A Survey on Shape Correspondence

Oliver van Kaick[1], Hao Zhang[1], Ghassan Hamarneh[1], Daniel Cohen-Or[2]

[1] Simon Fraser University, [2] Tel Aviv University

Abstract

We present a review of the correspondence problem and its solution methods, targeting the computer graphics audience. With this goal in mind, we focus on the correspondence of geometric shapes represented by point sets, contours or triangle meshes. This survey is motivated by recent developments in the field such as those requiring the correspondence of non-rigid or time-varying surfaces and a recent trend towards semantic shape analysis, of which shape correspondence is one of the central tasks. Establishing a meaningful shape correspondence is a difficult problem since it typically relies on an understanding of the structure of the shapes in question at both a local and global level, and sometimes also the shapes' functionality. However, despite its inherent complexity, shape correspondence is a recurrent problem and an essential component in numerous geometry processing applications. In this report, we discuss the different forms of the correspondence problem and review the main solution methods, aided by several classification criteria which can be used by the reader to objectively compare the methods. We finalize the report by discussing open problems and future perspectives.

1. Introduction

Finding a meaningful correspondence between two or more shapes is an important task whose specializations appear under a variety of different names in the literature, such as registration, alignment and matching. The correspondence problem can be generally stated as: *given input shapes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N$, establish a meaningful relation between their elements*. Figure 1 presents an example of a correspondence, where the elements are feature points on the surface of the shapes and the relation links pairs of these points.

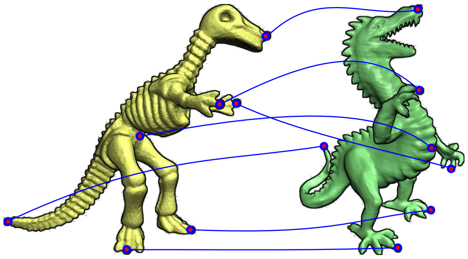


Figure 1: *An example of a meaningful correspondence between two shapes automatically computed with the approach in [ZSCO*08]. The correspondence is shown as the blue lines that relate a sparse set of feature points on the two shapes (shown as the dots).*

The properties that a correspondence has to satisfy to be deemed meaningful depend on the task at hand. Our task can range from the simpler case of finding portions of the shapes that are geometrically similar, to the more complex one where we seek to relate elements that represent the same parts on the shapes or serve the same function. An example of such a semantic correspondence problem is shown in Figure 2. In general, computing a correspondence is a hard problem, since it involves understanding the global structure of the shapes at local and global levels, and possibly the functionality of its parts.

The correspondence problem has been traditionally studied in the computer vision and image analysis communities. However, in this report, our goal is to provide a survey targeted towards the computer graphics audience. We focus on methods directed at datasets that we denominate *shapes*, which are represented as points sets, contours, or triangle meshes, as opposed to digital images and volumes. The key property of these datasets is that they provide explicit geometry information (with or without connectivity information), but generally lack a simple parameterization domain. Another distinction is that the analysis of images benefits from rich local descriptors based on color and texture, while on shapes the local descriptors that can be computed are not as

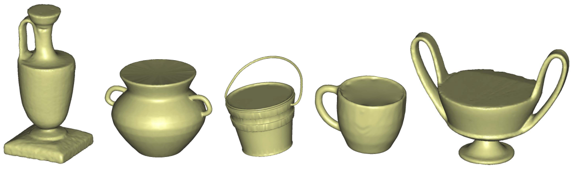


Figure 2: *An example of a collection of man-made shapes (liquid containers) for which computing a correspondence is a challenging problem. Note how the shapes can be constituted by different types and numbers of parts (e.g., one or two handles), how the parts of a same type can vary in their geometry (e.g., long vs. short handles), and how they can connect to each other in different manners.*

distinctive. In this report, we also present a comprehensive coverage of the correspondence of 3D datasets, which we think needs special attention due to the importance of 3D models in computer graphics and geometry processing, and given the inherent increase in complexity that these datasets introduce to the problem.

Correspondence methods have been commonly used for graphics applications such as 3D scan alignment or shape morphing. However, there have been a number of recent advances in the field that have prompted the development of this survey in the present-day. These innovations have been driven by new applications, such as time-varying surface reconstruction and markerless motion capture, which are now possible due to advances in acquisition techniques.

Firstly, there have been emerging techniques for surface deformation that work directly on the input shapes and efficiently obtain a deformed surface. The potential application of these techniques to non-rigid registration of surfaces has been realized by different authors [HAWG08, ZSCO*08]. Secondly, progress in matching approximately isometric shapes has also been exposed in recent works [BBBK08, LF09]. Finally, there appears to be a recent trend in the literature towards the development of techniques that go beyond low-level geometry and extract high-level semantic information from the shapes, such as segmenting a mesh into parts [Sha08], finding analogies between these parts [SSSCO08], transferring information from one shape to another [SP04], or extracting the high-level structure of the shapes for manipulation or deformation [XWY*09, GSMCO09]. Computing a correspondence between shapes is one of the key problems that can benefit from semantically-driven techniques, since our aim is to understand the structure of the shapes in order to find a meaningful correspondence between their parts.

In this report, we provide an up-to-date survey on shape correspondence which should be beneficial to those seeking familiarity with the topic. Although several comprehensive surveys have been written on image registration, e.g., [Bro92, MV98, ZF03], only few aimed specifically at the correspondence of shapes and 3D datasets. Therefore, this survey focuses on this topic and also intends to cover all of the recent developments in the field.

Our report is directed at researchers in computer graphics who would like to obtain an overview of the different approaches for correspondence or who need to select the most suitable method to solve a problem at hand. For the latter purpose, we present several classifications of the correspondence methods in our discussion, so that readers are able to objectively compare the different techniques that are available, and understand their strengths, weaknesses, and applicability. We also cover relevant techniques from other fields, such as computer vision and machine learning, in order to give a complete overview of the most important methods for shape correspondence to the targeted audience. Finally, we elaborate on challenges and future perspectives.

2. Problem statement

There is a variety of problems that can be classified under the scope of *shape correspondence*. As we will see in Section 3, significant differences exist between these problems and the computational paradigms used for their solution also differ. However, since all these different problems comprise the same fundamental task, we can view them all in a unified manner by considering the following problem statement: *given input shapes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N$, establish a meaningful relation $\mathcal{R}$ between their elements*. When two elements are related to each other (i.e., $(s,z) \in \mathcal{R}$, for elements $s \in \mathcal{S}_i$ and $z \in \mathcal{S}_j$, with $i \neq j$), we say that they are in correspondence or that they match to each other. The relation can be constrained in different manners, such as asking for a one-to-one, one-to-many, or many-to-many correspondence.

Therefore, shape correspondence is generally defined as identifying homologous elements of two or more shapes, i.e., the elements that possess the same or similar structure in terms of their local appearance and context. The meaning of *elements* and *similar structure* depend on the problem at hand. However, a shape (represented as a point set, surface, skeleton, etc.) will be usually composed of multiple elements (primitives such as points, feature points, faces, skeletal features, or higher-level entities such as parts, composed of a group of primitives). To establish the similarity between the elements, we resort to measures of geometric or structural similarity, although in certain applications our ultimate goal behind this step is to recognize the elements and infer their semantics. The forms that the components of the problem definition can assume to make our problems more concrete are discussed in detail in Section 4.

One important distinction to raise here is the difference between the problems of shape correspondence and retrieval. Retrieval is the task where, given a query shape, we wish to find shapes from a database that are similar to the query, tolerating a variety of transformations and possibly considering partial similarities. Therefore, we are only interested

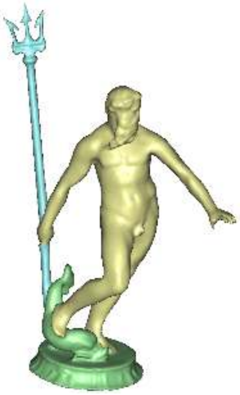


Figure 3: *An example of partial correspondence between parts of shapes: the goal here is to find a correspondence between Neptune's statue (left) and the human (right), by relating the parts in yellow and ignoring the extra parts shown in green and blue.*

in quantifying the similarity between shapes, while in correspondence we want to explicitly relate the elements of one shape to those of the other. The evaluation of shape similarity in retrieval is sometimes referred to as *shape matching* in the literature [KFR04], which conflicts with the nomenclature followed by other authors who refer to shape matching as computing a correspondence. Perhaps the reason for the interchangeable use of these terms in the literature is that retrieval and correspondence are closely related problems, since computing a correspondence and evaluating its quality is one possible manner of assessing shape similarity.

The general correspondence problem stated above can be specialized in different ways, which we discuss as follows.

Full vs. partial correspondence: following our problem definition, here we focus on the inclusion properties of the relation $\mathcal{R}$ defined between shape elements. We can require our correspondence to be full (defined for all the shape elements) or partial (defined for a subset of the elements). The motivation for the latter is that the shapes under consideration may be constituted by different parts (i.e., one shape can have missing or additional parts when compared to the others), and so it may not be meaningful to establish a correspondence between all their elements. These parts can differ by their geometry, scale, and connecting location on the overall object. An example is shown in Figure 3.

Computing a partial correspondence is a more difficult problem than computing a full correspondence, due to the combinatorial explosion in the solution space. If we consider all one-to-one assignments between shapes with n elements, the solution space is composed of $n!$ correspondences. If we add the possibility for partial matching, the solution space includes all the possible subsets of these $n!$. Moreover, searching for the right subset increases the complexity of the problem and also requires a careful definition of the optimality criterion.

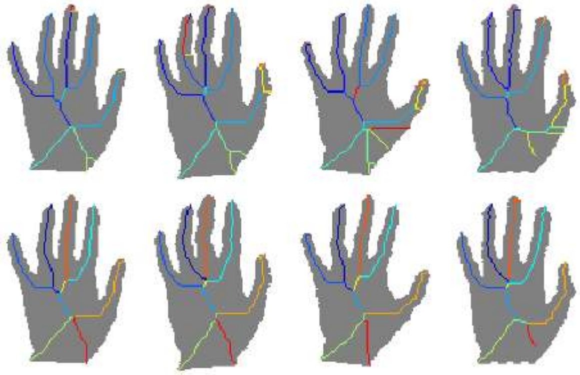


Figure 4: *Motivation for group correspondence: in this example, we compare the skeletonization computed individually for each 2D shape (top row) to the result of pruning these skeletons with the method in [WH09] (bottom row), which takes group information into account. We see how using group information improves the coherence of the skeletons across shapes and also the quality of the branch correspondence (shown by coinciding colors).*

Dense vs. sparse correspondence: another aspect to take into account is the *density* of the relation $\mathcal{R}$. A *dense correspondence* is defined for all the elements or primitives on the shape (e.g., faces). A *sparse correspondence* is defined for a small number of pre-selected elements. The elements are usually a set of features. For example, to infer the semantics of human shapes, it is sufficient to match representative points located at the legs, arms, head and body of the shapes (a sparse correspondence). On the other hand, for applications such as morphing or attribute transfer between two shapes, we require a dense correspondence in order to guarantee global smoothness in the morphing or transfer result.

It is generally accepted that computing a sparse correspondence between representative points is as hard a problem as computing a dense one between two shapes, since both problems necessarily involve taking into consideration the global structure of the shapes [ZSCO*08]. However, a typical approach for the dense case is to obtain a solution from a sparse correspondence with techniques such as the computation of a cross-parameterization or interpolation, involving decisions at a more local level [Ale02].

Group correspondence: a specialized form of shape correspondence involves a group of shapes simultaneously (i.e., $N > 2$, in our problem definition). Although this problem can be straightforwardly solved by computing correspondences between all the pairs of shapes in the group, considering all the shapes at once can have advantages in the process. For example, as illustrated in Figure 4, the group information

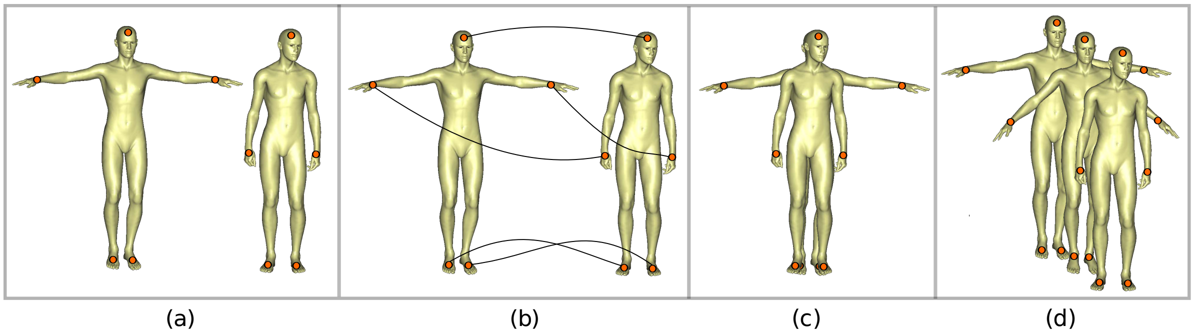


Figure 5: *Different manners of solving the correspondence problem for the input shapes shown in (a) and their feature points (indicated by the circles): (b) computing a correspondence without explicitly bringing the shapes into alignment, (c) computing a global rigid transformation to align the two shapes, and (d) computing local non-rigid transformations for the shape primitives to deform one shape into the other.*

can act as a reinforcement to distinguish what structures or parts are common to all the shapes and should be considered in the correspondence, and which parts can be characterized as noise and ignored in the computation [WH09]. A robust correspondence between a group of shapes is the desired input for tasks such as building a statistical description of the group of shapes [DTC*02].

3. Overview of correspondence problems

Now, we proceed to give an overview of correspondence problems by discussing them in a more concrete manner, while also describing some example applications. We can obtain a correspondence directly from the similarity of the elements, or we can first align the shapes and then derive a correspondence from the proximity of the aligned elements. Moreover, we can also iterate between the two procedures. These options directly affect how we solve the correspondence problem. It is worth noting that the alignment between the shapes is a side product of the computation which is useful and sometimes essential to the underlying application. We will start by considering the case in which no alignment is utilized, and then we discuss the scenarios where the relation $\mathcal{R}$ is derived from aligning the shapes in a rigid or a non-rigid manner. We illustrate the distinction between these three cases in Figure 5. We finalize with the instance where the time dimension is added to the input datasets.

Similarity-based correspondence: one of the most fundamental ways of computing a correspondence is to estimate the similarity between pairs of shape elements or feature points collected from the shapes and derive a correspondence from those estimates, which is sometimes called the *feature matching* approach. The elements are commonly characterized by shape descriptors. A correspondence is then obtained by selecting pairs of assignments between elements while optimizing an objective function composed of two terms. The first term seeks to maximize the similarity between the descriptors of corresponding elements, while the second term seeks to minimize the distortion that would be introduced in the shapes if they were deformed to align with each other. However, the second term can be estimated from the correspondence without the need to explicitly align the shapes. Ideally, satisfying these objectives should translate into a solution that is geometrically or semantically meaningful. Such a solution is typically obtained with an optimization method.

This approach can be applied in any context where it is possible to compute a set of descriptors for the elements. Example applications include registration of 3D scans [CCFM08] and deforming surfaces [ASP*04], or skeleton matching [BMSF06]. Moreover, this approach is not restricted to its own domain and can be combined with alignment-based approaches to provide a proper initialization to these methods [RL01], or to restrict the size of the solution space [GMGP05, AMCO08, CZ08, ACOT*10].

Rigid alignment: under certain assumptions, it is possible to pose the correspondence problem as finding a geometric transformation that aligns the shapes. One example application is the rigid alignment of geometry scans used for shape acquisition. Our goal here is to capture a real-world 3D shape and obtain its digital representation. However, we may not be able to capture the entire object in a single scanning pass due to self-occlusions and physical constraints of the scanner, so it might become necessary to acquire multiple scans and optimally align them to reconstruct the full object [TL94, RL01, GMGP05, AMCO08]. The key characteristic of this alignment problem is that our objects do not change from one scanning pass to another. Thus, we assume that each scan can be transformed with a *single rigid transformation* in order to align it perfectly with the other scans. Rigid transformations comprise mainly translations and rotations, and one of their important characteristics is that they reside in a low-dimensional space.

Scan alignment is just one example of many applications where we assume rigidity in the datasets. If the input shapes are given as sets of 3D points, the problem of rigid alignment can be posed as: for each point set $\mathcal{S}$, find the rigid transformation that, when applied to $\mathcal{S}$, maximizes the number of points in $\mathcal{S}$ that align to points in the other sets. This goal is usually dependent on a threshold ε that indicates when two points are close enough and can be considered as matching to each other [IR96]. Since finding the best aligning transformation might be an involved task, we can make use of the feature matching approach to aid in the search of the optimal alignment.

Non-rigid alignment: sometimes it might be necessary to lift the assumption that each scan can be perfectly aligned with a rigid transformation, e.g., when large amounts of noise are present in the scans. More significant examples of datasets that cannot be perfectly aligned with a rigid transformation include the correspondence of articulated shapes [EK03, ASP*04, JZvK07, CZ08, HAWG08], where certain parts of the shapes can bend independently, the correspondence of anatomical shapes (e.g., organs) [AFP00], which can deform in an elastic manner and introduce stretching to localized portions of the shape, and finally the correspondence between shapes with different geometries but that represent a same class of objects or which have parts that are semantically related [ACP03, ZSCO*08]. In the latter case, we can see the problem as that of establishing a correspondence between shapes that can differ in both local stretching and bending.

In this setting, it becomes necessary to add more freedom to how the shapes can be brought into correspondence. This can be achieved by generalizing two aspects of the problem. First, we deviate from the rigid case and allow non-rigid (possibly non-linear) transformations to be taken into consideration, e.g, thin-plate splines [CR03]. Secondly, these transformations can be applied separately to local portions of the shape. For example, we can represent the transformation applied to a shape as a displacement vector associated to each vertex on the shape [PMG*05]. Then, finding the best transformation amounts to computing the displacements that bring each vertex in correspondence with the target shape. The distinction to the rigid case is that the space of geometric transformations being considered is now inherently high-dimensional.

Time-varying registration: due to recent technological advances, an application that is attracting increasing attention is the reconstruction of 3D shapes acquired over time while moving and deforming. In this setting, a fixed number of scans is acquired per unit of time, and these scans have to be registered to allow the reconstruction of both the object and the motion sequence [MFO*07, WJH*07, SAL*08, PG08, dAST*08, LAGP09, GSdA*09, CZ09, TBW*09, ZST*10]. Although this may sound as another instance of the non-rigid alignment problem, there are certain particularities that make this problem unique. In the classic registration problem, it is assumed that all the scans can be registered to compose a single and coherent object. On the other hand, the time-varying setting introduces the additional difficulty that the shape might have deformed significantly from one frame to the other. Therefore, scans acquired later in time may only be registered to the earlier scans if the deformation is taken into account. Moreover, additional challenges are the large amount of missing data (due to occlusion) that can be present in each frame [PG08], and datasets that were captured over sparse time frames [CZ09, ZST*10].

Having established the basic notions and definitions related to shape correspondence, we proceed to the next section to divide the problem into its relevant components and analyze each of them in a detailed manner.

4. Classification of correspondence methods

In this section, we present several ways of classifying the correspondence methods discussed in this report. The purpose of these classifications is to allow readers to compare the methods not only by their algorithmic aspects, but also by the properties of the problems that can be handled and the requirements of the methods. These include questions such as: what subproblems the method can handle (e.g., partial correspondence, rigid registration, etc.), what datasets can be handled, what initialization the method requires, etc.

We group the classification criteria based on the components of the problem statement, which we review here once more: *given input shapes $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N$, establish a meaningful relation $\mathcal{R}$ between their elements*. To make this statement more concrete, we have to define the meaning for the keywords that appear in it. We will do that by associating them to the following questions.

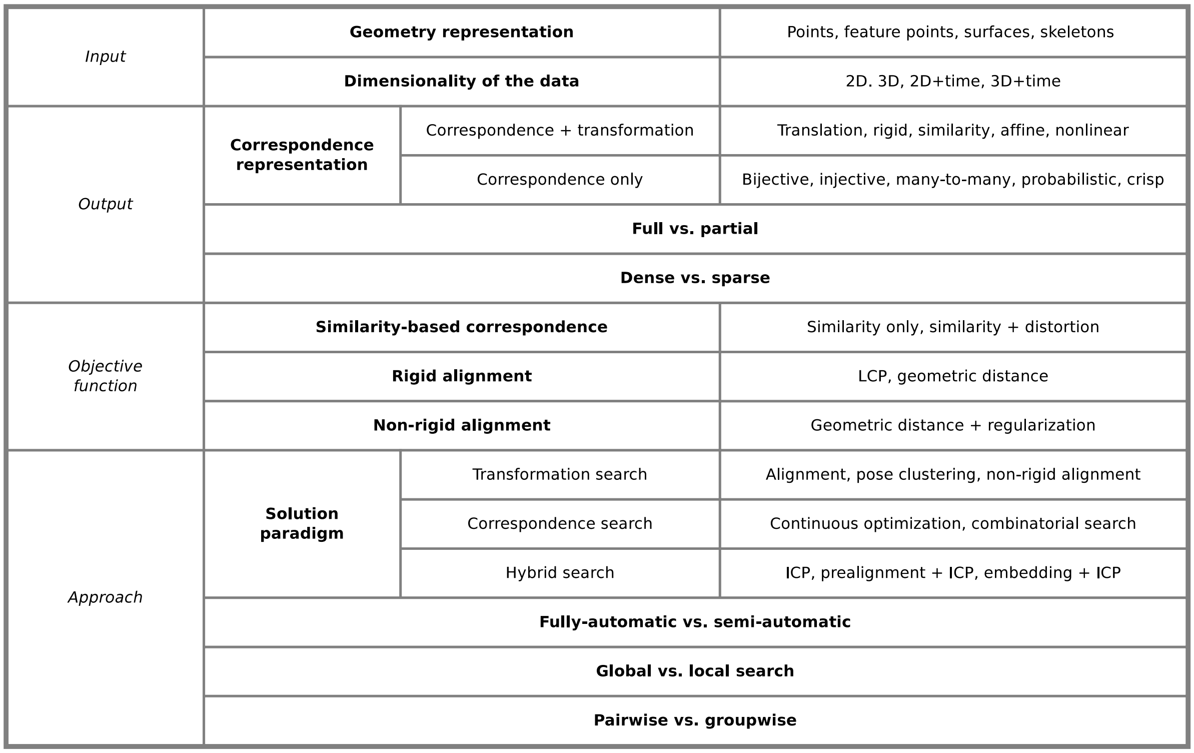
Keyword → Question

1. *shapes* → how are the input shapes represented?
2. *relation* → how is the output correspondence represented and what properties does it possess?
3. *meaningful* → which correspondence to select (which correspondence is closer to our objective)?
4. *establish* → what approach is used to compute the correspondence?

Answering these questions will lead to the different classification categories. We follow with a detailed discussion of these classifications, summarized in Figure 6.

4.1. Input shapes

The geometry of the input datasets can be represented as point sets (feature points or not), oriented points (points with normals), surfaces, skeletons, images, volumes, sets of parts. Their dimensionality can be 2D, 3D, or they can be spatio-temporal: 2D + time or 3D + time. We focus on 2D and 3D shapes (point sets, surfaces, and skeletons). Surfaces can be



Input	**Geometry representation**		Points, feature points, surfaces, skeletons
	Dimensionality of the data		2D. 3D, 2D+time, 3D+time
Output	**Correspondence representation**	Correspondence + transformation	Translation, rigid, similarity, affine, nonlinear
		Correspondence only	Bijective, injective, many-to-many, probabilistic, crisp
	Full vs. partial		
	Dense vs. sparse		
Objective function	**Similarity-based correspondence**		Similarity only, similarity + distortion
	Rigid alignment		LCP, geometric distance
	Non-rigid alignment		Geometric distance + regularization
Approach	**Solution paradigm**	Transformation search	Alignment, pose clustering, non-rigid alignment
		Correspondence search	Continuous optimization, combinatorial search
		Hybrid search	ICP, prealignment + ICP, embedding + ICP
	Fully-automatic vs. semi-automatic		
	Global vs. local search		
	Pairwise vs. groupwise		

Figure 6: *Schematic view of the classification criteria followed in this work (a hierarchy read from left to right). The first column are the components of the problem statement used to group the different criteria.*

represented implicitly (e.g., level sets) or explicitly (e.g., triangle meshes). One extra distinction in the case of triangle meshes is whether they need to represent well-defined manifolds or if they can be triangle soups.

Classic registration methods such as RANSAC [FB81], geometric hashing [WR97], pose clustering [Ols97], and alignment [HU90] typically work with point sets. Surfaces are the common representation for recent methods based on deformation [HAWG08, ZSCO*08], methods that work on isometric surfaces [BBBK08, LF09] or articulated shapes [CZ08], and graphics applications based on template matching [ACP03, PMG*05, LAGP09]. Time-varying surfaces are the focus of works on motion reconstruction of deforming surfaces [MFO*07, WJH*07, SAL*08, PG08, GSdA*09, LAGP09, ZST*10]. Skeleton is a more general name for shape representations such as the medial axis, Reeb graphs, and curve skeletons [SP08, CSM07, HSKK01, SSGD03, BMSF06, ACOT*10].

These datasets can arise from a variety of sources, such as 3D scanners (based on LASER, structured light, physical contact, etc.) that provide point clouds, manual modeling via software which commonly results in triangle meshes, or images and volumes obtained with different imaging equipment (digital cameras, ultrasound, magnetic resonance imaging (MRI), computed tomography (CT), etc.) that provide surfaces via image segmentation or isosurfacing. Skeletons are commonly obtained from surfaces by an extra processing step [CSM07], and they capture more structural information (such as shape parts represented as branches and their associated thicknesses).

Shape descriptors: instead of making use of the datasets in their original representation, we can sometimes characterize them by extracting representative points (features) from the shapes and computing descriptors for these points. These descriptors will typically be scalar values or vectors of scalars that capture some property of the shape around the neighborhood of the interest point [TH08]. The shape descriptors are then used to indirectly establish the similarity between the datasets by assessing the similarity between the descriptors. Ideally, if two descriptors are similar, their corresponding points should also be similar. Alternatively, the descriptors can be used to guide the search for initial solutions, while the final verification of the correspondence quality is performed with the original dataset [AMCO08]. Note that we can see the approach of comparing descriptors as a general correspondence framework, since the positions of the feature points themselves can be regarded as descriptors.

A variety of descriptors have been proposed in the liter-

Descriptor	Type of dataset
Shape context [BMP00, KPNK03]	Point sets
Spin images [JH99]	Oriented points
Multi-scale features [LG05]	Oriented points
Spherical harmonic-based [FS06]	Oriented points
Curvature maps [GGGZ05]	Surfaces
Integral invariants [GMGP05, MCH*06]	Surfaces
Salient geometric features [GCO06]	Surfaces
Part-aware metric [LZSCO09]	Surfaces
Heat Kernel Signature [SOG09]	Surfaces

Table 1: *An extract of shape descriptors proposed in the literature that can be used for shape correspondence.*

ature and a discussion of their characteristics can be found in [VH01, BKS*05, TV08], while a comparison of descriptors for the image case is presented in [MS05]. In Table 1, we list a partial set of descriptors, which can be computed at a local level and used in conjunction with shape correspondence for 2D surfaces or 3D point sets. Such a set of descriptors is the typical input to optimization- or search-based matching methods used in vision and graphics [MC03, ASP*04, BBM05, LH05, GMGP05, ZSCO*08]. Another interesting related problem is how to choose the set of descriptors that gives the best correspondence results, a problem known as *feature selection* in the machine learning literature [GE03, WH07].

4.2. Output correspondence

The correspondence computed by the methods can be represented in different manners, and it can also differ in other properties, such as whether it is a full, partial, dense, or sparse correspondence.

Correspondence representation: We can represent a correspondence as a transformation applied to the shapes (either a single or multiple transformations) or simply as a relation between elements of the datasets, i.e., a set of pairwise assignments between vertices, parts, etc.

Correspondence + transformation: when utilizing a single transformation or a set of transformations, one of the distinguishing factors is the type of transformation that is utilized. These transformations can be ordered by increasing number of degrees of freedom: translation, rigid transformation (includes rotations), similarity transformation (includes isotropic scaling), affine transformation (includes shearing), and nonlinear deformation (includes nonlinear transformations). Notice that the group of rigid transformations also includes reflections.

A rigid transformation preserves the pairwise distances between the transformed points, and can be decomposed into translations, rotations and reflections. Therefore, it is the common choice when dealing with problems such as scan registration [RL01, GMGP05]. Similarity transformations incorporate the possibility of uniform scaling into the rigid transformations, which might be necessary in certain contexts such as matching patterns to limited portions of larger datasets (e.g., repeated pattern detection as in [PMW*08]). Moreover, affine transformations extend the previous set of transformations by also taking into consideration the possibility of shearing, which can be used at a global [IR96, AMCO08] or local [ACP03, SP04] level. When matching non-rigid surfaces that possess the topology of a sphere, the natural transformation to use is the group of homographic or Möbius transformations (Figure 7), which are conformal (preserve angles) and contain the group of isometric transformations (which preserve distances) [LF09] . Finally, in more general cases of non-rigid alignment, it might be necessary to allow the shape elements to move freely in order to match with the corresponding dataset, which can be seen as assigning a nonlinear deformable transformation (e.g., a displacement) to each element [PMG*05].

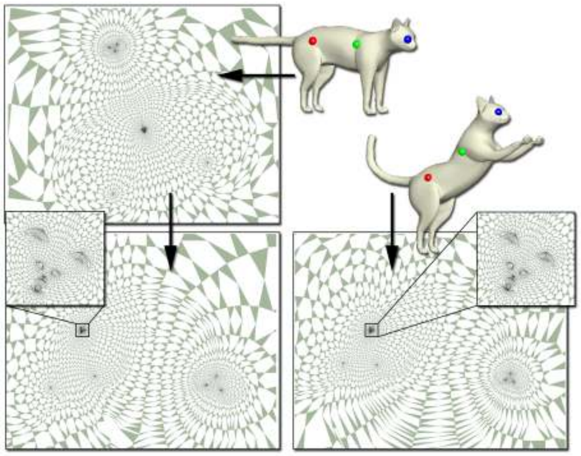


Figure 7: *Utilizing Möbius transformations for shape correspondence [LF09]: the meshes for a cat in two different poses are flattened onto the complex plane. Next, a Möbius transformation can be defined from three corresponding pairs of points (red, green, and blue). By applying the transformation to the flattening of the first pose (top-left), the result at the bottom-left is obtained, which looks similar to the flattening of the second pose (bottom-right).*

Another distinction here is whether the transformation is applied to the shape as a whole (a *global* transformation), or whether it is applied in a *local* manner, i.e., to parts or elements of the shape. The global case is a central concept in the problem of rigid alignment, while the local case is usual in tasks of a non-rigid nature.

Correspondence only: in the case of working only with a correspondence, we can limit the relation $\mathcal{R}$ to represent a bijection (one-to-one mapping), an injection (the relation has to be defined for every element of one of the shapes taken as reference, but it can be a one-to-many mapping), or we can allow a full relation to be defined (many-to-many). Moreover, we might require a one-to-one or one-to-many mapping only for a subset of elements, such as in a partial correspondence context where we allow some elements not to be part of any assignment.

Certain methods allow to select which type of mapping we desire, such as [LH05], where we obtain the final correspondence by filtering an initial result according to the mapping constraints. Other approaches assume that we are interested in a specific type of mapping and build upon that assumption, e.g. [MC03, BBM05], which pose the correspondence problem as an optimization where we constrain the mappings to be one-to-one.

We can also distinguish a correspondence by whether we assign a confidence value to the pairwise assignments. The assignments can be characterized as *crisp* (either an assignment is part of the correspondence or not), or they can have a degree of confidence attached to them (a measure of *fuzziness* or *probability*). The first case can be seen as that where a binary confidence is associated to each assignment. The type of confidence measure that is available depends on the correspondence algorithm. For example, the method in [ZS08] directly returns a probability associated to each assignment, since it formulates the correspondence problem in a probabilistic manner, while [MC03, BBM05] return mainly binary outputs, since they define the correspondence problem as an integer optimization.

Full vs. partial correspondence: some methods are only suitable for contexts in which the full extent of the shapes is considered, while others are able to compute a partial correspondence. In general, if a method can find solutions for the partial case, it will also be applicable to the full correspondence case. Since computing a partial correspondence is an important problem, we discuss such methods in detail in Section 5.5.

Dense vs. sparse: the advantage of defining the problem in terms of a sparse correspondence is that the complexity of the computation (in time and space requirements) might be reduced by considering smaller sets of elements. Some techniques were designed by taking this view into consideration, such as the search-based methods described in [GMGP05, ZSCO*08]. Despite their associated exponential search space, these methods can be utilized in practice by considering a sparse set of feature points extracted from the shapes. Other methods do not make a strong distinction between these two cases and work interchangeably for computing both sparse and dense correspondences, since their complexity can be thought of as increasing linearly with the number of elements in the shapes (e.g., the optimization methods discussed in Section 5). Finally, a whole collection of methods were created to compute a dense correspondence from an initial sparse one, such as the cross-parameterization methods surveyed in [Ale02] or more general algorithms which take an initial set of markers as input for computing a non-rigid correspondence [ACP03, SP04, PMG*05, SSB05].

4.3. Objective function

The objective function provides a measure of how good a given correspondence is, or how far it is from the desired solution. It can sometimes be referred to as the *error measure*, the *cost function*, or the *energy*, in the case of methods that formulate the problem as the minimization of some energy function. Its formulation depends on the type of input dataset that we are considering (points, surfaces, etc.), and also on the specific problems that we desire to solve (partial correspondence, rigid alignment, etc.). Here, we will mainly look at the objective function from the perspective of the overview given in Section 3.

Similarity-based correspondence: when we are interested in finding a correspondence between two datasets without first aligning the shapes, we need to utilize shape descriptors and intrinsic measures to quantify the quality of the correspondence. Therefore, for two shapes $\mathcal{P}$ and $\mathcal{Q}$ and a correspondence relation $\mathcal{R}$, the objective takes the form

$$\mathrm{Obj}(\mathcal{P},\mathcal{Q},\mathcal{R}) = \mathrm{Sim}(\mathcal{P},\mathcal{Q},\mathcal{R}) + \alpha\,\mathrm{Distor}(\mathcal{P},\mathcal{Q},\mathcal{R}), \quad (1)$$

with a similarity term that is linear on the number of feature points and a distortion term that is usually quadratic on the number of feature points, since it commonly involves comparing properties of pairs of points. The weight α controls the influence of each term in the objective function. Automatically setting α to a value that reflects the user goal can also be a challenging problem [CMC*09].

Similarity term: this term encodes the similarity of the shape descriptors of points in correspondence. The descriptors can encompass geometric attributes such as point normals or local frames, which can give an indication of whether the orientation of the points is coherent across the two matched point sets [ASP*04].

Distortion term: the distortion term quantifies how much the shapes would be deformed if brought into correspondence. A common candidate for a distortion measure is the disparity in the distances between pairs of matched points. The disparity is an approximate way of measuring the distortion introduced by the correspondence without having to first align the shapes. It can be expressed as

$$\mathrm{Dispar}(\mathcal{P},\mathcal{Q},\mathcal{R}) = \sum_{\substack{\{p_1,p_2\}\subset\mathcal{P}\\ \{q_1,q_2\}\subset\mathcal{Q}}} \mathrm{Dispar}(p_1,p_2,q_1,q_2), \quad (2)$$

where $(p_1,q_1)\in\mathcal{R}$ and $(p_2,q_2)\in\mathcal{R}$. The disparity term between two pairs $\{p_1,p_2\}$ and $\{q_1,q_2\}$ can be given by the difference in the distances between the pairs of points. Any appropriate distance measure can be used. Examples include

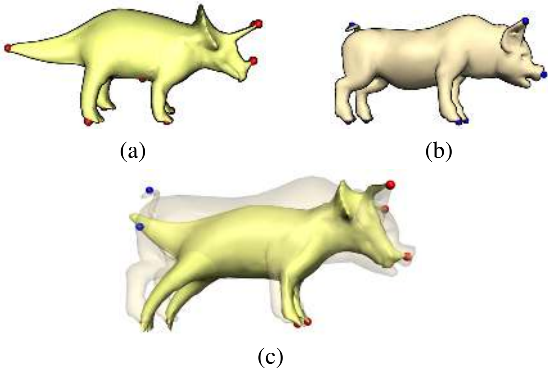


Figure 8: *Example of using a deformation to assess the quality of a correspondence (with the method described in [ZSCO*08]): the triceratops in (a) is deformed to match the pig in (b), according to a set of pairwise assignments between feature points (colored in red and blue). The result of the deformation is shown in (c).*

Euclidean distance [CZ08],

$$\text{Dispar}(p_1,p_2,q_1,q_2) = \big|\|p_1-p_2\| - \|q_1-q_2\|\big|, \quad (3)$$

or geodesic distance, in the case of surfaces [ASP*04],

$$\text{Dispar}(p_1,p_2,q_1,q_2) = \left|\text{geod}_{\mathcal{P}}(p_1,p_2) - \text{geod}_{\mathcal{Q}}(q_1,q_2)\right|, \quad (4)$$

where $\text{geod}_{\mathcal{S}}(s_1,s_2)$ is the geodesic distance between s_1 and s_2 on the surface $\mathcal{S}$.

Deformation: a more elaborate form of quantifying distortion is to use a global deformation measure, such as by the methods described in [HAWG08,ZSCO*08]. Once we know which feature points should be matched, we deform one shape into the other so that the matched points are aligned. An example of this procedure is shown in Figure 8. Notice that this is different from the non-rigid deformation case, since here we already have a correspondence and just need to estimate its distortion, while on the former case we deform the shapes to establish the actual correspondence. For this step, we can utilize one of the recently proposed deformation methods (as in the surveys [Sor06, SSP07]). Then, measuring how much the surfaces had to deform to align to each other (an intrinsic rigidity energy) gives an indication of the distortion introduced by the correspondence. An advantage of using a surface-based deformation energy is that it is able to differentiate between correspondences that switch symmetric parts of the shape (Figure 9), which usually pass undetected when only pairwise distances are utilized.

Rigid alignment: for the problem of rigid alignment between two or more datasets (e.g., point sets), the objective is commonly defined in terms of the number of matching points, or given by a metric that quantifies how well the datasets align to each other.

Largest Common Pointset (LCP): here we are interested in finding a transformation that brings the largest number of points into correspondence [IR96, AMCO08], given a threshold ε which indicates if two points are close enough and can be considered as matching to each other. Therefore, the objective is to maximize the cardinality of the set of matched points, which can be expressed for two point sets $\mathcal{P}$ and $\mathcal{Q}$ as

$$LCP(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}} \text{Match}(p,\mathcal{Q}), \quad (5)$$

where

$$\text{Match}(p,\mathcal{Q}) = \begin{cases} 1 & \text{if } \exists q \in \mathcal{Q}, \text{ s. t. } \|p-q\| < \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

for some distance measure $\|\ldots\|$.

Geometric distance: another common objective does not rely on such a parameter ε but minimizes the alignment error given by the sum of squared distances between points. That is, for each point in the transformed set, we find the closest point in the reference set and add the distance between these two points to the error measure, expressed as

$$\text{Dist}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}} \text{Dist}(p,\mathcal{Q}), \quad (7)$$

where

$$\text{Dist}(p,\mathcal{Q}) = \min_{q\in\mathcal{Q}} \|p-q\|. \quad (8)$$

This is the common measure utilized in algorithms such as the Iterated Closest Point (ICP) [RL01]. Variants of this scheme can also be utilized, e.g., by adding orientation or surface information [CM92], where $\text{Dist}(p,\mathcal{Q})$ is replaced by a more elaborate point-to-surface measure when our datasets $\mathcal{P}$ and $\mathcal{Q}$ are given as surfaces.

From the above, the LCP formulation has the advantage that partial matching can be directly handled by the objective function, since the largest set of matching points will correspond to the region of overlap between the two point sets. The sum of squared distances will necessarily consider all the points in the objective, unless we also incorporate some estimate of the amount of overlap between the point sets or a threshold to quantify points that are too far away from each other [RL01].

Non-rigid alignment: in the case that the shapes are matched to each other by deforming them in a non-rigid manner, the objective will have to incorporate terms to quantify when such a transformation is meaningful. That is, if each vertex can move freely according to its own transformation or displacement, we need to enforce some form of global consistency (regularization). Such a regularization can be obtained by limiting the number of degrees of freedom of the transformations or by penalizing large deformations. The error measure in the non-rigid case resembles the one for similarity-based matching (Equation 1), and will typically incorporate the following two terms.

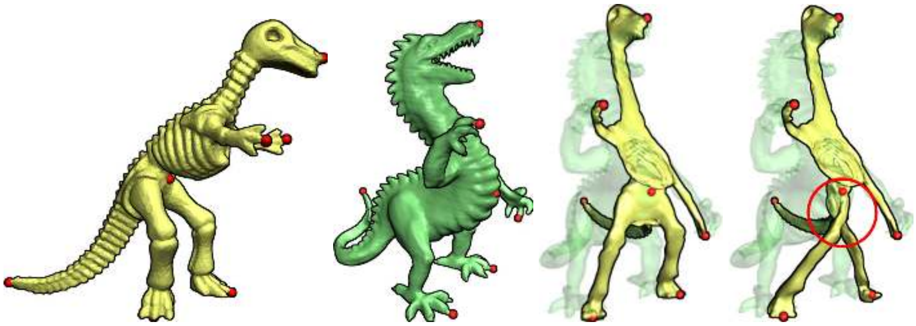


Figure 9: *When searching for a correspondence between the two dinosaur models shown on the left, a rigidity energy assigns a higher value to a correspondence that switches the limbs of the dinosaur (to the right), than to a correspondence that does not switch any symmetric parts of the shape [ZSCO*08].*

Alignment quality: a measure to quantify how well the datasets are aligned by the transformation, given by a measure of geometric distance (similar to the rigid case discussed above) [SP04, PMG*05] or distance to plane in the case of surfaces [ACP03].

Regularization: a term which enforces the global consistency of the transformations by demanding that the transformations of neighboring vertices are similar (which provides a smooth transition of transformations from one vertex to the other). Such a transformation similarity can be measured in a direct manner (e.g., by the norm between the matrix representations of the transformations [ACP03, SP04]) or according to derivatives of the transformations [PMG*05].

4.4. Solution approach

There is a variety of techniques that can be utilized to search for the best correspondence. In terms of the solution paradigm, there are methods that search for a transformation that aligns the shapes, methods that only consider the pairwise assignments between elements and find a solution using well-known optimization methods or search techniques, and methods that perform a hybrid search, alternating between alignment and correspondence computation. We discuss these solution strategies in detail in Section 5. Here we consider additional classification criteria based on particular properties of the solution approaches.

Fully-automatic vs. semi-automatic: semi-automatic methods require user input, such as a proper initialization or a set of corresponding landmarks between the shapes. Automatic methods do not require any user input besides a few parameter values. Semi-automatic methods include the approaches for cross-parameterization [Ale02] and methods that take markers as input [ACP03, SP04, PMG*05, SSB05].

Although user input is required for the proper initialization of certain methods, this can also be seen as a necessary requirement when the semantics of the shapes cannot be easily inferred. Therefore, a track open for future research is that of methods based on a feedback loop, where the user gradually improves a correspondence based on his or her preferences. Ideally, such a method would minimize the amount of user interaction and provide hints on what information is missing to refine the correspondence.

Global vs. local search: the distinction here is whether the method explores the whole solution space in search of a good solution (e.g., by performing an exhaustive search [GMGP05, ZSCO*08]) or whether the results of the method depend directly on its initialization. The initialization can be given by a user, such as in the case of semi-automatic methods. However, it can also include automatic initializations which will be used as the starting point for the local search of the algorithm.

The most prominent example of the local search category is the ICP [RL01] algorithm, which alternates the computation of correspondences between points (given by the closest points) with the computation of an aligning transformation. Since this iterative process follows a single path in the solution space, it can end up with a result that is a local minima. Thus, the initial state clearly influences the final result of the algorithm, and therefore different forms of initialization have been proposed for this algorithm (which take the form of computing a prealignment between the shapes to be matched). These are discussed in detail in Section 5.

Another example of algorithms that perform local search are the methods for non-rigid alignment based on explicitly computing the transformation for each shape element. Since these transformations are computed with a method based on gradient descent or Newton's optimization, the initialization will also necessarily influence the final correspondence result [ACP03, SP04, PMG*05].

Pairwise vs. groupwise: methods for group correspondence appear predominantly in the computational anatomy community [HM09], where it is important to obtain a coherent correspondence for a group of shapes, so that an accurate

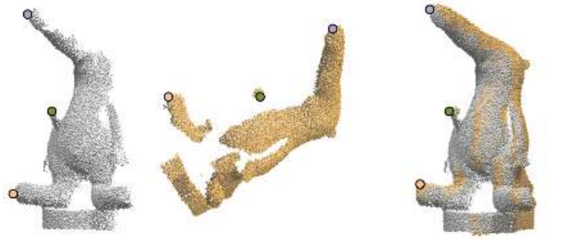


Figure 10: *Example of the alignment procedure in 3D for two partial scans of the Coati model (shown to the left): three points are sampled on each scan (the colored dots) and a rigid transformation is derived. If the transformation aligns the largest possible number of points (to the right), then we have found the transformation sought [AMCO08].*

statistical model can be constructed. Therefore, methods targeted at this problem seek to optimize the group correspondence so that the generated statistical model is as compact as possible [DTC*02]. This optimization typically consists in making local adjustments to the correspondence of prealigned shapes (a process known as landmark sliding).

Although the term *group correspondence* is not used in the field of time-varying reconstruction, a certain class of methods applied to this problem can also be seen as following this approach, since all the scans are considered simultaneously in the registration. The difference to the case of anatomical shapes is that each scan can deform over time and there can be a significant amount of missing data between frames, while in the anatomy case we typically search for a full correspondence between complete shapes, which are seen as variations from the same mean shape of an organ or bone. These methods pose the time-varying reconstruction problem as the reconstruction of a space-time surface [MFO*07, SWG08, SAL*08], or obtain a skeleton that is coherent for all the time frames [ZST*10]. The advantage of such a formulation is that missing data can be filled in with data from frames that are further away in time.

5. Main representative methods

We use the classification based on how the correspondence is obtained as the starting point to cover the discussion of the individual methods. Thus, the methods are primarily classified into those that search for an aligning transformation, those that search directly for a correspondence without performing alignment, and the ICP method, which works in a hybrid manner alternating between transformation search and correspondence search. Next, we discuss the use of embeddings, which can be applied for non-rigid alignment by combining them with methods for the rigid case, and then we conclude with a discussion on methods for computing partial correspondences.

It is worth noting that most of the methods discussed here are also applicable to images. The main differences are in the procedures that extract feature points, compute shape descriptors, and quantify the distortion introduced by a correspondence, which are dependent on the data representation. In the case of images, the problems of measuring distances and preserving the neighborhood structures of elements are simplified by the regular parameterization that is enforced by these datasets [TH08]. Furthermore, we might also have to consider different types of transformations (e.g., projection of a 3D shape onto a 2D plane), for problems such as matching stereo images or registration of images taken from different viewpoints [Bro92].

5.1. Transformation and alignment search

Recall that the methods in this class first search for a transformation that aligns the shapes, and then derive the correspondence from the proximity of the aligned elements.

Rigid alignment: the methods in this class rely on the fact that the transformations used for alignment can be derived from a small set of sample of points. For example, if considering a rigid transformation between two point sets in 3D, we can derive its parameters (given by a rotation matrix and a translation vector) from an initial configuration of three points and their transformed positions. After sampling a transformation, we can either *verify* its quality in aligning the shapes or *vote* on it.

Sample and verify: under assumptions of rigidity, a naive alignment algorithm is given by a direct application of the sampling idea above [HU90]. Specifically, we take three points from the first shape, three points from the second shape, derive a rigid transformation, and test how well the transformation aligns the two shapes (e.g., with one of the objectives discussed in Section 4.3). This is illustrated in Figure 10. After testing all possible triplets of points in the two shapes, we return the best transformation. This naive algorithm in 3D has a complexity of $O(m^3n^3)$ for sampling the triplets of points and $O(m \log n)$ for the verification, yielding a total complexity of $O(m^4n^3 \log n)$ for aligning two point sets of size m and n.

Clearly, such an algorithm is far from efficient. Thus, different modifications have been proposed to improve upon this idea. Approaches following the philosophy of the random sample consensus (RANSAC) method propose to randomize the different steps in the procedure described above [FB81]. Instead of sampling all possible triplets of points, we can consider only a random sample of points in one shape, reducing the complexity by a factor of $O(m^3)$. Furthermore, we can also randomize the verification step, reducing the complexity by another factor of $O(m)$ in the typical case [IR96].

We can also explore the geometric invariances maintained by the transformations. One such case is the ratio between three coplanar points, which is preserved by rigid and affine

Approach	Method	2D		3D	
		Time	Space	Time	Space
Alignment	Naive algorithm	$O(m^3 n^2 \log n)$	–	$O(m^4 n^3 \log n)$	–
	Randomized	$O(mn^2 \log n)$	–	$O(mn^3 \log n)$	–
	Randomized verification	$O((n^2 r + lm) \log n) \approx O(n^2 \log n)$	–	$\approx O(n^3 \log n)$	–
	Sets of 4 coplanar points	$O(n^2 + k)$	$O(n)$	$O(n^2 + k)$	$O(n)$
Pose clustering	Naive algorithm	$O(m^2 n^2 + h)$	$O(h)$	$O(m^3 n^3 + h)$	$O(h)$
	Randomized	$O(mn^2 + h)$	$O(h)$	$O(mn^3 + h)$	$O(h)$

Approach	Method	Pre-processing	Space	Query
2D Geometric hashing	Original algorithm	$O(m^3 \log m)$	$O(m^3)$	$O(n^3 \log n)$
	Randomized	$O(r^3 \log r)$	$O(r^3)$	$O((n^3 + lm) \log n) \approx O(n^3 \log n)$

Table 2: *Complexity of rigid registration methods for two sets with m and n points. r is the size of a subset of points used for random verification. l is the number of times that such a subset of points match consistently and further verification is needed. k is the size of the output. h denotes the size of the accumulation table in pose clustering.*

transformations. Thus, the problem of searching for triplets of points that provide the optimal transformation can be transposed to that of finding four sets of coplanar points that share the same ratios [Hut91]. By pre-processing these invariances and keeping them in appropriate data structures that allow for efficient retrieval, output sensitive methods can be achieved [AMCO08], reducing the complexity of the alignment problem even further to $O(n^2 + k)$, where k is the size of the reported output.

Sample and vote: instead of sampling a transformation and evaluating its quality, we can simplify the verification step by voting on the transformation. For this purpose, pose clustering utilizes an accumulation table [Sto87, Ols97]. After enumerating two triplets of points and deriving a transformation, a vote indexed by the parameters of the transformation is stored in the table. At the end of this $O(m^3 n^3)$ process, the cells with most votes correspond to the best candidate transformations that align the point sets. Note that processing the accumulation table requires an extra $O(h)$ step dependent on the size h of the table.

Geometric hashing is a voting-based method that makes use of the concept of pre-processing to speed up the alignment [WR97]. The main idea here is to store in a hash table all the possible configurations of a group of reference point sets, so that when we seek the reference point set that best matches to a query point set, this search can be performed efficiently. We can informally look at this method as breaking the $O(m^3 n^3)$ complexity of the naive enumeration into an $O(m^3 \log m)$ pre-processing phase (which samples all the possible configurations of a reference set and stores them in the hash table) and an $O(n^3 \log n)$ query phase (which samples all the possible configurations of the query set and accumulates votes in the hash table to allow the retrieval of the best matching reference set). The increase in speed in the query phase is gained at the expense of utilizing more memory resources.

Remarks: as we see from the discussion, these techniques are all closely related and their complexity can be compared as in Table 2. Randomized versions of geometric hashing and pose clustering are also proposed in the literature. Note that a special case of rigid alignment arises when we wish to minimize the mean squared error between two point sets. In this case, we do not need to search for the best aligning transformation, but we can obtain it directly via the singular value decomposition of a covariance matrix [Ume91]. Another observation is that the principle of sampling and verifying transformations also applies to other contexts with different types of transformations. For example, in [LF09], Möbius transformations (which define a mapping from one Riemann sphere to another) are used to establish the correspondence between shapes that are approximately isometric. These transformations can also be derived from triplets of points sampled from each shape (as illustrated in Figure 7).

Piecewise-rigid alignment: the methods discussed so far utilize one global transformation to match one shape to another. A different class of methods generalizes this idea by applying transformations to local portions of the shapes. In [CZ08], these transformations are applied in a piece-wise rigid manner to establish a correspondence between articulated shapes. The problem is formulated as labeling the vertices of the shapes with candidate transformations, which where estimated from aligning the local frames of vertices with similar descriptors (Figure 11). Since now we are restricting the vertices to possess a transformation from a predefined set, this greatly simplifies the solution search, in contrast to the methods that allow any transformation to be assigned to a vertex. By also adding a regularization term to the labeling optimization, a grouping of the vertices into rigid components is guaranteed. An alternative to this approach is

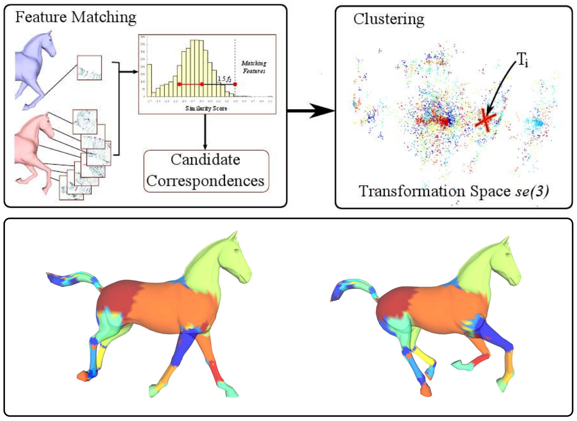


Figure 11: *Example of non-rigid alignment of articulated shapes: candidate correspondences are obtained by matching vertices with similar shape descriptors (top left). The candidates are used to derive transformations, which are then clustered (top right) and assigned to the shape vertices according to a labeling algorithm [CZ08]. The result of labeling two corresponding shapes is shown on the bottom.*

to explicitly fit the shapes to a kinematic skeleton of articulated bones, so that the skeleton can be used to track the movement of the shape and also infer in which regions there is missing data [PG08, CZ09, GSdA*09].

Non-rigid alignment: for the methods described in [ACP03, SP04, PMG*05], different transformations are assigned to each vertex on the shape. The problem is formulated as finding the best transformation that brings each vertex in a reference shape close to its counterpart in the target shape. The optimization is solved with a Newton-based method, and a regularization term is added to enforce the similarity of transformations across neighboring vertices. The difficulty in this setting is to avoid solutions that are local minima. This is achieved by initializing the methods with a set of corresponding marker points and solving the optimization in a multi-level fashion. Alternatively, we can also utilize Markov Random Fields for this optimization [PH03].

Instead of computing local deformations or displacements for the vertices, in [SSB05], the displacements are implicitly obtained by learning a function that warps one shape into the other. The warp is obtained by solving a convex optimization problem similar to learning a support vector machine classifier (which includes a form of regularization in its definition). Thus, global minima are avoided.

An extension of this class of methods which circumvents the need of marker points is proposed in [LSP08], where the alignment between two shapes is performed with two separated transformations: a global rigid transformation which roughly aligns the shapes and per-vertex affine transformations that bring the non-rigid shapes into full alignment. A robust alignment can also be obtained by deforming one shape into the other in terms of a 3D optical flow [dATSS07] or a Laplacian deformation of the meshes [dAST*08].

Image registration: deforming one shape into the other can also be achieved by adopting methods developed for image registration [MV98]. First, the shapes are transformed into 2D images (or 3D volumes) by mapping each feature point to its nearest pixel (or voxel). The value that is assigned to the pixel can be a vector of descriptors computed at the point [TH08], or the generated image can represent a level set function of the shape [HNM06]. Finally, the resulting images or volumes are registered by computing a global alignment followed by a non-rigid deformation. The advantage of such an approach is that we transform the shapes into a parameterized representation where a variety of registration algorithms can be utilized [MV98]. However, creating a volume with enough resolution to capture all the details on the shapes can imply considerable memory consumption in 3D.

5.2. Correspondence search

The main characteristic of the methods discussed in this section is that they work mainly with the pairwise assignments between feature points, without considering transformations that align the shapes. The correspondence problem is typically posed as optimizing an objective function of the form $\mathrm{Obj}(\mathcal{P},\mathcal{Q},\mathcal{R}) = \mathrm{Sim}(\mathcal{P},\mathcal{Q},\mathcal{R}) + \alpha \mathrm{Distor}(\mathcal{P},\mathcal{Q},\mathcal{R})$, as described in Section 4.3, which is based on the quality of pairwise assignments (a linear term) and the compatibility between pairs of such assignments (a quadratic term). The solution is found by using well-known discrete or continuous optimization methods. A special group of methods in discrete optimization utilize a tree-based search to explore the solution space. We discuss this specific category of methods in a special section.

Optimization: if the objective being optimized is only composed of a similarity term $\mathrm{Sim}(\mathcal{P},\mathcal{Q})$, then the formulation becomes a Linear Assignment Problem (LAP). This simplified objective can be solved by the simplex algorithm, since it is a special case of a linear program [PS82]. However, if we constrain the correspondence to a one-to-one mapping, the problem becomes that of finding an optimal matching in a weighted bipartite graph, which can be solved more efficiently by the Hungarian algorithm in $O(n^3)$ time, where n is the number of feature points in each shape [PS82].

On the other hand, if the objective comprises both the linear and quadratic terms, we arrive at a Quadratic Assignment Problem (QAP), which is known to be NP-hard [PRW94]. Several techniques have been proposed to compute approximate solutions to this problem. One group of methods poses the problem as an integer optimization, which is relaxed to the continuous setting and solved with a continuous optimization technique, e.g., the softassign technique [GR95] (which iteratively normalizes rows and columns of an affinity matrix), concave programming [MC03], approximations

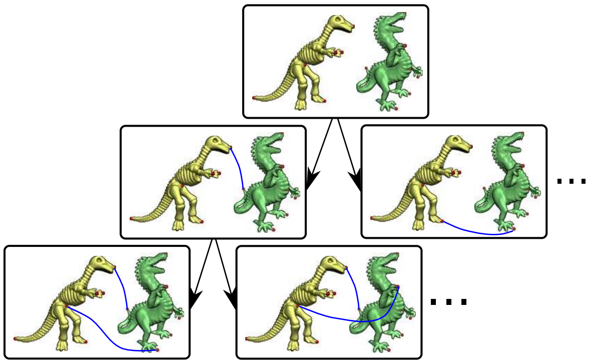


Figure 12: *Example of search-based correspondence for a set of feature points [ZSCO*08]: each node of the tree encodes a partial correspondence. All possible assignments are added when expanding a new level of the tree.*

based on linear programming [BBM05], spectral clustering [LH05], or relaxation labeling [ZD06]. It can also be formulated in probabilistic terms and solved as a convex optimization problem [ZS08].

Another group of methods solves the problem in the discrete setting without resorting to the continuous domain. One common solution approach in the discrete case is to solve the problem by computing an optimal labeling of a graph, e.g., we can pose the problem in terms of a Markov network where the set of labels corresponds to matching points on the target shape [ASP*04, ZST*10]. Other methods make use of metaheuristics for combinatorial optimization, such as ant colony optimization [vKHZW07]. We can also sample the space of correspondences in search of a solution, guided by geodesic distances and importance sampling [TBW*09].

2D contour correspondence: a considerable part of the literature has also focused on the specific case of 2D contour correspondence, since these datasets can be easily extracted from 2D images. A collection of techniques were developed by taking into account the fact that the vertices on a contour can be linearly ordered. This observation is used in combination with optimization techniques such as dynamic programming [LWZ*04, SN06], which can also be used to solve the problem posed in terms of computing shortest paths [MCH*06], and graph cuts [STCB07]. The last two techniques are able to find an optimal correspondence for two contours.

Tree-based search: one specific group of methods in discrete optimization find a solution by making use of tree-based search techniques, such as branch-and-bound, priority search, etc [GMGP05, FS06, ZSCO*08, ACOT*10]. During the expansion of the tree, each node represents a partial solution. A full solution is found by following the path from the root of the tree to one of its leaves. These techniques usually involve three important steps: expanding a node that represents a new partial solution (branching), estimating how far the partial solution is from the optimum solution (bounding), and eliminating nodes that will not lead to the optimum solution (pruning).

In the case of correspondence, solutions are mainly represented as collections of assignments between pairs of feature points, and the expansion step involves adding a new pairwise assignment to a given solution (Figure 12). Bounding and pruning can be performed by verifying the quality of the registration given by the current solution, either by aligning the shapes [GMGP05] or by deforming one shape into the other [ZSCO*08]. Other pruning methods include testing the compatibility between pairwise assignments, such as quantifying the distortion introduced in the Euclidean [GMGP05, FS06] or geodesic distances [ZSCO*08, ACOT*10] between pairs of points, or testing the agreement in the spatial configuration of the shapes [ACOT*10]. Naturally, the descriptors computed for the feature points are also considered in the bounding and pruning steps.

When a hierarchical or multi-resolution structure can be extracted from the shape representation, this information can also be considered in the solution search. Skeletons are commonly represented as trees or graphs for which a tree can be easily extracted. Therefore, in this context, it is common to resort to search-based algorithms that take this hierarchy into account [SSGD03]. Methods of a more greedy nature can also benefit from such hierarchical [BMSF06] or coarse-to-fine representations of the shapes [HSKK01].

5.3. ICP and variants

This section discusses mainly the ICP method, which iteratively computes a correspondence by alternating between two steps. In the first step, we search for an alignment between the shapes, while in the second step, we derive a correspondence from the alignment. Finally, we reiterate this procedure by using the correspondence to estimate a new aligning transformation. Thus, we call it a hybrid search method, since it searches for both alignment and correspondence solutions, which in turn affect each other. The different variants of the ICP algorithm are obtained when the two steps are solved in different manners (e.g., by changing the type of transformation or the way in which we determine the correspondences) [RL01].

Rigid alignment: in the "classic" variant of the ICP algorithm for rigid alignment, given two point sets $\mathcal{P}$ and $\mathcal{Q}$, we establish a correspondence between every point $p \in \mathcal{P}$ and its closest point in $\mathcal{Q}$, according to a given distance metric. Next, from all the pairwise assignments that were defined in the previous step, we estimate the best rigid transformation that aligns the two point sets (by solving a linear system) and realign the point sets according to this transformation. Finally, we repeat the two-step procedure, stopping when there is no significant change in the estimated transformation.

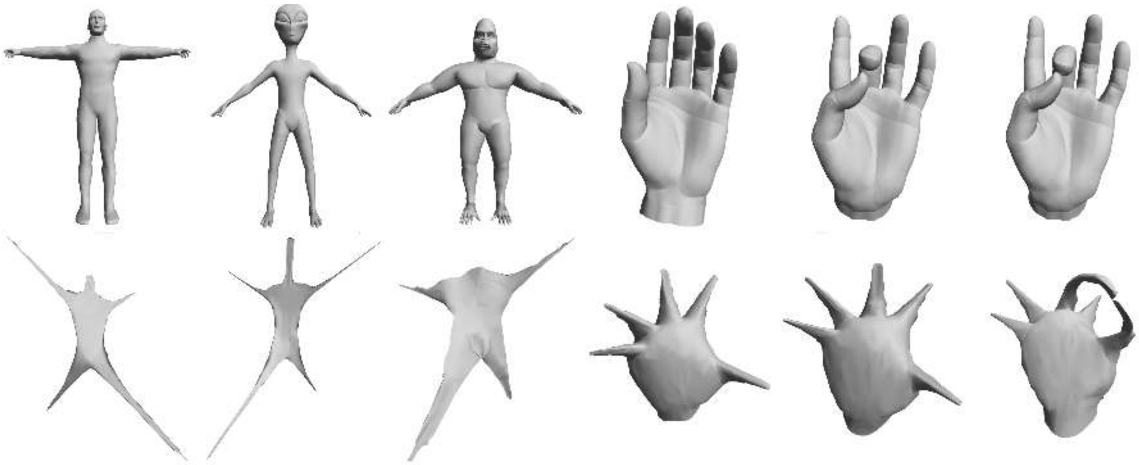


Figure 13: *Example of shape normalization: by applying MDS to the meshes on the top row, we obtain the embeddings on the bottom row. Note how the pose of limbs and fingers is normalized, turning the non-rigid alignment problem into that of rigid alignment. The last column shows an example containing topological noise, where two fingers were connected after the reconstruction [GSCO07].*

As we can infer from this description, the initial position of the point sets tremendously influences the final result of the ICP algorithm, since the first correspondence is derived from this initial configuration. Thus, a crucial step in ICP-based methods is to perform a prealignment of the shapes so that the algorithm does not get trapped in local minima. Different forms of prealignment have been proposed in the literature to address this issue. The classic solutions are to rely on a set of matching feature points, an initial set of markers given by a user, or to automatically prealign the shapes with Principal Component Analysis (PCA) [RL01]. Recently, prealignment based on the reflectional symmetry axes of the shapes has been suggested as another effective solution [PSG*06].

Non-rigid alignment: the ICP method can also be used for non-rigid alignment by modifying some of its components. One set of methods computes a weighted correspondence where each assignment has an associated confidence value. These confidences are crucial for robust outlier detection. After rigidly aligning the shapes with the weighted ICP, the shapes are non-rigidly deformed into each other by computing a warp function based on thin-plate splines [CR03, BR07]. In [HAWG08], the rigid transformation of the original ICP algorithm is substituted by a deformation based on rigid-body components, in order to address the registration of approximately isometric shapes. Moreover, the alignment of articulated shapes can also be obtained by first embedding the shapes in a space that normalizes them for bending (discussed in Section 5.4), and then rigidly aligning the shapes in this space.

Recently, variants of ICP are also utilized in the context of time-varying surface reconstruction to align the geometry of adjacent time frames [WJH*07, PG08, WAO*09, LAGP09]. If a sufficient number of scans is acquired per unit of time, we can assume that only small changes take place in the spatial configuration of the shapes (i.e., we can consistently track their rigid-body components), and so the initial alignment of each frame is not a strong issue in the registration.

5.4. Use of embeddings

The non-rigid alignment of shapes, especially that of articulated shapes, can also be accomplished by first embedding the shapes in a space where the configuration of the rigid parts is normalized, and then treat the problem simply as a case of rigid alignment in this embedding space (shown in Figure 13). The rigid alignment can then be obtained by any of the methods discussed so far. The key to create such an embedding is to obtain an intrinsic representation of the shape which is invariant to bending (e.g., by collecting geodesic distances between surface points) and then utilize this representation to embed the shape in a new ambient space, so that the intrinsic geometry of the shape is translated into its extrinsic geometry in this new space. This embedding can be obtained with techniques such as Multi-Dimensional Scaling (MDS) [EK03, BBK06, BBBK08] or the spectral transform [JZvK07, MHK*08, SY10]. A comprehensive coverage on different forms of embeddings is given in [ZvKD07]. Embedding the shapes for normalization can also be used for other types of datasets (Figure 14).

5.5. Partial correspondence

Since computing a partial correspondence is an important specialization of shape correspondence, we discuss in this

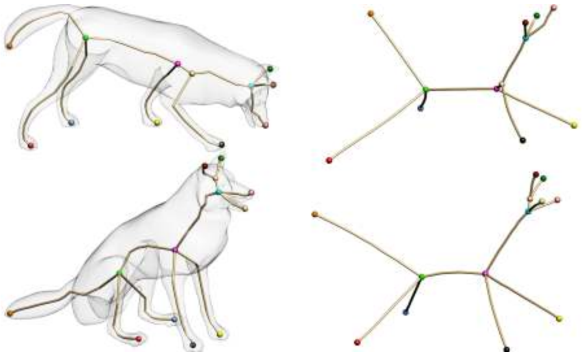


Figure 14: *The skeletons of the dog in two different poses (to the left) are normalized by applying MDS (to the right). The embeddings are used to measure the spatial consistency of a correspondence, which acts as a pruning test in the search-based method of [ACOT*10].*

section a few strategies proposed for this task. The problem is defined as finding a subset of shape elements for which a meaningful correspondence can be computed, as opposed to finding a full correspondence that could include additional parts or features which do not exist in both shapes. This task can be seen as composed of two subproblems: searching for an optimal subset of k feature points that match consistently, and finding the correspondence between these k elements.

One approach to determine the subset is to examine the objective function in search of sharp increases in the alignment error, which appear when an outlier point is added to the set of matched points [GMGP05, ZSCO*08]. Alternatively, we can add to the optimization problem an estimate on the number of outlier features, which limits the number of points that appear in the computed correspondence [MC03, BBM05]. Such an estimate can be derived from the data itself [OEK08].

Another strategy to determine the subset of features is to rely on voting [LF09, ACOT*10]. Here, we compute a series of candidate correspondences and cast a vote on the pairwise assignments that constitute each candidate. At the end of this process, the assignments that are certain emerge as the ones with the highest number of votes, while the assignments relating points that do not have any meaningful matching possess a negligible quantity of votes. The reason for sampling candidate correspondences is that this procedure acts as a form of group reinforcement, where we only vote on an assignment if it can be part of a consistent correspondence.

It is also natural to look at the partial correspondence problem as that of matching two graphs. We can consider the feature points on a shape or skeleton as the nodes in a graph, and connect every pair of nodes with an edge whose weight is proportional to some geometric quantity (e.g., distance between the nodes). Then, partial matching becomes the problem of subgraph isomorphism, whose decision variant is known to be NP-complete. Since different heuristics have been proposed to address this problem, it is also possible to use these methods for shape correspondence. For example, the notion of matching two graphs by finding a set of operations that transform one graph into the other (e.g., by merging nodes [NB07]) is used to derive heuristic algorithms for skeleton matching in 2D [SKK04] and 3D [BMSF06]. Then, the subset of matching features is given by the nodes that are not removed during the editing of the graph.

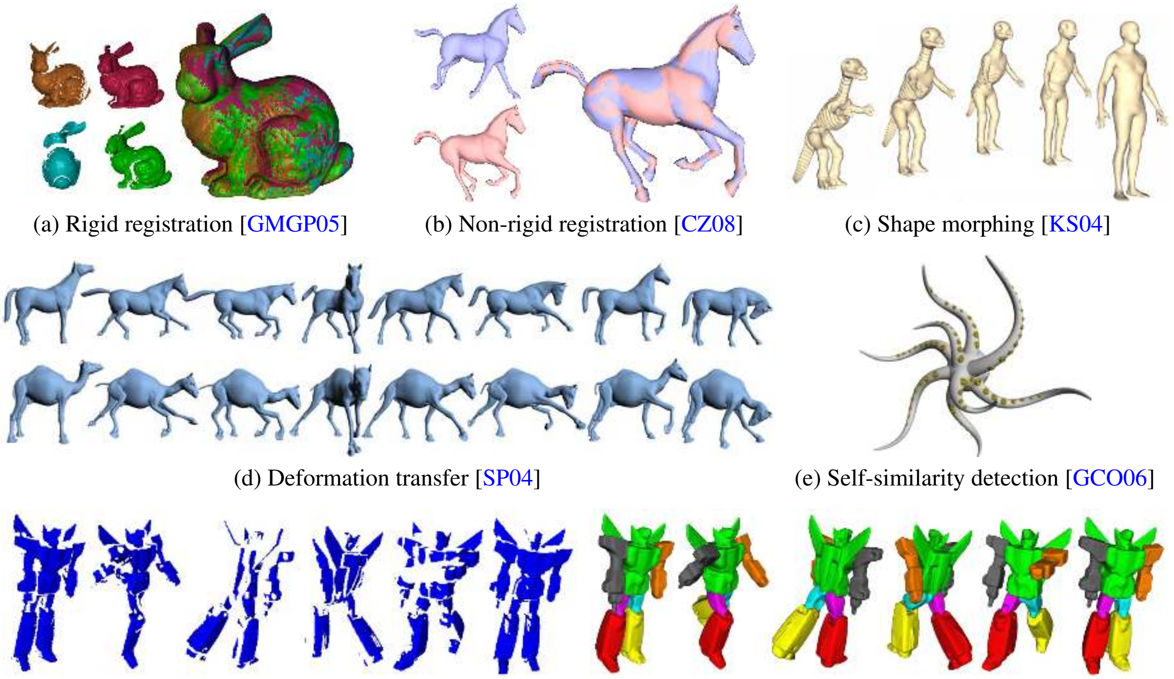
6. Validation of correspondence methods

Validation is an important and necessary aspect of the correspondence problem, since we need to be able to effectively compare the results of the different methods. The most common form of validation is certainly the visual inspection of the results. Displaying a morph between shapes is also a similar way of assessing the visual quality of the correspondences, where we expect to see a smooth transition from one shape to the other for a good correspondence [KS04, ZSCO*08]. Such procedures allow for a qualitative comparison of the results. However, since this form of evaluation can be subjective, more objective or quantitative procedures are also sought.

One possibility for more objective comparisons is to use the output of an objective function (Section 4.3) to derive a similarity measure, so that the correspondence method can be indirectly evaluated in terms of retrieval [FS06, JZ07]. The assumption is that the accuracy of the retrieval results will be proportional to the accuracy of the computed correspondences. Additionally, in the computational anatomy community, it is also common to assess the computed correspondences in terms of the quality of the statistical shape models that they generate, which can be evaluated with measures such as generality, specificity, or compactness [DTC*02, SRN*03, KE06]. As in the case of retrieval, the assumption is that more accurate correspondences will lead to better statistical models.

However, to allow for a direct comparison of the results, the most reasonable procedure is to utilize a set of shapes already in correspondence, so that the computed correspondences can be compared against the ground-truth. This can be achieved with a discrete measure such as the Hamming loss (counting the number of points incorrectly matched [CMC*09]) or more continuous measures such as the endpoint error (where, for each point, we compute the distance from its matching point to its known ground-truth). We can add up the endpoint errors [KE06] or compute statistics on them [CMC*09].

These considerations lead us to the question of a benchmark for shape correspondence. Benchmarks have been developed for other important problems, such as mesh segmentation [CGF09], however, there is no specific benchmark



(a) Rigid registration [GMGP05]

(b) Non-rigid registration [CZ08]

(c) Shape morphing [KS04]

(d) Deformation transfer [SP04]

(e) Self-similarity detection [GCO06]

(f) Time-varying surface reconstruction [PG08]

Figure 15: *Examples of applications that make use of correspondence methods: (a) a set of scans (to the left) is rigidly aligned to reconstruct the shape of the bunny (to the right), (b) the horse in two different poses (to the left) is non-rigidly aligned (result shown to the right), (c) the dino-skeleton is morphed into the human, (d) the motion defined on the horse (top line) is transferred to the camel (bottom line), (e) an application of partial matching: the suction cups on the tentacles of the octopus are detected as being similar (highlighted in yellow), (f) a set of range scans of an object in motion (shown to the left in blue) provides a single reconstructed model on which the motion is defined (shown to the right in color).*

for general shape correspondence problems. A number of datasets developed for shape retrieval and analysis have been utilized for this task. Shapes from the well-known *Princeton Shape Benchmark* [SMKF04] and *McGill 3D Shape Benchmark* [SZM*08] have been widely used for the comparison of methods that work on articulated shapes. Moreover, the datasets related to the various tracks of the *Shape Retrieval Contest* (SHREC) have also been considered. These datasets are aimed at various problems such as partial shape retrieval and retrieval of specific models, e.g., CAD models and faces. Recently, datasets involving non-rigid deformations have been made available, such as the *SCAPE dataset* [ASK*05], the *Non-rigid World Dataset* [BBK06] and the *TOSCA* dataset [BBK08].

From the above datasets, the TOSCA dataset can be characterized as a true benchmark for correspondence, since the models have a compatible triangulation (i.e., the ordering of vertices corresponds across all the deformations of a same model), and thus the correct correspondences can be implied from the models. The creation of a dataset with ground-truth correspondences has also been considered for anatomical shapes, where a reduced set of shapes in correspondence was enlarged by applying random (yet realistic) deformations on the original shapes [HJT08].

However, there is still the absence of a benchmark directed at more general correspondence problems (e.g., including shapes of a same class that possess significant geometric differences). Such a dataset would have to include a collection of challenging correspondence cases and also a list of prominent features on the shapes and their correspondence across different shapes.

7. Applications of correspondence

In this section, we discuss what we view as the most important applications that make use of correspondence methods. Examples of these applications are illustrated in Figure 15.

Shape registration: given a number of scans in arbitrary initial positions, the goal of registration is to match regions that correspond across the scans, so that the scans can be aligned and the target object can be fully reconstructed. We can either assume that the shapes do not change during the scanning (rigid registration) [RL01, AMCO08] or that they are free to deform during the acquisition (non-rigid registration) [ASP*04, JZvK07, BR07, CZ08, HAWG08].

Time-varying surface reconstruction: the challenge in this area is to take the great amount of generated data and organize it into one single model that represents a deforming shape. Correspondence methods are central to time-varying surface reconstruction, since the point sets of different frames have to be registered to yield the final model [MFO*07, WJH*07, SAL*08, PG08, LAGP09, CZ09, TBW*09, ZST*10].

Shape interpolation: the goal in interpolation or morphing is to gradually transform one shape into another. The transformation has to satisfy certain aesthetic requirements, so that the gradual change of the shape has a visually pleasing aspect [Ale02]. One important property in this aspect is that the correspondence relating the starting shape to the target shape should be meaningful, i.e., it should correspond parts in the shapes that are semantically equivalent.

Information transfer: A task that is becoming common is to transfer information from a source 3D object to a target 3D object, especially to enable the reuse of attributes or motion information associated to the source shape. Examples include transferring the deformation or morph from one mesh to another [SP04], or transferring textures while deforming a mesh [DYT05]. Such a task clearly requires a correspondence, since we need to obtain a motion or attribute for each element in the target shape from its corresponding element in the source shape.

Recognition and retrieval: understanding a scene described by a range image is certainly one of the classic challenges in computer vision [FP03]. Shape correspondence is one of the approaches that can be used for this task. Given a query object in the scene, we compute a correspondence between the query and models in a database. Next, the identity of the object is inferred from the best match to one of the models (according to a correspondence quality measure). A similar procedure can be utilized for shape retrieval [FS06].

Statistical shape modeling: the analysis of anatomical structures such as organs or bones can be facilitated when a statistical shape model is available. These models are useful for extracting shapes from images, since they are able to describe the valid variations in the appearance and the size of a shape. Such a model can be constructed from a group of shapes that represent the same structure. One of the requirements of building such models is an accurate correspondence between the group of shapes [DTC*02, HM09].

Change detection: another application of correspondence is to track changes in a shape (e.g. displacements, growth) over time. Examples include, in the medical field, tracking the change in the number and density of moles on a patient's skin (for cancer prediction), which is posed as a problem of point cloud correspondence [MHL09], and in remote sensing, tracking the change over time in the structure of cities and their land usage [LLFM00].

8. Previous surveys on shape correspondence

Surveys that reviewed a partial list of the topics covered in this report have appeared before. We discuss the most recent and relevant references here.

Tangelder and Veltkamp [TV08], Iyer et al. [IJL*05], and Bustos et al. [BKS*05] provide comprehensive surveys on content based 3D shape retrieval, mostly discussing global shape signatures. Biasotti et al. [BFF*07] present a tutorial on 3D shape description based on properties of real functions, which can be used as signatures for shape matching.

In terms of correspondence surveys, Alexa [Ale02] covers recent advances in mesh morphing, where the computation of dense correspondences from an initial set of sparse correspondences is also part of the discussion. Veltkamp and Hagedoorn [VH01] discuss similarity measures and algorithms for shape matching. This is certainly the publication that is the most similar to this report, and includes a solid introduction to the correspondence problem. However, most of the survey focuses on similarity metrics, while the discussion on the algorithms is brief, and important new developments have appeared since the publication of that survey.

The book by Bronstein et al. [BBK08] is a recent source for a comprehensive discussion on the analysis of non-rigid shapes. It covers problems such as determining the similarity between non-rigid shapes and techniques for correspondence based on MDS.

Audette et al. [AFP00] present an overview of surface registration techniques for medical imaging. The survey discusses in detail techniques based on applying geometric transformations to the data (rigid-body motion, etc), but does not go into techniques that work in the correspondence space. Heimann and Meinzer [HM09] survey the topic of statistical shape modeling and briefly review techniques for registration and group correspondence.

Some of the aforementioned surveys only discuss the related topics of shape retrieval and shape descriptors. Moreover, the publications that focus specifically on shape correspondence only cover a subset of the available methods, and the recent developments in the field are not discussed.

9. Open problems and future perspectives

Finding a meaningful correspondence between shapes belonging to the same class but differing geometrically, structurally, even topologically remains a challenge, since methods based on assumptions of rigidity, isometry or direct geometric similarity are inadequate. The more challenging scenario involves man-made shapes, where the objects differ not only by geometric deformations, but also by the part constitution of the shapes (as the example shown in Figure 2). In this case, shape correspondence departs from the low-level sphere of geometry analysis and becomes the higher-level problem of semantic reasoning, where we seek to recognize the parts of shapes and infer their semantics or functionality. A meaningful correspondence can then be established between the recognized parts. The utilization of prior knowledge is one of the possible solutions for this problem, where the main difficulty is how to model the knowledge and make use of it effectively.

Works which made the first steps towards knowledge-driven shape correspondence have recently appeared. Knowledge can be incorporated by utilizing a set of examples where a few landmark points have already been matched by an expert user [WH07], or by using a set of examples already in full correspondence [PDT07]. Another direction is to learn how the terms in the various objective functions should be weighted, depending on the restricted domain of the problem that we are considering [CMC*09]. One more possibility is to consider group information when performing correspondence-related tasks, such as skeletonization [WH09]. Finally, it would also be interesting to apply ideas related to feature selection to the correspondence problem [GE03]. Given a set of different shape descriptors computed for a dataset, these techniques seek to learn which subset of the descriptors (or alternatively, which weighted combination of the descriptors), gives the best correspondence results for a class of shapes.

On the other hand, a great variety of effective correspondence methods are already available, yet their objective validation is difficult. Therefore, we see the urgent need for a benchmark designed specifically for shape correspondence, so that automatic methods can be compared in a qualitative and quantitative manner, and their strengths and deficiencies clearly exposed. The envisioned benchmark would possibly contain different classes of problems (rigid, non-rigid) and should certainly also contain a collection of man-made objects (as shown in Figure 2), which would serve as the set of examples that are one of the current challenges in the field. To alleviate the amount of work required to create such a benchmark, one possible strategy is to take a reduced set of shapes for which the correspondence ground-truth is known and deform the shapes with random (yet statistically or physically plausible) transformations, similar to what is done for landmarks of anatomical shapes [HJT08].

We have seen the recent development of a benchmark for mesh segmentation [CGF09], where human participants provided a set of segmentations that they considered meaningful. This naturally leads to the question of how humans tend to segment shapes and then establish a correspondence between these segments. Such a correspondence reflects the process of recognition. Hence it seems apparent that there is a deep connection among the three tasks: segmentation, recognition and correspondence. Therefore, we conjecture that the ultimate approach for the semantic analysis of shapes would not treat these problems separately, but solve them all simultaneously and with the aid of group or prior information.

Acknowledgements

We would like to thank the reviewers for their valuable comments and the authors of the publications whose figures were reproduced in this report.

References

[ACOT*10] Au O. K.-C., Cohen-Or D., Tai C.-L., Fu H., Zheng Y.: Electors voting for fast automatic shape correspondence. *Computer Graphics Forum (Proc. EUROGRAPHICS) 29*, 2 (2010).

[ACP03] Allen B., Curless B., Popović Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 587–594.

[AFP00] Audette M. A., Ferrie F. P., Peters T. M.: An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis 4*, 3 (2000), 201–217.

[Ale02] Alexa M.: Recent advances in mesh morphing. *Computer Graphics Forum 21*, 2 (2002), 173–198.

[AMCO08] Aiger D., Mitra N. J., Cohen-Or D.: 4-points congruent sets for robust surface registration. *ACM Trans. on Graphics (Proc. SIGGRAPH) 27*, 3 (2008), 1–10.

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: SCAPE: shape completion and animation of people. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 408–416.

[ASP*04] Anguelov D., Srinivasan P., Pang H.-C., Koller D., Thrun S., Davis J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS* (2004).

[BBBK08] Bronstein A. M., Bronstein M. M., Bruckstein A. M., Kimmel R.: Partial similarity of objects, or how to compare a centaur to a horse. *Int. J. of Computer Vision* (2008).

[BBK06] Bronstein A. M., Bronstein M. M., Kimmel R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing 28*, 5 (2006), 1812–1836.

[BBK08] Bronstein A. M., Bronstein M. M., Kimmel R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[BBM05] Berg A. C., Berg T. L., Malik J.: Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Conf. on CVPR* (2005), pp. 26–33.

[BFF*07] Biasotti S., Falcidieno B., Frosini P., Giorgi D., Landi C., Marini S., Patané G., Spagnuolo M.: 3D shape description and matching based on properties of real functions. In *Eurographics Tutorial* (2007).

[BKS*05] Bustos B., Keim D. A., Saupe D., Schreck T., Vranić D. V.: Feature-based similarity search in 3D object databases. *ACM Comput. Surv. 37*, 4 (2005), 345–387.

[BMP00] Belongie S., Malik J., Puzicha J.: Shape context: A new descriptor for shape matching and object recognition. In *NIPS* (2000), pp. 831–837.

[BMSF06] Biasotti S., Marini S., Spagnuolo M., Falcidieno B.: Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design 38*, 9 (2006), 1002–1019.

[BR07] BROWN B. J., RUSINKIEWICZ S.: Global non-rigid alignment of 3-D scans. *ACM Trans. on Graphics (Proc. SIGGRAPH) 26*, 3 (2007).

[Bro92] BROWN L. G.: A survey of image registration techniques. *ACM Comput. Surv. 24*, 4 (1992), 325–376.

[CCFM08] CASTELLANI U., CRISTANI M., FANTONI S., MURINO V.: Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum (Proc. EUROGRAPHICS) 27*, 2 (2008), 643–652.

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3 (2009).

[CM92] CHEN Y., MEDIONI G.: Object modelling by registration of multiple range images. *Image Vision Comput. 10*, 3 (1992), 145–155.

[CMC*09] CAETANO T. S., MCAULEY J. J., CHENG L., LE Q. V., SMOLA A. J.: Learning graph matching. *IEEE PAMI 31*, 6 (2009), 1048–1058.

[CR03] CHUI H., RANGARAJAN A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89* (2003), 114–141.

[CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 13*, 3 (2007), 530–548.

[CZ08] CHANG W., ZWICKER M.: Automatic registration for articulated shapes. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008), 1459–1468.

[CZ09] CHANG W., ZWICKER M.: Range scan registration using reduced deformable models. *Computer Graphics Forum (Proc. EUROGRAPHICS) 28*, 2 (2009).

[dAST*08] DE AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. In *ACM Trans. on Graphics (Proc. SIGGRAPH)* (2008).

[dATSS07] DE AGUIAR E., THEOBALT C., STOLL C., SEIDEL H.-P.: Marker-less deformable mesh tracking for human shape and motion capture. In *Proc. IEEE Conf. on CVPR* (2007).

[DTC*02] DAVIES R. H., TWINING C. J., COOTES T. F., WATERTON J. C., TAYLOR C. J.: A minimum description length approach to statistical shape modeling. *IEEE Trans. on Medical Imaging 21*, 5 (2002), 525–537.

[DYT05] DINH H. Q., YEZZI A., TURK G.: Texture transfer during shape transformation. *ACM Trans. on Graphics 24*, 2 (2005), 289–310.

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE PAMI 25*, 10 (2003), 1285–1295.

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395.

[FP03] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. In *Proc. Symp. on Geom. Processing (SGP)* (2006), pp. 131–142.

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. on Graphics 25*, 1 (2006), 130–150.

[GE03] GUYON I., ELISSEEFF A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research 3* (2003), 1157–1182.

[GGGZ05] GATZKE T., GRIMM C., GARLAND M., ZELINKA S.: Curvature maps for local shape comparison. In *Proc. Conf. on Shape Modeling and Applications* (2005), pp. 244–253.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Proc. Symp. on Geom. Processing (SGP)* (2005), pp. 197–206.

[GR95] GOLD S., RANGARAJAN A.: Softmax to softassign: neural network algorithms for combinatorial optimization. *J. Artif. Neural Netw. 2*, 4 (1995), 381–399.

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose-oblivious shape signature. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 13*, 2 (2007), 261–271.

[GSdA*09] GALL J., STOLL C., DE AGUIAR E., THEOBALT C., ROSENHAHN B., SEIDEL H.-P.: Motion capture using joint skeleton tracking and surface estimation. In *Proc. IEEE Conf. on CVPR* (2009).

[GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3 (2009).

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008), 1449–1457.

[HJT08] HAMARNEH G., JASSI P., TANG L.: Simulation of ground-truth validation data via physically- and statistically-based warps. *Lecture Notes in Computer Science (Proc. MICCAI) 5241* (2008), 459–467.

[HM09] HEIMANN T., MEINZER H.-P.: Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis 13*, 4 (2009), 543–563.

[HNM06] HUANG X., N. N. P. D., METAXAS: Shape registration in implicit spaces using information theory and free form deformations. *IEEE PAMI 28*, 8 (2006), 1303–1318.

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. SIGGRAPH* (2001), pp. 203–212.

[HU90] HUTTENLOCHER D. P., ULLMAN S.: Recognizing solid objects by alignment with an image. *Int. J. of Computer Vision 5*, 2 (1990), 195–212.

[Hut91] HUTTENLOCHER D. P.: Fast affine point matching: an output-sensitive method. In *Proc. IEEE Conf. on CVPR* (1991), pp. 263–268.

[IJL*05] IYER N., JAYANTI S., LOU K., KALYANARAMAN Y., RAMANI K.: Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design 37*, 5 (2005), 509–530.

[IR96] IRANI S., RAGHAVAN P.: Combinatorial and experimental results for randomized point matching algorithms. In *Proc. Symp. on Computational Geometry* (1996), pp. 68–77.

[JH99] JOHNSON A., HEBERT M.: Using spin-images for efficient multiple model recognition in cluttered 3D scenes. *IEEE PAMI 29*, 5 (1999), 433–449.

[JZ07] JAIN V., ZHANG H.: A spectral approach to shape-based retrieval of articulated 3D models. *Computer-Aided Design 39*, 5 (2007), 398–407.

[JZvK07] JAIN V., ZHANG H., VAN KAICK O.: Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling 13*, 1 (2007), 101–124.

[KE06] KARLSSON J., ERICSSON A.: A ground truth correspondence measure for benchmarking. In *Proc. Int. Conf. on Pat. Rec. (ICPR)* (2006), vol. 3, pp. 568–573.

[KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Shape matching and anisotropy. *Proc. SIGGRAPH* (2004), 623–629.

[KPNK03] KÖRTGEN M., PARK G.-J., NOVOTNI M., KLEIN R.: 3D shape matching with 3D shape contexts. In *Proc. 7th Central European Seminar on Computer Graphics* (2003).

[KS04] KRAEVOY V., SHEFFER A.: Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 861–869.

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* (2009).

[LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3 (2009).

[LG05] LI X., GUSKOV I.: Multi-scale features for approximate alignment of point-based surfaces. In *Proc. Symp. on Geom. Processing (SGP)* (2005).

[LH05] LEORDEANU M., HEBERT M.: A spectral technique for correspondence problems using pairwise constraints. In *Proc. Int. Conf. on Comp. Vis. (ICCV)* (2005), vol. 2, pp. 1482–1489.

[LLFM00] LECLERC Y. G., LUONG Q.-T., FUA P. V., MIYAJIMA K.: Detecting changes in 3-D shape using self-consistency. In *Proc. IEEE Conf. on CVPR* (2000), vol. 1, pp. 395–402.

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008).

[LWZ*04] LIU L., WANG G., ZHANG B., GUO B., SHUM H.-Y.: Perceptually based approach for planar shape morphing. In *Proc. of Pacific Graphics* (2004), pp. 111–120.

[LZSCO09] LIU R., ZHANG H., SHAMIR A., COHEN-OR D.: A part-aware surface metric for shape analysis. *Computer Graphics Forum (Proc. EUROGRAPHICS) 28*, 2 (2009), 397–406.

[MC03] MACIEL J., COSTEIRA J. P.: A global solution to sparse correspondence problems. *IEEE PAMI 25*, 2 (2003), 187–199.

[MCH*06] MANAY S., CREMERS D., HONG B.-W., YEZZI A. J., SOATTO S.: Integral invariants for shape matching. *IEEE PAMI 28*, 10 (2006), 1602–1618.

[MFO*07] MITRA N. J., FLORY S., OVSJANIKOV M., GELFAND N., GUIBAS L., POTTMANN H.: Dynamic geometry registration. In *Proc. Symp. on Geom. Processing (SGP)* (2007), pp. 173–182.

[MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *Proc. IEEE Conf. on CVPR* (2008), pp. 1–8.

[MHL09] MIRZAALIAN H., HAMARNEH G., LEE T.: Graph-based approach to skin mole matching incorporating template-normalized coordinates. In *Proc. IEEE Conf. on CVPR* (2009), pp. 2152–2159.

[MS05] MIKOLAJCZYK K., SCHMID C.: A performance evaluation of local descriptors. *IEEE PAMI 27*, 10 (2005), 1615–1630.

[MV98] MAINTZ J. B. A., VIERGEVER M. A.: A survey of medical image registration. *Medical image analysis 2*, 1 (1998), 1–36.

[NB07] NEUHAUS M., BUNKE H.: *Bridging the gap between graph edit distance and kernel machines*. World Scientific, 2007.

[OEK08] OLSSON C., ENQVIST O., KAHL F.: A polynomial-time bound for matching and registration with outliers. In *Proc. IEEE Conf. on CVPR* (2008), pp. 1–8.

[Ols97] OLSON C. F.: Efficient pose clustering using a randomized algorithm. *Int. J. Comput. Vision 23*, 2 (1997), 131–147.

[PDT07] PITIOT A., DELINGETTE H., THOMPSON P. M.: Learning shape correspondence for n-d curves. *International Journal of Computer Vision 71*, 1 (2007), 71–88.

[PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum (Proc. EUROGRAPHICS) 27*, 2 (2008), 399–408.

[PH03] PAULSEN R. R., HILGER K. B.: Shape modelling using markov random field restoration of point correspondences. *Lecture Notes in Computer Science 2732* (2003).

[PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L. J.: Example-based 3D scan completion. In *Proc. Symp. on Geom. Processing (SGP)* (2005).

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH) 27*, 3 (2008).

[PRW94] PARDALOS P., RENDL F., WOLKOWICZ H.: The quadratic assignment problem: a survey and recent developments. In *Quadratic assignment and related problems*. Amer. Math. Soc., 1994, pp. 1–42.

[PS82] PAPADIMITRIOU C., STIEGLITZ K.: *Combinatorial Optimization: Algorithms and Complexity*. Pretice Hall, 1982.

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *Proc. 3rd Int. Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152.

[SAL*08] SHARF A., ALCANTARA D. A., LEWINER T., GREIF C., SHEFFER A., AMENTA N., COHEN-OR D.: Space-time surface reconstruction using incompressible flow. In *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* (2008).

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556.

[SKK04] SEBASTIAN T. B., KLEIN P. N., KIMIA B. B.: Recognition of shapes by editing their shock graphs. *IEEE PAMI 26*, 5 (2004), 550–571.

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton shape benchmark. In *Proc. Conf. on Shape Modeling and Applications* (2004), pp. 167–178.

[SN06] SCOTT C., NOWAK R. D.: Robust contour matching via the order-preserving assignment problem. *IEEE Trans. on Image Processing 15*, 7 (2006), 1831–1838.

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum (Proc. SGP) 28*, 5 (2009).

[Sor06] SORKINE O.: Differential representations for mesh processing. *Computer Graphics Forum 25*, 4 (2006), 789–807.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004), 399–405.

[SP08] SIDDIQI K., PIZER S. (Eds.): *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2008.

[SRN*03] STYNER M. A., RAJAMANI K. T., NOLTE L.-P., ZSEMLYE G., SZÉKELY G., TAYLOR C. J., DAVIES R. H.: Evaluation of 3D correspondence methods for model building. *Lecture Notes in Computer Science: Information Processing in Medical Imaging* (2003), 63–75.

[SSB05] Schölkopf B., Steinke F., Blanz V.: Object correspondence as a machine learning problem. In *Proc. 22nd Int. Conf. on Machine Learning* (2005), pp. 776–783.

[SSGD03] Sundar H., Silver D., Gagvani N., Dickinson S.: Skeleton based shape matching and retrieval. In *Proc. Conf. on Shape Modeling and Applications* (2003), pp. 130–142.

[SSP07] Sumner R. W., Schmid J., Pauly M.: Embedded deformation for shape manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 26*, 3 (2007).

[SSSCO08] Shalom S., Shapira L., Shamir A., Cohen-Or D.: Part analogies in sets of objects. In *Proc. of Eurographics Symposium on 3D Object Retrieval* (2008), pp. 33–40.

[STCB07] Schmidt F. R., Töppe E., Cremers D., Boykov Y.: Efficient shape matching via graph cuts. In *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition* (2007), pp. 39–54.

[Sto87] Stockman G.: Object recognition and localization via pose clustering. *Comput. Vision Graph. Image Process. 40*, 3 (1987), 361–387.

[SWG08] Süssmuth J., Winter M., Greiner G.: Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008), 1469–1476.

[SY10] Sahillioğlu Y., Yemez Y.: 3D shape correspondence by isometry-driven greedy optimization. In *Proc. IEEE Conf. on CVPR* (2010). To appear.

[SZM*08] Siddiqi K., Zhang J., Macrini D., Shokoufandeh A., Bouix S., Dickinson S.: Retrieving articulated 3-D models using medial surfaces. *Machine Vision and Applications 19*, 4 (2008), 261–275.

[TBW*09] Tevs A., Bokeloh M., Wand M., Schilling A., Seidel H.-P.: Isometric registration of ambiguous and partial data. In *Proc. IEEE Conf. on CVPR* (2009).

[TH08] Tang L., Hamarneh G.: SMRFI: Shape matching via registration of vector-valued feature images. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (2008), pp. 1–8.

[TL94] Turk G., Levoy M.: Zippered polygon meshes from range images. In *Proc. ACM SIGGRAPH* (1994), pp. 311–318.

[TV08] Tangelder J. W. H., Veltkamp R. C.: A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications 39*, 3 (2008), 441–471.

[Ume91] Umeyama S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE PAMI 13*, 4 (1991), 376–380.

[VH01] Veltkamp R. C., Hagedoorn M.: State of the art in shape matching. In *Principles of visual information retrieval*. Springer-Verlag, 2001, pp. 87–119.

[vKHZW07] van Kaick O., Hamarneh G., Zhang H., Wighton P.: Contour correspondence via ant colony optimization. In *Proc. Pacific Graphics* (2007), pp. 271–280.

[WAO*09] Wand M., Adams B., Ovsjanikov M., Berner A., Bokeloh M., Jenke P., Guibas L., Seidel H.-P., Schilling A.: Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data. *ACM Trans. Graph. 28*, 2 (2009).

[WH07] Ward A. D., Hamarneh G.: Statistical shape modeling using MDL incorporating shape, appearance, and expert knowledge. *Lecture Notes in Computer Science (Proc. MICCAI) 4791* (2007), 278–285.

[WH09] Ward A. D., Hamarneh G.: The groupwise medial axis transform for fuzzy skeletonization and pruning. *IEEE PAMI Accepted for future publication* (2009).

[WJH*07] Wand M., Jenke P., Huang Q.-X., Bokeloh M., Guibas L., Schilling A.: Reconstruction of deforming geometry from time-varying point clouds. In *Proc. Symp. on Geom. Processing (SGP)* (2007), pp. 49–58.

[WR97] Wolfson H. J., Rigoutsos I.: Geometric hashing: an overview. *IEEE Computational Science & Engineering 4*, 4 (1997), 10–21.

[XWY*09] Xu W., Wang J., Yin K., Zhou K., van de Panne M., Chen F., Guo B.: Joint-aware manipulation of deformable models. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3 (2009).

[ZD06] Zheng Y., Doermann D.: Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE PAMI 28*, 4 (2006), 643–649.

[ZF03] Zitová B., Flusser J.: Image registration methods: a survey. *Image and Vision Computing 21*, 11 (2003), 977–1000.

[ZS08] Zass R., Shashua A.: Probabilistic graph and hypergraph matching. In *Proc. IEEE Conf. on CVPR* (2008).

[ZSCO*08] Zhang H., Sheffer A., Cohen-Or D., Zhou Q., van Kaick O., Tagliasacchi A.: Deformation-driven shape correspondence. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008), 1431–1439.

[ZST*10] Zheng Q., Sharf A., Tagliasacchi A., Chen B., Zhang H., Sheffer A., Cohen-Or D.: Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum (Proc. EUROGRAPHICS)* (2010), to appear.

[ZvKD07] Zhang H., van Kaick O., Dyer R.: Spectral methods for mesh processing and analysis. In *Proc. of Eurographics State-of-the-art Report* (2007), pp. 1–22.