Coarse Registration of Surface Patches with Local Symmetries

Joris Vanden Wyngaerd[1] and Luc Van Gool[1,2]

[1] University of Leuven, ESAT-PSI, Kasteelpark Arenberg 10,
B-3001 Leuven, Belgium
joris.vandenwyngaerd@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/psi/visics
[2] Kommunikationstechnik, ETZ, Gloriastr. 35, ETH Zentrum,
CH-8092 Zürich, Switzerland
vangool@vision.ee.ethz.ch

Abstract. Most 3D recording methods generate multiple partial reconstructions that must be integrated to form a complete model. The coarse registration step roughly aligns the parts with each other. Several methods for coarse registration have been developed that are based on matching points between different parts. These methods look for interest points and use a point signature that encodes the local surface geometry to find corresponding points. We developed a technique that is complementary to these methods. Local descriptions can fail or can be highly inefficient when the surfaces contain local symmetries. In stead of discarding these regions, we introduce a method that first uses the Gaussian image to detect planar, cylindrical and conical regions and uses this information to compute the rigid motion between the patches. For combining the information from multiple regions to a single solution, we use a a Hough space that accumulates votes for candidate transformations. Due to their symmetry, they update a subspace of parameter space in stead of a single bin. Experiments on real range data from different views of the same object show that the method can find the rigid motion to put the patches in the same coordinates system.
Keywords : Surface Registration, Surface geometry, Shape.

1 Introduction

Three-dimensional (3D) models of objects usually are not acquired in one piece. We will refer to a partial reconstruction from a single view as patch. An important step in the procedure to integrate them is the registration. There are good algorithms to do this, e.g. ICP [2,5] or the maximization of mutual information [19]. Nevertheless, such registration algorithms require an initial, coarse registration. Starting from very different initial positions they might not converge to the correct solution. Providing such initial coarse registration automatically is the goal of the work reported here.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 572–586, 2002.

1.1 Previous Work

Registration and 3D model-based recognition are closely related. For both applications, correspondences between 3D surfaces must be determined. Two approaches can be taken to represent surfaces for these tasks, global or local.

Global representations represent the entire surface. Some represent the entire surface by a parametric surface [15]. Another global method is the extended Gaussian image for which matching is based on correlating spherical maps [11, 13]. Because global representations represent the entire surface, they are not well suited for the registration of several partial surfaces.

Local methods can handle partial surfaces better. They typically use a local description of the surface geometry around special surface features. Correspondences are determined by comparing these point signatures [6,7,8,12,21] and the correct relative position between the patches is computed based on these corresponding points. Several methods have been developed to find these point correspondences. A good example is the work by Johnson and Hébert [12]. They use so called spin images for describing the local geometry of all the surface points. Yamani and Farag use a similar approach but apply a selection process based on surface curvature to calculate the surface signatures only for points of interest [21]. This work can also be seen as an extension to our previous work [18] where we use bitangent curve pairs as landmarks on the surface. Bitangent curve pairs represent a point pair with a common tangent plane sliding over the surface. The type of surfaces we analyze here are degenerate cases of bitangency, where bitangent curve pairs are not defined. For example, in a planar part all points have the same tangent plane.

1.2 Problem and Chosen Approach

Point signatures are created based on local information. This is necessary to have a compact description and for robustness against occlusion. However, such local descriptions can become useless when the surface contains (local) symmetry. For example, all points in a plane or on a cylinder will have exactly the same signature and cannot be used to find corresponding points in this kind of regions. Yamani and Farag [21] only calculate surface signatures in points with higher curvature. They discard the points with low curvature because they *"are redundant and do not serve as landmarks of the object"*. This is a problem when all points of a patch have low curvature. The method we report here is complementary because it focuses on points with low curvature, more precisely points that have zero Gaussian curvature. We developed a method that detects planar, cylindrical and conical parts in the surface patches and uses this information to calculate the unknown transformation. Computing rigid motion from many point correspondences has been studied by several authors [10,14]. The problem they solve is typically highly overdetermined because many point correspondences are available. These methods can not be used in the presence of local symmetry, because no individual points can be identified and matched. In this paper we introduce a method for computing the rigid motion by combining

the constraints imposed by the detected planar, cylindrical and conical regions in the surfaces. For example, two planes can be registered correctly while keeping three degrees of freedom, two translational components parallel to the plane and a rotation around the plane's normal vector. They do not define a unique transformation, but they do impose constraints on the possible transformations. We analyze these constraints and combine them by updating a Hough space where votes are accumulated for candidate transformations.

This paper is organized as follows. Section 2 describes how we detect the planar, cylindrical and conical regions in the surfaces. Section 3 describes how we calculate the transformation. Section 4 shows the results of the algorithm. Section 5 concludes this paper.

2 Detecting Planes, Cylinders, and Cones

2.1 The Gaussian Image

The Gaussian image [4] maps the surface normal onto the Gaussian sphere. It provides a unique representation for convex objects. If G represents this mapping, δO a patch on a surface and $G(\delta O)$ the corresponding patch on the Gaussian sphere , the Gaussian curvature K can be defined as

$$K = \lim_{\delta O \to 0} \frac{G(\delta O)}{\delta O} \tag{1}$$

Several extensions have been made to encode more information. The extended Gaussian image (EGI) takes into account the area on the surface of points with the same normal [11]. The complex extended Gaussian image (CEGI) adds the distance of the tangent plane to the origin [13]. The spherical attribute image (SAI) developed by Hébert, Ikeuchi and Delingette [9] also encodes surface connectivity to provide a unique representation of the surface, up to rotation, translation and scale. These representations are useful for recognition [11] and pose determination [4] [13]. Sun and Sherrah use the EGI for symmetry detection [16].

When more information is encoded, creating the representation will be more demanding. For some applications, a more complex representation is justified. When building a database for object recognition for example, using more complex models is reasonable because they can be created off-line and are used many times. In our application, the models are created at runtime and used only once. Using complex models is not practical, especially when many patches are included.

As complementary information to that which our earlier work in [18] can provide, we only want to detect the presence of planar, cylindrical and conical regions. For this, we use the orientation histogram, which is a discrete version of the extended Gaussian image. It is used to find candidate regions for fitting a shape. For these candidates, the type of shape and their parametric representation will be computed. We will refer to the planar, conical and cylindrical regions that we look for as 'shapes'. For a region to be classified as a shape, we require that it is connected.

2.2 Finding Special Regions

The orientation histogram is used for detecting planar, conical and cylindrical regions in a surface. The Gaussian image G maps a surface patch δO in a planar, cylindrical or conical region on a region on the Gaussian sphere with zero area, and points in such regions therefore have a zero Gaussian curvature. Because of this, the shapes we look for generate high values in the orientation histogram. All points in a planar part have the same surface normal and are mapped on the same point on the Gaussian sphere. A cone or cylinder is mapped onto a circle on the Gaussian sphere. This is illustrated in fig. 1.

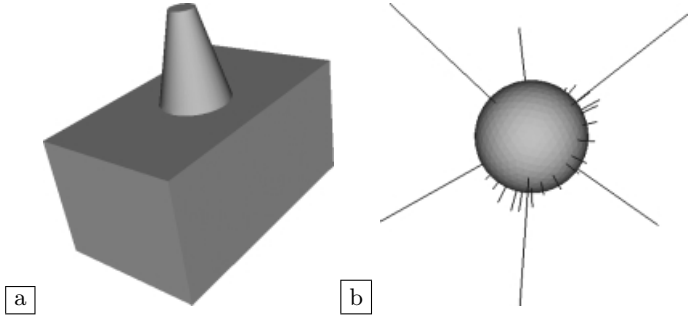

Fig. 1. (a) An example object. (b) The orientation histogram or needle map. Triangles of the unit sphere are histogram bins. The normal to the triangle is also shown with a length proportional to the histogram value. The isolated big peaks correspond to the planes of the box. The circular configuration of smaller peaks corresponds to the cone.

To locate candidate regions we detect peaks in the orientation histogram. Neighboring bins are combined to form clusters. For non-convex objects, such a cluster can contain different shapes. For example, all points in parallel planar regions will fall in the same cluster. The clusters are split so that they only contain one shape. In a first step we split clusters based on the requirement that the points in the shape have to be connected. It is easy to see that two parallel but non-coplanar planar regions cannot be connected on the surface without leaving the cluster. However, this can be possible for two cylindrical regions. When a candidate region has been identified, we have to determine the type of shape and its parametric representation. In a first step, we determine whether the region can be planar or not. This is done by observing that for a plane, all normals point in the same direction, while this is not the case for cones and cylinders. If a region is classified as non-planar, we take a second step by splitting the region into convex parts.

To validate the hypotheses, we will fit a parametric model and calculate the Euclidean distance between the points that were used for fitting and the shape represented by the parametric model.

2.3 Fitting Models to the Data

We represent a plane by its equation

$$P : ax + by + cz + d = 0 \tag{2}$$

$\mathbf{N}(a, b, c)$ represents the normal to the plane. We require that it is a unit vector. Like this, the absolute value of the residue of this equation represents the Euclidean distance between a point and the plane. We also require that the normal is directed outward. This is done by giving it the same direction as the corresponding bin in the orientation histogram.

Fitting a 3D plane to a set of points is a text book least square problem. Suppose we have a number of points $\mathbf{X}(x, y, z)$ and have to determine the parameters (a, b, c, d). Least squares will minimize the residue of this equation and as such minimize the Euclidean distances of the points to the plane. Fitting of cones and cylinders to data is a lot more complicated. One approach is to use the generic equation of a quadric :

$$q(x, y, z) = ax^2 + by^2 + cz^2 + 2\,hxy + 2\,gxz + 2\,fyz + 2\,ux + 2\,vy + 2\,wz + d = 0 \tag{3}$$

The parameters of this equation can also be determined by a linear least square method, similar to the plane. This however poses two problems. First, the residue of this equation is not the Euclidean distance, but an algebraic distance. Bolle and Cooper [3] use a different parameterization of cylinders such that the residue represents the Euclidean distance. The solution is found with a non-linear optimization, which is computationally expensive. Secondly, no guarantee exists that the resulting quadric represents a cone or a cylinder. Especially when only partial data is available and in the presence of noise, the fitting can result in other types of quadrics. This is illustrated in figure 2. Werghi *et al.* [20] developed a method for constraining the solution to for example a cylinder. They introduce Lagrange multipliers to impose constraints on the unknown parameters. The nonlinear constrained optimization problem is solved with a Levenberg-Marquardt method. This leads to a computationally expensive numerical algorithm.

The reason why a procedure that minimizes the Euclidean distance is complicated is that no closed form expression exists. We illustrate this by explaining the procedure we use to calculate the distance between a point $\mathbf{X_1}(x_1, y_1, z_1)$ and a quadric represented by (3). Let $\mathbf{X}(x, y, z)$ denote the (unknown) point on the quadric that is closest to $\mathbf{X_1}$. This point can be found by minimizing the square of the Euclidean distance between $\mathbf{X_1}$ and $\mathbf{X}$:

$$d^2(x, y, z) = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \tag{4}$$

subject to the constraint (3), i.e. that $\mathbf{X}$ lies on the quadric. The function to be minimized becomes:

$$d^2(x, y, z) + \lambda\, q(x, y, z) \tag{5}$$

Partial differentiation with respect to the unknowns (x, y, z, λ) leads to the following set of equations :

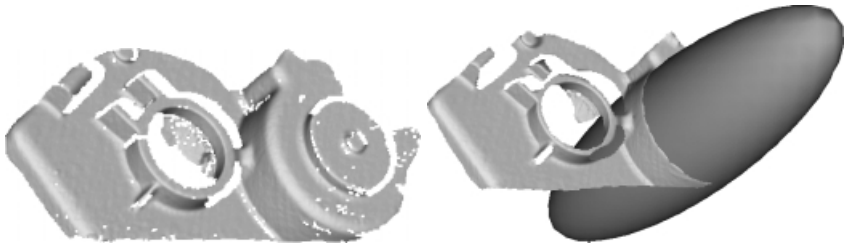

Fig. 2. (a) Surface with a conical region. Due to the partial nature of our data and the geometry of the object, only a very small part of the mathematical cone is present in the data. (b) Due to measurement noise, the quadric fitted to these points is not a cone, but an ellipsoid.

$$\begin{cases} x - x_1 + \lambda\, ax + \lambda\, hy + \lambda\, gz + \lambda\, u = 0 \\ y - y_1 + \lambda\, hx + \lambda\, by + \lambda\, fz + \lambda\, v = 0 \\ z - z_1 + \lambda\, gx + \lambda\, fy + \lambda\, cz + \lambda\, w = 0 \\ ax^2 + by^2 + cz^2 + 2\, hxy + 2\, gxz + 2\, fyz + 2\, ux + 2\, vy + 2\, wz + d = 0 \end{cases} \tag{6}$$

The first three equations are linear and can be solved for x,y and z. Substitution in the last equations leads to a polynomial in λ of degree 6. The real roots are substituted in (4) and the one with the smallest distance is chosen.

It is clear that this distance is not a closed form solution. Using this for calculating the residue in an iterative optimization procedure would be computationally expensive. We made a trade-off between accuracy and speed. As will be clear in the next section, for our registration application there is no need to have the most accurate parameters. For the parameter fitting we minimize the algebraic distance. Only in a validation step we will calculate the Euclidean distances.

3 Registration

In this section we explain how we compute the correct relative position of two patches. We need to determine six degrees of freedom, three translation parameters and three rotation parameters. Assume we found a number of shapes and their parametric representation as explained in the previous section. Because of the symmetry of the shapes, a single pair of corresponding shapes does not result in a unique transformation to align the surfaces. A pair of corresponding shapes imposes constraints on the possible transformations, but will leave some degrees of freedom. For example, when two planes are aligned, three degrees of freedom remain. The planes can still be translated in directions parallel to the plane and rotated around the plane's normal $\mathbf{N}$. Combining the constraints imposed by all shapes can lead to a unique solution. For example, three non-parallel planes

with different angles between them result in a unique solution. The second plane still leaves a translation along the intersection line which is solved by the third plane.

To combine the constraints imposed by all the shapes, we use a Hough space [17]. Tian and Shah used a Hough space to recover 3D motion in 2D images. They divide the input images in patches and for every patch, a candidate transformation is computed and generates a single vote in a single bin in parameter space. In our application, updating the parameter space for candidate corresponding shapes is more complicated, because many possible transformations exist and many bins have to be updated. The benefit of using a voting scheme is that no matching is needed, and matching for example partial planar parts would be difficult. For all shapes found on one surface, we update parameter space for all shapes of the same type on the other surface, unless we have additional information that can eliminate some possibilities.

For practical reasons, we do not directly work in the six dimensional parameter space. Instead, we separate rotation and translation, each in a three dimensional parameter space. We will first determine the rotation and then the translation. Separating rotation and translation has several advantages. First, a 6D parameter space would be memory and computationally expensive. Second, initially we do not know the bounds of the possible translations. After applying the rotation we can calculate the bounds for the translation.

We will describe our method by the constraints imposed by a pair of corresponding planes. The extension for cones and cylinders is straightforward. Let S_R be the reference surface and S_M the moving surface. We will determine the Euclidean transformation that, when applied to S_M, best aligns both patches. We denote the unknown rotation and translation that aligns S_M to S_R by $\mathbf{R}_{MR}$ and $\mathbf{T}_{MR}$ respectively. We represent a translation as $\mathbf{T}(t_x, t_y, t_z)$ and a rotation around an axis $\mathbf{A}(a_x, a_y, a_z)$ over an angle α is represented by three angles

$$\mathbf{R}(\alpha, \beta, \gamma). \tag{7}$$

β and γ represent the direction of the axis, α the angle of rotation around this axis. Without loss of generality, we assume

$$a_z \geq 0, \| \mathbf{A} \| = 1, \tag{8}$$

If $a_z < 0$, we invert the axis $\mathbf{A}$ and the angle α. β represents the angle of the projection of the axis in the xy plane and the x axis, γ represents the angle of the axis with the xy plane. We require that

$$0 \leq \alpha < 2\pi, -\pi \leq \beta < \pi, 0 \leq \gamma \leq \tfrac{\pi}{2} \tag{9}$$

β and γ are calculated as

$$\begin{aligned} \beta &= \arctan\left(\tfrac{a_x}{a_y}\right) \\ \gamma &= \arccos\left(\sqrt{a_x^2 + a_y^2}\right) \end{aligned} \tag{10}$$

3.1 Aligning Planes

Assume we have a pair of planes for which we will investigate the constraints imposed on the transformation parameters. Let P_m be a plane on the moving surfaces that we want to align to the plane P_r on the reference surface, each with their parametric representation

$$\begin{array}{l} P_r : \quad a_R x + b_R y + c_R z + d_R = 0 \\ P_m : a_M x + b_M y + c_M z + d_M = 0 \end{array} \tag{11}$$

Their respective unit normal vectors (pointing outward) are $\mathbf{N_R}$ and $\mathbf{N_M}$. One possibility to determine these constraints is to use the general representation of a rotation and translation, determine how the parameters of P_m are transformed and identifying with the parameters of P_r. This leads to a system of four equations which is not easy to solve symbolically because of two reasons. First, because of the rotation, the problem becomes nonlinear in the unknown parameters. Second, we have less equations than unknowns. Since the equations can be made polynomial, Gröbner bases [1] allowed us to find a solution. However, solving for planes is hard, but solving this for quadrics becomes infeasible. We found the same solution by looking at it geometrically. The two planes can be aligned in two steps :

1. We rotate P_m such that it becomes parallel to P_R. This is the same as making the normals parallel and can be done by a rotation $\mathbf{R}_1$ with axis $\mathbf{A}_1 = \mathbf{N_R} \times \mathbf{N_M}$ over an angle $\alpha_1 = \arccos(\mathbf{N_R} \cdot \mathbf{N_M})$.
2. We translate P_m such that it aligns to P_R. This is done by translating along $\mathbf{N_R}$ over a distance $|d_M| \; - \; |d_R|$,

$$\mathbf{T}_1 = (|d_M| \; - \; |d_R|)\mathbf{N}_R \tag{12}$$

We extend this solution to all possible solutions by adding two operations :

1. A rotation $\mathbf{R}_p$ around the reference $\mathbf{N_R}$, the normal of the reference plane, over a free angle α_p
2. A translation $\mathbf{T}_p(t_{xp}, t_{yp}, t_{zp})$ parallel to the reference plane. This can be expressed as

$$a_R t_{xp} + b_R t_{yp} + c_R t_{zp} = 0 \tag{13}$$

We conclude that any transformation that aligns the planes can be described as a sequence of four operations : $\mathbf{T}_p \circ \mathbf{R}_p \circ \mathbf{T}_1 \circ \mathbf{R}_1$. In general, rotations and translations are not commutative, but here we can change the order of $\mathbf{R}_p$ and $\mathbf{T}_1$, because $\mathbf{T}_1$ is a translation along the axis of $\mathbf{R}_p$. Translations and rotations can be combined to a single rotation followed by a single translation : $\mathbf{T}_{MR} \circ \mathbf{R}_{MR}$ with $\mathbf{T}_{MR} = \mathbf{T}_p \circ \mathbf{T}_1$ and $\mathbf{R}_{MR} = \mathbf{R}_p \circ \mathbf{R}_1$.

3.2 Determining the Rotation $\mathbf{R}_{MR}$

To find a rotation that can align all shapes and hopefully aligns the surfaces we first examine the constraint imposed by a pair of corresponding planes that have

to be made parallel. For a plane pair, $\mathbf{R}_{MR}$ has one free parameter, α_p. This α_p is a parameterization of a 3D curve in the parameter space (α, β, γ). This is illustrated for a simple surface with 2 planar parts in figure 3. Every possible plane pair (P_m, P_r) generates such a curve. Rotations that can align all planes are found at intersections of curves that correspond to matching plane pairs. For finding these intersections, the three dimensional rotation parameter space

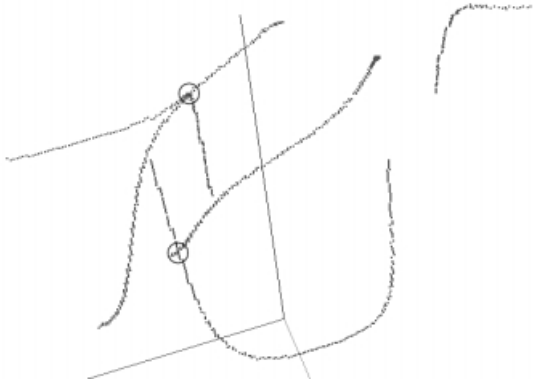

Fig. 3. The rotation parameter space showing the constraints imposed by 2 plane pairs. This generates four mathematically closed curves in parameter space. In this visualization, the curves are not closed because the values represented are angles defined in the interval $[0, 2\pi[$. The circles indicate two intersection points, which correspond to the two possibilities to align two pairs of planes.

is discretized by dividing the volume defined by (9) with a rectilinear grid. This grid subdivides the volume into a number of bins. The center of the bin serves as a quantization for all points in the volume defined by the bin. The size of the bins determines the accuracy of the solution. The bins will be used to count the number of curves that visit them.

For all possible pairs (P_m, P_r), we update the bins that are visited by their constraint curve. The free parameter α_p is varied from 0 to 2π and for all values, the corresponding $\mathbf{R}_{MR}$ is calculated. The bin containing this rotation is updated. This process is repeated for all possible plane pairs (P_m, P_r). The resulting parameter space is illustrated in figure 5 (a) for a surface with 6 planar parts as shown in figure 4. The bins that have been visited by the highest number of curves are used as candidate rotations. The accuracy of the rotation is limited by the resolution of the bins. For candidate rotations we do a refinement step. For such a candidate some of all the possible plane pairs can be eliminated because the candidate rotation does not align them. A Levenberg-Marquardt optimization is used in which the sum of the angles between remaining plane combinations serves as the error function.

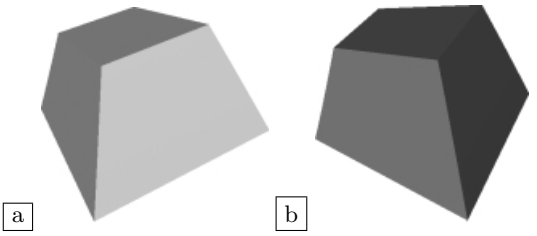

Fig. 4. (a) A test surface consisting of six planes. It is a box that tapers off along x, y and z. The surface does not have global symmetry such that a unique solution exists. (b) The surface after rotation and translation.

3.3 Determining the Translation $\mathbf{T}_{MR}$

We first examine the constraint on the possible translations imposed by a pair of corresponding planes. Since $\mathbf{T}_{MR} = \mathbf{T}_1 + \mathbf{T}_p$,

$$\mathbf{T}_p = \mathbf{T}_{MR} - \mathbf{T}_1. \tag{14}$$

Substituting (14) in (13) results in a constraint in the parameter space for $\mathbf{T}_{MR}(t_x, t_y, t_z)$:

$$a_R t_x + b_R t_y + c_R t_z - a_R t_{x1} - b_R t_{y1} - c_R t_{z1} = 0, \tag{15}$$

or by substituting (12) :

$$a_R t_x + b_R t_y + c_R t_z - (|d_M| \ - \ |d_R|) = 0 \tag{16}$$

This means that the constraint imposed by aligning two planes represents a plane in the translation space. The translation that is able to align all corresponding planes is found as the intersection of all their corresponding constraint planes.

To find this solution, we take a similar approach as used for finding the rotation. The translation parameter space is discretized and for every constraint plane the visited bins are updated. An update in the translation space will update more bins, since we have two degrees of freedom. An additional complication is that initially, no bounds for the solution exist. Updating is done efficiently by using a recursive subdivision scheme. We also take into account the candidate rotations found as described in the previous section. This allows us to calculate the bounds for the translation. It also allows us to reduce the number of plane pairs for which we have to update the translation parameter space. Only plane pairs are kept which, after applying the rotation, are parallel.

Figure 5 (b) shows an example of the translation space. The bins that have been visited by the highest number of constraint planes are selected. The translation represented by a bin is an approximation due to the discretization of the parameter space. However, we can perform an inexpensive refinement step. We select the constraint planes that intersect the selected bin and the intersection of the remaining constraint planes is calculated with a least square method.

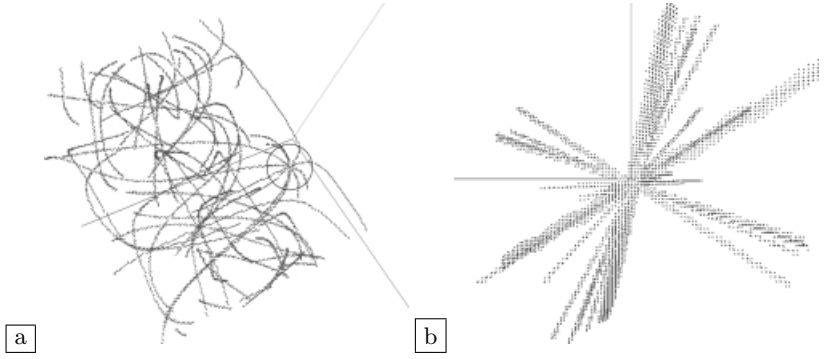

Fig. 5. Parameter spaces for the surfaces of figure 4. (a) The orientation parameter space. The circle indicates the intersection point of 6 curves. (b) The translation parameter space. Only bins which have been visited by at least two constraint planes are shown.

3.4 Extension to Cones and Cylinders

After aligning two cones or cylinders, a rotational degree of freedom around their axis remains. Theoretically, the position of the top of a cone can fix the translation. However, as illustrated in figure 2, we have very partial data and often it is impossible to get an accurate estimate of the top of the cone. Because of this, the top is not used in determining the transformation and we only use the axes of cones and cylinders. Aligning these axis is very similar to aligning planes. Updating the rotation parameter space is the same as for a pair of planes, but in stead of aligning normals, the axes are aligned. The difference is that the axes do not have a direction and because of this, two possibilities for the initial rotation R_1 have to be taken into account. The translation constraint imposed by a pair of axes is a straight line in translation parameter space. The point-direction representation of this constraint line is

$$\mathbf{T}(t_x, t_y, t_z) = \mathbf{X}_R - \mathbf{X}_M + t\mathbf{A}_R, \tag{17}$$

in which $\mathbf{X}_M$ represents a point on the axis of the reference shape. $\mathbf{X}_M$ is a point on the axis of the moving shape after applying the rotation that was computed. $\mathbf{A}_R$ represents the axis direction of the reference shape.

3.5 Validation

The procedure as explained in the previous sections does not always lead to a unique solution although the data presents enough information. For example, because the rotation is calculated independently, sometimes more than one rotation exists that can make all planes parallel, even though only one of them leads to the correct translation. A typical example is data containing planes in three perpendicular directions, such that three possible rotations exist. When

multiple possibilities remain, the best one is chosen by 'measuring' the distance between the patches after applying the corresponding transformation. The distance measure used is the number of points on S_M that find a partner point on S_R at close range.

4 Experiments

Figure 6 shows the orientation histogram and the shapes that are found for the surface in figure 2 (a). The shapes found provide enough constraints for registration. Figure 7(a) shows the same surface subject to a known rotation and translation. This transformation is correctly found by our method as illustrated in figure 7(b). The orientation space has $180 \times 180 \times 45$ bins resulting in a one degree accuracy. The translation space has $100 \times 60 \times 70$ bins. The solution is correct taking into account the discretization of the parameter space.

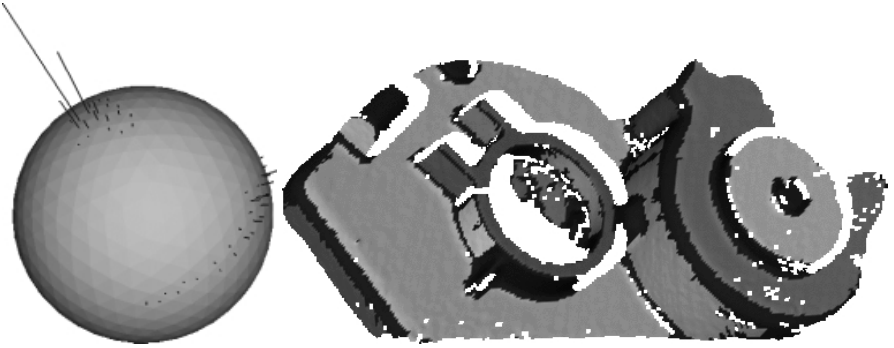

Fig. 6. (a) The orientation histogram for the partial reconstruction from an engine casting from figure 2 (a). The largest peak corresponds to the large parallel planar regions. On the lower side of the sphere, a circle of smaller peaks can be observed. They correspond to the conical part (b) The regions in which a shape was found (gray). The darkest parts do not belong to any shape. 13 shapes are detected on this surface. 11 are classified as planar. The most planar conical part has been incorrectly classified as planar.

Figure 8 shows two partial reconstructions of an engine casting taken from different views. In this example, the rotation histogram results in two candidate rotations. This results in two possible solutions, shown in 9(a) and (b). The second solution is chosen in the validation step.

5 Summary and Conclusions

We developed a technique for the coarse registration of surface patches that is complementary to methods that look for features of interest and describe the

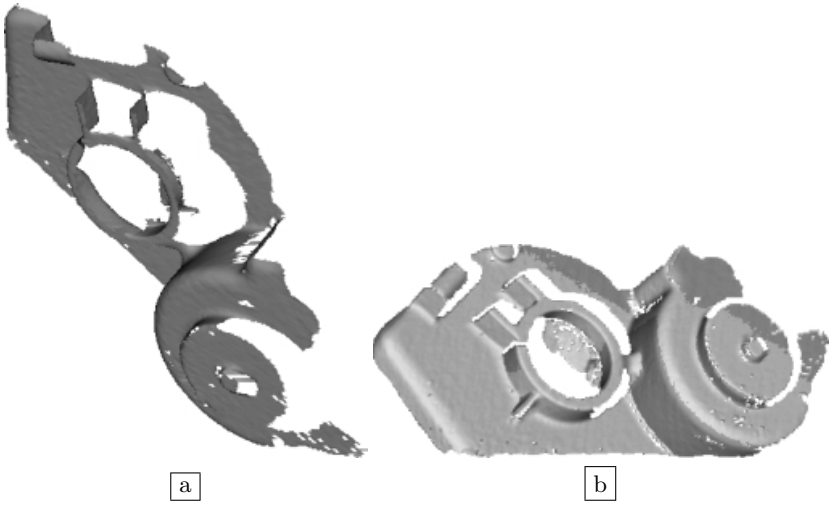

Fig. 7. (a) The surface of figure 2 (a), after applying a translation $\mathbf{T}(5, 5, 5)$ and a rotation of 100 degrees around an axis with direction $(1, 1, 1)$. (b) Both surfaces after applying the transformation that was found.

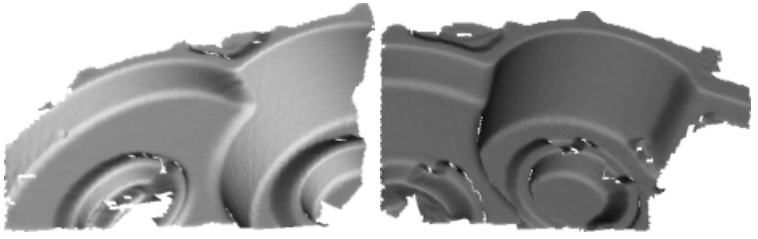

Fig. 8. Two partial reconstructions of an engine casting.

local geometry around them. Such methods could fail for surfaces containing regions with local symmetries, whereas the proposed method exploits their information to perform registration. Our method uses an orientation histogram to detect regions on a surface that are planar, conical or cylindrical. A parametric surface is fitted to the points in such a region. A symmetric shape present in both patches leaves degrees of freedom but does pose constraints on the possible transformations. For combining the constraints imposed by the special regions, we use a Hough space where votes are accumulated for candidate transformations. Transformations that have high votes are validated and the transformation that best aligns both surfaces is selected. Experiments show that this method is able to position partial views that were taken over a wide baseline.

Future work will include the integration of the technique reported here with feature signature methods, which also generate candidate transformations that could be integrated in the voting scheme. The points of the surface can be split

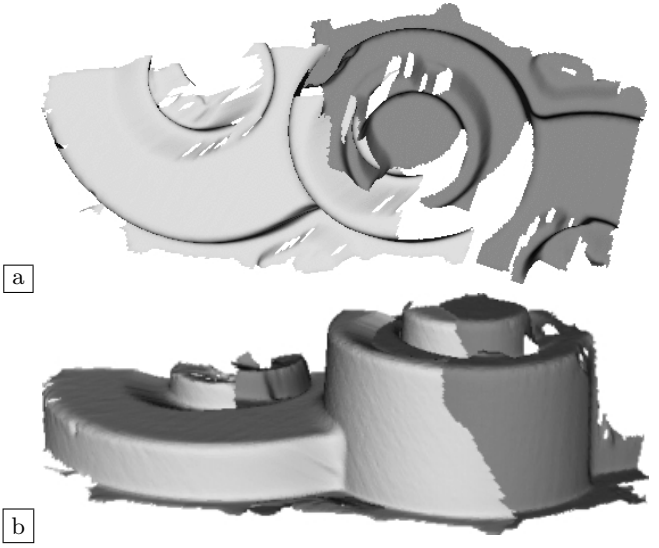

Fig. 9. The solution found for the surfaces of figure 8. For this example, two possible rotations are found. These possibilities differ by a rotation of 180 degrees around the axis of symmetry of the central part. (a) Solution when the wrong rotation is used. (b) Correct solution.

based on curvature. For points with high curvature, a point signature method could be used, and the method proposed here for the low curvature points.

Acknowledgments. Joris Vanden Wyngaerd gratefully acknowledges support by a grant of the Flemish Instute for the Advancement of Science in Industry 'IWT'. The authors also gratefully acknowledge support from the Belgian Federal Government IUAP4/24 "IMechS project" and the GOA project 'VHS+', financed by Kath.Un.Leuven.

References

1. W.W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*, Graduate Studies in Mathematics, Vol. 3., American Mathematical Society, 1994
2. P. Besl, N. McKay, *A method of registration of 3-D shapes*, IEEE Trans. PAMI 12 (2) pp. 239-256, 1992
3. R. M. Bolle and D. B. Cooper. *On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data.* IEEE Trans. PAMI 8 (5) pp. 619-638, 1986
4. P. Brou, *Using the Gaussian Image to Find the Orientation of an Object*, Int'l J. Robotics Research, vol. 3, pp. 89-125, 1983
5. Y. Chen and G. Medioni, *Object modeling by registration of multiple range images*, Proc. Int. Conf. on Robotics and Automation, pp. 2724-2729, 1991

6. C.S. Chua and R. Jarvis, *Point signatures : A new representation for 3D object recognition*, Int. J. of Computer Vision, 25(1), pp. 63-85, 1997
7. J. Feldmar and N. Ayache, *Rigid, affine and locally affine registration of free-form surfaces*, TR INRIA Epidaure, No. 2220, 1994
8. J. Feldmar, N. Ayache, and F. Betting, *3D-2D projective registration of free-form curves and surfaces*, TR INRIA Epidaure, No. 2434, dec. 1994
9. H. Hebert, K. Ikeuchi, and H Delingette, *A spherical representation for recognition of free-form surfaces*, IEEE Trans. PAMI 17 (7) pp. 681, 1995
10. Berthold K. P. Horn (1987), *Closed-form solution of absolute orientation using unit quaternions*, Journal of the Optical Society of America A, 4:629-642
11. K. Ikeuchi., *Recognition of 3-D Objects Using the Extended Gaussian*, Image. In Proc. of Seventh IJCAI, pages 595-600, 1981.
12. A. Johnson and M. Hebert, *Recognizing objects by matching oriented points*, Proc. Conf. Computer Vision and Pattern Recognition, 684-689, San Juan, 1997
13. S.B. Kang and K. Ikeuchi, *3-D Object Pose Determination Using Complex EGI*, T.R. CMU-RI-TR-90-18, Robotics Institute, Carnegie Mellon University, 1990
14. Y. Liu and M.A. Rodrigues, *Essential Representation and Calibration of Rigid Body Transformations*, Machine Graphics and Vision Journal Vol 9 (2000).
15. A. P. Pentland, *Perceptual Organizations And The Representation Of Natural Form*, Artificial Intelligence, vol. 28, no. 2, pp. 293-331, 1986
16. C. Sun, and J.Sherrah, *3-D Symmetry Detection Using The Extended Gaussian Image*, IEEE Trans. PAMI 19 (2) pp. 164-168, 1997
17. Tina Y. Tian and Mubarak Shah. *Recovering 3d motion of multiple objects using adaptative hough transform.* IEEE Trans. PAMI, 19(10):1178
18. Vanden Wyngaerd, J., Van Gool, L., Koch, R., Proesmans, M., 1999. Invariant-based registration of surface patches. Proc. International Conference on Computer Vision, IEEE Computer Society Press, pp. 301–306.
19. P. Viola and W. Wells, *Alignment by maximisation of mutual information*, Proc. Int. Conf. on Computer Vision, pp. 16-23, 1995
20. N. Werghi R.B. Fisher, A. Ashbrook and C. Robertson, *Faithful Recovering of Quadric Surfaces from 3D Range Data*, Proc. 2nd Int. Conf. on 3-D Digital Imaging and Modeling, Ottawa, Canada, pp 280-289, October 1999
21. S. Yamany and A. Farag, *Free-form surface registration using surface signatures*, Int. Conf. on Computer Vision, pp. 1098-1104, 1999