3D Non-rigid registration for deformable data

Zhi-Quan Cheng
Cheng.zhiquan@gmail.com
Outlines

- Introduction
- Related work
- Non-rigid registration in 3D implicit vector space
- Non-rigid registration for large-difference data
- Conclusion
1. Introduction

Welcome to Non-rigid World
Non-rigid shapes everywhere

Volumetric medical data
Deformable shapes
Articulated shapes
1.1 Non-rigid Registration

Two registration examples:
Left: registering the open partial shape. (a) scanned data (b) alignment
Right: the closed shapes. (c) data from computer animation, (d) the sequence of transformation, (e) Correspondence

Non-rigid registration: accurate aligning the discrete data:
• mapping the points in different shapes (correspondence)
• Matching the shapes naturally (non-rigid transformation)
1.2 Correspondence

- The search space is large, $O(N!)$ for $N$ input points, assuming no outlier.
- How to measure the feature

Partial matching: The points of input data is always not equal, there exists noise and missing data.

[Partial matching: The points of input data is always not equal, there exists noise and missing data]

1.3 Transformation

- The difficulty of the transformation stems from the deformable shapes

A transformation is a family of smooth/natural deformation
- preserve detail
- preserve volume
- ......

1.3 Transformation

- Essential problem: non-rigid deformation
  - Deducing local rotations of the surface based on position displacements constraint

Inherently a non-linear problem!

Natural

Artifact!
1.3 Transformation

- The transformation solving methods
  - Non-rigid ICP (iterative closest point) algorithms
    - Spatial approximate heuristic
    - Show limitation at the non small-difference shapes
    - Show limitation at the missing data
  - As-rigid-as-possible algorithms
    - Embedded graph nodes deformation [Sumner et al. 2007]
    - Linear blending skinning [James and Twigg 2005]
    - As-rigid-as-possible deformation [Sorkine and Alexa 2007]

1.4 Non-rigid registration

- Non-rigid registration deals with the problem of finding a partial aligning for input deformable data, such that a certain optimality criterion is achieved.
  - Coherent correspondence
  - Natural transformation

\[
\arg\min_{C,T} f(C,T) \quad \text{s.t. constraints}
\]
2. Related work

- Marker-assisted algorithms
  - Provide the correspondences
  - Avoid wrong alignments greatly


Taken from [Allen et al. 2002, 2003]
2. Related work

- Template-based algorithms
  - Partial-whole mapping (correspondence)

A smooth template mesh is registered to each of the input scans using a non-linear, adaptive deformation model. Small-scale detail coefficients are estimated and integrated into the template. The final reconstruction is obtained through detail aggregation and filtering to propagate detail into occluded regions and separate salient features from noise. Taken from [Li et al. 2009].

H. Li, B. Adams, L. Guibas, and M. Pauly, “Robust single-view geometry and motion reconstruction,” in Proc. SIGGRAPH AISA, 2009, Article No.: 175.
3. Non-rigid registration in 3D implicit vector space

- 3D Implicit vector representation
- Global-to-local Algorithm
  - Framework
  - Global alignment
  - Local deformation
- Conclusion
3.1 Implicit Representation

- Curves and surfaces can be represented in implicit forms:

\[ C = \{(x, y) : f(x, y) = 0\}, f : \mathbb{R}^2 \rightarrow \mathbb{R} \]

\[ Surface: S = \{(x, y, z) : f(x, y, z) = 0\}, f : \mathbb{R}^3 \rightarrow \mathbb{R} \]
3.1 The Signed Distance Function (SDF)

\[ f(X) = \begin{cases} 
+D(X) & \text{if inside} \\
0 & \text{on the boundary} \\
-D(X) & \text{if outside}
\end{cases} \]

\(D\) is the \textit{min} Euclidean distance between \(X\) and the contour.

\(f\) is continuous and differentiable around the zero level.
3.1 The Vector Distance Function (VDF)

- For the curve shown, we can define the function:

\[ f_{VDF}(X) : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \]

\[ f_{VDF}^{-1}(X) = X - X_0 \]

- \( X_0 \) is the closest point to \( X \).
- It can represent open/closed boundaries.

- An example for a planar circle is shown below:

3.1 The VDF and SDF Relationship

- They have same magnitude.
- The VDF have a direction equal to the gradient of the SDF for the closed shape.
- The following relation holds

\[ f_{VDF} = f_{SDF} \nabla f_{SDF} \]

where \( \| \nabla f_{SDF} \| = 1 \)

- The main advantage of VDF: no sign problem of SDF.

3 Shape Registration in 2D implicit space

- SDF has been widely used in the 2D registration
  - Many papers (hundreds!)
- VDF was firstly deduced and applied in the 2D registration by Abd EL Munim and Farag

3.2 3D implicit registration

- The extension from 2D to 3D is nontrivial, since transformations are more complex in 3D.
  - For example, 3D rotations do not commute and cannot be linearized as in the 2D case.
- Show how to use the VDF to register both the open partial and closed shapes.
- The basic framework of our algorithm
  - The global alignment for the static and deformable shapes
  - The local registration using IFFD space deformation model.
3.2 Algorithm framework

3D Vector distance function representation

Global alignment

Local non-rigid matching using IFFD

国防科学技术大学可视计算研究组
NUDT Visual Computing Team

2010-03-21
3.2.1 Global alignment

- Given two shapes \( S \) and \( T \) represented by the vector distance function \( f_S \) and \( f_T \).

- A transformation \( A \) with parameters \( \Phi \) (including rotation \( T \) and translation \( T \)) is to be calculated to transform one point \( x \) in \( S \) to its corresponding point on \( T \): \( A(\Phi, x) = Rx + T \).

- To any pair of points \( x, p_S(x) \in \Omega_S \) results in a corresponding pair of points \( x', p_S(x') \in \Omega_T \)

\[
f_T(A(\Phi, x)) = x' - p_T(x') \\
= Rx + T - (R p_S(x) + T) \\
= R(x - p_S(x)) \\
= R f_S(x).
\]
	rigid-invariant under global alignment
3.2.1 Global alignment

- The dissimilarity function:
  \[ r_d(\Phi; x) = Rf_S(x) - f_T(A(\Phi; x)) \]

- By using all points of a sampling grid in the vector space, a global Sum-of-Square-Difference (SSD) energy functional:
  \[ E_q(\Phi) = \int_{\Omega_S} r_d^T r_d \, dx \]

- The global alignment problem is converted into a nonlinear optimization to minimize the SSD.
  - Could find the main rigid motion between deformable shapes since VDFs downgrades the importance of local deformations.
3.2.1 Global alignment

- **Non-linear levenberg-marquardt optimization to the SSD energy**
  - To solve the non-linear rotation, we first relax the problem to find an optimal linear transformation, and then extract the rotation part from it.
  - Use the quaternion to express the rotation.
  - Use polar decomposition to extract pure rotation

---

**Algorithm 1** Nonlinear optimization of global VDF alignment

```
repeat
    initialize \( \Phi \)
    while (energy decreases AND not too many iterations)
        use \( J_r \) to update estimate of \( \Phi \)
        obtain rotation \( R \) from \( q \)
        transform \( S \) by \( (R, T) \)
    until (energy is low enough OR too many iterations)
```
3.2.1 global alignment

Static (left) and dynamic (right) model. (a) input data, (b) ICP variant algorithm [Rusinkiewicz and Leyov 2001], (c) our VDF method

3.2.1 Global alignment

- **Convergence testing using ground truth**
  - Random generate 40 pair static bunny and walker, and compare the final result to the ground truth, the relative error <3%

- **Robust to the large-difference for the deformable shapes, works well.**

- **Comparision**

<table>
<thead>
<tr>
<th></th>
<th>ICP variant</th>
<th>global VDF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSD</td>
<td>Time</td>
</tr>
<tr>
<td>bunny</td>
<td>0.0093</td>
<td>1.2s</td>
</tr>
<tr>
<td>rigid walker</td>
<td>0.0162</td>
<td>0.4s</td>
</tr>
<tr>
<td>tetrahedron</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>open walker</td>
<td>185.9</td>
<td>3.3s</td>
</tr>
<tr>
<td>closed walker</td>
<td>40.46</td>
<td>1.4s</td>
</tr>
</tbody>
</table>

The time and accuracy is determined by the sampling grid resolution. The higher resolution, the accurate result, it also means the high time costs.

500 sampling points VS 70*70*70 sampling grid
3.2.2 Local non-rigid registration

- **FFD deformation: 2D illustration**

  - Lattice resolution is M X N.
  - P=P_{m,n} is the control point where m={1,...,M} and n={1,...,N}.
  - The deformation field is represented by the deformation of the control lattice:
    \[ \Theta(X) = \sum_{i=0}^{3} \sum_{m=0}^{3} B_i(u) B_m(v) \partial P_{i+j,m} \]
  - \((i,j)\) is the index of the nearest node to \(X\).
  - \((u,v)\) is the relative displacement field from that node.
3.2.2 Local non-rigid registration

- IFFD parameters $\Theta = \delta P$ are the translations of the control lattice points in each of the $x, y, z$ directions.
- IFFD deformation: $P = P^0 + \Theta$
- As the control lattice deforms from $P^0$ to $P$, the deformed position $x$ in the embedding space is defined by a tensor product of cubic B-splines

$$L(\Theta; x) = \sum_{a=0}^{3} \sum_{b=0}^{3} \sum_{c=0}^{3} B_a (u) B_b (v) B_c (w) (P^0_{i+a, j+b, c+k} + \delta P_{i+a, j+b, c+k})$$

$$= \sum_{a=0}^{3} \sum_{b=0}^{3} \sum_{c=0}^{3} B_a (u) B_b (v) B_c (w) P^0_{i+a, j+b, c+k} + \sum_{a=0}^{3} \sum_{b=0}^{3} \sum_{c=0}^{3} B_a (u) B_b (v) B_c (w) \delta P_{i+a, j+b, c+k}$$

$$= x + \sum_{a=0}^{3} \sum_{b=0}^{3} \sum_{c=0}^{3} B_a (u) B_b (v) B_c (w) \delta P_{i+a, j+b, c+k}$$

$$= x + \delta L(\Theta; x)$$
3.2.2 Local non-rigid registration

- Non-rigid registration is achieved by incrementally evolving $P$ in such a way that the deformation parameters $\Theta$ minimize the non-rigid dissimilarity for $x$, now measured using
  \[
  r_n(\Theta; x) = f_S(x) - f_T(L(\Theta; x))
  \]

- Final energy function

\[
E_{Local}(\Theta) = \int_{\Omega} r_n^T r_n d\Omega + \alpha \int_{\Omega} \left( \left\| \frac{\partial \delta L(\Theta; x)}{\partial x} \right\|^2 + \left\| \frac{\partial \delta L(\Theta; x)}{\partial y} \right\|^2 + \left\| \frac{\partial \delta L(\Theta; x)}{\partial z} \right\|^2 \right) d\Omega
\]

- SSD Smoothness constraint
3.2.2 Local non-rigid registration

- The calculus of variations and a gradient descent method

\[ \frac{\partial}{\partial \Theta_i} E_{\text{Local}}(\Theta) = 0 \]

- If incremental step of each control point \( x \) was not large, then we can use the following Taylor series expansion to approximate the vector representation:

\[ f_T = f_T(x + \delta L) \approx f_T(x) + (\nabla f_T(x))^T \delta L \]

- .......

- Final closed-form solution for the \( P=P^0+ \Theta \)
3.2.2 Local non-rigid registration

- Final closed-form solution for the \( P=P^0+\Theta \)

\[
K = QP \\
\text{where} \\
K_{row} = \int_{\Omega_S} f^T(x)(\nabla f_T(x))^T \frac{\partial \delta L}{\partial \theta_i} \, dx, \\
Q_{row, col} = \int_{\Omega_S} ((\nabla f_T(x))^T \delta L')^T (\nabla f_T(x))^T \frac{\partial \delta L}{\partial \theta_i} \, dx \\
+ \alpha \int_{\Omega_S} ((\frac{\partial \delta L'}{\partial x})^T \frac{\partial \delta L}{\partial x})(\frac{\partial \delta L}{\partial x}) \\
+ ((\frac{\partial \delta L'}{\partial y})^T \frac{\partial \delta L}{\partial y})(\frac{\partial \delta L}{\partial y}) \\
+ ((\frac{\partial \delta L'}{\partial z})^T \frac{\partial \delta L}{\partial z})(\frac{\partial \delta L}{\partial z}) \, dx.
\]
3.2.2. Local non-rigid registration

- Q is a symmetric indefinite matrix with ill-conditioned number, generally larger than $10^{36}$

- Solving:
  - LU decomposition with Pivoting
  - SVD decomposition
  - Jacobi Iteration
  - Improved Jacobi Iteration
3.2.2 Local non-rigid registration

Global-to-local non-rigid registration for closed dancer models in front view (top row) and side view (bottom row). (a) Initial poses (source: blue, target: brown). (b) Global alignment using SSD minimization, showing aligned shapes. (c) Iterations of local non-rigid registration. (d) Final non-rigid registration using IFFD; the transformed source shape (green) is shown overlaid on the target shape (brown). (e) Dense correspondences produced by global-to-local registration (221 corresponding points).
3.2.2 Local non-rigid registration

haoTorso: (a) original position, (b) results from two viewpoints.

noise arm, (a) Original position, (b) global alignment, (c) final non-rigid registration.
3.2.2 Local non-rigid registration

➢ Results

<table>
<thead>
<tr>
<th>dataset</th>
<th>elephant</th>
<th>dancer</th>
<th>arm</th>
</tr>
</thead>
<tbody>
<tr>
<td># points</td>
<td>24k</td>
<td>10k</td>
<td>15k</td>
</tr>
<tr>
<td># sampling resolution</td>
<td>70</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td># band size</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td># IFFD resolution</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td># iteration</td>
<td>8</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>global alignment</td>
<td>7.4s</td>
<td>9.2s</td>
<td>8.9s</td>
</tr>
<tr>
<td>local registration</td>
<td>16.4s</td>
<td>17.5s</td>
<td>13.9s</td>
</tr>
</tbody>
</table>

For the haoTorso and elephant objects, our approach spends less than 17s to finish the non-rigid registration, while [Li et al. 2008] takes over one minute to obtain similar registration quality.

3.3 Conclusions

- A global-to-local registration approach for moving and deforming shapes.
  - Do not require any user specified markers, or prior templates;
  - Achieves accurate rigid and non-rigid registrations even in the presence of modest transformations.
  - Work well both for open partial and closed objects, even for imperfect noisy data with gaps.
3.3 Conclusions

**Limitation**

- Relying on linearizing the non-rigid deformation energy functional by the first Taylor expansion.
- Using the feature constraints to solve the artifacts.

\[
E_f = \sum_{i=1}^{N_f} r_i^T r_f, \quad r_f = x_{s_i} - L(\Theta; x_{T_i})
\]

\[x_{s_i}\text{ and } x_{T_i}\text{ is one corresponding feature pair}\]

One walker (source: blue, target red) with (1) and without (2) feature constraints marked by black spheres. (a) Original poses, (b) final results.
4. Non-rigid registration for large-difference data

- Basic idea
- Algorithm
  - Salient feature correspondences
  - Global optimization
- Conclusion
4.1 Basic idea

- Aim to develop a **robust** pairwise non-rigid registration for the **complex motion data**:
  - Large-scale deformation
  - Small-scale time-varying shape details
  - Close/Open noisy models
4.1 Basic idea

Explicit feature correspondence + Global optimization

Robust non-rigid registration
4.2 Algorithm overview

- Large-difference motion data
- Preprocessing:
  - Keypoint detection
  - Feature matching
  - Initial rigid alignment
- Global optimization:
  - Deformation error
  - Correspondence evaluation
- Non-rigid registered surfaces

Keypoint detection & feature matching

Initial rigid alignment

Global optimization
4.2.1 Salient feature correspondence

- **Slippage keypoints detection** [Bokeloh et al. 2008]

- **Spectral-RANSAC feature matching**
  - First sample some accurate correspondences extracted by spectral matching method [Leordeanu et al. 2008],
  - Then many correspondences are progressively detected by the RANSAC [Fischler et al. 1981] technique

---


4.2.1 Salient feature correspondence

- **Spectral-RANSAC feature matching**
  
  - For the spectral matching algorithm, although it could prune a set of correspondences in the fast way, only a few correspondences (always less than twenty) are found.
  
  - The RANSAC could obtain many correct correspondences, however, the randomly sampling is time-costing.

---

**Input:** consistency matrix $C=\{c_{ij}\}$

**Output:** a set of correct correspondences $K$

```plaintext
K := nil
consensus_set := nil
H := nil
extract the $N$ highest-score correspondences
add them to $H$
for each 3-element combination \{i, j, k\} of H do
  consensus_set := \{i, j, k\}
  remove the scores of i,j,k from spectral_scores
  while spectral_scores is not empty do
    corr := the correspondence with the highest score
    if corr is consistent with all elements in $K$ then
      add corr to consensus_set
    end if
    remove the score of corr from spectral_scores
  end while
  if consensus_set.size() > $K$.size() then
    $K := \text{consensus_set}$
  end if
end for
return $K$
```
4.2.1 Salient feature correspondence

- **Briefly analyze the time complexity of relevant matching algorithms.**
  - For the RANSAC [Fischler et al. 1981], if \( p(N) \) denotes the probability of 3-element matching has been correctly detected from \( N \), the expected probability \( pt \) after \( M \) sampling can be expressed as \( M = \ln(1-pt)/\ln(1-p(N)) \). The formal expression of \( p(N) \) is \( 1/\binom{N}{3} \).
  - For the RANSAC-like algorithm [Tevs et al. 2009], \( N \) could be reduced as 0.2\( N \).
  - For our spectral-RANSAC algorithm, \( N \) is constantly 20 set as the output of spectral algorithm.

- **Therefore,**
  - if \( pt = 0.95 \) and keypoints number \( N = 1000 \), then sampling number for the RANSAC-like [Tevs et al. 2009] and our spectral-RANSAC algorithm is 3, 934, 593 and 3, 413 respectively. In general, the number of slippage keypoints is far more than 1000 for one model with 10K vertices, this means that our algorithm is always 1,000 times faster than the RANSAC-like.

4.2.1 Salient feature correspondence

- **Results**
  - A performer (top) with 23 correspondences,
  - An elephant (middle and bottom) with 234 correspondences.

- It’s stable for the large difference deformable data.

- Time cost is about 20 seconds, while [Tevs et al. 2009] spent over 20 hours for similar data.

4.2.2 Non-linear optimization

- The embedded graph nodes deformation model [Sumner et al. 2007] is used.
- The correspondence terms are similar to [Li et al. 2008]
- The final energy function are defined as

\[
E_{total} = \alpha_{rigid}E_{rigid} + \alpha_{smooth}E_{smooth} + \alpha_{ini}E_{ini} + \alpha_{ICP}E_{ICP} + \alpha_{conf}E_{conf}
\]

(deformation terms)

(correspondence terms)


4.2.2 Deformation Terms

- **Pure rotation:**
  
  \[ E_{\text{rigid}} = \sum_i \text{Rot}(R_i) \]

  - each of its three columns must be unit length, and all columns must be orthogonal to one another

- **Smooth regulation:**
  
  - Neighboring nodes should agree on where they transform each other

\[ E_{\text{reg}} = \sum_{j=1}^{m} \sum_{k \in N(j)} \alpha_{jk} \left\| R_j (g_k - g_j) + g_j + t_j - (g_k + t_k) \right\|_2^2 \]

  - where node \( j \) thinks node \( k \) should go
  - where node \( k \) actually goes

Taken from [Sumner et al. 2007]
4.2.2 Correspondence Terms

- The correspondence:
  - Initial feature correspondence
  - Dense ICP correspondence

- The confidence of the correspondence

\[ E_{conf} = \sum_{i=1}^{k} ||1 - w_i||_2^2 \]

- Our final energy function: Levenberg-Marquardt method

\[ E_{total} = \alpha_{rigid} E_{rigid} + \alpha_{smooth} E_{smooth} + \alpha_{ini} E_{ini} + \alpha_{ICP} E_{ICP} + \alpha_{conf} E_{conf} \]

(deformation terms)

(correspondence terms)
4.2 Results

- The correspondences and non-rigid registration of a guy. Top row correspondences show the similar visual comparison with [Tevs et al. 2009] (Left two snaps), the black rectangle illustrates the correspondences with localization unaccuracy. Bottom row demonstrates the registering steps: original positions, initial rigid alignment, and final non-rigid registration result.

4.2 Results

- The explicit correspondences and nonrigid registration of a cat: Our correspondences are shown on the top row. The wrong matching corresponding points on the crura marked by orange and yellow spheres. (Middle row) Results of [Lipman and Funkhouser 2009] for a similar cat. Note their algorithm also generates uncorrect correspondences (points in green). (Bottom row) Final non-registration result of cat. Wrong correspondences could be automatically detect and remove by our optimization method.

4.2 Results

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Size(S/T)</th>
<th>#Cand.</th>
<th>kp. Detect</th>
<th>Geodesic</th>
<th>ft. Match</th>
<th># Corr.</th>
<th># GraphNode</th>
<th># Iterations</th>
<th>Registr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>performer(Fig.1)</td>
<td>10k/10k</td>
<td>474</td>
<td>4.3s</td>
<td>23.1s</td>
<td>21.4s</td>
<td>147</td>
<td>265</td>
<td>14</td>
<td>15.2s</td>
</tr>
<tr>
<td>guy(Fig.4)</td>
<td>10k/10k</td>
<td>575</td>
<td>4.1s</td>
<td>25.4s</td>
<td>32.9s</td>
<td>197</td>
<td>322</td>
<td>14</td>
<td>19.4s</td>
</tr>
<tr>
<td>cat(Fig.5)</td>
<td>7k/7k</td>
<td>324</td>
<td>5.1s</td>
<td>31.9s</td>
<td>6.2s</td>
<td>133</td>
<td>393</td>
<td>12</td>
<td>18.0s</td>
</tr>
<tr>
<td>elephant(Fig.6)</td>
<td>24k/24k</td>
<td>209</td>
<td>13.2s</td>
<td>17.6s</td>
<td>1.9s</td>
<td>90</td>
<td>237</td>
<td>37</td>
<td>35.8s</td>
</tr>
<tr>
<td>haoTorso(Fig.6)</td>
<td>135k/135k</td>
<td>780</td>
<td>180s</td>
<td>347s</td>
<td>112s</td>
<td>37</td>
<td>505</td>
<td>21</td>
<td>36.9s</td>
</tr>
</tbody>
</table>

- Numberical resuslt: test data set, data size, number of candidate correspondeces, keypoint detection time, geodesic distance computation time, feature matching time, number of final correspondences, number of graphnodes, number of iterations, registration time. The machine is a 2.4GHz Intel Core 2 Quad PC with 2GB of RAM.
4.3 Conclusion

- A complete and practical non-rigid registration algorithm:
  - that automatically registers complex and large-difference 3D objects with smallscale shape details, e.g., a performer wearing loose-fitting apparel;
  - that draws its strength from a spectral-RANSAC feature matching method, which could both achieve the effectiveness of RANSAC and the efficiency of spectral matching algorithm, and an iterative global deformable optimization method.
5 Conclusion

- **Non-rigid registration.**
  - The prime technique to reconstruct one dynamic model.
  - The basic technique to edit one dynamic model.

- The problem of non-rigid registration is ill-posed and no algorithm is applicable to all scenarios.

- How to conceive novel algorithms
  - Correspondence
  - Non-rigid transformation.
5. Conclusion

➢ Trends:

✓ Minimal prior information:
  • No markers, no template, no manual intervention

✓ Accuracy:
  • The details are almost neglected

✓ Robustness:
  • The input data almost has no noise
  • Large difference
  • Large missing data

✓ Time performance
  • STAR algorithm spends about 30 seconds per registration

✓ Streaming processing
  • Gigabit level input data.
Many thanks