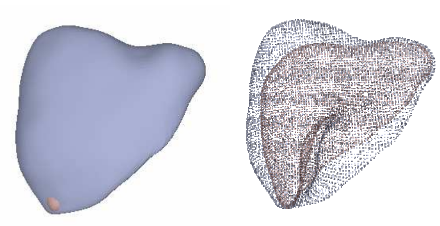
Average distance error: 0.0184

Experimental results (4)

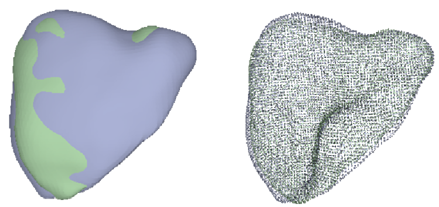
- Human organs
  - Liver
- (Initial)
- Before registration
- (Transformed: green)
- After registration

Source & Target





Average distance error: 7.3192

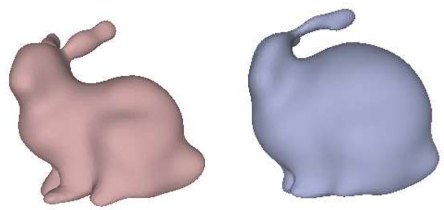


Average distance error: 1.975

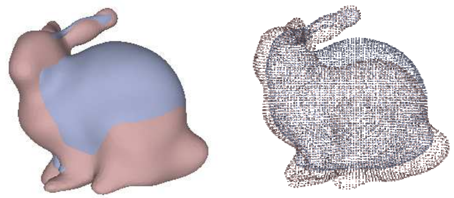
Experimental results (5)

- Sculpture models
  - Bunny

Source & Target

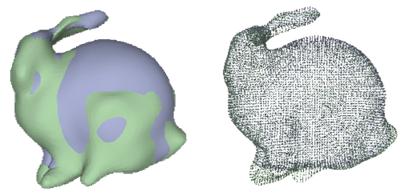


- (Initial)
- Before registration



Average distance error: 0.0769

- (Transformed: green)
- After registration



Average distance error: 0.0279

Conclusion and Future work

- A new 3D non-rigid registration algorithm for implicit surfaces is proposed.
- We do not sample points for registering two implicit surfaces.
- The continuous deformation function is determined by the proposed algorithm in an energy minimization framework.

- In the future, we would like to combine registration and generation of implicit surfaces in a unified framework.

Thank you!