An Automatic Assembly and Completion Framework for Fragmented Skulls

Zhao Yin\textsuperscript{1,2}, Li Wei\textsuperscript{1}
\textsuperscript{1}Department of Automation
Xiamen University
\textsuperscript{2}Department of Geography & Anthropology
LSU FACES Laboratory
Louisiana State University

Mary Manhein
Department of Geography & Anthropology
LSU FACES Laboratory
Louisiana State University

Xin Li\textsuperscript{*}
\textsuperscript{2}Dept. Electrical & Computer Engineering
Center for Computation and Technology
Louisiana State University
Email: xinli@lsu.edu

Abstract

We develop a completion pipeline for fragmented and damaged skulls. The goal of this work is to convert scanned incomplete skull fragments to a complete skull model for subsequent forensic or archeological tasks such as facial reconstruction. The proposed assembly and completion algorithms can also be used to repair other fragmented objects with inherent symmetry. A two-step assembly framework is proposed: (1) rough assembly by an ICP-like template matching algorithm integrated with the slippage features and spin-image descriptors; (2) assembly refinement by a global optimization on least square transformation error (LSTE) of break-curves. The assembled skull is finally repaired by a symmetry-based completion algorithm. Experiments on repairing scanned skull fragments demonstrate the efficacy and robustness of this framework.

1. Introduction

Geometric data completion is an important problem in 3D computer vision. In this project, we focus on developing an effective framework to complete damaged and even fragmented skulls. Skulls provide precious information in anthropological, archeological, and forensic research and applications. For example, forensic anthropologists reconstruct faces from skulls to reveal their appearances \cite{16}. The subject skulls from museums are precious and should not be contaminated. Therefore by 3D scanning, these skulls can be represented in computer and analyzed/processed in the digital environment.

The state-of-the-art manual pipeline to construct a face from a skull consists of the following phases: (1) the collection, cleaning, and preparation of a skull; (2) repairing the damaged skull; and (3) the facial tissue/skin clay modeling procedure. We try to migrate the entire pipeline into the digital environment. The main challenges in our skull completion pipeline are two-folded. (1) Skulls are sometimes fragmented and regions with rich geometric information could be fragile. (2) The repaired skull is expected to possess high conformity especially in the frontal facial region with subtle substrate.

The main contributions of this paper are as follows, and the entire pipeline is illustrated in Fig. 1.

- We present an efficient assembling algorithm, especially suitable for fragmented skulls. It involves two components: the initial rough assembling using template matching; and assembly refinement process based on break-curve matching.
- We develop a reliable data completion algorithm. Missing regions will be repaired using inherent symmetry of the skull when possible; if damages exist on both sides with respect to the symmetric plane, a template-based repair is performed to ensure the robust completion.

2. Background Review

2.1. Skull Assembly

Excavated skulls are sometimes fragmented or damaged due to environmental or animal activities. A reliable skull assembly algorithm is highly desirable for subsequent completion, analysis, and modeling.

Cooper et al. \cite{8} proposed a framework of assembling 3D pot fragments based on 3D measurement and matching of break-curves and sherd normals. This approach was later improved by Willis et al. \cite{23}, who used Bayesian analysis in pots assembly based on semi-automatic matching. Their experiments demonstrate curve matching applies very nicely on pottery assembly. Curves can also be matched using optimization algorithms \cite{21}, which also minimizes least-square estimation of point patterns.

We propose a curve matching optimization pipeline for skulls, with reduced searching space, compared with \cite{23}. Our modification is based on the difference between pottery assembly and skull assembly: (1) The skull geometry is different from the pottery. Pottery curve matching algorithms \cite{8, 23} handle fragmented pots that are axially symmetric and the revolving curves are relatively smooth and simple. In contrast, skulls are not axially symmetric and they have...
subtle facial geometry. On the other hand, unlike potters who may have various unpredictable shapes, a template skull (from similar category of races and sexes) can be obtained to assist the assembly. (2) Pottery assembly problems considered in [8, 23] may need to handle multiple objects at the same time. Only one skull is considered in our work. Therefore, we use a template skull to perform a rough assembly which significantly reduces the searching space of curve matching. A reliable registration between the fragments and template is important. 3D registration between a shape $S$ and a template $T$ has been widely studied in vision literature. While we only need a nice fitting to the template of each fragment under a rigid transformation, the Iterative Closest Point (ICP) algorithm [5] is one of the most popular approaches. The basic concept of ICP is to find corresponding closest points on $T$ for points on $S$, and then to estimate the locally optimal transformations that minimizes these distance errors. Many effective variants of ICP [1, 2, 7] have been proposed to find rigid or non-rigid correspondence between shapes. A main disadvantage of ICP is that it is sensitive to the initial spatial positions of $S$ and $T$. Therefore, an initial coarse alignment usually benefits. Johnson et al. [12] designed the Spin Image as a local shape descriptor that can help identify corresponding regions with similar geometry.

### 2.2. Skull Completion

Extensive study has been conducted on scanned data completion, especially for the filling of holes caused by scanning occlusions. A thorough review on data completion is beyond the scope of this work. Most completion algorithms are based on the geometry near the holes (e.g. by explicit [15] or implicit [13] fairing). Many of such approaches tend to produce smooth repair, possibly losing geometry subtlety of subject skulls, and are therefore not suitable for the purpose of accurate repair. Being used by forensic anthropologists, the state-of-the-art manual skull completion uses the skull symmetry to recover missing regions. We adopt this idea and thus will briefly review symmetry detection literature. If the skull is severely damaged, e.g., both sides are missing, and no suitable region from the model itself can provide credible repair, then to transplant a corresponding patch from a template, and deform it according to the geometry of the subject model is a natural second choice. We will also briefly discuss the template-based completion.

Mitra et al. [17] detected symmetry of models through a voting scheme; they use a clustering algorithm to identify the symmetry indicated by randomly sampled and paired surface points. This was later generalized by Pauly et al. [18] for discovering the structural regularity. Simari et al. [20] used GM-estimator to detect symmetry regions and introduced a folding tree data structure which is a hierarchical union of planar symmetric and asymmetric parts. Raviv et al. [19] used heat diffusion to induce the shape metric and define non-rigid symmetries as self-isometries with respect to the diffusion metric.

Many template-based completion algorithms [11, 1, 3] have been proposed to repair incomplete models with missing data with the help of a template model. The template is usually deformed to match the subject through a non-rigid registration, often composed by a global rigid/affine transformation and a local non-linear deformation. Our algorithm is similar to the framework of [1], where a local affine transformation is computed on each vertex of the template, minimizing a quadratic energy composed of a fitting error and a smoothing error.

### 3. Fragments Assembly

We propose a two-step assembling pipeline for fragmented skulls. (1) We matches and rigidly transforms each fragment to a template skull and roughly reassemble the skull (Section 3.1). (2) By analyzing break curves, we refine assembled fragments to reduce artifacts such as self-intersections and gaps (Section 3.2).

#### 3.1. Rough Assembly

Given a set of fragments $\{S_i\}$ (see Fig. 1(a)), we want to compute rigid transformations $\{T_i\}, T_i : S_i \rightarrow M$ from these fragments to a template skull $M$. Every model $S_i$ (also $M$) is represented as a triangle mesh. Transformed fragments $\{T_i(S_i)\}$ should reassemble the original geometry of the subject skull (Fig. 1(b)).

![Figure 1. Skull Assembly and Completion Pipeline. Skull fragments (a) are first assembled (b) with the help of a template, then refined (c) based on break curve analysis, and finally completed (e) using symmetry (d).](image-url)
prove the reliability of the features extraction against noise, the transformed fragment on slippage feature points we also compute their curvatures, optimal transformation coarse matching between corresponding feature points can provide a more robust feature extraction (compared with models with similar spin images tend to be corresponding points. Fig. 2(b) shows the spin-image of a point \( p \) on a skull fragment (a). The possibility of its matching regions on the template is colored on the template skull (c).

However, computing and matching the spin-image on every mesh vertex are too time consuming. We first extract salient feature points on \( \{ S_i \} \), denoted as set \( V'_i^F \), using the slippage features [6], which are critical points that have maximized stability under the rigid matching of the surface with itself locally. Then we compute the spin-image on these extracted features on \( \{ S_i \} \) and \( M \). To further improve the reliability of the features extraction against noise, on slippage feature points we also compute their curvatures, and use a threshold \( \varepsilon_c \) to filter out point pairs with different local curvatures, which are unlikely to be corresponding pairs. Slippage features with similar spin image signatures and similar principal curvatures are matched. This approach provides a more robust feature extraction (compared with [22]) and helps increase the accuracy of feature matching. Then we apply the mean shift algorithm to cluster corresponding points on \( M \) for each extracted feature point on \( S_i \), and then use a graph matching algorithm [4] to get the final feature correspondence.

Finally, we search for the optimal transformation on each fragment using ICP while the matching errors between corresponding features contribute a feature term in the ICP energy: \( E_{\text{feature}} = \sum_{v_j \in \mathcal{V}_j} d^2_i(T_i(v_j), M) \), where \( d^2_i(x, M) \) is the distance from point \( x \) to surface \( M \).

Fig. 3 illustrates an example of the above process on a female skull. Assembled fragments are visualized from two viewing directions.

Selecting the template \( M \) is important for the initial assembling. A template skull having high geometrical similarity with the subject \( \{ S_i \} \) is desirable. A skull database is organized; skulls are classified by sex, age, cranial form, etc., and stored as a tree with two main branches: male and female. Then each branch splits into different sub-branches representing different races etc. Since no geometrical comparison can be done between \( M \) and \( \{ S_i \} \) before \( \{ S_i \} \) is assembled, we rely on the anthropological characteristics including sex, race, age to pick the most suitable template. Some of such information can be identified using different methods by specialists.

### 3.2. Curve Matching using LSTE

With the help of template skull, the fragments are roughly assembled (Section 3.1). However, this assembled skull needs further refinement, since gaps and intersections between adjacent patches can exist (see Fig. 3) due to potential misalignment (e.g., caused by the disparity between the template and the subject skull). On the other hand, the rough assembly does correctly reveal the adjacency relationship between fragments and provide a relatively good initial guess. In the next step, we refine the rough assembly result based on the geometry of the subject skull itself by analyzing the break-curves of the fragments.

Break-curves are defined as the locations along which the skull surface breaks. They provide useful geometrical information between adjacent fragments [8, 23]. Willis and Cooper [23] assemble axis-symmetric potteries by integrating the manually labeled break-curve segments and normal directions based on Bayesian analysis. When assembling fragmented skulls, however, since the skull is not as smooth as pots (especially, the front facial area of a skull has subtle yet important geometry), and is not axis-symmetric, directly applying this existing approach often fails. Especially when the following two cases happen: (1) matching ambiguity: break-curves from non-adjacent fragments may be geometrically similar, and therefore could be matched incorrectly; (2) matching error: corresponding break-curves of adjacent regions may have geometric inconsistency due to local damages, and therefore could be treated as not well matched.
In order to robustly find the pairing of matched curve segments from different fragments, it is necessary to build our curve matching upon the roughly assembled skull, in which spatial relationship among fragments can be induced. We denote the set of break-curves by $C = \{C_1, \ldots, C_N\}$, where $C_i$ is the boundary loop for the $i$th fragment $S_i$. Usually when two fragments $S_i$ and $S_j$ are adjacent to each other, parts of their break-curves $C_i$ and $C_j$ can be matched well. We use $\beta^{ij} \subseteq C_i$ to denote the segment of $C_i$ that matches a segment $\beta^{ji} \subseteq C_j$. The segment $\beta^{ij}$ is represented by a set of consecutive sampled points $V(\beta^{ij}) = \{\beta_k^{ij}\}, k = 0, \ldots, N_k = |\beta^{ij}|$ along $\beta^{ij}$ where $N_k = |\beta^{ij}|$ is the total number of sampled points on $\beta^{ij}$. Then the least square transformation error (LSTE) between two break-curves can be formulated as

$$d(\beta^{ij}, \beta^{ji}, T) = \sum_{v_k \in V(\beta^{ij})} d_T^2(\beta^{ij}, T(\beta_k^{ji}))$$

(1)

where $T$ is the transformation applied on $\beta^{ji}$ to be solved, and $d_T^2(\beta^{ij}, v_k)$ is the distance from point $v_k$ to curve $\beta^{ji}$.

Assembled skull is usually incomplete; it can still have missing regions, gaps, or self-intersections between re-assembled fragments. An example is shown in Fig. 6 (a). In order to get an accurate completion, we propose to use symmetry for the skull repair. We develop a reliable symmetry detection and completion algorithm (see Fig. 6 (b,c) and Section 4.1) for most damaged regions; if damages exist on both sides with respect to the symmetric plane, a template-based repair (see Fig. 6 (d,e) and Section 4.2) is performed to ensure the completion of the skull.

4. Skull Completion

The optimization proposed in [23] iteratively glues a fragment that best fits the assembled pot region; this assembly is slow (about 6.75 hours for assembling 10 fragments) and a false matching during the iteration may significantly affect the subsequent gluing of unassembled fragments. According to the rough assembling we can derive the set of adjacent fragments $A_i = \{S_{j_1}, S_{j_2}, \ldots, S_{j_m}\}$ of a given fragment $S_i$. Without searching within entire $C$, we only search for the best fragment in $A_i$ for $S_j$. Meanwhile, unlike [23], normal direction is not considered in Equation 2. Because the geometry of the skull surface is subtle, adding the normal term in skull assembling is insignificant. By this we reduce the searching space and can now globally optimize LSTE on all fragments simultaneously. Fig. 5 illustrates the result refined from the coarse assembly in Fig. 3.

4.1. Symmetry-based Completion

The inherent symmetry of skulls usually can provide a relatively accurate repair on damaged regions. In contrast, popular hole filling approaches based on fairing tend to smooth out subtle geometric details, and template-based methods (Section 4.2) tend to fill holes using the geometric variance of template instead of the substrate of the subject skull itself. Preserving subtlety of skulls is critical to the accuracy of the subsequent tasks such as facial reconstruction/superimposition. Therefore, symmetry has been widely adopted by forensic specialists in their manual com-
Figure 6. Skull Completion Pipeline. Given an incomplete skull (a), we detect the symmetric plane (b), and perform a symmetry-based completion (c). For the holes cannot be repaired using symmetry (the lower holes in (a)), we perform a template-based completion (e). (d) is the template we used. The symmetry error is color-encoded.

Figure 7. Symmetry Detection on Damaged Skulls. (a) Symmetry detection on a complete model (PC only); (b) symmetry detection on model with big missing regions (PC only); (c) symmetry detection on model with big missing regions (PC+SDF); and (d) completed skull.

Detection of skulls, and it naturally becomes our first choice.

Given the incomplete skull \( M \), first we need to find the symmetric plane \( L : ax + by + cz + d = 0 \) of the skull. Then the reflection of \( M \) about \( L \), denoted as \( M' = R_L(M) \), is used to repair \( M \). Suppose the complete skull is denoted as \( \hat{M} \). The ideal \( L \) actually should minimize the Hausdorff distance between \( \hat{M} \) and \( \hat{M}' = R_L(\hat{M}) \). In other words, \( L \) should be able to nicely reflect \( \hat{M} \), instead of \( M \) whose optimal symmetric plane may be different.

Detecting such a symmetric plane on significantly damaged models is nontrivial, and many existing symmetry-detection algorithms are not directly applicable. Mitra et al. [17] design a geometry signature based on principal curvatures. The principal directions \( c_1, c_2 \) along with the normal direction \( n \) define a local frame on a surface point. If a reflection transforms the local frame on a point \( p_1 \) to an-
other point \( p_2 \), then a corresponding symmetric plane is uniquely determined. Ideally, if two points are reflective to each other with respect to \( L \), they should have the reflective local geometry, i.e., local frames. Therefore, the algorithm of [17] pairs up surface points with similar principal curvatures (PC), uses all these pairs to vote (by a clustering algorithm) for the optimal symmetric plane. This symmetry detection algorithms works very well for complete shapes. However, it turns out to be less reliable for significantly damaged objects, whose sampling could be biased. Fig. 7 shows an example. The above symmetry detection works nicely for a complete skull (a), but is undesirable for a damaged model (b).

To detect symmetry on a damaged model more robustly, we include a second component to the descriptor of each local point that evaluates the volume information of the model. The Shape Diameter Function (SDF) [10] evaluates the approximate diameter of the object’s volume and is robust even when the object is incomplete with holes (see Fig. 8). We pair two points as potential corresponding reflective points when both their principal curvatures (local surface bending) and shape diameters (volume property) are similar (in our experiments, both disparity should < 10%). Finally, we cluster all computed transformations to get the optimal symmetric plane using Mean-Shift Cluster Method as suggested by [17]. As shown in Fig. 8, principal curvature signature works well for complete data after clustering (a). But it fails to detect symmetry correctly on incomplete data (e), because the green points and the red points have similar curvature and thus are mismatched. After considering the SDF signature(not sensitive to holes, (h,l)) of the points, the green points and red points are no longer matched. So the new pruning method (principal curvature + SDF) is more reliable for incomplete data (Fig.7 (c), Fig.8 (k)).

**Measuring Symmetry Error.** We measure the symmetry error \( e_{sym}(M, L) \) of a model \( M \) about a symmetric plane \( L \) using the average distance from the reflection of each vertex \( R_L(v), \ v \in M \) to \( M \) itself. The reflection \( R_L(v) \) of \( v \) about \( L \) can be easily computed in a closed-form representation. The distance from \( R_L(v) \) to \( M \) can also be evaluated efficiently through the implicit representation of \( M \). We compute the distance field in \( \mathbb{R}^3 \) to \( M \) using the algorithm introduced in [14]. It can be used for quick query of this distance \( d_s(v, S) \) from vertex \( v \) to surface \( S \). We color-encode the symmetry error and illustrate it on the skull model, to visualize the detected symmetry.

**4.2. Template-based Completion**

Despite its accuracy, symmetry-based completion cannot repair skulls whose both sides (with respect to the symmetric plane) are missing. See Fig. 6 (a) (the lower two holes on the skull) for example. On a skull damaged like this, the template shall be used to fill the hole. When a mapping between the template \( M \) and the subject \( S \) is computed, for each missing region \( H \) on \( S \), we can transplant its corresponding region \( H' \) on \( M \) to fill it.

First, we use a non-rigid registration computation similar to [1] to transform the template \( M \) to subject \( S \). We define a local transformation on every vertex and minimizes a registration energy composed of three quadratic terms: (1) data fitting error \( E_{data} \), (2) smoothness (shape-preserving) error \( E_{smooth} \), and (3) feature alignment error \( E_{feature} \). \( E_{data} \) (and \( E_{feature} \)) naturally measures the deviation of each point (and feature) from closest point (corresponding feature) on the target surface, after the transformation. \( E_{smooth} \) then balances the fitting accuracy and the smoothness of the transformation; it prevents transformations applied on adjacent points from being too dramatic. The error terms can be formulated as:

\[
\begin{align*}
E_{data} &= \sum_{v_i \in M_V} d_s^2(T_i + v_i, S), \\
E_{feature} &= \sum_{v_i \in M_{VF}} d_s^2(T_i + v_i, S), \\
E_{smooth} &= \sum_{\{v_i, v_j\} \in M_E} |T_i - T_j|^2,
\end{align*}
\]

where \( M_V, M_{VF}, \) and \( M_E \) are the vertex, feature, and edge sets of mesh \( M \) respectively; \( T_i \) is \( v_i \)'s transformation to be solved. The final objective function is \( E = \mu_1 E_{data} + \mu_2 E_{smooth} + \mu_3 E_{feature} \) which can be solved efficiently.

Second, inspired by [24], we further use harmonic functions to obtain natural deformation and smooth gluing of
H’ from M. In [24], harmonic vector fields are computed using a harmonic-guided diffusion technique enforcing boundary constraints. In our problem, we need to compute the transformation of patches H’ = \{H’_k\}, where each H’_k is a connected component. The translation vector defined on each vertex is also composed of 3 components \(x, y, z\). We can solve them efficiently following the classical discrete harmonic function computation [9]. The boundary constraints are transformations computed on boundary vertices of \{H’_k\}. We now briefly recap this harmonic function computation on triangle meshes. For each patch H’_k, let \(T^k = (t^k_1, t^k_2, \ldots, t^k_n)^T\) be the set of unknown translation vectors, \(t^k_i\) is a \(3 \times 1\) vector on each vertex \(v^k_i\) of H’_k. \(T^k\) is computed by minimizing the quadratic energy: 
\[
T^k = \arg \min_{T} \left\{ \frac{1}{2} T^T \Delta T \right\},
\]
where L is the cotangent Laplacian matrix defined as \(L = D - W\). \(D\) is a diagonal matrix such that
\[
d_{ij} = \begin{cases} 
0 & i = j \\
\sum_j w_{ij} & \text{otherwise} 
\end{cases} \quad (4)
\]
and \(w_{ij} = \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})\) is the cotangent weight [9] defined on the edge \([v^k_i, v^k_j]\) \(\in H’_k\), where \(\alpha_{ij}\) and \(\beta_{ij}\) are two angles opposite to the edge \([v^k_i, v^k_j]\). The harmonic stretching energy is minimized and \(H’_k\) is transformed to obtain smooth deformation. Fig. 9 illustrates the effect of completion without (c) and with (d) harmonic refinement after the ICP-like template registration. A more natural completion is obtained when the harmonic refinement is applied. After the template patch is deformed, we cut it and from M and glue it to S.

5. Experimental Results

We have used our assembly and completion pipeline to repair three sets (two female and one male) of excavated skull fragments. The completion of the female skulls \(S_1, S_2\) is shown in Fig. 1 (e), Fig. 10 (d), and the completion of the male skull \(S_3\) is shown in Fig. 7 (d). We also perform experiments on a skull model \(S_0\) from Aim@Shape Repository: we simulate the handling of completion of various fragmented cases and holes. The \(S_0\) is normalized to a unit box and the completion error can be measured between the repaired model \(\tilde{S}\) and \(S_0\) by \(D(S_0, \tilde{S}) = 1/N \sum_{v \in V(S_0)} d^2(v, \tilde{S})\), where \(V(S_0)\) is the vertex set of \(S_0\). A statistical runtime table is given in Table 1. As we can see, the pipeline is efficient and the completion error (of the simulation on \(S_0\)) is quite small. Our algorithm is efficient (< 6 minutes for 8 fragments).

Fig. 10 shows a side-by-side comparison of three hole-filling approaches we implemented in this work. On the same upper right hole of \(S_2\) (a), the direct hole-filling [15] (b) and template-based completion (c) are less natural compared with the symmetry comparison (d).

Table 1. Runtime Table: \#\(\Delta\)(K): the number of (thousands) triangles in the mesh; \#F: number of fragments; \(T_f\): feature extraction time; \(T_{asm}\): total assembling time (ICP+curve matching); \(T_{com}\): completion time (symmetry+template); \(\varepsilon\): completion error. The experimental time is measured in seconds.

<table>
<thead>
<tr>
<th>Skulls</th>
<th>#(\Delta)(K)</th>
<th>#F</th>
<th>(T_f)</th>
<th>(T_{asm})</th>
<th>(T_{com})</th>
<th>(\varepsilon(10^{-4}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>70.6</td>
<td>5</td>
<td>23.2</td>
<td>113.6</td>
<td>105.7</td>
<td>N/A</td>
</tr>
<tr>
<td>S2</td>
<td>71.3</td>
<td>8</td>
<td>27.7</td>
<td>182.7</td>
<td>109.5</td>
<td>N/A</td>
</tr>
<tr>
<td>S3</td>
<td>75.2</td>
<td>7</td>
<td>32.1</td>
<td>213.5</td>
<td>113.0</td>
<td>N/A</td>
</tr>
<tr>
<td>S0</td>
<td>31.6</td>
<td>4</td>
<td>11.2</td>
<td>80.1</td>
<td>30.8</td>
<td>0.96</td>
</tr>
<tr>
<td>S3</td>
<td>30.5</td>
<td>7</td>
<td>16.8</td>
<td>103.8</td>
<td>29.7</td>
<td>1.17</td>
</tr>
<tr>
<td>S0</td>
<td>31.2</td>
<td>4</td>
<td>12.7</td>
<td>82.7</td>
<td>30.4</td>
<td>1.13</td>
</tr>
</tbody>
</table>

6. Conclusion

We introduce a framework for fragmented skull completion. The framework has two components: fragments assembly and model repair. (1) We compute a rough assembly based on registration between fragments and the template, we then refine it by globally optimizing break-curve matching in a reduced searching space. (2) We develop a reliable symmetry detection algorithm on damaged models, and use it to repair the skull. We demonstrate the efficacy of our proposed algorithms on both steps, with respect to existing approaches.

A limitation of our current framework is in the assembly of tiny fragmented pieces. When a fragment is too small, especially without salient geometric features, the fragment-template matching (Section 3.1) is not reliable. Right now we do not conduct rough assembly for these pieces, but directly put them in the break-curve matching optimization. Sometimes it may still fail to find correct locations of such pieces on the subject. Then, we give up assembling such pieces, and let the completion algorithm finish the repair. This problem might not be easily solved using pure geometric methods; we will explore effective integration of domain knowledge into this completion system. In the near future, we will also work on the modeling of facial tissues, and evaluate the effectiveness of this entire computer-aided face reconstruction pipeline.

Acknowledgement

This project is partially supported by Louisiana Board of Regents (LA-BOR) Research Competitiveness Subprogram (RCS) LEQSF(2009-12)-RD-A-06, LA-BOR PFund: NSF(2011)-PFund-236, LSU Faculty Research Grant 2010, and LA-BOR Post Katrina Funds. This work was done when the first author was a visiting student at Louisiana State University.

References


