Rigid registration of different poses of animated shapes

Marco Livesu
Dipartimento di
Matematica e Informatica
University of Cagliari
via Ospedale, 72
09124 – Cagliari, Italy
marco.livesu@unica.it

Riccardo Scateni
Dipartimento di
Matematica e Informatica
University of Cagliari
via Ospedale, 72
09124 – Cagliari, Italy
riccardo@unica.it

ABSTRACT

Different poses of 3D models are very often given in different positions and orientations in space. Since most of the computer graphics algorithms do not satisfy geometric invariance, it is very important to bring shapes into a canonical coordinate frame before any processing. In this paper we consider the problem of finding the best alignment between two or more different poses of the same object represented by triangle meshes sharing the same connectivity. Firstly, we developed a method to select a region of interest (ROI) which has a perfect alignment over the two poses (up to a rigid movement). Secondary, we solved a simplified version of the Largest Common Point-set (LCP) problem with a-priori knowledge about point correspondence, in order to align the ROIs. We eventually align the poses performing least square rigid registration. Our method makes no assumption about the starting positions of the objects and can also be used with more than two poses at once. It is fast, non-iterative, easy to reproduce and brings the poses into the best alignment whatever the initial positions are.

Keywords

Pose registration; Mesh alignment; Numerical methods.

1 INTRODUCTION

Objects are often given in arbitrary position, orientation and scale in space. *Registration* is the process of finding the geometric transformation which brings different sets of data into a congruent coordinate system.

Registration find its application in several fields, like medicine, where data acquired with different modalities (CT, MRI) have to be aligned for joint analysis [31] [33] [20]; image mosaicking [24], where more images have to merged seamlessly to produce a single, wider, image; creation of super-resolution images [9], where many different images of the same scene contribute to compose a super-resolution representation; computer vision [23], visualization [8], segmentation [10], object recognition, shape matching and retrieval [12], and so on.

In this work we focus our attention on a particular aspect of registration: our goal consists in finding the best alignment between two triangle meshes having the same connectivity and representing the same object in different poses. This is quite important for algorithms that deal with multiple poses, because usually they need them to be aligned. For example, in [25] Marras and colleagues considered a set of poses of the same object as seen from different points of view, in order to find the rigid parts of it and calculate its *motion based segmentation*. As all the computations rely on the silhouettes the alignment of the shapes is fundamental to properly catch the parts involved in the movement and discard the static ones. This is exactly the kind of algorithm we are targeting: it uses multiple poses, it assumes that the

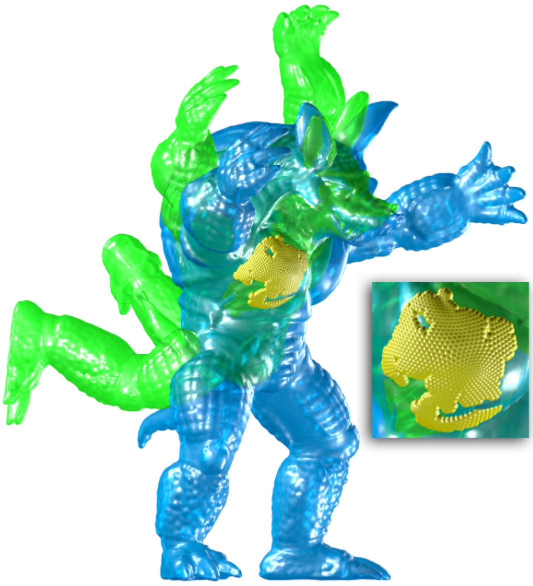


Figure 1: An example of pose registration of two different armadillos, achieved using our method. In the inset the ROI used for registering.

connectivity is invariant over them and it requires them to be aligned.

Registration is also a fundamental building block of many shape interpolation techniques. One can think for example, given two poses, at the problem of generating the complete sequence which describes the movement bringing from one pose to the other [32]; or, given a set of simple poses, at the problem of summarizing them into a more articulated one [32]. Depending on the algorithms used, for all these settings, a pre-alignment could be either mandatory or suggested. Once again, it is quite common in this scenario to have the same connectivity among the models, because probably they have been generated by the same reference triangle mesh.

The main question to answer to is: when are two poses well aligned? Intuitively a good alignment scheme should take into account only the parts of the object which are not involved in the movement suggested by the considered poses; leaving aside all the parts for which, because of the change of pose, a perfect alignment does not exist any longer. This is definitely true, but not sufficient. Think for example at the horses in Figure 2: there are more patches along the surface which remain unchanged over the two poses. Moreover, it is clear from that image, that such patches are likely to ask for a different rigid registration scheme each other.

How can we handle this? One additional consideration must be done. Our final idea about pose registration is the following: *a good rigid pose registration scheme should take into account, among all the patches which are common in the considered poses, only the patch covering the largest area.*

Indeed, there are a lot of good registration algorithms in literature, but none of them is designed to face this particular problem in which we need to align only a subset of primitives and point correspondences are known a priori. Global methods, like the ICP algorithms provided by Besl and McCay in [11] and by Chen and Medioni in [13] are not suitable to accomplish this task; first of all because they are *global*, that is, they use all the available data to estimate the alignment scheme; secondary, because ICP algorithms always converges monotonically to the nearest local minimum of a mean square distance metric. This means that a coarse pre-alignment is mandatory to avoid to fall in the wrong local minima.

In recent years many other registration algorithms have been proposed to the scientific community. In [4] Aiger and colleagues proposed a randomized alignment scheme which is based on approximately coplanar 4-points sets. Their algorithm is fast and resilient to noise. However, we made some tests with the implementation provided in [2], and we found that the results were not satisfactory. The motivation is that such algorithm has been designed to align range maps, which usually does not have a priori correspondences and also require a high percentage of overlapping, say more than 40%. Since poses could be strongly different each other, we need algorithms able to perform well, even in presence of lower overlapping percentages.

Other algorithms are based on global quantities, such as Principal Component Analysis or centroids (see for example the method proposed by Chaouch and Verroust-Blondet in [12]). They are not suitable too. The reason is that a global descriptor computed over a shape bears an arbitrary relationship to the value that would be computed for a different pose of the same shape. It follows that is impossible to align poses employing this kind of techniques.

1.1 Main contribution

Our **main contribution** to solving the problem of rigid pose registration is that we provide a simple method to localize the **greatest surface patch which preserve its appearance over the poses**. Such method is based on the discrete Gaussian curvature, which is one of the most familiar of all local shape descriptors (to have another example of application of the Gaussian curvature one can refer to [19]). The pose registration is eventually achieved by applying a state of the art least square rigid registration algorithm.

The rest of the paper is organized as follows: in Section 2 we briefly recap the mathematical notions over which we built our work. In Section 3 we summarize our proposal. In Sections 4 and 5 we discuss all the technical details of the core of our algorithm. In Section 6 we analyze three different approaches to multiple pose registration. In Section 7 we present and discuss the results we have obtained. In Section 8 we discuss on the limitations our method have, and, finally, in Section 9 we draw the conclusions.

2 MATHEMATICAL BACKGROUND

In this section we summarize the basic notions useful to better understand our approach. After a brief introduction to the differential geometry of surfaces, focusing in particular on discrete Gaussian curvature, we introduce the Largest Common Point-set (LCP) problem along with its most popular metrics.

2.1 Curvature in Differential Geometry

Let S be a $\mathscr{C}^\infty$ surface embedded in $\mathbb{R}^3$: curvatures describe the local bending of the surface. For every point $s \in S$ the two *principal curvatures* k_1 and k_2 are respectively the maximum and the minimum normal curvatures, measured in their orthogonal *principal directions* $\mathbf{e}_1$ and $\mathbf{e}_2$. The *mean curvature* can then be defined as

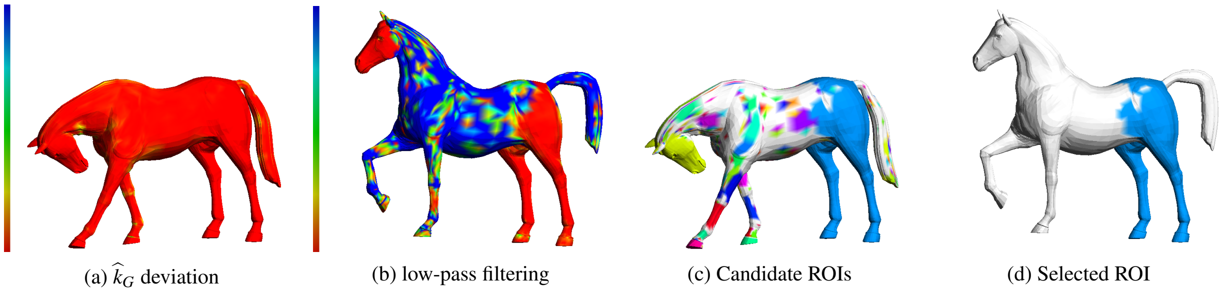

(a) $\widehat{k}_G$ deviation (b) low-pass filtering (c) Candidate ROIs (d) Selected ROI

Figure 2: Region of interest selection. From left to right: Gaussian curvature displacements between the considered poses; Gaussian curvature displacements after the cutting-off of the 20% most affected vertices; all the candidate ROIs and, finally, the ROI selected to align the poses.

$k_H = (k_1 + k_2)/2$. The *Gaussian curvature* k_G, instead, is defined as the product of the two principle curvatures $k_G = k_1 \cdot k_2$. Mean and Gaussian curvatures are among the most important local properties of a surface [16] [34].

Moving from $\mathscr{C}^\infty$ surfaces to meshes (which one can view as $\mathscr{C}^0$ surfaces) the definitions provided above need to be reformulated. In discrete differential geometry the geometric properties of the surface at each vertex are considered as spatial averages around this vertex. Such average is usually restricted to the triangles incident to the vertex itself, which is often referred as the 1-ring or star neighborhood.

Meyer and colleagues [26] provide a discretization of the Gaussian curvature formula; the derived pointwise discrete Gaussian curvature operator is

$$\hat{\kappa}_G(\mathbf{x}_i) = \frac{1}{\mathscr{A}} \left(2\pi - \sum_{j=1}^{\#f} \phi_j \right)$$

where ϕ_j is the angle of the j-th face at the vertex $\mathbf{x}_i$, and $\#f$ denotes the number of faces around this vertex.

As we previously said, the curvature over a mesh vertex is calculated as the spatial average around the vertex itself. The definition of area we use is known as *Voronoi area* and, for a vertex $\mathbf{x}_i$, is defined in function of its neighbors $\mathbf{x}_j$ as

$$\mathscr{A}_{\text{Voronoi}} = \frac{1}{8} \sum_{j \in N(\mathbf{x}_i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \parallel \mathbf{x}_i - \mathbf{x}_j \parallel ,$$

where α_{ij} and β_{ij} are the angles measured in the opposite corners with respect to the edge joining $\mathbf{x}_i$ and $\mathbf{x}_j$. This area proved to be sufficient in all our experiments. However, our approach does not depend on the particular choice of $\mathscr{A}$. Mixed area, for instance, would accomodate obtuse triangles and its application is straightforward [26].

2.2 The LCP problem

The problem of finding the largest set of points which is congruent to a subset of each input data is well known in literature under the name of *Largest Common Point-set problem* (LCP) [6] [7]. It has been studied both from the theoretical and practical points of view and it is widely used in different fields like computational biology and chemistry (there is a frequent need to extract a common pattern from multiple data), and surface reconstruction from 3D scans (different range maps partially overlapped have to be aligned before extracting the final surface [4]).

Given two point sets A and B, the LCP between A and B is the maximal subset $A' \subseteq A$ which is geometrically congruent to some subset B' of B. This implies that two subsets are said to match only when the underlying geometric transformation takes each point of A' exactly onto one point of B'. This statement is also known as largest common point set problem with *exact matching metric* [22] [27] [5].

In real applications, congruence between subsets is a restriction too tight, a similarity metric (ε-congruence) is used instead. Two common metrics for quantifying the notion of similarity are the *Hausdorff distance* [14] [15] [21], and the *bottleneck matching metric* [17].

The former is defined as the maximum distance between a point in one set and its nearest neighbor in the other set; the latter seeks a perfect bipartite matching between two equal cardinality point sets such that the maximum distance between any two matched points is minimized.

It is worth noticing that most of the problems we want to solve with the LCP scheme, especially those involving three and higher dimensional point sets, demand a one-to-one matching between two point sets, making the Hausdorff metric ill-posed. This motivates the study of the problem using the bottleneck matching metric.

3 OVERVIEW

Our method can align pairs of meshes representing different poses of the same object. This implies that the

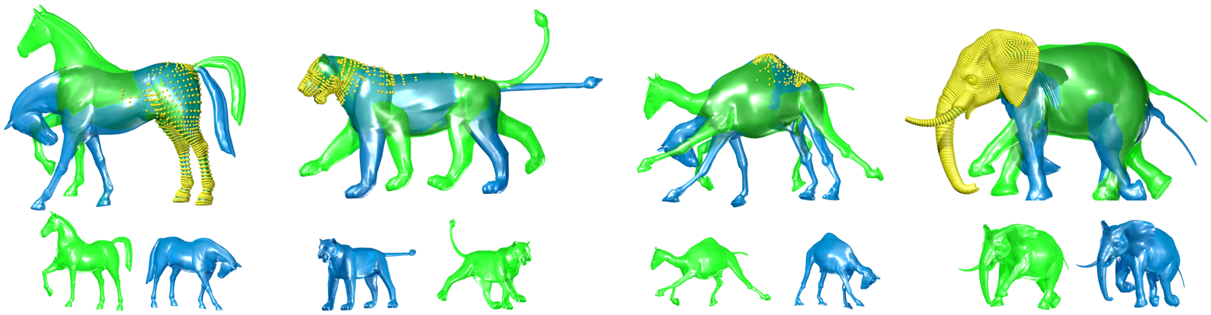


Figure 3: From left to right: mesh registration between two horses, two lions, two camels and two elephants. In the bottom row there are the meshes in the original position and orientation, before registration.

cardinality of the vertices vector is the same, and the two meshes share an identical connectivity.

Let $\mathscr{M} = (V, K)$ and $\mathscr{M}' = (V', K)$ be the meshes we want to align, where K describes the connectivity and $V = \{\mathbf{v}_1, ..., \mathbf{v}_n\}$ and $V' = \{\mathbf{v}'_1, ..., \mathbf{v}'_n\}$ describe the geometric positions of the vertices in $\mathbb{R}^3$. In our approach we take advantage of the particular nature of the problem, setting up a simplified version of the largest common point set problem (LCP) with bottleneck matching metric and a priori knowledge about point correspondences between the vertices of the two shapes. In this simplified framework, once defined the areas we want to overlap, we only have to solve for the isometry which performs such alignment.

We can summarize the algorithm in four steps:

1. Calculation of the Gaussian curvature for each vertex of $\mathscr{M}$ and $\mathscr{M}'$;
2. Selection of the subset of vertices we want to overlap among those having similar Gaussian curvature in both poses (see Section 4);
3. Estimation of the rigid movement which makes $\mathscr{M}$ and $\mathscr{M}'$ ε-congruent (see Section 5);
4. Pose registration achievement by applying the geometric transformation obtained at the previous step.

4 REGION OF INTEREST (ROI)

Let $V_{k_G} \subseteq V$ be the subset of vertices of $\mathscr{M}$ having similar Gaussian curvature in both meshes, up to a tolerance threshold ε

$$V_{k_G} = \left\{ \mathbf{v}_i \in V \;\middle|\; |\,\hat{\kappa}_G(\mathbf{v}_i) - \hat{\kappa}_G(\mathbf{v}'_i)\,| \le \varepsilon \,,\ \mathbf{v}'_i \in V' \right\} .$$

The distribution of the points in $V_{k_G} \subseteq V$ among the surface identifies which parts of the surface have common behaviour in both poses. Starting from this points we perform region growing in order to define a set of candidate regions of interest for the alignment. Let $\mathbf{v}_i$ be a vertex in a candidate ROI and let $N(i)$ be the set of vertices sharing an edge with it (i.e., its one ring). If the average Gaussian curvature deviation

$$\overline{\Delta\kappa_G}(\mathbf{v}_i) = \frac{\sum_{j \in N(i)} |\,\hat{\kappa}_G(\mathbf{v}_j) - \hat{\kappa}_G(\mathbf{v}'_j)\,|}{\#N(i)}$$

is lower than a predefined threshold σ, the candidate ROI expands over $N(i)$. We repeated this process iteratively starting from each seed point until convergence. In Figure 2c the candidate ROIs for two poses of the horse model.

The ROI selection depends on the threshold σ. As one can note from Figure 2a, curvature deviation is ill-distributed over the surface: few vertices have a huge displacement, thus flattening the rest of the distribution. In this scenario the automatic research of a good σ value can be very difficult. We observed that vertices with high curvature deviation are located in the areas involved in the change of pose and, thus, they are meaningless for the registration problem. We therefore decided to restrict the research to a subset of vertices with small curvature deviation. Let $\mathscr{LD}$ be the indexes of the vertices less affected by curvature deviation (80% of the total number of vertices); we eventually compute the σ value as

$$\sigma = 0.2 \ \operatorname*{arg\,max}_{i \in \mathscr{LD}} |\,\hat{\kappa}_G(v_i) - \hat{\kappa}_G(v'_i)\,| .$$

Cutting-off of the vertices with greater deviation makes the error distribution easier to be treated algorithmically (see Figure 2b). All the results shown in this paper have been achieved using the automatic σ calculation described above.

We have now a problem with selecting the best ROI to guide the whole registration process. Our idea is to consider only the widest patch, thus we have to estimate the area of each of them. The discrete Gaussian curvature measured in a vertex of a mesh is the average of the Gaussian curvatures computed in the area $\mathscr{A}$ around it (see section 2). This area becomes now twice important: first because it is involved in the discrete $\hat{\kappa}_G$ formula; second because also gives the possibility to measure the global area of each candidate ROI as the sum of the areas associated to each vertex in it. This is quite important because makes the measure of the patch's size independent from the underlying triangulation we are dealing with.

5 ALIGNMENT

Given a region of interest $\mathscr{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ in the first pose, we want to find the rigid movement that optimally aligns $\mathscr{P}$ and its dual in the second pose $\mathscr{P}'$. The transformation $T : \mathscr{P} \to \mathscr{P}'$ we are looking for has the form

$$T = \begin{pmatrix} R & \mathbf{t} \\ 0^T & 1 \end{pmatrix},$$

where $R \in \mathbb{R}^{3\times3}$ is a pure rotation matrix (i.e. $R^T R = I$) and $\mathbf{t} \in \mathbb{R}^3$ is a translation. If $\mathscr{P}$ and $\mathscr{P}'$ were strictly congruent, we would have $\|\mathbf{p}'_i - T(\mathbf{p}_i)\| = 0$, $i = 1, \dots, n$. Unfortunately, congruence is very unlikely to happen; in the common scenario a perfect alignment does not exists, therefore we should find the best approximating rotation and translation that fit $\mathscr{P}$ and $\mathscr{P}'$ in the least square sense, i.e. minimizes

$$\left(\tilde{R}, \tilde{\mathbf{t}}\right) = \arg\min_{R,\mathbf{t}} \sum_{i=1}^{n} \|\left(R\,\mathbf{p}_i + \mathbf{t}\right) - \mathbf{p}'_i\|^2 . \qquad (1)$$

This is a very well known problem in literature and there are several existing algorithms to solve it. A comparison between four of them can be found in [18] while a weighted instance of the problem has been studied by Sorkine and Alexa in [29].

The best translation can be found by taking the derivative of (1) w.r.t. $\mathbf{t}$ and searching for its roots. One can eventually find that

$$\tilde{\mathbf{t}} = \overline{\mathbf{p}}' - R\,\overline{\mathbf{p}} ,$$

with

$$\overline{\mathbf{p}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i \qquad\qquad \overline{\mathbf{p}}' = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}'_i .$$

In other words the optimal translation $\tilde{\mathbf{t}}$ maps the transformed centroid of $\mathscr{P}$ in the centroid of its dual $\mathscr{P}'$. To find the best rotation we can now consider the centered vectors

$$\mathbf{x}_i = \mathbf{p}_i - \overline{\mathbf{p}}_i \qquad\qquad \mathbf{y}_i = \mathbf{p}'_i - \overline{\mathbf{p}}'_i$$

$i = 1, \dots, n$, and the Singular Value Decomposition (SVD) of their 3×3 covariance matrix XY^T (X and Y are two $3 \times n$ matrices containing all the centered vectors ordered by column)

$$XY^T = U\Sigma V^T$$

The rotation we are looking for is

$$R = V \begin{pmatrix} 1 & & \\ & 1 & \\ & & \det(VU^T) \end{pmatrix} U^T .$$

The last term in the diagonal matrix is setted to $\det\left(VU^T\right)$ instead of 1 in order to avoid reflections. A formal and clear demonstration of the whole minimization process can be found in [28].

6 REGISTRATION OF MULTIPLE POSES

Our method can be also used to align more than two poses at once. Given a set of poses $\mathscr{M}_1, \dots, \mathscr{M}_n$, three different strategies are possible:

1. if we need to select a *reference pose* $\mathscr{M}_{\text{ref}}$, it can be used to align all the other poses;

2. if there is a time-sequence leading to changes in the poses, they can be registered in chain, that is, $\mathscr{M}_1$ vs $\mathscr{M}_2$, $\mathscr{M}_2$ vs $\mathscr{M}_3, \dots, \mathscr{M}_{n-1}$ vs $\mathscr{M}_n$;

3. if no reference pose and time-sequence exists, the poses can be registered all together.

What is the best multi-registration schema? Actually, each strategy has pros and cons: the best choice depends on the applications. To summarize multiple simple poses into an articulated one, the first scheme is likely to be the best one. To add frames at a discrete sequence of poses describing a movement, the second scheme would be better.

7 RESULTS AND DISCUSSION

We implemented our methods in C++, using the VCG Library [3] for the manipulation of geometric data structures, and the Eigen library [1] for the numerical computations. Experiments were run on a iMac equipped with 2.66GHz Intel Core 2 Duo and 4GB RAM, using a single core. We used as dataset the shapes provided by Sumner and Popović in their deformation transfer for triangle meshes's web page [30]. For each shape we considered many different poses: from a minimum of 8 poses (horse), to a maximum of 47 poses (elephant gallop).

Our tests have been organized as follows: for each couple of poses we run our algorithm three times, each time applying a random rigid movement to the shapes, in

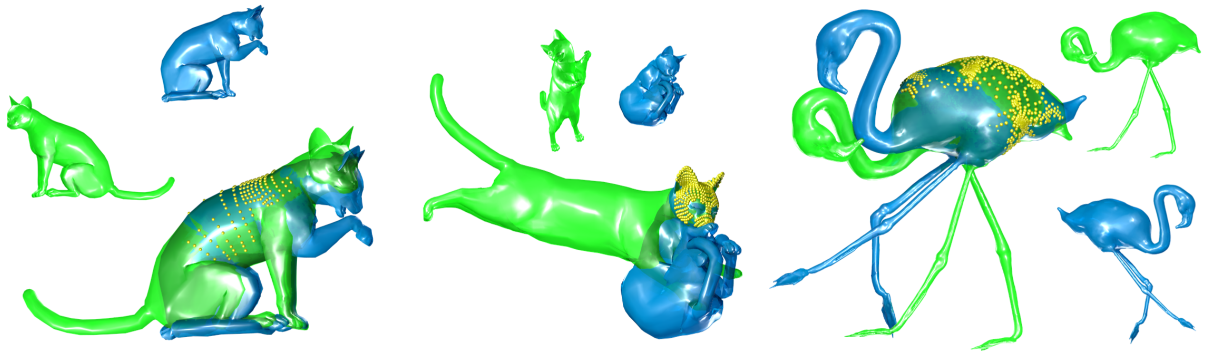


Figure 4: Some examples of registration achieved with our algorithm. In yellow, the subsets of mesh vertices used to align the poses.

order to put them in general position. We had been able to find a good alignment in every test we made (see Figures 3 and 4). In particular, for each couple of poses, we always get the same numerical results, emphasizing the robustness of our shape descriptor (i.e. discrete Gaussian curvature displacements) with respect to isometries. In Table 1 we report timings for each registration shown in this paper. One can note that, since the most time consuming task is the registration step, the complexity of the whole algorithm depends on the size of the ROI rather than the complexity of the mesh (i.e. the number of vertices).

7.1 Multiple poses

For our experiments, we used the horse gallop sequence dataset. Both reference pose and time sequence are available for it, so we had been able to test each possible strategy.

Theoretically, the third method presented in Section 6 is more computationally expensive than the first two: it it intrinsically quadratic, while the others are linear. But, what we actually do in the third method, is the registration of all the poses versus a reference one using a ROI computed on all the meshes. This leads to timing that are comparable each other. As we can see in Table 2, all the methods require the same time for the computation of the curvature and alignment and the last one (all vs all), is definitely the fastest for ROI computation (due to the smaller number of vertices in the ROI). An example of multiple registration is given in Figure 5.

We studied the quality of the registration schemas, measuring the Mean Square Error (MSE) metric over the vertices of the ROI:

$$MSE(\mathscr{M}_a, \mathscr{M}_b) = \frac{\sum_{i \in \mathrm{ROI}_{ab}} \left\| v_i^{(a)} - v_j^{(b)} \right\|_2^2}{\#\mathrm{ROI}_{ab}}.$$

We computed MSE errors over each possible couple of poses, in order to have a reference ROI and a reference error measure. We, then, tested each multi-registration strategy, and measured the registration errors over the, previously determined, reference ROI.

For each pair of horse meshes that we tested, MSE error was very close to zero (Figure 6a), emphasizing the registration capabilities of the method. Multiple pose registration over a reference pose behaves well, however some error peaks between two particular non-reference poses can occur, for example between the third and the eighth horses (Figure 6b). Chain registration suffers the same problem: error is low for most of the poses but there are peaks between first and last poses of the chain (Figure 6c). Finally, as one could expect, we found that the best error distribution over the poses have been achieved considering all the poses together (Figure 6d).

8 LIMITATIONS

The proposed algorithm performs well with all the shapes we tested on. It has, however, some limitations, in the sense that there are no guarantees that the registration provided will be *good* for an end user, in other words, the one the user expected. In particular, if the model contains more parts that move rigidly and cover more or less the same area the choice between these parts as candidate ROI will be unstable. Moreover, in case the selected ROI lies in a peripheral area of the shape (e.g., a foot, a head or a hand) the result could be unnatural. The definition of *natural registration* is

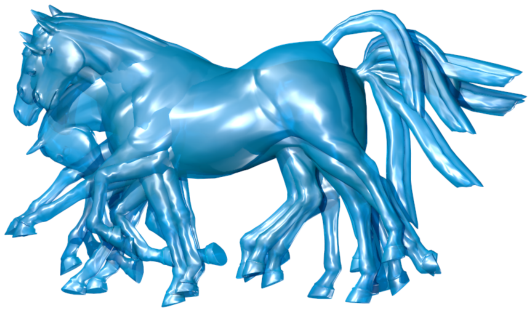
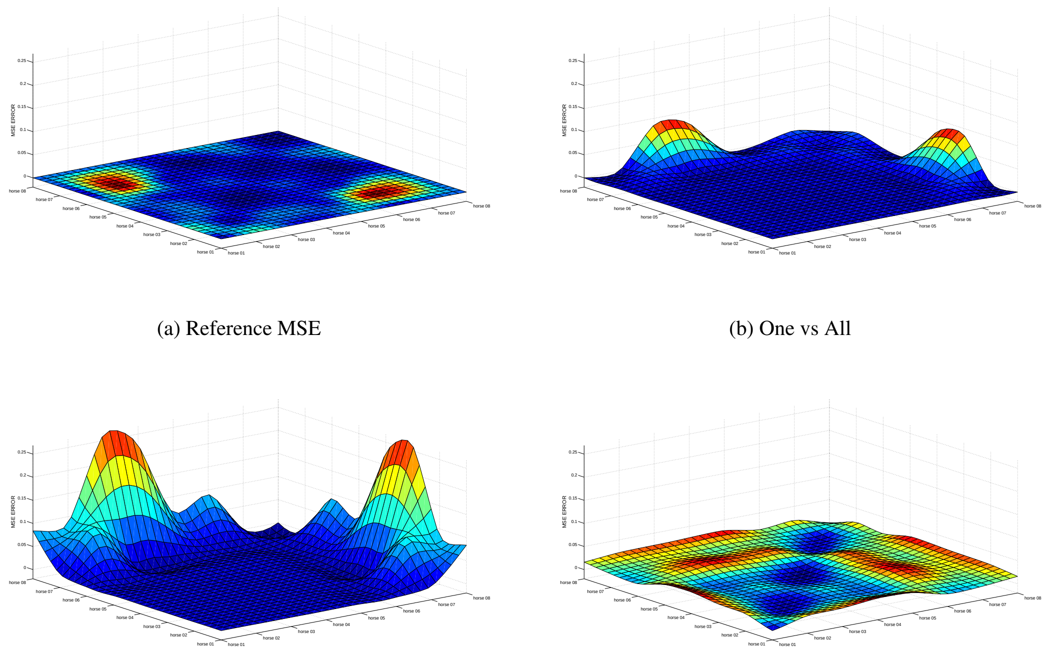


Figure 5: Multiple registrations of eight poses of the Horse mesh.

Model	Vertices	ROI	Curvature (ms)	ROI (ms)	Alignment (ms)	Total (ms)
Armadillos	165,954	3,541	307	126	6	439
Elephants	42,321	19,339	189	98	28	315
Flamingos	26,907	1,092	120	44	10	174
Camels	21,887	376	72	34	25	131
Horses	8,431	2,491	35	27	11	73
Cats 1	7,207	965	32	23	4	59
Cats 2	7,207	544	32	22	2	56
Lions	5,000	1,751	24	15	8	47
Cactus	5,261	410	22	2	3	27

Table 1: Running times of our algorithm, in milliseconds. In the rightmost column there is the total time needed to align the poses while in the previous three columns there are the timings needed, respectively, for Gaussian curvature calculation (Curvature), Region Of Interest determination (ROI), and the least square registration calculation (Alignment). Second and third columns show, for each pair of poses, respectively the number of mesh Vertices (Vertices) and the cardinality of the Region Of Interest (ROI) we used to align.



(a) Reference MSE

(b) One vs All

(c) Chain

(d) All vs All

Figure 6: Mean Square Error plots for multiple pose registration. From left to right, up to down: reference MSE error measured between each possible pair of the sequence (a); MSE error obtained aligning all the poses in the sequence against a reference pose (b); MSE error obtained aligning all the poses in chain (c); and MSE error obtained comparing all the poses together (d).

somehow related to the human perception, it is the kind of registration that a human can imagine when looks at two different poses of the same object. We have tried to set up a mathematical model which works like a human should work in the general case but, unfortunately, natural registrations are as easy to do for humans as incredibly difficult for machines, because is a matter of shape understanding and matching.

One second limitation regards noise. The algorithm is noise insensitive as long as all the models involved in the computations are affected by the same kind and the same amount of noise. In any other case, the Gaussian curvature values would be totally unreliable, leading to wrong registrations. However, this is usually not a big deal: in shape interpolation and modelling one tends to use almost the same meshes, deformed and readapted to the new pose, thus bearing the same amount of noise.

9 CONCLUSIONS

Pose registration is the process of finding the best possible alignment among different meshes representing the same shape in different positions. Since poses can be strongly different each other the best alignment method should be able to work only with local patches of the surface, that is the regions of the shape having the same properties (mainly curvature) in the given poses.

In this paper we presented a new, non-iterative, pose registration scheme, which employs Gaussian curvature as local shape invariant and we show that the proposed method can be fast, robust and accurate in the most relevant cases of pose registration: interpolation among different poses and modelling.

10 ACKNOWLEDGMENTS

The authors would like to thank Robert Sumner and Jovan Popović for making available on their website their 3D shapes, which were useful for our work. We would like also to thank our colleagues, Marianna Saba, Enrico Puppo and Daniele Panozzo, for the fruitful discussions had during this work.

Method	Curvature (ms)	ROI (ms)	Alignment (ms)	Total (ms)
one vs all	160	209	46	415
chain	158	190	55	403
all vs all	160	53	72	285

Table 2: Comparison of the the three multiple poses registration methods.

11 REFERENCES

[1] http://eigen.tuxfamily.org/

[2] http://meshlab.sourceforge.net/

[3] http://vcg.sourceforge.net/

[4] Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. In: ACM SIGGRAPH 2008 papers, SIGGRAPH '08, pp. 85:1–85:10. ACM, New York, NY, USA (2008)

[5] Akutsu, T.: On determining the congruence of point sets in d dimensions. Comput. Geom. Theory Appl. **9**, 247–256 (1998)

[6] Akutsu, T., Halldórsson, M.M.: On the Approximation of Largest Common Subtrees and Largest Common Point Sets. In: Proceedings of the 5th International Symposium on Algorithms and Computation, ISAAC '94, pp. 405–413. Springer-Verlag, London, UK (1994)

[7] Akutsu, T., Tamaki, H., Tokuyama, T.: Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. In: Proceedings of the thirteenth annual symposium on Computational geometry, SCG '97, pp. 314–323. ACM, New York, NY, USA (1997)

[8] Allen, P., Feiner, S., Troccoli, A., Benko, H., Ishak, E., Smith, B.: Seeing into the Past: Creating a 3D Modeling Pipeline for Archaeological Visualization. In: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium, 3DPVT '04, pp. 751–758. IEEE Computer Society, Washington, DC, USA (2004)

[9] Arican, Z., Frossard, P.: Super-resolution from unregistered omnidirectional images. In: International Conference on Pattern Recognition, pp. 1–4 (2008)

[10] Attene, M., Katz, S., Mortara, M., Patane, G., Spagnuolo, M., Tal, A.: Mesh Segmentation - A Comparative Study. In: Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on, p. 7 (2006)

[11] Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**, 239–256 (1992)

[12] Chaouch, M., Verroust-Blondet, A.: A novel method for alignment of 3D models. In: Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on, pp. 187 –195 (2008)

[13] Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image Vision Comput. **10**, 145–155 (1992)

[14] Chew, L.P., Dor, D., Efrat, A., Kedem, K.: Geometric Pattern Matching in *d*-Dimensional Space. Discrete & Computational Geometry **21**, 257–274 (1999)

[15] Chew, L.P., Goodrich, M.T., Huttenlocher, D.P., Kedem, K., Kleinberg, J.M., Kravets, D.: Geometric pattern matching under Euclidean motion. Comput. Geom. Theory Appl. **7**, 113–124 (1997)

[16] Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice Hall (1976)

[17] Efrat, A., Itai, A., Katz, M.J.: Geometry helps in bottleneck matching and related problems. Algorithmica **31**, 1–28 (2001)

[18] Eggert, D.W., Lorusso, A., Fisher, R.B.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. Mach. Vision Appl. **9**, 272–290 (1997)

[19] Guo, K., Li, M.: A Novel Shape Descriptor: Gaussian Curvature Moment Invariants. In: Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering, ICISE ’09, pp. 1087–1090. IEEE Computer Society, Washington, DC, USA (2009)

[20] Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration. Physics in Medicine and Biology **46**(3), R1 (2001)

[21] Huttenlocher, D.P., Kedem, K., Sharir, M.: The upper envelope of Voronoi surfaces and its applications. In: Proceedings of the seventh annual symposium on Computational geometry, SCG ’91, pp. 194–203. ACM, New York, NY, USA (1991)

[22] Irani, S., Raghavan, P.: Combinatorial and experimental results for randomized point matching algorithms. In: Proceedings of the twelfth annual symposium on Computational geometry, SCG ’96, pp. 68–77. ACM, New York, NY, USA (1996)

[23] Jin, H., Favaro, P., Soatto, S.: A semi-direct approach to structure from motion. The Visual Computer **19**, 377–394 (2003)

[24] Kourogi, M., Kurata, T., Hoshino, J., Muraoka, Y.: Real-Time Image Mosaicing from a Video Sequence. In: International Conference on Image Processing, pp. 133–137 (1999)

[25] Marras, S., Bronstein, M.M., Hormann, K., Scateni, R., Scopigno, R.: Motion-based mesh segmentation using augmented silhouettes. Graphical Models **74**(4), 164–172 (2012)

[26] Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In: H.C. Hege, K. Polthier (eds.) Visualization and Mathematics III, pp. 35–57. Springer-Verlag, Heidelberg (2003)

[27] de Rezende, P.J., Lee, D.T.: Point Set pattern matching in d-dimensions. Algorithmica **13**, 387–404 (1995)

[28] Sorkine, O.: Least-Squares Rigid Motion Using SVD. Tech. rep. (2009)

[29] Sorkine, O., Alexa, M.: As-Rigid-As-Possible Surface Modeling. In: A. Belyaev, M. Garland (eds.) SGP07: Eurographics Symposium on Geometry Processing, pp. 109–116. Eurographics Association, Barcelona, Spain (2007)

[30] Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Trans. Graph. **23**(3), 399–405 (2004)

[31] Tao, J.X., Hawes-Ebersole, S., Baldwin, M., Shah, S., Erickson, R.K., Ebersole, J.S.: The accuracy and reliability of 3D CT/MRI co-registration in planning epilepsy surgery. Clinical Neurophysiology **120**, 748–753 (2009)

[32] Winkler, T., Drieseberg, J., Alexa, M., Hormann, K.: Multi-Scale Geometry Interpolation. Computer Graphics Forum **29**(2), 309–318 (2010)

[33] Yang, Z., Liu, L., Liu, W., Yang, W.: Research of CT/MRI Tumor Image Registration Based on Least Square Support Vector Machines. In: International Conference on Intelligent Computing, pp. 1078–1086 (2008)

[34] Yamauchi, H., Lee, S., Lee, Y., Ohtake, Y., Belyaev, A., Seidel, H.P.: Feature Sensitive Mesh Segmentation with Mean Shift. In: Proceedings of the International Conference on Shape Modeling and Applications 2005, pp. 238—245 (2005)