

# Slippage Features

Martin Bokeloh, Alexander Berner, Michael Wand,  
Hans-Peter Seidel, Andreas Schilling

email: bokeloh@gris.uni-tuebingen.de  
berner@gris.uni-tuebingen.de  
mwand@mpi-inf.mpg.de  
hpseidel@mpi-sb.mpg.de  
schilling@uni-tuebingen.de

WSI-2008-03  
Juni 2008

Graphisch-Interaktive Systeme  
Wilhelm-Schickard-Institut  
Universität Tübingen  
D-72076 Tübingen, Germany  
WWW: <http://www.gris.uni-tuebingen.de>

© WSI 2008  
ISSN 0946-3852

## Abstract

In this report, we present a novel feature detection technique for unstructured point clouds. We introduce a generalized concept of geometric features that detects locally uniquely identifiable keypoints as centroids of area with locally minimal slippage. We extend the concept to multiple scales and extract features using multi-scale mean shift clustering. In order to validate matches between feature points, we employ a two stage technique that first sorts out unlikely matches, followed by an approximate alignment between remaining features by a rotational cross-correlation analysis and a local iterative closest point (ICP) registration. The resulting residuals are then used as final similarity measure. The proposed combination of techniques results in a robust and reliable correspondence detection technique that yields registration results in situations where previous techniques are not able to detect usable feature correspondences. We provide a detailed empirical analysis of the method, and apply the technique to global registration, symmetry detection and deformable matching problems.

# 1 Introduction

Feature detection techniques have become an indispensable tool in geometry processing. The task of a feature detector is to identify a set of representative regions on a geometric object that are subsequently used to solve a geometric problem more efficiently on this sparse, discrete set. The main application area of geometric features is solving correspondence problems, such as global rigid registration [HFG<sup>+</sup>06, GMGP05, LG05, LGB06, HH01, MGGP06], deformable registration [HTB03, ASP<sup>+</sup>04, WJH<sup>+</sup>07] or symmetry detection [MGP06, PSG<sup>+</sup>06, MSHS06, SKS06]. Reducing the problem to a sparse set of correspondence candidates is often a key concept for solving such problems efficiently.

The task of a feature detection algorithm can be split into two stages: First, a set of *keypoints* is identified that are locally unique. This means, if the same feature detection algorithm is applied to a similar piece of geometry, it will retrieve the same keypoint at the same position. The local area associated with each keypoint in order to define this property is called a *geometric feature*. Second, a *feature descriptor function* is defined that allows for comparing two different such features. The function descriptor gets two features as input (maybe including precomputed descriptive values per feature, called *descriptors*) and outputs a measure of distance, which can be used to establish initial correspondence candidates. Typically, a subsequent global optimization algorithm will be employed to verify the candidate correspondences obtained by local feature comparison from a more global point of view. Common techniques for solving global correspondence problems are RANSAC- or forward search-based sampling techniques [FB87, HFG<sup>+</sup>06], belief-propagation based matching techniques [ASP<sup>+</sup>04] and spectral graph matching [LH05]. For our prototype applications, we will use the latter technique that yields very good results and is easy to implement.

An important problem for feature-based techniques is the ability to *reliably* detect a sufficient number of keypoints: For traditional applications such as performing rigid registration of piecewise scanned point clouds [HFG<sup>+</sup>06, MGGP06, GMGP05, LG05, LGB06, HH01], a small number of correctly established correspondences is sufficient, as a rigid motion is uniquely determined by specifying point-to-point correspondences between two triples of non-collinear points. However, some application areas of geometry processing that recently gained importance require many more correspondences: For example, deformable alignment of surfaces [WJH<sup>+</sup>07, BR07] has many more degrees of freedom so that a larger number of correspondences, covering all parts of the object, is necessary to make the alignment problem globally well-defined. In our experience, this can be a serious issue for traditional keypoint detection techniques. Another interesting recent research direction is symmetry detection, where correspondences within one and the same object are established in order to reveal redundancies and reoccurring parts within the object. Again, for such an application, a much larger number of correspondences is necessary. Most established feature detection techniques detect keypoints at points of maximum curvature [Low03, GMGP05, LG05, YF02, WG02]. For the aforementioned application areas, this notion is often not general enough so that an insufficient number of keypoints is extracted for many input models.

The objective of this report is to generalize the notion of keypoints to include as many locally uniquely localizable regions as possible without compromising

stability and reliability of detection. The basic idea of our technique is that a feature must have the property that within a local neighborhood the position of the keypoint must be uniquely defined so that the notion of feature correspondences is locally well defined. In addition, for many applications it is also useful to have a well-defined rotational alignment in order to establish correspondences with corresponding coordinate frames rather than just point-to-point correspondences. This means that the rotational degrees of freedom should also be well defined for the feature within its local neighborhood.

In order to determine such points we examine the local geometry at each surface point contained within a ball of fixed radius. The radius determines the scale at which the feature detection takes place. In order to detect regions with well defined translation and rotation, we employ the *slippage analysis* of Gelfand et al. [GG04] to compute a measure of how well constrained an auto-alignment problem of the feature regions is. Keypoints are then defined as local maxima of this constraintness measure. We extract these maxima using a meanshift clustering algorithm and verify the stability by filtering out points where the maxima are not pronounced enough to allow a stable detection. This criterion for determining keypoints is more general than just looking for local maxima in curvature of the geometry itself; a piece of geometry containing a complex geometric pattern might be strongly constrained in its auto-alignment although it might appear flat in average, not providing a local extremum in curvature. As the local well-constraintness of the alignment in translation (and, depending on the application, also rotation) is a necessary criterion for a geometrically defined feature, our approach also defines a natural and very general class of features. In order to eliminate the free radius parameter of the detection radius, we generalize our technique to multi-scale feature detection and detect maxima in scale space, adopting the conceptual framework of SIFT keypoint detection [Low03].

The novel notion of general keypoint detection based on a slippage analysis is the key idea and main contribution of this report. We complement our technique by adding a similarly general descriptor technique: In order to compute descriptors for the obtained features, we employ a two stage filtering approach: First, we compute local curvature histograms of the feature area [GGZ05] to obtain a filtering criterion to rapidly reject unlikely matches. Afterwards, for the smaller set of remaining matches, we employ a local iterated closest points (ICP) alignment to compute a more precise matching score [HFG<sup>+</sup>06]. In order to make this alignment computation fast and reliable, we employ a cross-correlation technique based on the fast Fourier transform (FFT) in order to compute a good initial rotational alignment.

We demonstrate the benefits of our technique by examining example applications in both rigid and non-rigid alignment, and also examine the potential for symmetry detection based on generalized keypoints. In addition, we provide a thorough empirical analysis of stability and matching precision of the proposed technique in comparison to related work.

## 2 Related Work

Feature detection for geometric models has first been examined in the context of rigid registration. Gelfand et al. [GMGP05] base their technique on curvature

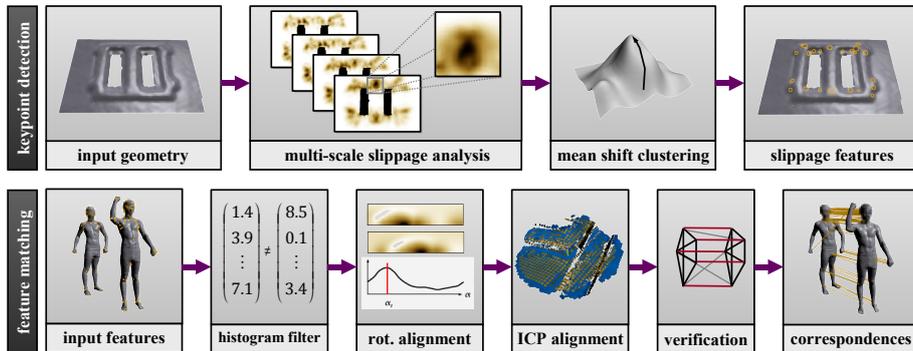


Figure 1: Overview of our feature matching pipeline: We compute keypoints as local maxima of local slippage using mean-shift clustering. Feature matching aligns detected features using a curvature histogram based filter, an FFT-based rotational alignment and an ICP registration. A global validation algorithm extracts a consistent set of matches.

on multiple scales and filter descriptors by preferring features that are more unique on each of the pieces to be registered. A drawback of their feature detection approach is that keypoints might not be well constrained along curves of constant mean curvature, so that random variations in feature positions might result. In addition, a local measure of curvature alone has only limited discriminative power. In contrast, Huber and Hebert choose a dense sampling approach: they compute a surface descriptor (spin-images) for every vertex on a uniformly sampled subset of the input meshes and use a RANSAC technique to find corresponding vertices with similar descriptors [HH01]. The number of necessary samples can be reduced by referring points with specific surface properties like high curvature [YF02] or bitangent points [WG02].

The method proposed by Li and Guskov [LG05] adapts the concept of SIFT-Features [Low03] to geometric data: They build a multi-scale representation of the input data and extract local extrema in the difference levels as features points. Our work is a generalization of their technique to a more general notion of features. Huang et al. proposed a method for reassembling fractured objects using global geometric registration [HFG<sup>+</sup>06]. They compute a set of surface signatures for every data point and apply a flooding based clustering algorithm to extract regions of similar signature. The centroids of the resulting surface patches are used as features, and the main axis determine the rotational alignment. The technique is performed on multiple scales to create a large number of potential features and descriptors. For feature matching, a series of filtering heuristics is used to exclude obvious mismatches, including ICP-based alignment as final step. Global verification of the alignment is done with statistical forward search. The drawback of the approach is that the region growing based feature extraction is likely to produce a large number of both false positive and false negative matches, as the connectivity of the surface patches can easily change, leading to non-matching features. In their target application, this is not an issue, as only a small number of final matches are necessary and the forward search based global verification is able to compensate for this drawback easily.

In contrast, our method targets on finding reliable and reproducible keypoints; this property is important to reduce the chance of missing keypoints in problematic areas that can be decisive for more general correspondence problems such as deformable matching.

**Geometry descriptors:** As we already have a reference key point, the main requirement for a descriptor is typically rotational invariance (in the more general case of deformable matching, a local rigid matching is usually assumed to approximate the local effect of the global deformation). There are a large number of techniques to describe geometry with a small digest of data. Many of these matching techniques have been designed for efficient object retrieval from large data bases. Thus, in order to perform a large number of comparisons efficiently, these methods typically employ mappings of geometric objects to small vectors of characteristic numbers: [KFR03] employs power spectra of a spherical harmonics description of the geometry to compute a rotationally invariant geometry descriptor. [NK03] extends this approach to Zernike polynomials to reduce the invariant space that maps to the same descriptor. [GGGZ05] consider maps of local curvature values; we use a similar approach at the first stage of our filtering cascade. Another standard technique are spin images [JH99,HH01] that compute an average trace of geometry intersecting a plane rotating around the surface normal. A related approach based on spherical harmonics has been proposed by [FHK<sup>+</sup>04]. [MGGP06] propose a matching technique based on min-hashing that also allows for partial object matching. Our technique aims at geometry processing applications. In this context, we opt for a direct alignment computation to compare geometric regions, which yields optimal accuracy. In order to speed up the computation, we filter out most non-matching regions using a descriptor similar to [GGGZ05]. ICP alignment with filtering for feature matching was initially proposed by [HFG<sup>+</sup>06]; we complement their approach by efficient and more reliable rotational and translational prealignment.

### 3 Multi-scale Slippage Keypoints

In this section, we describe how keypoints are extracted. We start by briefly reviewing the slippage analysis technique of Gelfand et al. [GG04]. Afterwards, we discuss how this is used for extracting keypoints from surfaces.

#### 3.1 Local Slippage Analysis

Slippage analysis poses the problem whether a rigid matching of a piece of surface with itself is well constrained. Conceptually, a point-to-plane ICP error function, dependent on 3 rotation and 3 translation variables, is setup for a configuration where the surface is perfectly aligned with itself. Then, the Hessian matrix of this objective function reveals directions in which the problem is not well constrained: The eigenvectors of the zero eigenvalues of this matrix correspond to transformations that map the surface onto itself in an infinitesimal sense (instantaneous motion). For example, for a cylinder mapping to itself, there are one rotational and one translational degree of freedom, resulting in two eigenvectors with zero eigenvalue. The cylinder can be rotated around its rotational symmetry axis and shifted along this axis without changing its shape locally. In general, non-zero eigenvalues are a necessary condition for the

auto-alignment problem being well defined. For non-perfect data sets, such as scanned point clouds, we cannot expect zero eigenvalues but small eigenvalues indicate an underconstrained auto-alignment. As shown in , the hessian matrix of the alignment problem is given by:

$$C = \sum_{i=1}^n \begin{pmatrix} c_i c_i^t & c_i n_i^t \\ n_i c_i^t & n_i n_i^t \end{pmatrix} \quad (1)$$

where  $c_i = (p_i \times n_i)$ .  $p_i$  refers to the  $i$ -th surface point of a surface discretization and  $n_i$  refers to its normal. Please note that as slippage is derived from a point-to-plane matching, normal information is necessary to setup the Hessian matrix.

### 3.2 Feature Point Extraction

In order to apply this analysis technique for our purpose, we first discretize the surface into a cloud of points. In addition, we also have to compute surface normals. If our input is given as a triangle mesh, we start by computing a dense point cloud by a uniform random sampling of the surface. Next, for both triangle and point cloud input, we continue with a Poisson-disc resampling step that results in a uniform, well distributed sampling of the surface [WS02]. The position of the resampled points is projected back on a quadratic moving least squares surface fit of the original data points to optimize the representation accuracy. After this computation, a uniform surface sampling is obtained with known sampling spacing. In case of triangle meshes, we retain the original triangle normals as initial normals. For point clouds, we compute normal information by a PCA analysis. We weight the contributions of neighboring points with a Gaussian windowing function with a standard deviation of 1.5 times the median point spacing.

Subsequently, we go through each point of our surface discretization and compute the local slippage analysis of a spherical neighborhood of points of a fixed radius  $2\sigma$ . For correctly band limiting the computation, we add a Gaussian weighting window with standard deviation  $\sigma$  to each term of each surface point in Equation (1), i.e. the neighborhood radius is chosen to truncate this Gaussian window as soon as the weighting function becomes negligibly small. Before slippage computation, we normalize the extents of the region considered to a unit bounding box centered at the origin. This makes sure that rotational and translational degrees of freedom are always measured with the same relative weighting. From the eigenvalue decomposition, we consider only the smallest eigenvalue  $\lambda_6$  as a measure for constraintness. In order to make the measure invariant under the amount of local area gathered, we additionally divide by the largest eigenvalue  $\lambda_1$  which is guaranteed to be non-zero (any non-empty surface is at least translationally constrained in one normal direction). A low value of this ratio indicates slippable motion in at least one direction; a high value indicates a more constrained rigid motion. Please note that using the smallest eigenvalue of the Hessian (1) means that we will exclude keypoints that are not rotationally well constrained, such as rotationally symmetric bumps. It is straightforward to setup a similar slippage analysis problem that ignores rotational degrees of freedom altogether in order to extract translationally invariant keypoints only. For clarity of exposition, we restrict ourselves to the case of ro-

tationally well-constrained keypoints in this report. For the rest of our pipeline, we stick to this scenario and exclude features that are rotationally ambiguous. Filtering out points that are slippable under infinitesimal rotations is thus the first step, as this violates a necessary condition for unique matching.

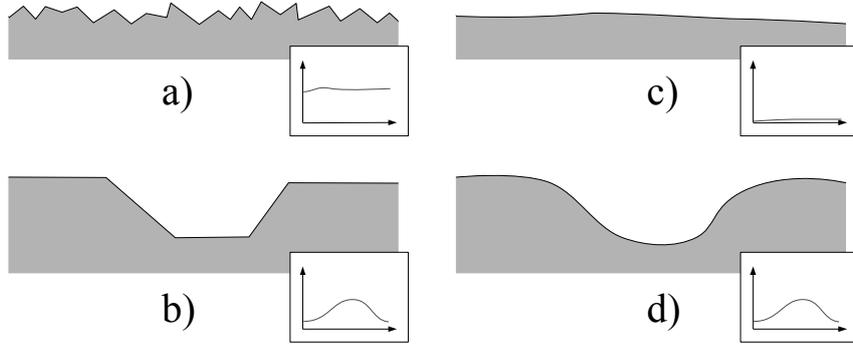


Figure 2: The normal smoothing problem (the small graphs show the slippage values schematically): Fine scale geometry variation can lead to low slippage values due to varying normals (a); at coarser scales, this is undesirable. Smoothing the normals resolves this issue (c). For large scale features (b), the normal smoothing will still preserve the detected feature (d).

Next, we consider the problem of estimating slippage values for multiple scales. At this point, we are facing the problem of coupling the normal variation to the geometric level of detail: In a multi-scale analysis, our objective is to examine geometric features in multiple frequency bands. For a coarse scale analysis, small scale features should not be taken into account. However, the slippage analysis is formulated as constraining rigid motion by aligning planes defined by points and normals. Therefore, small scale variations in the normals can make the auto-alignment strongly constrained although this is not warranted at the frequency level considered. The problem is illustrated in figure 2: The plane in figure (a) is non-slippable due to normal variations at a much smaller scale. In order to couple the normals to the scale of considerations, we have to band-limit the frequency of normal variation accordingly (figure 2 (c)). We do this by filtering the normals with a Gaussian low-pass filter of support proportional to the size of the region considered. Small scale variations will be removed, while large scale geometric features will still be retained (2 (b,d)) In experiments, it turns out that the best results are obtained from a rather aggressive filtering strategy, where only relatively low frequency normal variations are retained at each scale (we use  $\sigma_{smooth} = 1.25\sigma$ ). Please note that we do not need to filter the geometry in a similar way, as point positions with small scale noise will still produce realistically slippable results as long as the normals are smoothed correctly. Another way to formulate this fact is that normals and geometric positions are measured in different units, where normals are not affected by scaling while positions are.

Repeated subsampling, normal filtering and slippage evaluation will yield a sequence of scales with support  $\sigma_i = \sigma_0 F^i$  ( $F > 1$ ). In the following, we denote  $\theta_{i,s}$  the slippage indicator resulting from the local slippage analysis computed

at point  $x_i$  with support  $\sigma_i$ . In the scale-space domain we now search for local maxima. A natural technique for extracting this information in a robust way in the presence of noise, is mean shift clustering [CM02]. This technique first convolves the data points with a low-pass filter kernel and then performs gradient ascend to find maxima. All points ascending to the same maximum are comprised in the same cluster, and the location of the maximum becomes our representative key point. In our case, we employ a standard Gaussian windowing function as mean shift kernel and set the support radius to a fixed value of 2 times the median sampling spacing. In scale direction, we use a Gaussian window of standard deviation 0.5, where 1.0 corresponds to one scale level  $F$ . As we are only interested in the location of the maxima, we can optimize the computation: We first compute candidate points at discrete positions that are local maxima on each scale after filtering in the spatial domain only. Afterwards, we start a gradient ascend in scale-space to find final maxima at continuous spatial position and continuous scale values. Doublets and diverging points are removed at that stage. In a final step, we evaluate how well determined the extracted maxima are. We estimate the second order derivatives by finite differencing and delete points with small curvature (below a threshold  $H_{min}$ ) in the slippage value, as these points will not be detected reliably under the influence of noise and small scale random variations of geometry.

## 4 Descriptors and Feature Matching

Having determined a set of keypoints, we need to compare the local geometric neighborhood in order to determine candidate correspondences. In our case, the local neighborhood always corresponds to the geometry contained within a fixed distance of 10-15 times the median point distance. In order to match the generality of the keypoint detector, our feature comparison function is based on a direct geometric comparison. We proceed in three steps: Signature based pruning, rotational alignment, and geometric alignment.

### 4.1 Signature-based Pruning

As a quick test to reduce the number of matching candidates, and thus reduce the number of the later, more expensive alignment steps, we compute local feature signatures [AKKS99]. The main idea is to compute histograms of surface attributes over concentric rings around the keypoint. By construction, such a signature is rotationally invariant. As attribute value, we will use mean curvature; it is straightforward to generalize this technique in order to use additional attributes such as surface color, if available.

Let us assume, that we have an attribute  $a_i$  defined for every point  $x_i$  of the data set that is invariant under rotation and translation, such as mean curvature. Then, a pair of corresponding features should have roughly the same distributions of this attribute in their environments. Given a feature point  $f$ , we denote a ring histogram  $H_r$  as a histogram of the attributes  $a_i$  weighted with a spatial function  $\omega_r$ :

$$H_r(f) = \{H_{r_b}(f) | b \in \{1, \dots, N_b\}\} \quad (2)$$

$$H_{r_b}(f) = \sum_{i \in I} B_b(a_i) \omega_r(\|x_i - f\|)$$

A ring histogram contains  $N_b$  bins.  $B_b$  is a function which returns 1, if the given attribute  $a_i$  falls into bin  $b$  (0 otherwise). We can assume, that the euclidean distance between  $f$  and  $x_i$  is roughly equal for corresponding features. We take this into account by computing multiple histograms with different spatial support:

$$\omega_r(x) = \max(0, \lambda_h |r - x|) \quad (3)$$

For stability reasons, we smooth the histograms with a triangle filter. The signatures are invariant under rigid transformations, easy to compute and robust to spatial noise. In all of our examples, we use the mean curvature as attribute: For every point in the data set, we precompute an approximation of the curvature by fitting a quadratic patch onto the neighboring data points. We use the computed normal to define a local tangent coordinate system and compute a weighted least-squares fit of a bivariate quadratic polynomial. From the resulting coefficients, we compute the mean curvature analytically. In order to compare two signatures, we just compare the Euclidian distance of the scalar vectors formed by the corresponding histograms and reject the match if a certain threshold is exceeded. Implicitly, this criterion will put larger emphasize on the outer rings as the histogram entries will consist of larger numbers due to the larger number of neighboring points counted. This is desirable, because more area is covered by these rings.

## 4.2 Local Feature Alignment

Having rejected most incompatible correspondences by comparing the signature vectors we can still expect to retain a small number of false positive matches, as the histogram criterion is not injective. Therefore we proceed by a geometric registration of the features and compare the residual of the match.

**Rotational alignment:** The first problem we are facing is a correct rotational alignment of the features. So far, only the translation is known due to the keypoints, which is not sufficient to guarantee the convergence of the later ICP alignment step. Huang et al. address this problem by computing a principal component analysis from a feature cluster and used the resulting eigenvectors for alignment [HFG<sup>+</sup>06]. Another approach is to use the principal frame defined by the normal and the directions of maximal and minimal curvature [LGB06, MGP06], and drop features with ambiguous principal frames. Both techniques have the drawback that they might return no well-defined result in rotationally symmetric situations, even though the auto-alignment problem is still well defined. In order to improve upon this, we employ a spectral cross-correlation technique: We assign a orthogonal local frame  $F_i = (u_i, v_i, n_i)$  to a feature  $f_i$ , where  $n_i$  is the feature normal and  $u_i, v_i$  are arbitrary orthogonal tangents. Then we transform the local environment into this frame and consider the surface as 2D height field. When two features match then there exists at least one rotation angle  $\alpha_t$  that maximizes the scalar product

$$\arg \max_{\alpha_t} \langle H_1, R_{\alpha_t}(H_2) \rangle \quad (4)$$

where  $H_1$  and  $H_2$  are height fields and  $R_{\alpha_t}(H)$  is a transformation which rotates the height field  $H$  by  $\alpha$  around the normal. Since we have only a sparse set of surface points, we need to produce a dense representation for comparison. For this purpose, we use the *splat-pull-push* method proposed by [GGSC96] to create a regularly sampled array of height values. After resampling, we perform a polar coordinate transformation of the values, i.e. we resample the values into an additional array that is indexed by angle and distance to the origin. For each ring of constant distance, the cross correlation function defined above corresponds over the angle of rotation to the 1D convolution of the two signal vectors. As we are looking for the maximum of this value over all rings, we sum up the 1D convolution functions. Each such convolution can be computed efficiently in the Fourier domain [CM87]: For each feature, we precompute a fast Fourier transform of its height field values of constant distance, obtained from the polar coordinates transform. Then, a point-wise multiplication of the Fourier coefficients yields the convolution. Each rotational alignment can now be performed in  $O(n)$  time, where  $n$  is the number of entries of the transformed height field. Each feature needs  $O(n \log n)$  precomputation. This is much more efficient than performing an  $O(n^2)$  brute force test for each of the potentially quadratic number of candidate feature matches. For subpixel accuracy, we compute a local parabolic fit and refine the peak of the rotational correlation function. Having found one or more peaks, we compute the transformation between the two features with respect to the estimated angle. For most comparison purpose it is sufficient to take the biggest peak only. Some features however have multiple peaks due to similar structures. We reject all peaks that are less than 0.8 times the maximum peak. In case we still find multiple rotational matching candidates, we dismiss the keypoint altogether, as it can be considered to be not rotationally well constrained. Alternatively, it is also possible to keep and test multiple alignments, if rotationally uniqueness is not required for the target application.

**Geometric alignment (ICP):** With the estimated approximate transformations, we perform local point-to-plane ICP alignment [CM92] and take the minimum alignment residual (average of the point to plane distances) as feature distance. If the ICP alignment does not converge, we reject the correspondence pair. Figure 3 shows an approximate alignment result with two corresponding features on the buddha data set.

## 5 Applications

In order to evaluate the practical performance of the proposed feature detection scheme, we have implemented three prototypical example applications: global rigid registration, global deformable registration, and a simple feature based symmetry browser.

### 5.1 Global Rigid Registration

For global rigid registration, we first compute keypoints and matching features as described above. After that, we use a global validation algorithm to extract a consistent set. We employ the spectral graph matching technique of Leordeanu and Hebert [LH05]. The algorithm considers pairs of correspondences and as-

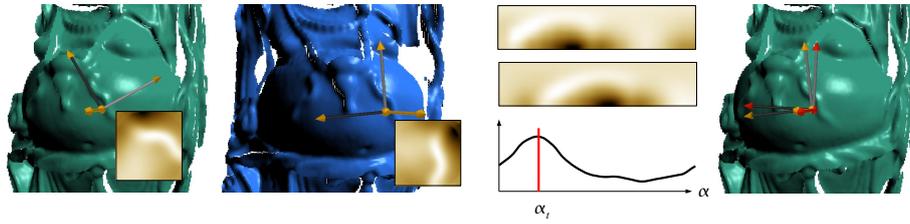


Figure 3: Approximate alignment of two features from different views of the buddha data set: We transform the parametric images (height fields) of the feature environments into the polar domain, use the phase correlation technique to find a good initial transformation (red coordinate frame) and perform local ICP to refine the transformation (yellow).

signs a plausibility score to these pairs based on the difference in Euclidian distance on source and target shape. This yields a compatibility matrix. The diagonal elements of this matrix refer to compatibility of a correspondence with itself; we insert normalized residuals from the ICP matching stage in order to evaluate the plausibility of the correspondence itself. Normalization and mapping of distance discrepancies are handled exactly as described in the original paper [LH05]. Next, the eigenvector with the largest eigenvalue is computed using a power iteration. Large entries in this vector correspond to correspondences that are compatible to a large set of other correspondences, while outlier correspondences receive low scores. A final quantization step (again following closely the original paper), sets the eigenvector entries to zero and one, deleting incompatible matches for which the Euclidian distance differs by more than a fixed threshold. Lastly, we compute a global rigid alignment from the validated correspondences using standard SVD-based matrix decomposition. We show the alignment results without a final ICP alignment in order accurately depict the obtained registration accuracy from feature matching only.

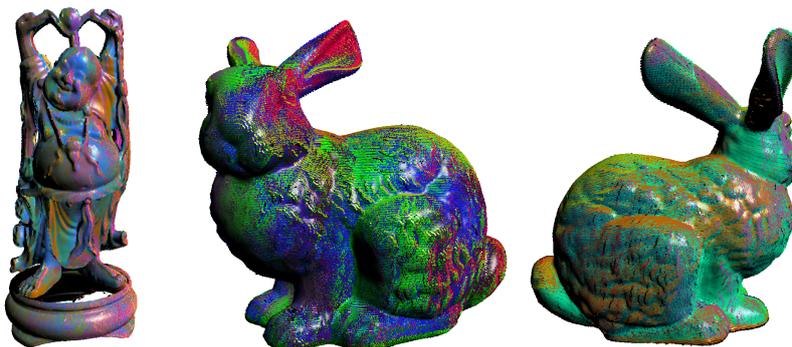


Figure 4: Results of the registration technique for the case of global rigid alignment.

## 5.2 Deformable Registration

Our deformable matching application uses exactly the same algorithm as the rigid case, just substituting geodesic distances along the surface for the Euclidian distances, as we expect the deformation to be at least approximately isometric [ASP<sup>+</sup>04]. Geodesic distances are computed in two steps: First, we form a graph that connects each surface point to its 20 nearest neighbors. Second, we run a standard Dijkstra algorithm to compute the distance from each keypoint to all other surface points, which yields approximate geodesic distances between all keypoints.

## 5.3 Symmetry Detection

As a third prototypical example application, we have implemented a simple symmetry browser: The user may select on feature on the object and the application returns all features that have been found on the same object with a matching local neighborhood. We do not implement a full, dense symmetry detection algorithm at this point, as this is out of scope of this paper, see [MGP06, PSG<sup>+</sup>06, MSHS06, SKS06] for more elaborated algorithms. However, feature matching is used as a subalgorithm in several of the existing detection techniques so that we would like to examine how reliably we can find reoccurring features within one and the same object.

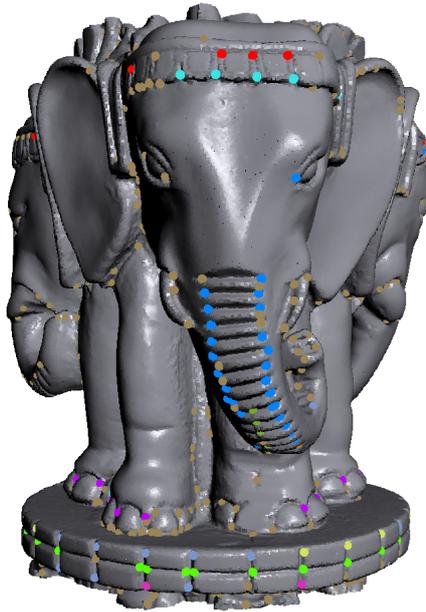


Figure 5: Symmetry example: features which are symmetric to a manually selected feature are shown in the same color (pale brown is unassigned).

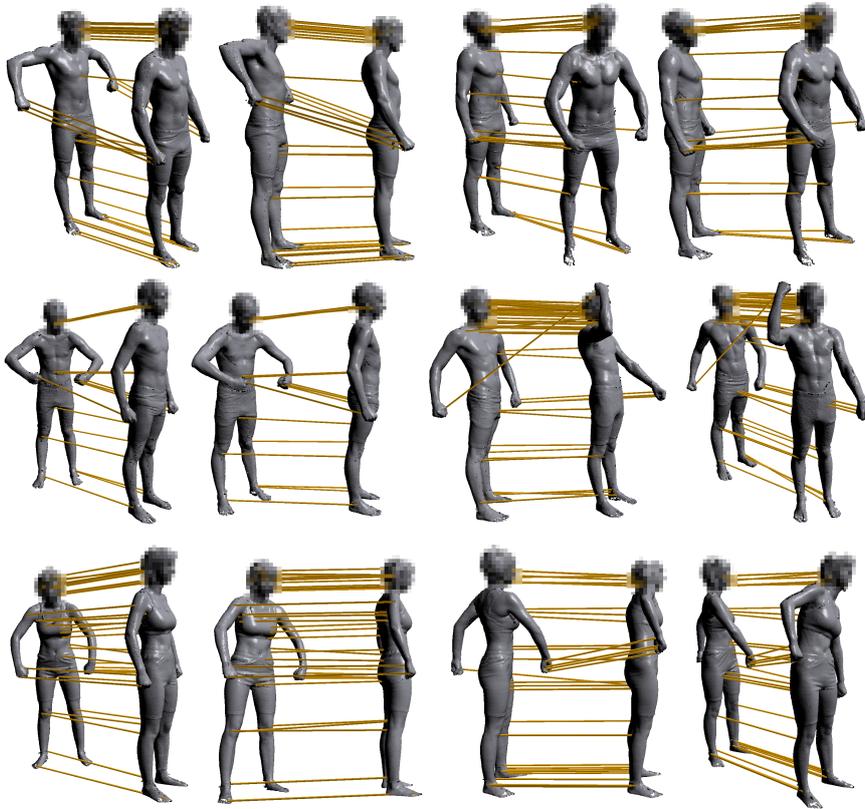


Figure 6: Result of the registration technique with deformable objects.

## 6 Results

### 6.1 Feature Stability

In real world scenarios we have to deal with noise in the data. We investigate this in the following test. Comparable to [LG05], we add uniform noise to the point positions in the normal direction with amplitude 0.5% of the maximum side length of the bounding box and perturb the normals between  $-0.05$  and  $0.05$ . We compare the features found in the noisy data set with the original ones by counting the number of corresponding features. Two features correspond, when their Euclidian distance is less than  $\sigma/2$  and their scale differ less than 1.

Data	points	scales	Smoothing	Slippage	Meanshift	Total
human	89147	10	20	80	31	131
bunny	40256	10	10	30	9	49
buddha	85000	6	10	26	18	54
thai-elephants	592084	4	40	80	127	247

Table 1: Timings for feature detection on a representative scan of several data sets (in seconds).

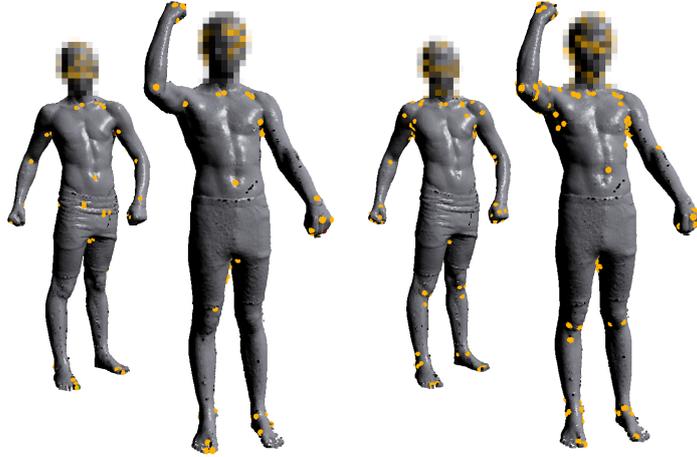


Figure 7: Comparison between DoG features (left) and slippage features (right).

Figure 10 shows the result of this comparison for the well known Stanford bunny and happy Buddha datasets. On the coarser scales one can see that most features can establish a correct correspondence. The peak in the last scale level can be explained by the multi-scale mean shift procedure: all features detected in the coarsest scale shift in the lower ones because the density still increases. If we would compute the a further scale level, these features would appear on their proper scale, but the truncation of the scale level makes all larger scale features occur at the final scale.

As we use the same technique for the stability analysis, our results are directly comparable to the curves in [LG05]. We obtain a similar stability of our generalized features, which shows that the generalization does not harm the stability of the results.

We also compare the feature matching results to those from [LG05]. For this, we have reimplemented the technique closely following the description in the paper. Figure 9 shows the results of our technique and the difference-of-gaussian (DoG) approach. Notice how the slippage features appear robustly at the ends of the notch while the other approach is unable to establish stable features, especially under noise (we used a simple thresholding to remove noisy features on both techniques, with thresholds decreasing from left to right). This behavior is expected, as a maximum curvature criterion of the DoG features is not well suited for lines of constant curvature. This effect can frequently be observed in a real data example, such as the well-known Thai Statue data set depicted in figure 8. Please note that a more stable feature detection is not restricted to the marks on the elephants trunk but we also obtain more and more reliable matches at the ornamental structures. This improved performance is critical for deformable matching (figure 7): Using a threshold that yields stable and reliable keypoints, the DoG features are not able to detect keypoints on all extremities of the model. If we decrease the threshold such that correspondences are detected everywhere, the positions of the keypoints become more random and unstable so that identifying corresponding points becomes much more complicated.

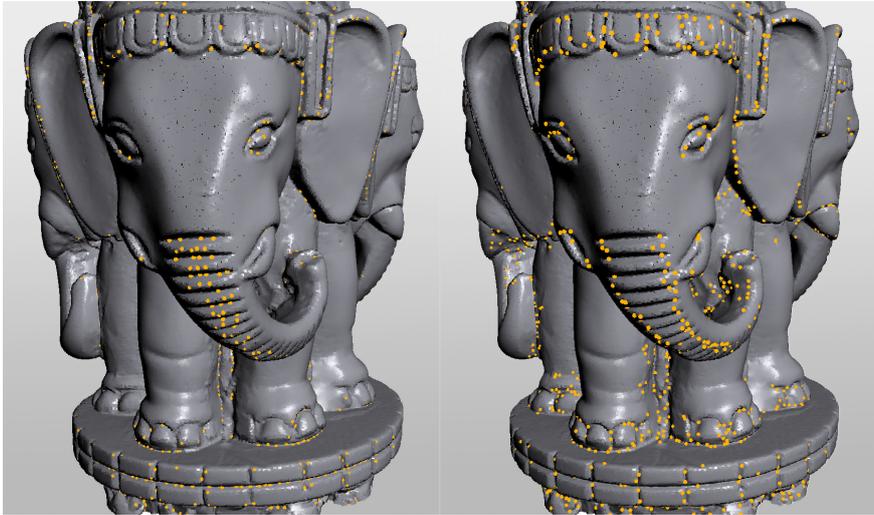


Figure 8: Comparison between DoG features (left) and slippage features (right) on a scanned data set.

## 6.2 Global Registration

We tested our registration pipeline for rigid alignment on the datasets bunny (10 views) and happy buddha (15 views). The most time consuming step is the computation of the slippage features. Timings can be seen in table 1 for a single view. For rigid registration we do not need to perform the full feature matching pipeline: We have skipped the ICP-based local alignment and used only the histogram based signatures to establish a set of initial correspondences. False positive matches are handled by the global validation algorithm. The results are shown in figure 4. Please note that we obtain fairly accurate global registration results without an ICP alignment of the parts; this shows that keypoints are retrieved accurately at corresponding positions.

## 6.3 Deformable Registration

We applied our technique to a set of scanned humans. Each individual was scanned in several poses. We established a set of correspondences between two randomly selected poses. The results are shown in figure 6. Unfortunately, the usage agreement of the data set requires not to show faces of the scanned individuals, so that we have to blur out the corresponding parts of the images. Nevertheless, we obtain a similarly consistent registration parts in the facial area. Notice that every extremity receives at least one correspondence, which is difficult to obtain with feature detectors that look for maximums in surface curvature. These global correspondences could be used as guidance for local, dense deformable registration algorithms such as [WJH<sup>+</sup>07, BR07].

## 6.4 Computational Costs

The runtime of our keypoint detection technique is comparable to that of the DoG features in [LG05], depending on the number of scales and points. Table

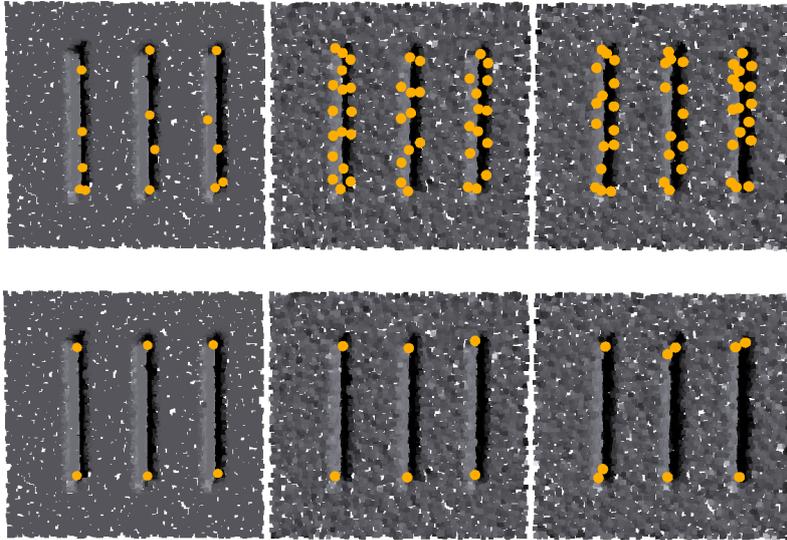


Figure 9: Comparison between DoG features (top) and slippage features (bottom) on a test example. Original example data (left), two noisy copies (middle, right).

1 shows the timings for the individual stages of the algorithm.

## 6.5 Symmetry Detection

We tested our simple symmetry detection scheme with the elephants of the thai statue data set. We selected a few features and detected successfully several symmetric parts. Figure 5 shows the results. Notice that even in complicated parts like the toes and the beak symmetries could be found.

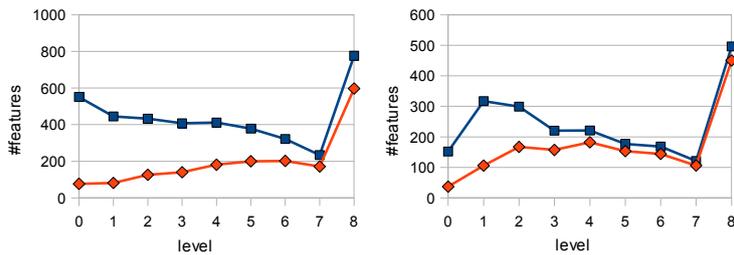


Figure 10: Feature stability for the datasets happy buddha (top) and bunny (bottom). All features (blue), features with correct correspondences (red).

## 7 Conclusions and Future Work

In this paper, we have presented a novel keypoint detection technique based on a local slippage analysis. The main advantage of our approach over the DoG geometric features we build our approach upon is the ability to detect *reliable* key points in more general situations than traditional techniques that these previous techniques. Additionally, we present a feature matching technique based on geometric matching that is able to accurately find feature correspondences of these generalized features. Overall, we obtain a conceptually clean framework to detect the very general class of feature correspondences characterized by well-defined translational and rotational geometry matching, avoiding any further assumptions such as local maxima in surface curvature. In future work, we would like to apply our technique to more sophisticated symmetry detection in geometric objects by considering graphs of features and maximal consistent subgraphs. In addition, we would also like to examine further the potential in deformable matching by complementing our prototype matching technique with a more general, multi-view global validation technique.

## References

- [AKKS99] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In R. Güting, D. Papadias, and F. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99*, volume 1651, pages 207–228, Hong Kong, China, 1999. Springer.
- [ASP<sup>+</sup>04] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, 2004.
- [BR07] Benedict Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), August 2007.
- [CM87] E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):700–703, 1987.
- [CM92] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.
- [CM02] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [FB87] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740, 1987.
- [FHK<sup>+</sup>04] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bulow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, May 2004.
- [GG04] Natasha Gelfand and Leonidas J. Guibas. Shape segmentation using local slippage analysis. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 214–223, New York, NY, USA, 2004. ACM.
- [GGGZ05] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Proc. Shape Modeling International*, page 244–256, 2005.
- [GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM Press.
- [GMGP05] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on Geometry Processing*, pages 197–206, 2005.

- [HFG<sup>+</sup>06] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. Graphics*, 25(3):569–578, 2006.
- [HH01] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. In *IEEE Computer Society Workshop on Computer Vision Beyond the Visible Spectrum (CVBVS 2001)*, December 2001.
- [HTB03] D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.
- [JH99] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.
- [KFR03] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, June 2003.
- [LG05] X. Li and I. Guskov. Multiscale features for approximate alignment of point-based surfaces. In *Symposium on Geometry Processing*, pages 217–226, 2005.
- [LGB06] Xinju Li, Igor Guskov, and Jacob Barhak. Robust alignment of multi-view range data to cad model. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, page 17, Washington, DC, USA, 2006. IEEE Computer Society.
- [LH05] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference of Computer Vision (ICCV)*, volume 2, pages 1482 – 1489, October 2005.
- [Low03] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.
- [MGGP06] Niloy J. Mitra, Leonidas Guibas, Joachim Giesen, and Mark Pauly. Probabilistic fingerprints for shapes. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 121–130, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [MGP06] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.*, 25(3):560–568, 2006.
- [MSHS06] Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François Sillion. Accurate detection of symmetries in 3d shapes. *ACM Transactions on Graphics*, 25(2):439 – 464, April 2006.

- [NK03] Marcin Novotni and Reinhard Klein. 3d zernike descriptors for content based shape retrieval. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 216–225, New York, NY, USA, 2003. ACM.
- [PSG<sup>+</sup>06] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006.
- [SKS06] Patricio Simari, Evangelos Kalogerakis, and Karan Singh. Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 111–119, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [WG02] Joris Vanden Wyngaerd and Luc Van Gool. Automatic crude patch registration: toward automatic 3d model building. *Comput. Vis. Image Underst.*, 87(1-3):8–26, 2002.
- [WJH<sup>+</sup>07] Michael Wand, Philipp Jenke, Qi-Xing Huang, Martin Bokeloh, Leonidas Guibas, and Andreas Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. Eurographics Symp. on Geometry Processing (SGP) 2007*, pages 49–58, 2007.
- [WS02] Michael Wand and Wolfgang Straßer. Multi-resolution rendering of complex animated scenes. *Computer Graphics Forum*, 21(3), 2002. Eurographics 2002.
- [YF02] Sameh M. Yamany and Aly A. Farag. Surfacing signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1105–1120, 2002.