A new 3D-matching method of non-rigid and partially similar models using curve analysis
Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot

To cite this version:

HAL Id: hal-00660801
https://hal.archives-ouvertes.fr/hal-00660801
Submitted on 17 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A new 3D-matching method of non-rigid and partially similar models using curve analysis

Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre and Olivier Colot

Abstract—The 3D-shape matching problem plays a crucial role in many applications such as indexing or modeling by example. Here, we present a novel approach to match 3D-objects in presence of non-rigid transformation and partially similar models. In this paper, we use the representation of surfaces by 3D-curves extracted around feature points. Indeed, surfaces are represented with a collection of closed curves and tools from shape analysis of curves are applied to analyze and to compare curves. The belief functions are used to define a global distance between 3D-objects. The experimental results obtained on the TOSCA and the SHREC07 datasets show that the system efficiently performs in retrieving similar 3D-models.

Index Terms—3D-shape matching, Curves analysis, Belief functions, Feature points.

1 INTRODUCTION

SINCE a few years, there is an increasing interest in analyzing shapes of three-dimensional (3D) objects. Advances in 3D scanning technologies, hardware-accelerated 3D graphics, and related tools, are enabling access to high quality 3D data. As technologies are improving, the need for automated methods for analyzing shapes of 3D-objects is also growing. Shape matching remains a central technical problem for efficient 3D-object search engine design. It plays an important role in many applications such as computer vision, shape recognition, shape retrieval and computer graphics.

In recent years, a large number of papers are concerned in 3D-shape matching. Most of the current 3D-shape matching approaches are based on global similarity. Some methods directly analyze the 3D-mesh — such as Antini et al. [1] who use curvature correlograms — and some others use a 2D-view based approach — such as Filali et al. [2] who propose an adaptive nearest neighbor-like framework to choose the characteristic views of a 3D-model. These approaches enable retrieval of similar objects in spite of rigid transformations. Other approaches are also enabling retrieval of objects that differ from non-rigid transformations such as shape bending or character articulation [3]. They are generally based on the graph or skeleton computation of a 3D-object [4], [5].

Other case of shape matching is encountered when partially similar shapes are presented. This kind of shape matching plays a crucial role in many applications such as indexing or modeling by example. It is usually approached using the recognition by part idea [6], [7]: segmentation of the shape in significant parts, and matching pairs of parts as whole shapes. Bronstein et al. [8] also proposed to consider the 3D-matching problem as a multi-criterion optimization problem trying to simultaneously maximize the similarity and the significance of the matching parts.

In this paper, we present a new approach for 3D-object matching based on curve analysis. An important line of work in face analysis has been to use shapes of curves that lie on facial surfaces [9], [10]. However, these works have been intended for face recognition only and not for shape analysis of generic 3D-surfaces. Our approach consists to represent 3D-objects as a set of parts. Each part is represented by an indexed collection of closed curves in $\mathbb{R}^3$ on the 3D-surface. Then we use shape analysis of curves in the matching process. First of all, given a 3D-object, a set of feature points is extract. Around each feature point, a set of closed curves is automatically computed. One feature point and its associated set of curves is called a 3D-part. Curves are level curves of a surface distance function defined to be the length of the shortest path between that point and the feature point. This function is stable with respect to changes in non-rigid transformations. In this paper, we analyze the shape of 3D-part surfaces by analyzing the shape of their corresponding curves. The belief functions are used to define a global distance between 3D-objects using the distance between their corresponding 3D-parts.

The paper is organized as follows. A feature point extraction algorithm, using geodesics, is introduced in the next section. Section 3 describes our framework for feature descriptors building, and explains how a set of curves is associated to each feature point already extracted. This section also presents a step-by-step procedure to compare 3D-part surfaces using the corresponding curves. In section 4, we exploit the previously presented 3D-part surface comparison schema to propose a 3D-object matching method using fusion tools given by belief function theory. In section 5, experimental results
are shown. Finally, section 6 contains the concluding remarks.

2 Feature points extraction

In this section, we present the algorithm used in the feature point extraction process. This algorithm is based on topological tools proposed by Tierny et al. [6]. Using a diversity of 3D-surfaces, the proposed algorithm produces robust and well-localized feature points. The concept of feature points has been introduced by several authors [11], [12]. However it is difficult to find a formal definition that characterizes this concept. In Katz et al. [12] feature points refer to the points localized at the extremity of a 3D-surface’s components. Formally, feature points are defined as the set of points that are the furthest away (in the geodesic sense) from all the other points of the surface. They are equivalent to the local extrema of a geodesic distance function which is defined on the 3D-surface. Figure 1 shows some 3D-surfaces with their corresponding feature points.

Let \( v_1 \) and \( v_2 \) be the farthest vertices (in the geodesic sense) on a connected triangulated surface \( S \). Let \( f_1 \) and \( f_2 \) be two scalar functions defined on each vertex \( v \) of the surface \( S \), as follows: \( f_1(v) = \delta(v, v_1) \) and \( f_2(v) = \delta(v, v_2) \) where \( \delta(x,y) \) is the geodesic distance between points \( x \) and \( y \) on the surface.

As mentioned by [13], in a critical point classification, a local minimum of \( f_1(v) \) is defined as a vertex \( v_{\text{min}} \) such that all its level-one neighbors have a higher function value. Reciprocally, a local maximum is a vertex \( v_{\text{max}} \) such that all its level-one neighbors have a lower function value. Let \( F_1 \) be the set of local extrema (minima and maxima) of \( f_1 \) and \( F_2 \) be the set of local extrema of \( f_2 \). We define the set of feature points \( F \) of the triangulated surface \( S \) as the closest intersecting points in the sets \( F_1 \) and \( F_2 \). In practice, \( f_1 \) and \( f_2 \) local extrema do not appear exactly on the same vertices but in the same geodesic neighborhood. Consequently, we define the intersection operator \( \cap \) with the following set of constraints, where \( \delta_n \) stands for the normalized geodesic distance function (to impose scale invariance):

\[
V \in F = F_1 \cap F_2 \iff \begin{cases} 
\exists v_{F_1} \in F_1 \setminus \delta_n (V, v_{F_1}) < \epsilon \\
\exists v_{F_2} \in F_2 \setminus \delta_n (V, v_{F_2}) < \epsilon \\
\delta_n (V, v_i) > \epsilon \forall v_i \in F \\
\epsilon, \delta_n \in [0, 1]
\end{cases}
\]

This algorithm detects all the feature points required in the subsequent analysis. They are accurately localized and their localization is robust with respect to rigid and non-rigid transformations, because of the use of geodesic distance in \( f_1 \) and \( f_2 \) functions. In the experimental section, we will evaluate the localization accuracy of feature points detection within our global approach.
representation of curves. Other authors, including Yezzi and Mennucci [18], have also used Riemannian metrics on curve spaces. Their main purpose was to study curves evolution rather than shape analysis. Here, we adopt the Joshi et al’s approach [17] because it simplifies the elastic shape analysis. The main steps are: (i) define a space of closed curves of interest, (ii) impose a Riemannian structure on this space using the elastic metric, and (iii) compute geodesic paths under this metric. These geodesic paths can then be interpreted as optimal elastic deformations of curves.

We start by considering a closed curve $\beta$ in $\mathbb{R}^3$. Since it is a closed curve, it is parameterizable using $\beta : \mathbb{S}^1 \rightarrow \mathbb{R}^3$. We will assume that the parameterization is non-singular, i.e. $\| \beta(t) \| \neq 0$ for all $t$. The norm used here is the Euclidean norm in $\mathbb{R}^3$. Note that the parameterization is not assumed to be arc-length; we allow a larger class of parameterizations for improved analysis. To analyze the shape of $\beta$, we shall represent it mathematically using a square-root velocity function (SRVF), denoted by $q(t)$, according to: $q(t) = \beta(t) / \sqrt{\| \beta(t) \|}.$

$q(t)$ is a special function that captures the shape of $\beta$ and is particularly convenient for shape analysis, as we describe next. Firstly, the squared $L^2$-norm of $q$, given by: $\| q \|^2 = \int_{\mathbb{S}^1} \langle q(t), q(t) \rangle dt = \int_{\mathbb{S}^1} \| \beta(t) \| dt$, which is the length of $\beta$. Therefore, the $L^2$-norm is convenient to analyze curves of specific lengths. Secondly, as shown in [17], the classical elastic metric for comparing shapes of curves becomes the $L^2$-metric under the SRVF representation. This point is very important as it simplifies the calculus of elastic metric to the well-known calculus of functional analysis under the $L^2$-metric. In order to restrict our shape analysis to closed curves, we define the set $\mathcal{C} = \{ q : \mathbb{S}^1 \rightarrow \mathbb{R}^3 | \int_{\mathbb{S}^1} \| q(t) \| dt = 0 \} \subset L^2(\mathbb{S}^1, \mathbb{R}^3)$.

Here $L^2(\mathbb{S}^1, \mathbb{R}^3)$ denotes the set of all functions from $\mathbb{S}^1$ to $\mathbb{R}^3$ that are square integrable. The quantity $\int_{\mathbb{S}^1} \| q(t) \| dt$ denotes the total displacement in $\mathbb{R}^3$ as one travels along the curve from start to end. Setting it equal to zero is equivalent to having a closed curve. Therefore, $\mathcal{C}$ is the set of all closed curves in $\mathbb{R}^3$, each represented by its SRVF. Notice that the elements of $\mathcal{C}$ are allowed to have different lengths. Due to a nonlinear (closure) constraint on its elements, $\mathcal{C}$ is a nonlinear manifold.

We can make it a Riemannian manifold by using the metric: for any $u, v \in T_q(\mathcal{C})$, we define:

$$\langle u, v \rangle = \int_{\mathbb{S}^1} \langle u(t), v(t) \rangle dt.$$ (1)

We have used the same notation for the Riemannian metric on $\mathcal{C}$ and the Euclidean metric in $\mathbb{R}^3$ hoping that the difference is made clear by the context. For instance, the metric on the left side is in $\mathcal{C}$ while the metric inside the integral on the right side is in $\mathbb{R}^3$. For any $q \in \mathcal{C}$, the tangent space: $T_q(\mathcal{C}) = \{ v : \mathbb{S}^1 \rightarrow \mathbb{R}^3 | \langle v, w \rangle = 0, w \in N_q(\mathcal{C}) \}$, where $N_q(\mathcal{C})$, the space of normals at $q$ is given by: $N_q(\mathcal{C}) = \text{span} \{ q^1(t), q^2(t), q^3(t) / \| q(t) \| \} \subset \mathbb{R}^3$.

It is easy to see that several elements of $\mathcal{C}$ can represent curves with the same shape. For example, if we rotate a curve in $\mathbb{R}^3$, we get a different SRVF but its shape remains unchanged. Another similar situation arises when a curve is re-parameterized; a re-parameterization changes the SRVF of curve but not its shape. In order to handle this variability, we define orbits of the rotation group $SO(3)$ and the re-parameterization group $\Gamma$ as the equivalence classes in $\mathcal{C}$. Here, $\Gamma$ is the set of all orientation-preserving diffeomorphisms of $\mathbb{S}^1$ (to itself) and the elements of $\Gamma$ are viewed as re-parameterization functions. For example, for a curve $\beta : \mathbb{S}^1 \rightarrow \mathbb{R}^3$ and a function $\gamma : \mathbb{S}^1 \rightarrow \mathbb{S}^1$, $\gamma \in \Gamma$, the curve $\beta(\gamma)$ is a re-parameterization of $\beta$. The corresponding SRVF changes according to $q(t) \rightarrow \sqrt{\| \gamma(t) \|} q(t)$. We set the elements of the set: $[q] = \{ \sqrt{\| \gamma(t) \|} O q(\gamma(t)) | O \in SO(3), \gamma \in \Gamma \}$, to be equivalent from the perspective of shape analysis. The set of such equivalence classes, denoted by $\mathcal{S} = \mathcal{C} / (SO(3) \times \Gamma)$ is called the shape space of closed curves in $\mathbb{R}^3$. $\mathcal{S}$ inherits a Riemannian metric from the larger space $\mathcal{C}$ and is thus a Riemannian manifold itself.

The main ingredient in comparing and analyzing shapes of curves is the construction of a geodesic between any two elements of $\mathcal{S}$, under the Riemannian metric given in Eqn. 1. Given any two curves $\beta_1$ and $\beta_2$, represented by their SRVFs $q_1$ and $q_2$, we want to compute a geodesic path between the orbits $[q_1]$ and $[q_2]$ in the shape space $\mathcal{S}$. This task is accomplished using a path straightening approach which was introduced in [19]. The basic idea here is to connect the two points $[q_1]$ and $[q_2]$ by an arbitrary initial path $\alpha$ and to iteratively update this path using the negative gradient of an energy function $E[\alpha] = \frac{1}{2} \int_0^1 \| \dot{\alpha}(s), \dot{\alpha}(s) \| ds$. The interesting part is that the gradient of $E$ has been derived analytically and can be used directly for updating $\alpha$. As shown in [19], the critical points of $E$ are actually geodesic paths in $\mathcal{S}$. Thus, this gradient-based update leads to a feature point of $E$ which, in turn, is a geodesic path between the given points. We will use the notation $d(\beta_1, \beta_2)$ to denote the geodesic distance, or the length of the geodesic in $\mathcal{S}$, between the two curves $\beta_1$ and $\beta_2$. Shown in the bottom row of Figure 3 is an example of geodesic path between two 3D-curves extracted from two different 3D-part surfaces; the cow-head part (left side) and the horse-head part (right side).

3.2 3D-Parts matching

Now we extend ideas developed in the previous section for analyzing shapes of curves to the shapes of full 3D-part surfaces. As mentioned earlier, a 3D-part surface $P$ is represented with an indexed collection of the level curves of the function $f$. That is, $P$ is equivalent to the set $\{ c_\lambda \in [0, L] \}$, where $c_\lambda$ is the level set associated with the distance function value equal to $\lambda$. Through this relation, each 3D-part has been represented as an element of the
Fig. 3. Geodesic path between cow-head part and horse-head part.

set $c^{0,L}$. In our framework, the shapes of any two parts are compared by comparing their corresponding curves. Let $D$ denote the global distance between two parts.

Given any two surfaces parts $P_1$ and $P_2$, and their collection of curves $\{c_1^\lambda, \lambda \in [0,L]\}$ and $\{c_2^\lambda, \lambda \in [0,L]\}$, respectively, our idea is to compare the curves $c_1^\lambda$ and $c_2^\lambda$, and to accumulate these distances over all $\lambda$. Formally, we define a distance: $D : C^{0,L} \times C^{0,L} \to \mathbb{R}_{\geq 0}$, given by

$$D(P_1, P_2) = \int_0^L d(c_1^\lambda, c_2^\lambda) d\lambda.$$ 

Here, the distance inside the integral is the geodesic distance function between the shapes of any curves, described in the last section. In Figure 3, The top row shows the two 3D-part surfaces (cow-head and horse-head) and the resulting geodesic path between them. Middle surfaces denote five equally spaced points (in the space of all parameterized paths) along the geodesic path. With respect to the chosen Riemannian metric, this path denotes the optimal deformation from the cow-head part to the horse-head part.

4 3D-SHAPE MATCHING USING BELIEF FUNCTION

In this section, we use the proposed 3D-parts matching method in order to compute a global distance between 3D-objects. For that, a fusion method based on the Transferable Belief Model (TBM) framework [20], an interpretation of the Dempster-Shafer theory of evidence, is proposed and applied to perform the 3D-shape matching. The TBM is shown to provide a powerful framework, well suited to the merge process.

4.1 Transferable Belief Model concept

The TBM is based on a two-level model: a credal level where beliefs are entertained, combined and updated, and a pignistic level where beliefs are converted into probabilities to make decisions.

4.1.1 Credal level

Let $\Omega$ denote a finite set called the frame of discernment. A Basic Belief Assignment (BBA) or mass function is a function $m : 2^\Omega \to [0,1]$, such that: $\sum_{A \subseteq \Omega} m(A) = 1$. $m(A)$ measures the amount of belief that is committed to $A$. The subsets $A$ of $\Omega$ such that $m(A) > 0$ are called focal elements.

Given two BBAs $m_1$ and $m_2$ defined over the same frame of discernment $\Omega$ and induced by two distinct pieces of information, we can combine them using the Dempster Shafer Combination rule [21] given by:

$$(m_1 \otimes m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C).$$

for all $A \subseteq \Omega$.

4.1.2 Pignistic level

When a decision has to be made, the beliefs held at the credal level induce a probability measure at the pignistic level. Hence, a transformation from belief functions to probability functions must be done. This transformation is called the pignistic transformation. Let $m$ be a BBA defined on $\Omega$, the probability function induced by $m$ at the pignistic level, denoted by $BetP$ and also defined on $\Omega$ is given by:

$$BetP(\omega) = \sum_{\omega \in A} \frac{m(A)}{|A|}$$

for all $\omega \in \Omega$ and where $|A|$ is the number of elements of $\Omega$ in $A$.

4.2 3D-shape matching

Given a collection of $M$ 3D-objects $\Omega = \{O_j, 1 \leq j \leq M\}$ and a set of $N$ 3D-object parts $\{P_i, 1 \leq i \leq N\}$ extracted from a global 3D-object query $Q$, our aim is to compute a global similarity metric between the 3D-object query $Q$ and 3D-objects in the collection. This metric is based on the distance between parts presented in Section 3. In order to achieve this computation, each part $P_i$ is modeled by one BBA $m_{P_i}$. This BBA is defined on $\Omega$ as frame of discernment and it measures the amount of belief accorded to the assumption: $P_i$ belongs to $O_j$. Formally, $m_{P_i}$ is a mass function $2^\Omega \to [0,1]$ given by:

$$m_{P_i}(O_j) = \mu(P_i) \cdot \frac{1 - D_1(P_i, O_j)}{\sum_{l=1}^M (1 - D_1(P_i, O_l))}$$

$$m_{P_i}(\Omega) = 1 - \mu(P_i)$$

where $D_1$ is a distance measure defined between an object part $P_i$ of the query $Q$ and a global 3D-object $O_j$ in the collection. This measure represents the distance between the part $P_i$ and its closest one in $O_j$. Formally $D_1$ is given by:

$$D_1(P_i, O_j) = \min_{k, P_k \subseteq O_j} (D(P_i, P_k))$$

It is required to normalize this distance in order to obtain a distribution on $[0,1]$ which allows us a correct construction of the BBA.

In equation 4, $\mu(P_i)$ is a confidence coefficient on $[0,1]$. This coefficient is interpreted as the significance given to each part in a global 3D object.
As mentioned by Bronstein et al. [8], the quantification of the significance of an object’s part is a major challenge. Here, we define the significance of a shape’s part as a function of the partiality function used in [8]. The value $\mu(P_i)$ of a 3D-part can be computed as $\mu(P_i) = \exp(-n\text{partiality}(P_i)^2)$ where $n\text{partiality}$ is the normalized partiality function of the 3D-part $P_i$. It is given by $n\text{partiality}(P_i) = \frac{\text{area}(Q) - \text{area}(P_i)}{\text{area}(Q)}$.

When all parts $P_i$ of the query $Q$ are modeled by their corresponding mass functions, we can apply the Dempster Shafer combination rule in order to get a mass function which measures the amount of belief committed to the assumption: $Q$ is similar to $O_j$.

Lastly, a decision is made from the obtained BBA and resulting objects are sorted based on a pignistic probability distribution. Our approach is summarized in algorithm 1 and depicted in Figure 4. In this figure a 3D-object of a cat is divided into several parts $P_i$. A mass distribution BBA is constructed for each part over all objects in the collection. The combination of all masses gives us a similarity indication about the whole object with all objects in the collection.

Algorithm 1 : 3D-shape search algorithm

Given a 3D-object query $Q$.
1: Extract a set of $N$ feature points $V_i$ from $Q$.
2: for each feature point $V_i$ where $i = 1, ..., N$ do
3: Extract a set of closed curves (ie. part $P_i$).
4: Compute the distance between $P_i$ and all objects in $\Omega$ using eq 5.
5: Set a BBA value for all the objects $m_{P_i}(O_j)$ using eq 4.
6: end for
7: Combine all BBAs $m_{P_i}$ into a new BBA $m_Q$ using eq 2.
8: Compute the pignistic probability induced by $m_Q$ using eq 3.
9: Display 3D-objects in the order according to the pignistic probability.

5 Experiments and Results

In this section, we present the databases, some state-of-the-art shape-matching algorithms to compare with, the evaluation criteria and the experimental results. Data with noise are considered in the sequel and a quantitative analysis of the experimental results are reported. The algorithms that we have described in the previous sections have been implemented using MATLAB software. The framework encloses an off-line feature point extraction algorithm and a curve extraction algorithm, and an on-line retrieval process. To evaluate our method, we used two different databases. The TOSCA data set for non-rigid shapes and the SHREC07 shape benchmark for partially similar models.

5.1 TOSCA data set description

The TOSCA data set has been proposed by Bronstein et al. [22]. It is an interesting database for non-rigid shape correspondence measures. The database consists of 148 models, enclosing 12 classes. Each class contains one 3D-shape under a variety of poses (between 1 and 20 poses for each class). This classification is provided with the data set.

5.2 SHREC07 data set description

This benchmark is composed of a dataset of 400 manifold models and of a query set of 30 manifold models composed of composite models as shown in Figure 5. Hence, it is an interesting database for partial shape retrieval. The dataset exhibits diverse variations, from pose change, to shape variability within a same class or topology variation (notice 4 of the 20 classes contain non-zero genus surfaces) [23]. The ground truth is provided with the data set.

5.3 Some state-of-the-art algorithms

In order to evaluate the 3D-global matching approach, we compare our method with some state-of-the-art shape-matching algorithms.

- **Extended Reeb Graphs** (ERG): it is a structural based 3D matching method. It works with Reeb graph properties [23].
- **Ray-based approach with Spherical Harmonic representation** (RSH): Saupe and Vranic [24] use PCA to align the models into the canonical position. Then the maximal extents are extracted, and finally the spherical harmonic is applied.
- **The hybrid feature vector** (DSR): it is a combination of two view based descriptors: the depth buffer and the silhouette and radialized extent function descriptor [25].
- **The Geodesic D2**: it is an extension of the Euclidean D2 [26]. It is computed as a global distribution of geodesic distances in 3D-shapes.

5.4 Evaluation criterion

There are several different performance measures which can evaluate retrieval methods. In this paper, we test the robustness of our approach using the Precision vs Recall plots. They are well known in the literature of content-based search and retrieval. The precision and recall are defined as follow:

$$\text{Precision} = \frac{N}{A}, \text{Recall} = \frac{N}{Q}$$

1. http://tosca.cs.technion.ac.il/
where $N$ is the number of relevant models retrieved in the top $A$ retrievals, $Q$ is the number of relevant models in the collection, which is the number of models to which the query belongs to.

5.5 Results
During the off-line process each model in the database has been remeshed to get a regular model. Feature points have been extracted and their related sets of curves (parts) have been extracted and stored into indexed files. During the on-line process, feature points and related curves of the query object are extracted.

5.5.1 Results on the TOSCA dataset
In order to show the main contribution of our approach, some results are shown as a matrix in Figure 6. In this visualization of the matrix, the lightness of each element $(i; j)$ is proportional to the magnitude of the distances between 3D-objects $i$ and $j$. That is, each square, in this matrix, represents the distances between two 3D-objects. Darker elements represent better matches, while lighter elements indicate worse matches.

This matrix is not symmetric because our similarity measure based on belief function is not symmetric either. In this matrix, humans are similar together and partially similar to centaurs. Centaurs are similar together and partially similar to horses and humans. According to the lightness of the matrix square, we can easily do the distinction between three different classes. The first class contains the first three animals, the second class encloses centaurs and horse and the third class contains humans and centaurs.

5.5.2 Results on the SHREC07 dataset
From a qualitative point of view, Figure 9 gives a good overview of the efficiency of the framework. For example, in the first row, the query is a centaur and thus most of the top-results are humanoid and animal models.

Quantitatively, we compare the Precision vs Recall plot of our approach with other methods competing the contest. Such a plot is the average of the 30 Precision vs Recall plots corresponding to the 30 models of the query-set. Figure 10 shows these plots. As the curve of our approach is higher than the others, it is obvious that it outperforms related methods.

5.6 Robustness to query noises
We investigate the framework robustness against surface noise. For each element of the query-set, we added a surface noise. The noise is ±0.2% and ±0.5% the length of the model’s bounding box. It consists of three random
translations in the three directions X, Y and Z. Figure 11 shows some examples of the noise effect.

Precision vs Recall plot are computed for our approach under the noise and compared with the original Precision vs Recall plot. From this results shown in Figure 12, the noise addition effects can be observed. These plots demonstrate the stability of the algorithm despite random noise. Moreover, even with a random noise of ±0.5% it still outperforms the scores on clean data of two methods presented in Figure 8.

5.7 Robustness to feature points localization

In order to assess the robustness of our method under localization of feature points, we apply the following transformation for the query set. We select a random vertex located in a geodesic radius (1% and 2% of the geodesic diameter of the 3D-object) centered in the original feature points.

In Figure 13 an example of the feature points localization with noise addition is shown. The original feature point is colored in green. The perturbation consists of a random selection of one vertex among vertices colored in red. Figure 14 shows that the performance of our approach is robust with respect to 1% of perturbation. The 2% perturbation plot shows the limit of the algorithm in term of feature point perturbation. However our method performance is still comparable to state-of-the-art method performance on clean data.

5.8 Discussion and limitations

The feature point extraction step of the framework, which is based on geodesics, introduces a bias in the comparison process. To guarantee stability and performance, this method has to be stable within a same class of objects and moreover coherent with the dataset ground-truth. In practice, with the TOSCA and the SHREC07 datasets, feature extraction turns out to be homogeneous within most classes. However, the geodesic
We would like to thank Shantanu H. Joshi for providing us a MATLAB code for computing a geodesic path between curves. This work was partially supported by the project ANR-07-SESU-004 and the Contrat de Projet Etat-Région (CPER) Région Nord-Pas-de-Calais Ambient Intelligence.

6 Conclusion

In this paper, we have presented a geometric analysis of 3D-shapes in presence of non-rigid transformation and partially similar models. In this analysis, the preprocessing is completely automated: the algorithm processes the 3D-object, extracts a set of feature points and extracts a set of curves around each feature point to form a part. A 3D-part matching method is then presented to allow the calculation of geodesic distances between two arbitrary 3D-part surfaces. The length of a geodesic path between any two 3D-part surfaces is computed as the geodesic length between the set of their curves. This length quantifies the differences in their shapes. Moreover, we have presented a 3D-shape matching method for global 3D search and retrieval applications. Belief functions have been used in this matching framework to compute a global similarity metric based on the 3D-part matching. The results on the TOSCA and the SHREC07 datasets show the effectiveness of our approach. They also show the robustness to rigid and non rigid transformations, surface noise and partially similar models. For future work, we plan to extend our approach for the detection of intrinsic symmetries in

Acknowledgment

We would like to thank Shantanu H. Joshi for providing us a MATLAB code for computing a geodesic path between curves. This work was partially supported by the project ANR-07-SESU-004 and the Contrat de Projet Etat-Région (CPER) Région Nord-Pas-de-Calais Ambient Intelligence.

References