Computers & Graphics ■ (■■■■) ■■■–■■■

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on 3D Object Retrieval

# Landmark transfer with minimal graph

Vasyl Mykhalchuk [a], Frederic Cordier [b], Hyewon Seo [a,*]

[a] Université de Strasbourg, (UMR 7357 CNRS), France
[b] LMIA, Université de Haute Alsace, France

ARTICLE INFO

*Article history:*
Received 25 October 2012
Received in revised form
15 April 2013
Accepted 15 April 2013

*Keywords:*
Landmark
Feature point
Graph matching
Isometric deformation

ABSTRACT

We present an efficient and robust algorithm for the landmark transfer on 3D meshes that are approximately isometric. Given one or more custom landmarks placed by the user on a source mesh, our method efficiently computes corresponding landmarks on a family of target meshes. The technique is useful when a user is interested in characterization and reuse of application-specific landmarks on meshes of similar shape (for example, meshes coming from the same class of objects). Consequently, across a set of multiple meshes consistency is assured among landmarks, regardless of landmark geometric distinctiveness. The main advantage of our method over existing approaches is its low computation time. Differently from existing non-rigid registration techniques, our method detects and uses a minimum number of geometric features that are necessary to accurately locate the user-defined landmarks and avoids performing unnecessary full registration. In addition, unlike previous techniques that assume strict consistency with respect to geodesic distances, we adopt histograms of geodesic distance to define feature point coordinates, in order to handle the deviation of isometric deformation. This allows us to accurately locate the landmarks with only a small number of feature points in proximity, from which we build what we call a minimal graph. We demonstrate and evaluate the quality of transfer by our algorithm on a number of Tosca data sets.



## 1. Introduction

Landmarks play a central role in many algorithms, including correspondence computation, and shape analysis, which deal with highly relevant problems in shape retrieval. Consequently, a lot of attention has been paid to landmark extraction and matching problems during the past decade. While most existing landmark extraction methods use geometrical prominence as a main criterion of feature selection, landmarks can often be defined from the semantics that are specific to applications, independently from geometric saliency. This is particularly true for anthropometric studies [1] or computer animation [2]. Moreover, landmarks are often not persistent across pose changes or inter-subject variations.

So far, when a user is interested in characterization and selection of points on a mesh without a strongly distinguishable geometric saliency, we have often relied on manual labeling. Manual labeling has also been almost the only trustworthy way when the objective is to obtain a persistent set of landmarks across a set of multiple meshes. Existing techniques on automatic landmark extraction [3,4] and matching may not work well in such cases, since the geometric features are not necessarily persistent across deformations; for instance, in case of non-rigid deformations.

However, the work spent on manually labeling and associating landmarks is tedious and time consuming. Thus, in this work we aim at developing techniques to help with the reuse of the landmarks defined by the user, so that consistency can be assured with a minimal user input, regardless of geometric distinctiveness of the landmarks.

Our landmark transfer technique allows the user to define one or more custom landmarks on a source mesh, and efficiently computes meaningful correspondences on a family of target meshes that are approximately isometric. We develop our method for uniquely describing any given point on the shape, which is not necessarily geometrically significant. A good advantage of our method in comparison with relevant/existing techniques is its fast computation time. This is possible because our method is optimally designed for transferring a sparse set of landmarks on multiple target models while avoiding unnecessary full registration.

With the goal of optimal landmark transfer towards obtaining a consistent set of landmarks across multiple sets, we make several smaller contributions:

(1) We develop the idea of the minimal graph (Section 5), which is used for landmark transfer with minimum computation.
(2) Identification of landmark points using a newly defined geodesic coordinates (Section 6.1): in contrast to previous approaches, we do not rely solely on geodesic distances. Instead we develop a reliable method of updating geodesic distances, which compensates well for distance changes due to imperfect isometry and assures precise and consistent landmark location.

* Corresponding author. Tel.: +33 3 68 85 45 58.
*E-mail address:* seo@unistra.fr (H. Seo).

0097-8493/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2013.04.005

## 2. Related work

In some sense, the problem we are solving in our work can be seen as a sub-problem of full correspondence, although it should be noted that our technique is tailored for the fast transfer of a sparse set of user-defined landmarks. Thus, we give a brief review of recent techniques devoted to surface registration here. In most existing registration techniques, to make the problem tractable, a smaller number of sparse correspondences are preceded, before it can be extended to a full correspondence. This strategy is often adopted for both inter-subject deformations [5,6] and approximate/near isometries of the same object [7–9]. These landmarks, whether automatically sampled/extracted [7–9] or manually labeled [5,6], facilitate specifying the rough physical characteristics or poses, so that the matching is made easier especially when surfaces exhibit large deformations.

Methods handling the large deformation can be classified into two categories: those that deal with large deformations of the same object (inter-subject registration) and those that register large inter-subject deformations. Inter-subject registration often relies on isometry-invariant local descriptors to select geometric feature points; then finds a matching among them such that the pairwise geodesic distances between all feature point pairs are preserved. Chang and Zwicker [7] developed an algorithm that assumes skeleton driven deformation among the meshes. They use the spin image as a feature descriptor and measure similarity between descriptors to find matches on a subset of vertices. For each match they generate a rigid transformation and cluster the resulting set of candidate transformations to obtain the final set of transformations. Assigning the transformations to the shapes has been made by using graph cuts optimization. Non-rigid registration proposed by Huang and coworkers [8] can be seen as a variant of the ICP (iterative closest point) algorithm, it finds optimal matching among a subset of vertices by using the Euclidean and feature distances among matching pairs. Tevs and colleagues [9] presented a RANSAC-like matching algorithm. For a set of random source points, they select the corresponding target points according to a probability function that measures the accuracy of the matching. The matching is further extended by adding additional correspondence in a way that they do not violate the isometric matching criterion. Later, the authors have extended their idea [10] by proposing a planning step to find an optimal set of feature points, instead of choosing the source points randomly. These points are matched first so that matching process converges to the solution as quickly as possible. More recently, Ovsjanikov et al. [11] have shown how dense isometric maps can be found among nearly isometric surfaces from a single correspondence, by using the Heat Kernel Map (HKM).

Most intra-subject registration techniques in computer graphics have been devoted to matching among different scans of human bodies [5,6]. They assume manually labeled landmarks on the surface and cast the matching problem as an optimization one, by using the error terms: the sum of Euclidian distances among corresponding landmarks, surface distance, and distortions of the surface under deformation. Lipman and Funkhouser [12] use Möbius transformations defined by a set of three randomly sampled points on each of the two point sets, and produces correspondences via a voting algorithm. They have shown that the algorithm can automatically find dozens of point correspondences between different object types belonging to the same class in different poses. Kim et al [13] also adopted Möbius transformations on conformal maps of each mesh, which have been computed from subsets of previously found sparse correspondences among feature points to produce a number of maps. These maps are then blended with weights that are computed with an objective function that favors low-distortion everywhere.

While it is possible to eventually consider these methods for the landmark transfer problem, our setting is different from (sparse or dense) matching of isometric surfaces. First, we assume that a sparse set of landmarks is provided by the user. This allows the user to define application-specific landmarks, independently from the geometric saliency. Second, our method efficiently computes a coherent set of corresponding landmarks on a number of target models. Unlike most existing methods that focus on computing global optimal solution to the full correspondence, we perform the transfer in one-by-one basis while avoiding unnecessary and costly full registration.

Graph matching has been successfully adopted in shape matching [14] and symmetry detection [15]. In our work, we use graphs for assisting the matching of geometric feature points within and between meshes. Graphs are constructed using geometric feature points as nodes; edges between connected feature points are weighted by the geodesic distances between the two.

## 3. Overview

The different steps of our algorithm are illustrated in Fig. 1. First, we build a graph $G_F$ on the source mesh $M_S$, whose nodes are the set of automatically selected geometric feature points and the edges are composed of geodesic paths between the nodes (Fig. 1(a)). Then, given a user-specified landmark, we build what we call the minimal graph $G_M$, a subgraph of $G_F$ (Fig. 1(b)). The graph $G_M$ has three main properties: (1) it uniquely defines the user-provided landmark, (2) it is as small as possible in terms of number of nodes and geodesic distances, (3) it is a unique subgraph of $G_F$, i.e. there is no other subgraph in $G_F$ that matches with $G_M$.

Next, given a target mesh $M_T$, we select a set of points with the local shape signatures similar to the points from graph $G_M$. From these feature points we compute the graph $\widehat{G}_F$ by connecting the points which are within the maximum geodesic radius of $G_M$ (Fig. 1(c)). Then we use the approximate graph matching technique to find $\widehat{G}_M$, a subgraph of $\widehat{G}_F$, that best matches with $G_M$.

Finally, now that we have $G_M$ matched with $\widehat{G}_F$ on the target mesh, we can find the corresponding landmark location on the target mesh by using $\widehat{G}_M$ (Fig. 1(d)). This task would be made easier if the source and target meshes are perfectly isometric, since we can simply use the geodesic distances from each of the geometric feature points to be able to uniquely identify the landmark location. Unfortunately, the meshes are only approximately isometric and such a method may fail to estimate the landmark location reliably, especially when the deformation between the two meshes is large. We solve this problem by interpolating the updated geodesic distances on the target mesh in order to compensate changes in those that were induced due to non/roughly isometric deformation.

### 3.1. Assumptions

Like many existing non-rigid registration methods, we expect that the meshes are approximately/nearly isometric. Techniques developed with such an assumption are appreciated in many applications dealing with the matching of 3D scan data of deforming objects.

As is the case with many real-world applications, we assume that the landmarks are sparse and develop our algorithm that is optimally tailored for such cases. However, our method can be easily extended to complete matching, with the modification of the use of the minimal graph. We discuss this point in further detail in Section 8.

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

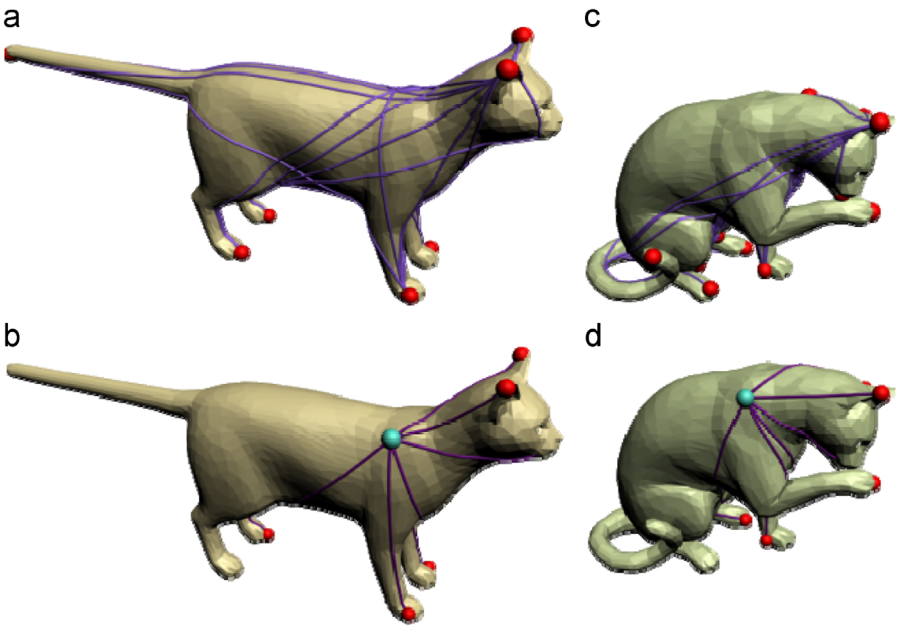


**Fig. 1.** Overview of our approach. Extracted geometric feature points are marked with red spheres, and the user landmarks as blue ones. (a) Geometric feature points are extracted on the source mesh, from which full graph $G_F$ is computed (see Section 4.3). (b) Provided a user landmark on the source mesh, minimal graph $G_M$ for the landmark is constructed (see Section 5). (c) Similarly to the source mesh, geometric feature points are extracted, and the full graph $\widehat{G}_F$ is computed on the target. (d) A corresponding point is computed on the target mesh using $\widehat{G}_M$, a partial matching of $G_M$ on $\widehat{G}_F$ (Sections 6.2, 6.3). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 4. Graph construction on the source mesh

### 4.1. Geodesic distance computation

Since our method relies heavily on the geodesic distance, its accurate computation is crucial. Later in this work, geodesic distances are used to compute isocurves of the local descriptors on the surface (Section 4.2), as well as to compute geodesic paths between geometric feature points (Section 4.3). Throughout this work, we adopt the MMP method of exact geodesic computation proposed by Mitchell et al. [16]. The algorithm itself is very similar to Dijkstra's approximation on the graph formed by vertices and edges of the mesh. This means that the surface of the mesh is regarded as locally planar on each face. As in Dijkstra's algorithm we start from a source vertex $v_S$ and compute the geodesic distance map in a region growing manner. Each edge in the graph representation of the mesh is subdivided into a set of segments by artificially adding vertices on the edge; on each Dijkstra's step, when updating the geodesic map, a geodesic path is allowed to pass through these artificial vertices. When there is no edge partitioning, we get standard Dijkstra's algorithm; the more partitioning vertices go through, the more precise is the geodesics computation. In the worst case the algorithm has time complexity $O(\log(n)n^2)$. Note that the partial computation is possible: we can stop computation when the path reaches certain distance or covers certain points on the surface of the mesh. This is an important point to reduce the computation time.

### 4.2. Feature point extraction

We use a local shape descriptor to identify the geometric features. Assuming approximate isometry between the source and target shapes, we are interested in a descriptor invariant to isometry and insensitive to the mesh discretization as much as possible. We employ the intrinsic wave descriptor proposed by Tevs et al. [10] and further refine it so that it is more robust to changes in mesh sampling.

For each vertex $x$ we compute a set of intrinsic geodesic isocurves of increasing the geodesic distance $r_i$ from $x$ with a fixed step $\Delta r$, by using the algorithm described in Section 4.1. The length $l_i$ of each curve is then normalized by $2\pi r_i$, the length of the geodesic isocurve on a flat surface. We sampled 16 isocurves as in Tevs et al. [10], resulting in the descriptor of a form $\boldsymbol{D_x} = (l_1/2\pi r_1, l_2/2\pi r_2, \ldots, l_{16}/2\pi r_{16})^T$. We approximate $l_i$ as a perimeter $P_i$ of a polygon whose edges connect intersection points of the real isocurve with triangle edges on the mesh. Next we take the inverse of the Euclidean norm of $\boldsymbol{D_x}$ in order to measure the geometric prominence (convexity/concavity) of vertex $x$:

$$\gamma(x) = \|\boldsymbol{D_x}\|_2^{-1}. \tag{1}$$

$\gamma(x)$ increases with growing "sharpness" of the shape in the neighborhood of $x$. Eventually $\gamma$ comes up to infinitely large values for a vertex on the tip of an infinitely sharp, needle-like shape.

Having computed the convexity all over the whole of the mesh (Fig. 2(a)), we sort its values $\gamma(x)$ in a descending order and retain only the first $\alpha{\cdot}n$ vertices with the highest values of $\gamma$, with $n$ being the number of vertices in the mesh. This gives us a set $X_\gamma$ of most prominent vertices with respect to convexity. We normally set a user-defined parameter $\alpha$ with a value of 0.3. As can be seen from Fig. 2(b), the vertices from $X_\gamma$ group around 'tips' of the mesh; we denote the number of these clusters as $n_F$.

From these most prominent vertices $X_\gamma$ on the mesh, we extract a set of feature points $V_F = \{f_i\}$, $i = 1 \ldots n_F$ in the following way. First, we assign vertices from $X_\gamma$ to a set of clusters $L = \{L_i\}$, $i = 1 \ldots n_F$ according to their local shape descriptor similarity and geometrical proximity. We chose the most prominent vertex $\tilde{x} \in X_\gamma$ and add it to an initial cluster $L_0 = \{\tilde{x}\}$. Then, we grow $L_0$ around $\tilde{x}$ according to the connectivity of vertices in $L_0$; the growing of the cluster stops when no more vertex $x$ is encountered that is adjacent to $L_0$ and satisfies $|\gamma(\tilde{x}) - \gamma(x)| < \gamma'$, where $\gamma'$ is a user-defined threshold. We repeat this process among those vertices that have not yet been labeled to construct subsequent clusters $L_i$ until all vertices in $X_\gamma$ have been assigned to a cluster. Second, in each of the clusters $L_i$ we identify a vertex $f_i \in L_i$ whose average geodesic distance to all

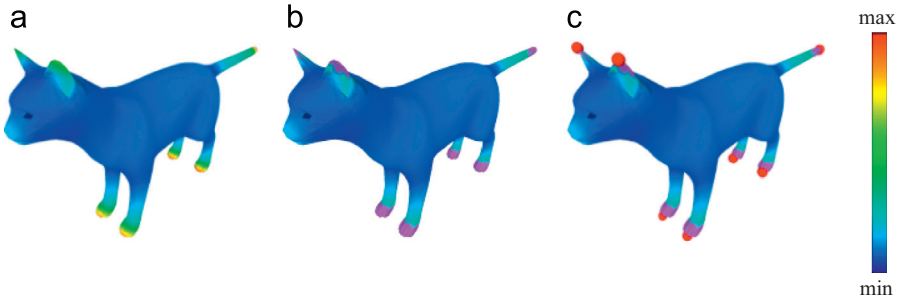


**Fig. 2.** Color-coded images of the convexity and geometric feature points on the mesh. (a) Visualization of the convexity field $\gamma(x)$ over the vertices of the mesh. (b) Set $X_\gamma$ of vertices with highest convexity is shown in purple. (c) Extracted feature points, which are the 'central' points of the regions of high convexity, are depicted as red dots. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the other vertices in its cluster is maximal, in preference to center points of the cluster. The extracted set $\{f_i\}$, $i=1\ldots n_F$ represent the 'tips' of the mesh (Fig. 2(c)).

### 4.3. Construction of the full graph $G_F$ on the source mesh

Given the feature points $V_F=\{f_i\}$, $i=1\ldots n_F$ we proceed with the construction of the full graph $G_F=(V_F,E_F)$. Each $f_i$ is connected by an edge $e\in E_F$ to all the others from $V_F$. So that, $G_F$ forms a complete graph of $n_F$ vertices. Let $\delta(v_i, v_j)$ be the geodesic distance between the vertices $v_i$ and $v_j$. We label each edge $(f,f')\in E_F$ with the corresponding geodesic distance $\delta(f,f')$.

## 5. Minimal graph construction

Given a full graph on the source mesh, we build what we call minimal graph $G_M$, a subgraph of $G_F$. The graph $G_M$ has two main properties: (1) it uniquely defines the user-provided landmark; i.e. position of a landmark can be uniquely identified by its geodesic distances to the nodes of $G_M$. (2) It is as small as possible in terms of the number of nodes and the geodesic distances it spans.

Given a landmark $v$ specified by the user, we build a minimal graph $G_M$ by iteratively adding nodes from the feature point set $V_F=\{f_i\}$, $i=1\ldots n_F$ one by one, in an order of proximity. We repeat this process until either all the following conditions are satisfied, or all nodes in $V_F$ are considered.

1 Position of $v$ is uniquely defined by its geodesic distances to each node in $G_M$.
2 $v$ is enclosed by the nodes of $G_M$.
3 $G_M$ is a unique subgraph of $G_F$ up to a symmetry.

Note that we cannot guarantee that $G_M$ always meets all these conditions. In such case, $G_M$ becomes equal to $G_F$. Algorithm 1 to Algorithm 3 summarizes the procedure for the minimal graph construction.

**Algorithm 1.** MinimalGraphConstruction

**Input:** $G_F$: the full graph of the source mesh,
$v$: a landmark specified by the user.
**Init:** $G_M=\varnothing$, $V_M=\varnothing$ ;
$V_F\leftarrow$ all nodes of $G_F$ sorted in an order of increasing geodesic distance from $v$;
**begin**
$n_{matching}\leftarrow\infty$//Number of matchings of $G_M$ to $G_F$;
$Err_{geod}\leftarrow\infty$//Geodesic error;
$g_0\leftarrow$first node from $V_F$;
$V_M\leftarrow V_M\cup\{g_0\}$;
$V_F\leftarrow V_F\backslash\{g_0\}$;
**repeat**
$g\leftarrow$fetch next node from $V_F$;
$V_M\leftarrow V_M\cup\{g\}$;
$V_F\leftarrow V_F\backslash\{g\}$;
$G_M\leftarrow$a complete graph with $V_M$ as nodes;
**if** ( countMatching $(G_M, G_F) < n_{matching}$) AND
(localizationErr $(v, G_M) < err_{geod}$)
$err_{geod}\leftarrow$localizationError $(v, G_M)$ ;
$n_{matching}\leftarrow$countMatching $(G_M, G_F)$ ;
**else**
$V_M\leftarrow V_M\backslash\{g\}$ ;
**endif**
$b_{enclosed}\leftarrow$ isEnclosedByNodes $(v, G_M)$;
**until** $(n_{matching}==1$ AND $err_{geod}<2\varepsilon$ AND $b_{enclosed==}$ **true**) OR $(V_F==\varnothing)$;
**return** $G_M$;
**end**

**Algorithm 2.** localizationError $(v, G_M)$.

// Estimate the range of regions in which $v$ can be localized in $G_M$.
**Input:** $G_M$: the current minimal graph,
$v$: a landmark specified by the user.
**Init:** $U\leftarrow\varnothing$,
$V_M\leftarrow$ all nodes of $G_M$;
$\varepsilon\leftarrow$ a small value;
**begin**
$U\leftarrow\{v_u|\ \|\delta(v,g_i)-\delta(v_u,g_i)\|<\varepsilon,\ \forall g_i\in V_M\}$;
$err\leftarrow$ the longest geodesic distance among $\{v_u\}$;
**return** $err$;
**end**

**Algorithm 3.** isEnclosedByNodes (v, $G_M$)

// Check if $v$ is enclosed by nodes of $G_M$.
**Input:** $G_M$: the current minimal graph,
$v$: a landmark specified by the user.
**Init:** $d\leftarrow$ a small value;
**begin**
$N_v\leftarrow\{p\in M_S|\|\delta(v,p)\|<d\}$;
**if** $\exists p\in N_V:\delta(p,g_i)>\delta(v,g_i),\forall g_i\in V_M$
**then**
**return false**;
**else**
**return true**;
**end**

**Unique coordinates condition**. Let $V_M=\{g_i\}$, $i=1..n_M$ be the set of nodes of $G_M$. We first find the set $U$ of all vertices $v_u$ whose geodesic distances to $g_i$ are of approximately equal length to those of $v$, as represented by:

$$\|\delta(v,g_i)-\delta(v_u,g_i)\|<\varepsilon \quad for \quad \forall g_i \in V_M \tag{2}$$

where $\varepsilon$ is a distances tolerance of equal geodesic length. If the largest geodesic distance among the vertices in $U$ is smaller than $2\cdot\varepsilon$, we consider that $G_M$ satisfies the unique coordinates condition, i.e. it can accurately locate $v$ (see Fig. 3).

**Enclose-by-nodes condition**. The second condition requires $v$ to be enclosed by the nodes of $G_M$, which makes the localization of $v$ more robust to small changes of geodesic distances. In Fig. 4(a), feature points $g_A$, $g_B$ and $g_C$ are located on one (the left) side of the landmark $v$. When the geodesic distances $\delta(v, g_A)$, $\delta(v, g_B)$ and $\delta(v, g_C)$ increase or decrease with the deformation, the estimated position of $v$ will move to right or left, respectively. In Fig. 4(b), on the other hand, by using additional feature point $g_D$ located on the other side of $v$, the distance changes influenced by $\delta(v, g_A)$, $\delta(v, g_B)$ and $\delta(v, g_C)$ will counterbalance with the change of $\delta(v, g_D)$. This strategy assumes simultaneous increase or decrease of geodesic distances, which has been usually the case in our experiments.

The algorithm proceeds as follows. First, we compute a set $N_V$ of all the vertices in the neighborhood (within certain geodesic distance) of $v$. If there exists a point $p \in N_V$ which is located further (with respect to $v$) to all the nodes of $G_M$, then $G_M$ does not satisfy the enclose-by-nodes condition with respect to $v$.

**Unique subgraph condition.** Primary step of our minimal graph construction algorithm is to ensure that user's landmark is defined uniquely on the source mesh. Several possible matching between $G_M$ and $G_F$ implies multiple matching between $\widehat{G}_M$ and $\widehat{G}_F$, and therefore multiple transferred landmarks that are computed from each matching. In order to avoid such an ambiguity, we build $G_M$ as a unique subgraph of $G_F$.

We initially tried to check the number of matching of the minimal graph with the full graph at each iteration, and stop growing the graph when we find only one matching. The graph matching algorithm we use is developed in spirit of Ullmann's tree-search subgraph isomorphism [17], which we describe in Appendix A. However, in the presence of symmetry in the mesh, which is often the case, the minimal graph will always be equal to the full graph, which is undesirable. Thus, we slightly modify our initial algorithm and add vertices to the current minimal graph only when the resulting graph reduces the number of matching to the full graph. This means that the our method does not guaranty the uniqueness of the matching of the minimal graph to the full graph, permitting the symmetric ambiguity. Further discussions related to this aspect can be found in Section 7.5.

## 6. Landmark transfer via graph matching

### 6.1. Construction of the full graph $\widehat{G}_F$ on the target mesh

Similarly to the source, we first extract the feature points on the target mesh using the convexity values computed from the modified intrinsic wave descriptor (Section 4.2). These feature points constitute the nodes of $\widehat{G}_F$, the full graph on the target mesh. To distinguish target mesh structures from the counterparts of the source, we use a 'hat' notation. Next, we compute the geodesic paths among them which serve as weighted edges of $\widehat{G}_F$. We avoid computing all the geodesic paths by limiting ourselves to those geodesic paths on the target mesh whose length is smaller or equal to $l_{max}$, the longest geodesic distance in $G_M$. An explanation for this is that paths that exceed $l_{max}$ will obviously not match with any path of the minimal graph $G_M$.

### 6.2. Matching the minimal graph $G_M$ to the full graph $\widehat{G}_F$

Having computed $\widehat{G}_F$, our goal now is to find a subgraph $\widehat{G}_M$ of $\widehat{G}_F$, that best matches with $G_M$. In general, due to imperfect isometries in the real world data sets, full graphs might not be consistent across the given meshes. We handle this problem again by using a variant of Ullmann's graph matching algorithm [17] (see Appendix A for a detailed description), with partial matching. This time, while building a search tree of possible matching solutions we consider partial matching as well, i.e. we look for a subgraph of

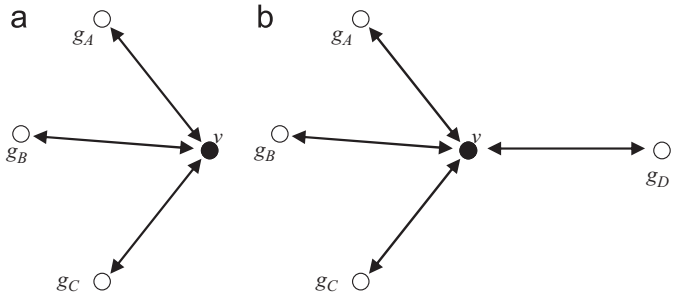


**Fig. 4.** (a) The landmark (dark dot) is not surrounded by feature points (white dots). In (b) by adding another feature point, the landmark is inside a convex hull; increases the confidence in the localization of the landmark.

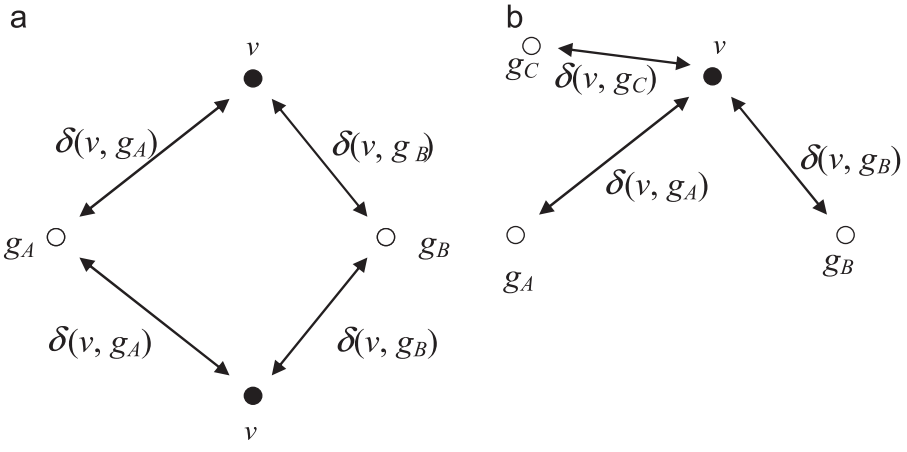


**Fig. 3.** Construction of the minimal graph. (a) By using two feature points (white dots), there are two possible position of landmark (black dot), thus the landmark is not uniquely defined.$\varepsilon$ value from $U$ is large. (b) With a minimal graph composed of three feature points, the landmark position is more unique ($U$ is very localized and $\varepsilon$ is small).

$G_M$ which is isomorphic to some subgraph of $\widehat{G}_F$. For each possible matching an error value is assigned, and the matching with the minimal error is chosen as the solution $\widehat{G}_M=(\widehat{V}_M,\widehat{E}_M)$, which is then used to locate the landmark on the target mesh.

### 6.3. Landmark transfer

If the source mesh $M_S$ and target mesh $M_T$ are perfectly isometric, then we are able to uniquely identify the landmark location by using solely the geodesic distances from each of the nodes of $\widehat{G}_M$. Let us denote a set of geodesic distances from a vertex $v\in M_S$ to the vertices from $V_M=\{g_i\}, i=1..\widehat{n}_M$, as

$$\delta_{\mathbf{M_S}}(v)=(\delta(v,g_1),\ldots,\delta(v,g_{\widehat{n}_M}))^{\mathrm{T}}, \tag{3}$$

and refer to it as feature point coordinates of $v$ (FP-coordinates). Then, we must be able to uniquely determine the location of the transferred point $\widehat{v}$ on $M_T$ that satisfies $\delta_{\mathbf{M_T}}(\widehat{v})=\delta_{\mathbf{M_S}}(v)$, where

$$\delta_{\mathbf{M_T}}(\widehat{v})=(\delta(\widehat{v},\widehat{g}_1),\ldots,\delta(\widehat{v},\widehat{g}_{\widehat{n}_M}))^{\mathrm{T}} \tag{4}$$

However, in practice, $M_S$ and $M_T$ are not isometric and in general case $\delta_{\mathbf{M_T}}(\widehat{v})\neq\delta_{\mathbf{M_S}}(v)$. This is partly due to the definition of the geodesic distance [18], the shortest surface distance between the two points. As illustrated in Fig. 5, the shortest geodesic path between the two points of interest changes as the shape deforms. Along the bending of the cylinder, we observed up to 9% of change in the geodesic distances.

Our solution to the above problem is to modify the geodesic distances on the target mesh in a way that they become similar to those of the source mesh. Let $\widehat{g}_i,\widehat{g}_j,g_i,g_j$ be two feature points on the target mesh and their corresponding counterpart on the source mesh respectively. Due to the non-isometric deformation, the distances $\delta(\widehat{g}_i,\widehat{v})$ are different from $\delta(g_i,v)$, with $v$ and $\widehat{v}$ being a vertex on the source mesh and its counterpart on the target mesh respectively. The idea is to modify the all geodesic distances from $\widehat{g}_i$ such that these geodesic distances become closer to those of the source mesh. That is, the distance $\delta(\widehat{g}_i,\widehat{g}_j)$ will become equal to $\delta(g_i,g_j)$. Similarly, the distance $\delta(\widehat{g}_i,v)$ of vertex $v$ in the close neighborhood of $\widehat{g}_j$ (i.e. $\delta(\widehat{g}_i,v)\,\delta(\widehat{g}_i,\widehat{g}_j)$ ) will be become close to $\delta(g_i,g_j)$.

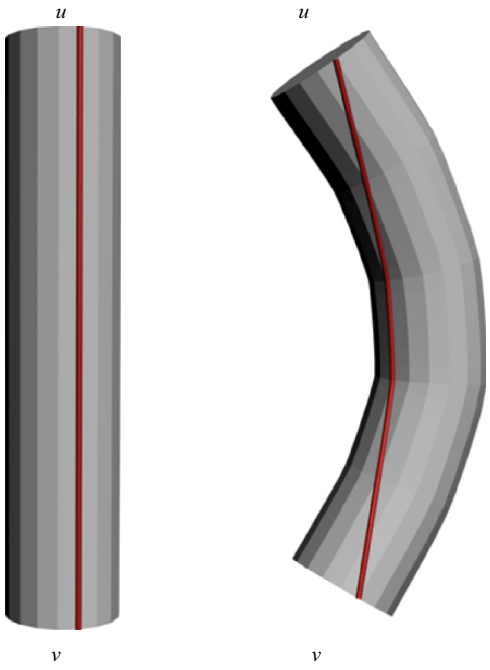


**Fig. 5.** The geodesic path as well as its distance between $u$ and $v$ change with the mesh deformation, from 100 to 90.7.

**Computation of the feature point coordinates using inverse distance weighting (IDW).** Let $g_i$ be a vertex of the minimal graph on the source mesh and $\widehat{g}_i$ its counterpart on the target mesh. We compute the geodesic distance histogram of these two vertices. The geodesic distance histogram $H(g_i)$ describes the distribution of the geodesic distances between the vertex $g_i$ and all the other vertices of the mesh $M_S$. As illustrated in Fig. 6, the histograms $H(g_i)$ and $H(\widehat{g}_i)$ might be dissimilar, although $g_i$ and $\widehat{g}_i$ are the same point on the shape. The main idea is to modify the geodesic distances of $\widehat{g}_i$ such that the histogram $H(\widehat{g}_i)$ becomes similar to $H(g_i)$. This is done by using the inverse distance weighting method and the geodesic distances of the minimal graph vertices.

We define the interpolated geodesic distance $\delta_I(\widehat{v},\widehat{g}_i)$ by means of inverse distance weighting:

$$\delta_I(\widehat{v},\widehat{g}_i)=\sum_{j=1}^{\widehat{n}_M}\frac{w_j(\widehat{v})(g_j,g_i)}{\sum_{k=1}^{\widehat{n}_M}w_k(\widehat{v})}, \tag{5}$$

where

$$w_j(\widehat{v})=\frac{1}{(\widehat{v},\widehat{g}_j)^{p_j}}$$

The value $p_j$ is a positive real number, called the power parameter. $\widehat{n}_M$ is the number of vertices of the minimal graph. Each vertex of the minimal graph is assigned a different power parameter. Greater values of $p_j$ assign greater influence of the vertex $\widehat{g}_j$. The geodesic distance $\delta_I(\widehat{v},\widehat{g}_i)$ is calculated with a weighted average of the geodesic distances between $\widehat{g}_i$ and other feature points on the source mesh. Intuitively speaking, as the vertex $\widehat{v}$ becomes closer to a feature point $\widehat{g}_i$, its interpolated geodesic distance to $\widehat{g}_i$ becomes closer to $\delta(g_j,g_i)$.

Using Eq. (5), we compute the interpolated geodesic distance of $\widehat{g}_i$ to all the other vertices and generate the corresponding histogram $H_I(\widehat{g}_i)$. An important step is to find the power parameters $p_j$ for each minimal graph vertex $\widehat{g}_j$ so as to minimize the difference between $H_I(\widehat{g}_i)$ and $H(g_i)$. We formulate this as a minimization problem where the unknown variables are the power parameters $p_j$ and the cost function is

$$d(H_I(\widehat{g}_i),H(g_i)), \tag{6a}$$

where $d$ is a metric to measure the distance between the two histograms. This minimization problem is computed for each minimal graph vertex $\widehat{g}_i$ separately.

One of the most common histogram metrics is the Earth Mover's distance [19]. In our implementation, we use a different version of the metric as follows. We compute a vector containing all the geodesic distances from $g_i$ sorted in an increasing order. The same vector is calculated for $\widehat{g}_i$. The distance between the two histograms is calculated as the norm of the difference of these two vectors. We assume that the vertex sampling on the source and target meshes is the same and the source and target meshes contain the same number of vertices.

To demonstrate the advantages of using the inverse distance weighting (IDW), we have compared the length of the geodesic paths before and after applying the IDW. Given a source and a target mesh whose correspondence is known, we have measured and compared the change of length of the geodesic paths between all pairs of vertices on the source mesh and their corresponding counterpart on the target. As shown in Fig. 7, the average of length variation and the standard deviation measured on the cat models is 10.91 and 2.87, respectively. After applying the IDW, they have been reduced to 3.9 and 1.11.

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

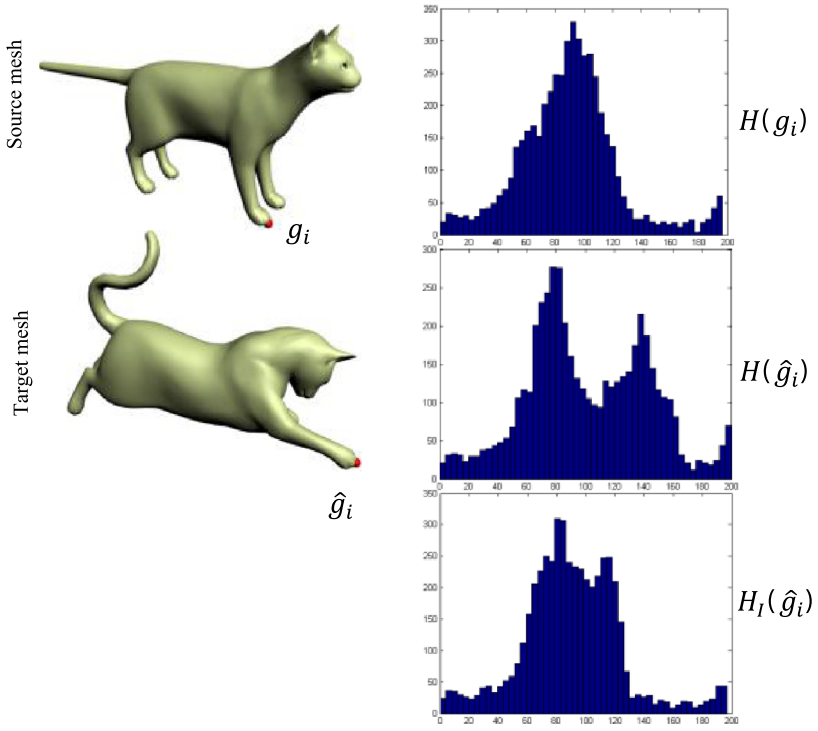


**Fig. 6.** Histograms of the feature point $g_i$ on the source mesh ($H(g_i)$) and the corresponding vertex $\widehat{g}_i$ on the target mesh ($H(\widehat{g}_i)$). $H_I(\widehat{g}_i)$ is generated with the inverse distance weighting.

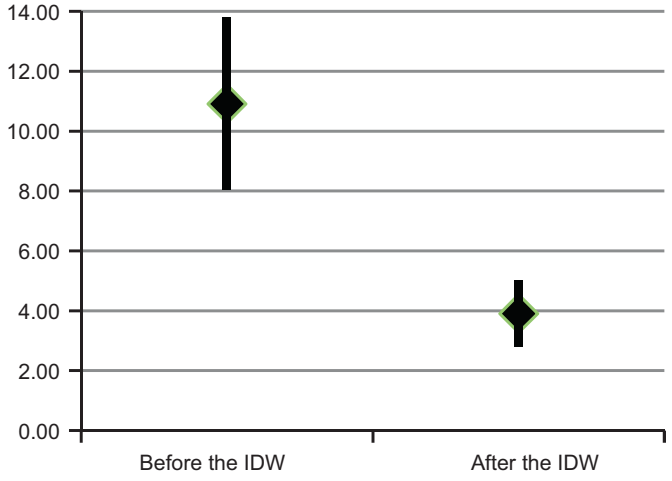


**Fig. 7.** The length variation of corresponding geodesic paths measured on the cat model, before and after applying the inverse distance weighting (IDW).

In Section 7.3, we further demonstrate the significantly improved performance of the landmark transfer with the use of IDW. These results clearly show that the IDW helps reducing the change of the geodesic distances caused by non-isometric deformation.

**Landmark transfer using the FP-coordinates.** Now that we have the interpolated geodesic distances on the target mesh, we proceed to the landmark transfer. Let $v$ be a landmark on the source mesh; the goal is to compute the location of corresponding landmark $\hat{v}$ on the target mesh. Note that $\hat{v}$ is generally not a vertex on the mesh.

(1) For each geometric feature points $\widehat{g}_i$, we determine a set $\mathbf{T_i}$ of all triangles which contain at least one vertex $v_t$ whose geodesic distances $\delta_I(v_t, \widehat{g}_i)$ are in the interval $[1/2\ \delta(v, g_i), 3/2\delta(v, g_i)]$. The set of triangles that are common in all $\mathbf{T_i}$'s $(i=1\ldots\widehat{n}_M)$ are considered for step 2. Needless to say, the process can be accelerated by limiting the subsequent range test for $\widehat{g}_{i+1}$ to those triangles in $\mathbf{T_i}$. The final pruned list of triangles $\mathbf{T}=\cap_{i=1}^{n_M}\mathbf{T_i}$ is used for further processing.

(2) For each triangle $t\in\mathbf{T}$, we compute a vertex $\hat{v}_t$ such that its FP-coordinates are as close as possible to those of the landmark $v$ on the source mesh. The FP-coordinates of the points inside $t$ are interpolated from the FP-coordinates of the vertices of $t$ using the barycentric coordinates. Let $v_{t1}$, $v_{t2}$ and $v_{t3}$ be the three vertices of the triangle $t$ and $w_c$, $c=1,0.3$ the barycentric coordinates of $\hat{v}_t$ that need to be determined. The barycentric coordinates of $\hat{v}_t$ such that its FP-coordinates are as close as possible to those of $v$ are given by

$$\begin{pmatrix} \delta_I(v_{t1},\widehat{g}_1) & \delta_I(v_{t2},\widehat{g}_1) & \delta_I(v_{t3},\widehat{g}_1) \\ \vdots & \vdots & \vdots \\ \delta_I(v_{t1},\widehat{g}_{\widehat{n}_M}) & \delta_I(v_{t3},\widehat{g}_{\widehat{n}_M}) & \delta_I(v_{t3},\widehat{g}_{\widehat{n}_M}) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} \delta(v,g_1) \\ \vdots \\ \delta(v,g_{\widehat{n}_M}) \end{pmatrix}$$

$$\text{with for } c=1\ldots3 \text{ and } \sum_{c=1}^{3} w_c = 1 \tag{6b}$$

If the number of feature points are three ($\widehat{n}_M=3$), we can compute the exact solution to the above equation. When $\widehat{n}_M>3$, $w_c$'s are determined by taking the least square solution.

(3) Finally, for each triangle $t\in\mathbf{T}$, we compute$(\hat{v}_t)$, the FP-coordinates of $\hat{v}_t$ by using the $w_c$'s we computed from (6b) and choose the one that minimizes the distance error as defined by

$$\hat{v}=\underset{\hat{v}_t}{\operatorname{argmin}}_{\hat{v}_t}\left\|\delta(\hat{v}_t)-\begin{pmatrix}\delta(v,g_1)\\ \vdots \\ \delta(v,g_{\widehat{n}_M})\end{pmatrix}\right\| \tag{7}$$

## 7. Results

In this section, we present results of experiments with the method described in the preceding sections. We implemented our algorithm using Matlab. All measurements were made on a Windows 7 Professional machine with 16 GB of memory and Intel Core i7–2600 processor running at 3.40 GHz. We tested our method on the models from the Non-Rigid World Benchmark and High-resolution Tosca data set [20], as well as on our own synthetic embossed plate models. Fig. 13 shows some of the results we obtained. Note that these models exhibit deformations which are only approximate isometries; also each family of objects have a mesh in an initial posture, which is labeled as a mesh in a rest posture (or source mesh). The landmarks have been chosen to be a set of 100 points evenly spaced over the surface using Poisson sampling. In order to avoid repetitive computation of the graph matching, the minimal graph is set to the full graph, so that, the graph matching is computed only once for the transfer of the 100 points. Compared to the minimal graph method, the computation time is about 2 to 3 times higher. We conducted a series of experiments and applied our algorithm to all the vertices in Poisson sampling sets. The quality of transfer is compared with the ground-truth correspondences from high-resolution Tosca models (Fig. 8), and with correspondences computed with existing methods (see Section 7.4).

### 7.1. Timing

In Table 1 an average computation time is shown, which was measured while transferring each of the landmarks; the time was measured and then averaged over landmark transfer to 10 cat models, 6 centaur models, 8 dog models, and 3 embossed plate models.

We clearly see that the computation of the updated geodesic distances (see Section 6.1) is the most time-consuming task. However, an update of the geodesic distances according to the histograms (see Fig. 6) is required only once per each target mesh. That is, it does not matter how many user landmarks are to be transferred from the source to the target (e.g. 10 or $10^3$), the geodesics are updated only once. i.e. it makes our technique very efficient when working with a number of landmarks. For example, once the geodesic distances are updated, it takes only 256 ms to transfer a landmark on the cat model (Table 1).

As shown in Table 1, the overall time of landmark transfer is just a matter of seconds; except for the centaur model. On the centaur our method shows rather a high computation time. The main reason is that this mesh has many more geometric feature points compared to other models (e.g. 15 feature points for the centaur vs. only 8 for the cat). Since $t_H$ is a function of a number of vertices and geometric features, our technique works fast on the embossed plate, which also has many geometric features, but the number of vertices is much less compared to the centaur model. In general, the computation time is an advantage of our method. It takes about 1 minute to find full correspondence for the cat model (on Matlab platform). This has been possible because (1) the update of the geodesic distances has been made only once, and (2) full graph has been used in place of minimal graph for every vertex.

### 7.2. Robustness

In order to measure the quality of the results of landmark transfer, we perform cross-validations by using the Tosca high-resolution dataset as ground truth. We applied our techique to a mixed data set of different subjects in different postures. Our test cases evaluate the maximum and mean errors while transferring 100 Poisson sampled landmarks: on the cat models (Fig. 8(a)), for the Centaur models (Fig. 8(b)). The error of landmark transfer is measured as a geodesic distance deviation from the corresponding ground truth, further normalized by square root of the mesh surface area. Average error values for all data sets are evaluated as shown in Fig. 8(c). Due to imperfect isometries, the error values vary from one posture to another (clearly visible in Fig. 8(a, b)). (Be reminded that these models are only nearly isometric. For example, some of the cat postures show up to 30% of the geodesic distance change with respect to the rest posture. With the embossed plate, maximum of 40% of the geodesic changes can be observed.)

As shown in Fig. 14, our method shows good quality of results on all the models. We can clearly see that the quality of landmark transfer depends on the landmark location with respect to the nodes of the minimal graph and degree of deformation in its neighborhood. In general, the best performance is obtained if the landmark location is close to the nodes of the minimal graph (tips of the limbs, tips of the breast). On the other hand, in the regions of highly non-isometric deformation the quality of transfer degrades (rear part and joints of humans, joints of animals). Note that we obtain a good quality of match on the embossed plate (Fig. 14(j)) despite its high degree of non-isometric deformation. This is especially true on the top center part, which is contributable to the fact that landmarks are well-surrounded by many geometric feature points. Bottom part of the plate lacks feature points, which explains higher errors on it.

Minimal graph plays one of the central roles in our landmark transfer algorithm; and naturally, the quality of transfer is correlated to the selection of $G_M$. In Fig. 9 (a, b) is shown a case when the user landmark was picked at the base of a human neck, which is close to the geometric feature points on the head, breast and hands. With such settings our minimal graph construction algorithm gives a compact $G_M$. On the other hand, when the user landmark is located far away from geometric feature points, as in Fig. 9 (c, d), $G_M$ turns out to be 'large' in terms of the geodesic distances of the edges. A 'small' minimal graph is preferred in our algorithm. First reason for this is that the shorter the graph edges are, the less distortion and error is introduced for corresponding graph on the other isometric mesh. Second, when $G_M$ is small, less geodesic computations are needed to find corresponding one on the target. Note that by configuring a maximum size and number of isocurves of the local shape descriptor, we can achieve detection of different number of feature points, according to our needs and mesh complexity. In our experiment, we were able to extract from just a few to dozens of feature points on the same human model.

**Experiments with genus-one model.** We have tested landmark correspondences between two genus-one surfaces. Embossed plate model with a hole has been used, which has been synthetically generated and deformed. Our algorithm has shown a good accuracy in such settings as well. (Fig. 10). The average geodesic error with respect to ground truth is 0.009, and the maximum error 0.028.

### 7.3. Inverse distance weighting scheme

As expected, the inverse distance weighting greatly improves the quality of the landmark transfer. In Fig. 11 color maps of the landmark transfer are presented for the cat and horse models. For the cat model, the maximum matching error is 0.17 without the IDW; however, when using the IDW it has been reduced to 0.08. For the horse model, maximum error has been reduced from 0.14 to 0.07 respectively. For these models, the matching quality is roughly two times better when using the IDW. More detailed comparison of the landmark transfer with the IDW and without

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

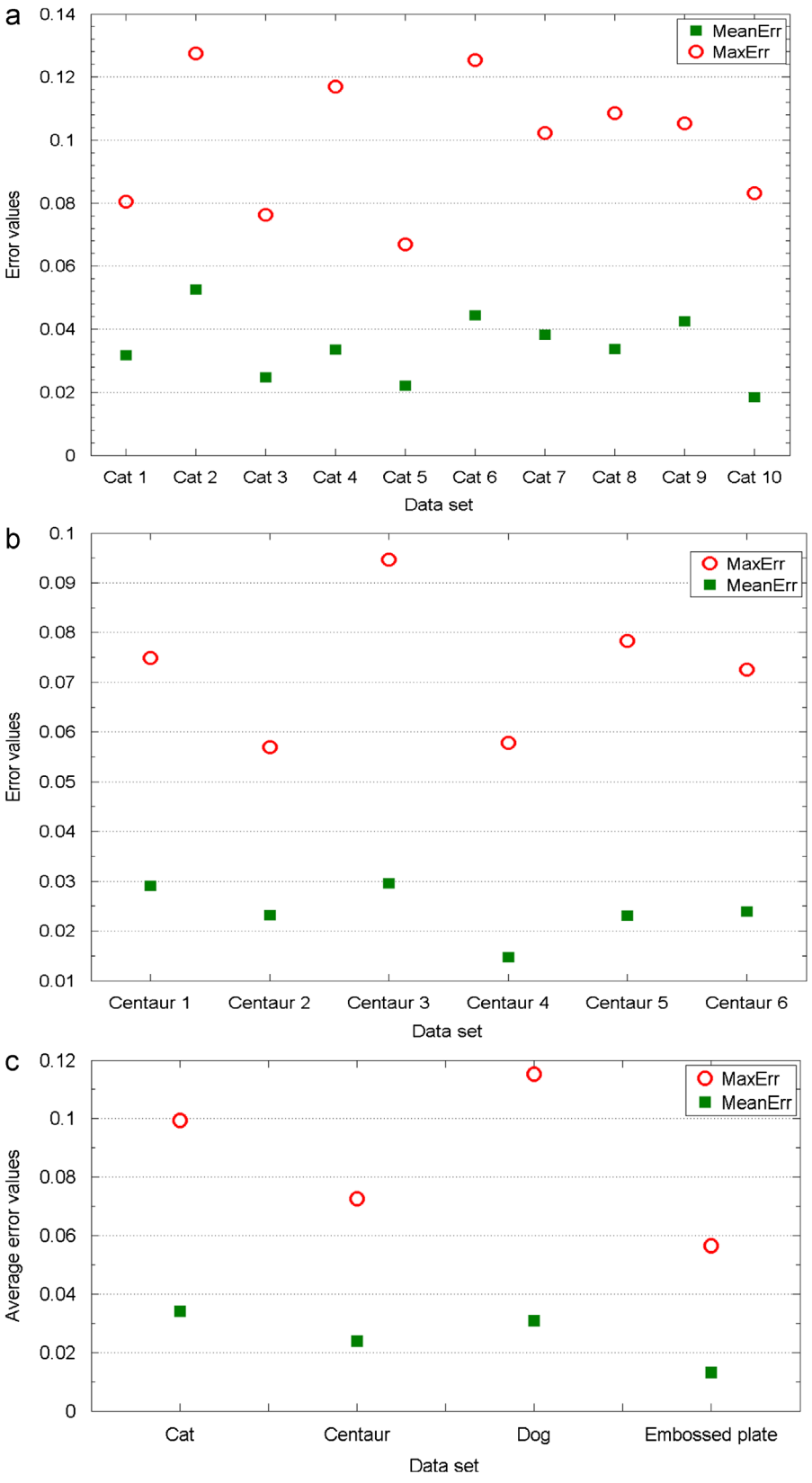


**Fig. 8.** Quality of landmark transfer with respect to the ground truth. We report mean and maximum error values for the (a) cat model and (b) centaur model. (c) Average mean and maximum error values are calculated for each data set.

the IDW is shown in Fig. 12, where the error plots of two landmark transfer with minimal graph (LTMG) implementations are shown. LTMG without the IDW yields approximately 50% correspondences exceeding the error value of 0.05, whereas with LTMG (with the IDW) this number is only 10% (which is better than the result obtained by Blended Intrinsic Maps [13] for the same error).

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

**Table 1**
Average computation time. $|X|$ is the number of vertices, and $|T|$ is the number of triangles in a mesh, $t_\delta$ is the average time needed to compute the geodesic distance matrix per each vertex, $t_{V_F}$ is the average time needed to extract feature points, $t_{G_M}$ is the average time needed to build a minimal graph, $t_H$ is the average time for updating the geodesic distances; hat over a symbol refers to it the target mesh.

| Data set | $\|X\|$ | $\|T\|$ | $t_\delta$(ms) | $t_{V_F}$(ms) | $t_{G_M}$(ms) | $t_{\widehat{V}_F}+t_{\widehat{G}_M}$(ms) | $t_H$(s) |
|---|---|---|---|---|---|---|---|
| Cat | 4994 | 9977 | 146 | 21 | 39 | 59 | 6.63 |
| Centaur | 5002 | 10,000 | 86 | 23 | 128 | 338 | 48.96 |
| Dog | 5000 | 9991 | 104 | 20 | 33 | 39 | 5.98 |
| Embossed plate | 1482 | 2960 | 19 | 7 | 85 | 1708 | 2.81 |

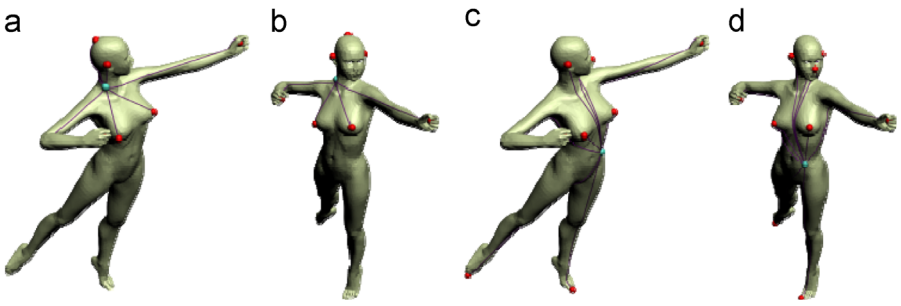


**Fig. 9.** Impact of the landmark location on the quality of transfer and performance. (a), (b) Landmark on the neck is closely surrounded by feature points. (c), (d) Landmark in the belly area has relatively large average distance to the feature points of the surface.

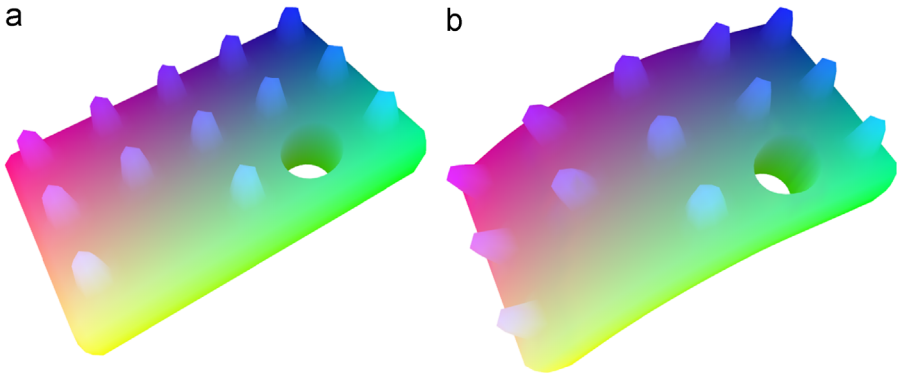


**Fig. 10.** Performance of our method on a genus-one plate model. (a) Each vertex in the source model is assigned with a color that corresponds to its position in a 3D color space. (b) On the target, the transferred locations are colored the same as their source vertices. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 7.4. Comparison with existing methods

Our technique is tailored for the fast transfer of a sparse set of user-defined landmarks on a one-by-one basis. This makes it a bit difficult to perform fair comparisons with existing methods, which mostly focus on computing global optimal solution to the full correspondence. Nevertheless, we used our landmark transfer technique for the full correspondence, in order to make quantitative comparisons tractable. We note however that our landmark transfer finds the corresponding location independently for each vertex, and therefore it is not competitive in terms of computation time when it comes to the full correspondence problem.

For the comparison we have chosen two state-of-the-art techniques—Blended Intrinsic Maps (BIM) [13] and Möbius Voting (MOB) [12]. Within each family of objects from the full Tosca dataset (11 cats, 9 dogs, 3 wolves, 8 horses, 6 centaurs, 4 gorillas, 39 human figures including one female and two male subjects), we arbitrarily selected a model as a source and computed full correspondences to the rest of the models using LTMG by treating each vertex on the source mesh as a landmark. The results of BIM benchmark and Möbius Voting were referred and reproduced, as presented in [13]. Comparative study of correspondence errors is illustrated in Fig. 12. Overall, LTMG and BIM show better accuracy than the Möbius Voting. LTMG shows comparable accuracy to Blended Maps. Compared to LTMG, BIM produces slightly larger number of correspondences in the error range of less than 0.03. However, in contrast to BIM, LTMG gives less outliers with errors higher than 0.04. Additionally, LTMG's plot is noticeably steeper and converges quickly to 100% of correspondences at the error value of 0.14 on the Tosca data set. On the contrary, BIM reaches 100% of correspondences only at the error value 0.25 on the same set.

**Comparison with PLANSAC.** We also compared our method to PLANSAC [10] and its predecessor RANSAC [9]. Provided by [10] the average error score $E(f)$ for the centaur model computed with PLANSAC and RANSAC methods is 0.032 and 0.113 respectively. With the LTMG method applied to the same data, we observe an average matching error of 0.027. This makes our method comparable to PLANSAC matching method, and actually better than RANSAC.

Note that Tevs and co-workers [10] run their algorithm on Poisson sampled centaur model and provide accuracy results in $\varepsilon$-units, where $\varepsilon$ is the minimum distance between two points in the discretization. Unfortunately the value of $\varepsilon$ is not provided.

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

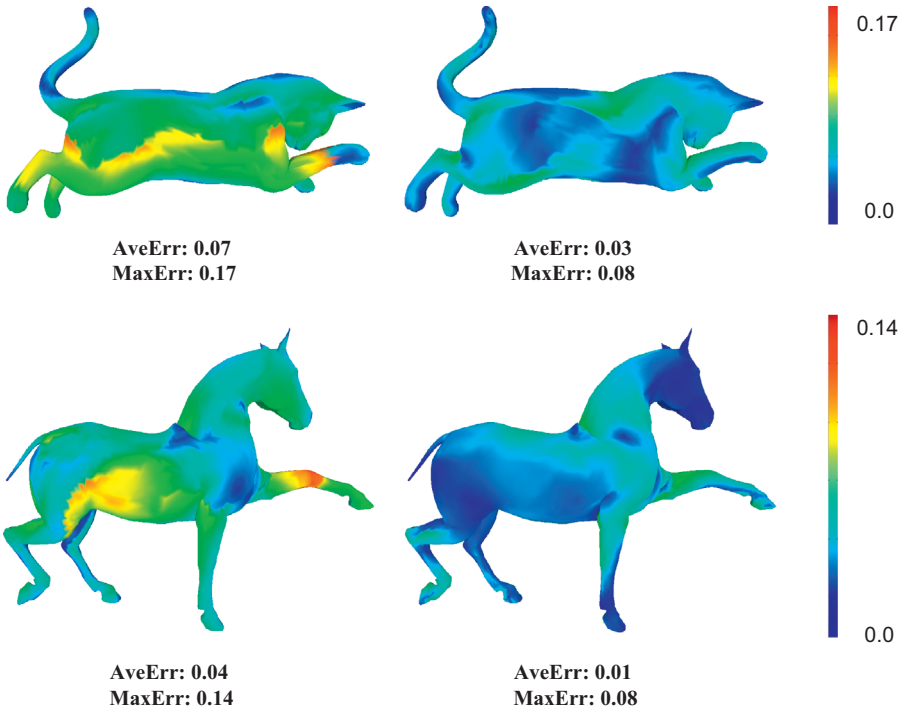


**Fig. 11.** Quality of landmark transfer without (left column) and with the IDW(right column) for the Cat1 (first row), and Horse7 (second row).

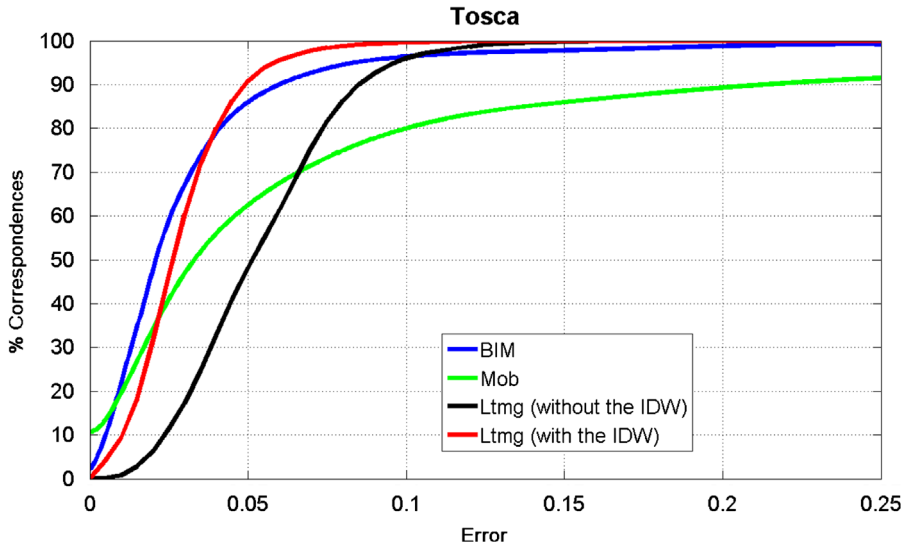


**Fig. 12.** Error plots from the BIM, MOB, LTMG (without IDW), LTMG (with IDW) methods on the full Tosca dataset. The *x*-axis represents the geodesic error and the *y*-axis the percentage of correspondences within the error from the ground truth. The IDW (red curve) shows a significant improvement on the performance of LTMG method (black curve). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

That is why, in order to compare error values, we first needed to bring them to a common scale. For this reason, we also applied uniform Poisson sampling to the centaur model and obtained discretization with the maximal number of samples $\sim$1000, as in PLANSAC settings. With this sampling, we have estimated the exact value of $\varepsilon$. Further normalization by the square root of the model's surface area has yielded directly comparable error values.

### 7.5. Limitations

The main limitation of our method is that it relies on the geometric features of the shape. If the shape does not have any prominent geometric points, our method is not applicable (for instance, our technique will not be able to give a result on a sphere model). The reason for this is that our technique relies on a local shape descriptor to extract feature points, which are then used as nodes of full and minimal graphs.

Apart from this, isometry between the source and target meshes is one of the main assumptions used in our landmark transfer. That is rather strong assumption, although it holds valid in many real-world situations of matching 3D scan data. When the meshes come from different objects, this assumption is violated and proposed method may not work well.

Another assumption we used in this work is sparse distribution of landmarks. The algorithm however can be easily extended to handle cases where the landmark set becomes more dense.

**Handling the symmetry**. In the presence of symmetry, the landmark transfer will propose only one of all possible solutions.

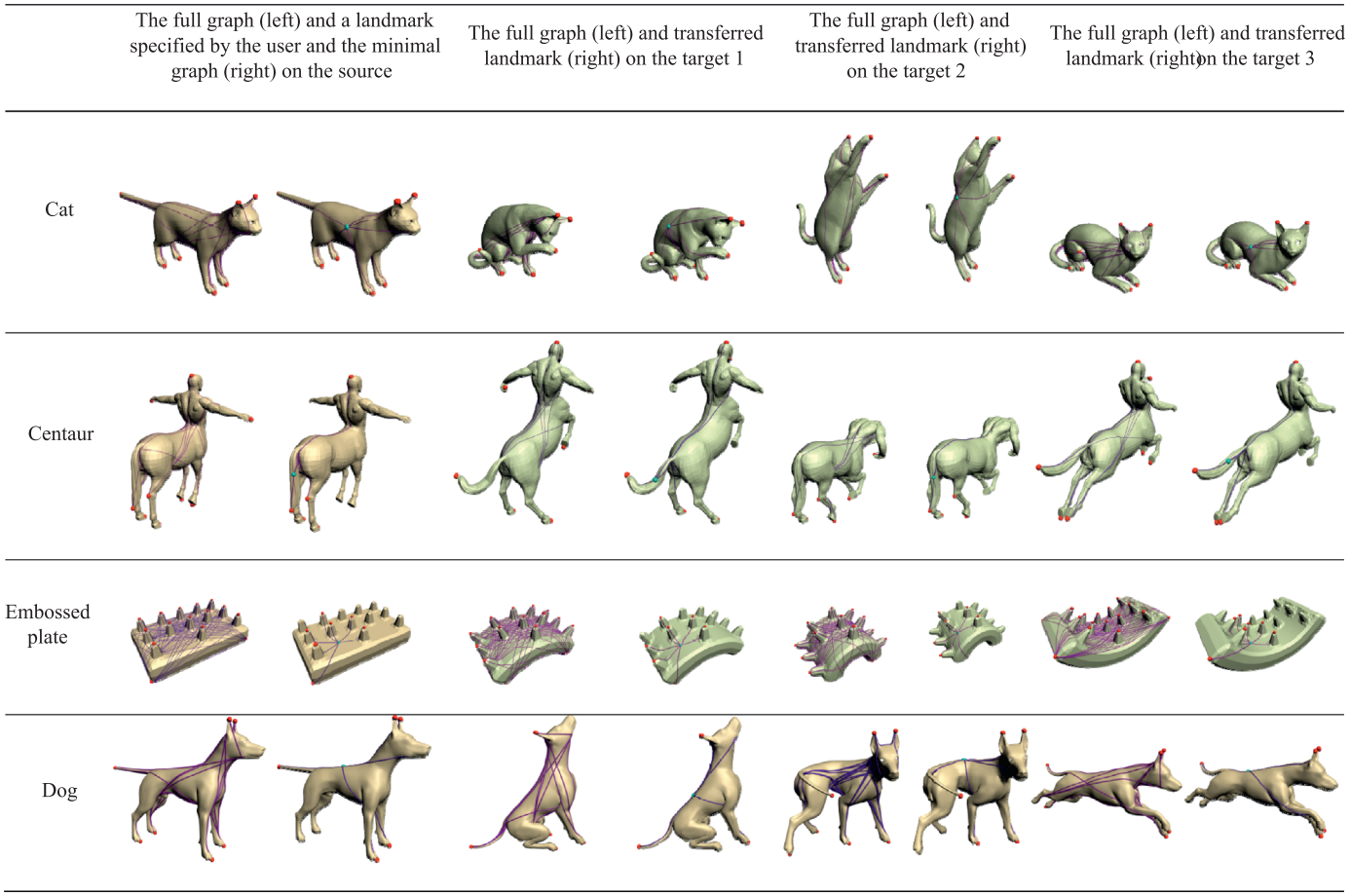


**Fig. 13.** Results obtained from our landmark transfer technique. For each dataset, 3 target postures have been chosen. For each posture, full graph (left) and the minimal (right) is illustrated.

This is related to the uniqueness condition of our minimal graph construction, which favors light computation in graph growing and graph matching, at the cost of permitting matching ambiguity, which is originated from the symmetry. Since our current implementation of minimal graph construction algorithm does not differentiate between the two symmetric minimal graphs given the same (graph-) matching error, sometimes the transferred landmark can be located at the mirror-reflection of the desired location of the transferred landmark. However, it would be easy to extend our method in a way that all possible transfers are proposed to the user. We can simply consider all matching of the minimal graph to the full graph on the target, and compute landmark transfer from each minimal graph. The user will then choose either one or all of them, depending on what s/he wants to have.

Note that in our robustness tests, the matching (landmark position) error has been measured on one half of the meshes, by considering one location and its reflective symmetry as identical.

### 7.6. *Executable*

We provide a sample LTMG application on the Collage authoring environment ([21]). At the moment, the cat data from Tosca is available for the test: one source model ("Experiment Data Item 1") and two target models. In "Experiment Code Item 1" a user can specify the landmark location by typing a vertex index. Running "Experiment Code Item 1" yields "Experiment Data Item 2", which shows the landmark on the source in red color. Given this landmark position, "Experiment Code Item 2" and "Experiment Code Item 3" invoke our algorithm and compute correspondences on target #1 and target #2, respectively. Transferred landmarks are further shown as red dots in "Experiment Data Item 4" (target #1) and "Experiment Data Item 6" (target #2), which are point cloud data in .pcd file format. Note that this implementation on Collage workbench runs in Octave, which is an interpreter and thus executes slower than a compiled version.

## 8. Conclusion

Given one or more custom landmarks on a source mesh, our landmark transfer technique efficiently computes their corresponding locations on target meshes that are approximately isometric. By assisting the user with the reuse of landmarks that have been manually defined once on the source mesh, our technique not only allows the user to define landmarks regardless of geometric distinctiveness, but also to assure consistency among landmarks across a family of meshes, with minimum user input.

Our method is optimally tailored for transferring landmarks that are presumably sparse, since it uses a minimum number of geometric features (i.e. minimal graphs) for each landmark that are necessary to accurately locate the user-defined landmarks and avoids performing unnecessary full registration. In addition, landmark transfer is made more robust thanks to the newly defined geodesic coordinates that makes use of histograms of the geodesic distances. Consequently, our method is ideal for 'one source to multiple target' sparse matching, rather than 'one source to one target' full correspondence. For instance, given a set of user-defined

Please cite this article as: Mykhalchuk V, et al. Landmark transfer with minimal graph. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.04.005

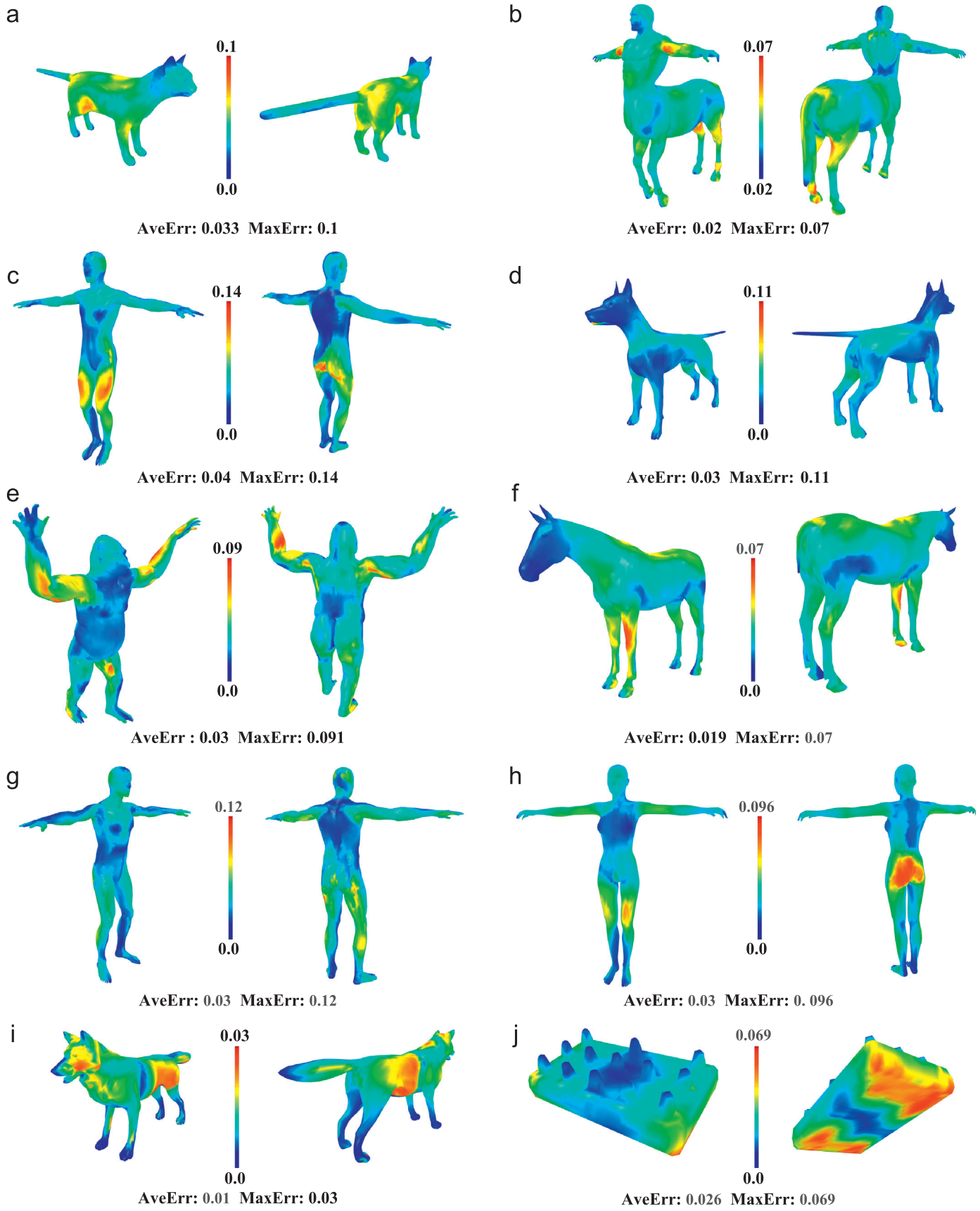


**Fig. 14.** Quality of transferred landmark shown as color map. For each class of Tosca models (a–i) and synthetic embossed plate models (j) we show 2 figures, along with the average and maximum errors. The highest maximum error of landmark transfer could be observed on the models of David (c) (MaxErr: 0.14) and Michael (g) (MaxErr: 0.12). The lowest maximum error was obtained for the Wolf (i) (MaxErr: 0.03) and Embossed plate (j) (MaxErr: 0.069). We can clearly see that the quality of landmark transfer depends on the landmark location with respect to the nodes of the minimal graph and degree of deformation in its neighborhood. In general, the best performance is obtained if the landmark location is close to the nodes of the minimal graph (tips of the limbs, tips of the breast). On the other hand, in the regions of highly non-isometric deformation the quality of transfer degrades (rear part and joints of humans, joints of animals). For the embossed plate (j) we obtain good quality of match on the top center part because landmarks are well-surrounded by the graph nodes. Bottom part of the plate lacks feature points, which explains higher errors on it.

landmarks our algorithm can find precise correspondences on a set of multiple meshes in a matter of seconds.

Although our work is indented for transferring a sparse set of landmarks, we have tested our method for the full, dense correspondence, in order to compare its robustness to other methods. Results show that our method can robustly transfer landmarks, with comparable time cost.

With small adjustments to the method, our landmark transfer technique can be extended to perform full matching, by considering every vertex on the mesh as 'landmark'. The key element lies in the minimum computation for the construction of minimal graphs and the maximum reuse of them for the transfer computation.

## Acknowledgments

We would like to thank Art Tevs and other authors of PLANSAC paper [10] for providing us with part of their codes, which allowed us to compare our landmark transfer technique to theirs. We also thank Vladmir Kim for having provided the ground truth correspondences [13] for the Watertight dataset. Paul Montgomery has proofread the paper.

This work has been supported by the French national project SHARED (Shape Analysis and Registration of People Using Dynamic Data, No.10-CHEX-014-01).

## Appendix A. Subgraph matching using Ullmann's Algorithm

Our subgraph matching method is similar in spirit to Ullmann's algorithm [17]. In the literature the subgraph matching problem is often called subgraph isomorphism. Given the source $G_S$ and target $G_T$ graphs, isomorphism is simply defined as a pair of injective mappings $f=(\alpha,\beta)$ between vertices and edges of $G_S$ and$G_T$. In Ullmann's algorithm subgraph matching is determined by a tree-search enumeration [17], i.e. by systematic generation of all possible matches between $G_S$ and $G_T$. Consider the source and the target graphs $G_S=(V_S,E_S,\vartheta_S)$, $G_T=(V_T,E_T,\vartheta_T)$, $n=|V_S|$, $m=|V_T|$, where and $\vartheta$ are node labels, $A$ and $B$ are weighted matrices of adjacency. A subgraph matching can be formally represented as a permutation matrix $M=[m_{ij}]$, $m_{ij}\in\{0,1\}, i=1\ldots n$, $j=1\ldots m$. If the value of $m_{ij}$ is equal to 1, it means that $i$th vertex of $G_S$ is mapped to $j$th vertex of $G_T$. Two sequential left-multiplications of $B$ by $M$, $M(M\cdot B)^T$ modify the target's graph adjacency matrix accordingly to the permutation matrix, in other words, the target's vertices are permutated according to $M$. Given $C\equiv M(M\cdot B)^T$, permutation matrix$M$ defines a subgraph isomorphism of $G_S$ to $G_T$, if $\forall i,j: A_{ij}\geq 0 => C_{ij}=A_{ij}$.

The valid permutation $M$ has a following set of properties:

- binary: $M$ contains only 0 and 1;
- injection: exactly one 1 in each row, and not more than one 1 in each column;

In order to support partial matches (i.e. only a subset of the source's vertices is mapped on the target) we modify the property set by removing the injection property and substituting it with a weaker condition: not more than one 1 in each row and column.

We use an iterative approach to find a permutation $M$ which corresponds to valid subgraph matching. First, we initialize the permutation $M^0$ with all ones (all permutations are possible). Then, we prevent the mapping of the source vertex to the target vertex which has a smaller degree:

$$M^0_{ij}=\begin{cases}1, \deg(v^j_T)\geq deg(v^i_S)\\ 0, \text{ otherwise}\end{cases},$$

where $\deg(v)$ denotes a degree of a vertex.

When the initialization is done, we generate systematically all valid permutation matrices $M^d$ by means of a depth-first tree search. $M^0$ is located in the root of a search tree; the tree node at level $l$ is binded with a partial permutation matrix, which maps precisely first $l$ vertices from $G_S$ to $G_T$. For each next $M^{d+1}$ we select a matching for $(l+1)th$ vertex from the source and check whether the weights of the new matching pair of nodes are consistent. If corresponding weights are within a user-defined error threshold, we continue going down the search tree. If the weight constraints are violated, we prune the search branch and come back to the parent search node. When the generation of permutations is done, as an output $\{M_1,\ldots,M_k\}$ we have a set of valid isomorphisms and a set of partial isomorphisms between $G_S$and $G_T$; or in case when there is no valid isomorphism, the output is an empty set $\varnothing$.

## References

[1] Robinette K, Boehmer M, Burnsides D. 3-D landmark detection and identification in the CAESAR project. In: Proceedings of the 3rd international conference on 3-D digital imaging and modeling; 2001. p. 393–8.
[2] ISO/IEC FCD 19774—humanoid animation, feature points for the human body. ⟨http://h-anim.org/Specifications/H-Anim200x/ISO_IEC_FCD_19774/FeaturePoints.html⟩.
[3] Zhang E, Mischaikow K, Turk G. Feature-based surface parameterization and texture mapping. ACM Trans Graph 2005;24(1):1–27.
[4] Zhou Y, Huang Z. Decomposing polygon meshes by means of critical points. MMM 2004:187–95.
[5] Seo H, Magnenat-Thalmann N. An automatic modeling of human bodies from sizing parameters. In: Proceedings of ACM symposium on interactive 3D graphics; 2003. p. 19–26.
[6] Allen B, Curless B, Popović Z. The space of human body shapes: reconstruction and parameterization from range scans. Proc ACM SIGGRAPH 2003:587–94.
[7] Chang W, Zwicker M. Automatic registration for articulated shapes. Comput Graphics Forum (Proc. SGP 2008) 2008:1459–68.
[8] Huang Q-X, Adams B, Wicke M, Guibas LJ. Non-rigid registration under isometric deformations. In: Proceedings of the symposium on geometry processing; 2008. p. 1449–57.
[9] Tevs A, Bokeloh M, Wand M, Schilling A, Seidel H-P. Isometric registration of ambiguous and partial data. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR '09); 2009.
[10] Tevs A, Berner A, Wand M, Ihrke I, Seidel H-P. Intrinsic shape matching by planned landmark sampling. Eurographics 2011.
[11] Ovsjanikov M, Mérigot Q, Mémoli F, Guibas L. One point isometric matching with the heat kernel, computer graphics forum. In: Proceedings of the Symposium on Geometry Processing; 2010.
[12] Lipman Y, Funkhouser T. Möbius voting for surface correspondence. ACM Transaction on graphics (Proc ACM SIGGRAPH); 2009.
[13] Kim VG, Lipman Y, Funkhouser T. Blended intrinsic maps. ACM Transaction on Graphics (Proc ACM SIGGRAPH); 2011.
[14] Wang C, Wang Y, Gu X, Samaras D, Paragios N. Dense non-rigid surface registration using high-order graph matching. Proc IEEE Comput Vision Pattern Recognition (CVPR) 2010:382–9.
[15] Berner A, Wand M, Mitra NJ, Mewes D, Seidel H-P. Shape analysis with subspace symmetries. Comput Graphics Forum (Eurographics) 2011;2(30).
[16] Mitchell JSB, Mount D-M, Papadimitriou C-H. The discrete geodesic problem. SIAM J Comput 1987;16(4):647–68.
[17] Ullmann JR. An algorithm for subgraph isomorphism. J Assoc Comput Mach 1976;23:31–42.
[18] Bosea P, Maheshwaria A, Shub C, Wuhrer S. A survey of geodesic paths on 3D surfaces. Comput Geometry—Theory Appl 2011;44(9):486–98.
[19] Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. Int J Comput Vision 2000;40(2):99–121.
[20] Tosca dataset, ⟨http://tosca.cs.technion.ac.il/book/resources_data.html⟩.
[21] Collage Authoring Environment, ⟨https://collage.elsevier.com/⟩.