1 Introduction

Matching of three-dimensional (3D) shapes is a principal function in such processes as object recognition, object surface analysis, geometric inspection, surface reconstruction, and reverse engineering of shapes. Typically, a match is sought between a measured object and an already available model, either in some library or generated synthetically. In this paper we propose a method for the direct, partial matching of 3D surfaces. Here, direct refers to the calculation of the matching criterion explicitly from the surface points, using the Hausdorff distance, which will be defined later. Partial means that it is sufficient that one of the shapes matches only some portion of the other shape. The motivation for this approach stems from the application that we support, namely the reuse of existing shapes as shape features in a new computer-aided design (CAD) model. This makes it necessary for the matching to depend not only on an affine transformation, but on intrinsic shape parameters as well. Before going into the details of the matching technique and of the driving application, we will briefly mention some existing shape matching methods. For a more elaborate overview the reader is referred to [1] and [16]. Shape matching depends on two functions, the calculation of shape dissimilarity and the search in some space to maximize shape similarity. For some applications the main task is to search among a large set of predefined discrete shapes $B_i$ for the one(s) that resembles most a given shape $A$. An example of such an application is the search for a fingerprint in a database. To perform this function some kind of indexing scheme can be introduced, based on global moments, Fourier descriptors, local geometric properties, or even strain energy [6, 9, 10]. If it is known beforehand that the set $\{B_i\}$ consists of shapes exhibiting particular features or singularities, then an indexing scheme can be based on the type and/or number of such features in each of the members $B_i$.

For the purpose of quality control a manufactured part $B$ can be compared with the CAD model $A$ it originated from. First the surface of the manufactured part is digitized and the point set is tested for matching the (surface or solid) model $A$. This involves a search among a continuous set of shapes $P(B)$, where $P$ denotes translations and rotations and possibly scaling as well, leading to a 6- or 7-dimensional search space. When the best alignment of $P(B)$ with $A$ is achieved, the positional deviation between the two shapes can be analyzed.
and visualized. For this purpose, commercial software systems are readily available [7]. Reverse engineering of a 3D part \( B \) comprises the reconstruction of a CAD model from measured points on \( B \). If the CAD model is hypothesized to consist of known types of surfaces \( A_i \), e.g. cylinders and planar sections, the surfaces \( A_i \) must be matched to (portions of) \( B \) [13, 14]. If the CAD model is of a more generic type \( A \), e.g. represented as a set of NURBS surfaces, then the task of reverse engineering is reduced to the reconstruction of \( A \) from data \( B \), without explicit shape matching [8, 15].

Although the applications listed above are in three dimensions, most of the literature on shape recognition and matching deals with applications and with algorithms tested in 2D, as e.g. in [4]. Many search strategies for 3D applications are based on 2D derivations from the models to be matched, including flat projections, silhouettes, and images, as e.g. in [3, 5].

As mentioned, search strategies should be distinguished from the actual similarity measure. The complexity of the similarity computation depends on the requirements of the application. If it is needed to select from a set \( \{ B_i \} \) the shape which is most similar to \( A \), then it may be sufficient to use a low-cost similarity measure. On the other extreme, for a best fit between shape \( A \) and a shape \( B \) from a continuously varying set of shapes, then the similarity computation will normally be complex. This is especially true for freeform shapes, properties of which can sometimes only be numerically approximated.

### 2 Requirements of the driving application

The requirements of both the similarity computation and the search strategy derive from the intended application. We have developed an approximation of the directed Hausdorff distance between 3D shapes, defined in Sect. 3. Here we explain this choice. The shapes of interest are portions of the boundaries of 3D solids. Hence the shapes can be considered as 2-manifolds. When a 3D solid or surface CAD model is developed, it is, in some situations, effective for the designer to reuse an existing shape. In doing so, the designer may save considerable time and effort. However, typically it is not sufficient to perform merely a copy-and-paste action. Instead the copied shape should be adapted to the target model. This adaptation can be partly automatic and should be partly controlled by the user, e.g. by modifying particular shape parameters. More detailed information about methodological issues of the shape reuse process can be found in [17]. The application comprises four basic steps:

1. A geometric representation \( S \) of the shape to be reused must be obtained. This involves either the digitization of a physical object or the selection of a shape that is already in digital form. In the latter case the shape may originate from the user’s local files, from a company’s shape library, or it may be found on the internet.

2. A static model of the shape \( S \) is not sufficient for the application. The representation \( S \) obtained in step 1 should be modeled as a parameterizable shape \( T(p) \), where those parameters \( p \) that the user will need later to adjust the shape before its insertion into the target model should be available to him/her. This involves the matching of shapes \( T(p) \) to shape \( S \). It should be noted that the multi-dimensional parameter \( p \) always includes the 6 DOF for rigid body transformation.

3. The user adjusts the parameter \( p \) to the value \( p = p' \) to obtain the shape \( T(p') \) meeting the current design requirements.

4. The shape \( T(p') \) is copied and inserted into the target CAD model.

In practice, the steps 3 and 4 are a mixture of automatic and semi-automatic processes in which the user is primarily concerned with the intrinsic shape parameters [19].

In step 2 of the application process the actual shape comparison takes place. The requirements of the measure of dissimilarity \( D(T(p), S) \) of shape \( T(p) \) to shape \( S \) are as follows:

(i) \( D \) needs to be generic, suited to any pair of 2-manifolds \( S \) and \( T(p) \). Therefore, the dissimilarity computation should not rely on presumed features of the shapes, on their representation forms, or on any correspondence information.

(ii) \( D \) needs to be a directed shape dissimilarity, since in general it is demanded that \( T(p) \) matches a part of \( S \), rather than \( S \) itself. If, instead of a directed shape dissimilarity, a symmetrical shape dissimilarity \( D \) is used, the following problem could occur. Suppose that the size of \( S \) is large compared with the size of \( T(p) \). If each point of \( T(p) \) is near \( S \) then there may still be points in \( S \) that are distant from