Approximation in the space of planes — Applications to geometric modeling and reverse engineering

Martin Peternell and Helmut Pottmann
Institut für Geometrie, Technische Universität Wien,
Wiedner Hauptstrasse 8–10, A–1040 Wien, Austria

Abstract

We discuss recent progress on approximation in the space of planes. Based on an appropriate distance measure introduced in this space, modeling problems with developable surfaces can be solved with curve approximation algorithms. Moreover, recognition and reconstruction of planar faces in clouds of measurement points appears as a clustering problem in the space of planes. Examples from applications illustrate the practical applicability of this concept.

Keywords:
reverse engineering, computer aided design, space of planes, planar faces, building reconstruction, reconstruction of developable surfaces.

peternell@geometrie.tuwien.ac.at, pottmann@geometrie.tuwien.ac.at

1 Introduction

Our original motivation for studying approximation in the set of planes came from the computational geometry of developable surfaces. There, it turned out that viewing these surfaces as envelopes of planes yields computational advantages [1, 5, 12, 13].

By projective duality, projective and algebraic properties of curves can easily be transferred to developable surfaces. NURBS curves correspond to

NURBS developable surfaces. However, when dealing with approximation, we need distance measures, and thus projective duality is not applicable. We have shown in earlier papers how to solve this problem by introducing a Euclidean metric in the space of planes [12, 11, 13].

In the present paper, we will further investigate the metric in the space of planes and present new applications of this concept. The first application concerns the recognition and reconstruction of planar faces in point clouds (scanned objects). We will show a solution based on the detection of clusters in the space of planes. The second application deals with the approximation of a point cloud or a surface by a developable surface. In contrast to earlier approaches [2, 5, 12] the data can come from a doubly curved surface. The solution is based on curve fitting to a point cloud in the dual space.

2 A Euclidean metric in the space of planes

In order to solve approximation problems in the set of planes, it is necessary to introduce an appropriate *distance* between two planes. Euclidean geometry does not directly provide such a distance function. All invariants are expressed in terms of the angle between planes and are inappropriate for our purposes. In view of applications, we are interested in the distances of points of the two planes which are near some *region of interest*, and this distance can become arbitrarily large with the angle getting arbitrarily close to zero at the same time.

We use the following well-known facts from projective geometry. If we extend real Euclidean 3-space E^3 by ideal points (points at infinity), i.e., intersections of parallel lines, we obtain a model of real projective 3-space P^3. All ideal points form a plane in P^3, the so-called ideal plane. The set of planes in P^3 is a projective space itself, the dual projective space. It is isomorphic to P^3.

Analytically, one uses homogeneous Cartesian coordinates (x_0, x_1, x_2, x_3) for points. For points not at infinity, i.e., $x_0 \neq 0$, the corresponding inhomogeneous Cartesian coordinates will be denoted by

$$x = \frac{x_1}{x_0}, \quad y = \frac{x_2}{x_0}, \quad z = \frac{x_3}{x_0}.$$

A *plane* with equation $u_0x_0 + u_1x_1 + u_2x_2 + u_3x_3 = 0$, or, equivalently, $u_0 + u_1x + u_2y + u_3z = 0$ can be represented by its *homogeneous plane coordinates* $U = (u_0, u_1, u_2, u_3)$.

2.1 Definition of a Euclidean metric in the space of planes

In the following, we will review results from [12, 11] on the introduction of a Euclidean metric in the dual space. We first have to obtain the structure of an affine space. Given a projective space P, we obtain an affine space if we remove a hyperplane from P. Thus, we have to remove the points of a plane from the dual space. Viewed from the original space P^3, this means we have to remove a bundle of planes from P^3. Since we actually want to remove the ideal plane, this bundle must have a vertex at infinity. Hence, if we remove all planes passing through a fixed ideal point (for example, planes through the ideal point of the z-axis = planes parallel to the z-axis), we get a set of planes which has the structure of an affine space. This is easily seen in the analytic model. Planes, which are not parallel to the z-axis, can be written in the form

$$z = u_0 + u_1 x + u_2 y, \tag{1}$$

i.e., they have homogeneous plane coordinates $U = (u_0, u_1, u_2, -1)$. We see that (u_0, u_1, u_2) are affine coordinates in the resulting affine space A^* (of planes, which are not parallel to the z-axis).

We will now introduce a Euclidean metric in A^*. Thereby we make sure that the deviation between two planes shall be measured within some region of interest. This region shall be captured by its projection Γ onto the xy-plane.

For a positive measure μ in $\mathbb{R}^2$ we define the distance d_μ between planes $A = (a_0, a_1, a_2, -1)$ and $B = (b_0, b_1, b_2, -1)$ as

$$d_\mu(A, B) = \|(a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)y\|_{L^2(\mu)}, \tag{2}$$

i.e., the $L^2(\mu)$-distance of the linear functions whose graphs are A and B. This, of course, makes sense only if the linear function which represents the difference between the two planes is in $L^2(\mu)$. We will always assume that the measure μ is such that all linear and quadratic functions possess finite integral.

A useful choice for μ is the Lebesgue measure $dxdy$ times the characteristic function χ_Γ of the *region of interest* Γ (Fig. 1). If $\mu = dxdy\chi_\Gamma$, we have

$$d_\mu(A, B)^2 = \int_\Gamma ((a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)y)^2 dxdy. \tag{3}$$

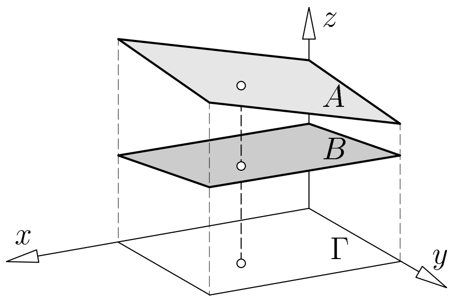


Figure 1: To the definition of the deviation of two planes.

We write $d_\Gamma(A,B)$ instead of $d_\mu(A,B)$. With $c_i := a_i - b_i$, equation (3) can be written as

$$d_\Gamma(A,B)^2 = (c_0, c_1, c_2) \cdot \begin{pmatrix} \int 1 & \int x & \int y \\ \int x & \int x^2 & \int xy \\ \int y & \int xy & \int y^2 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}. \tag{4}$$

This is a quadratic form, whose matrix depends on the domain of integration Γ for the integrals (where we omitted the differentials $dxdy$ for brevity).

Another possibility is that μ equals the sum of several point masses at points (x_j, y_j); see [4]. In this case we have

$$d_\mu(A,B)^2 = \sum_j ((a_0 - b_0) + (a_1 - b_1)x_j + (a_2 - b_2)y_j)^2. \tag{5}$$

Theorem 1 *The distance d_μ defines a Euclidean metric in the set of planes of type (1), if and only if μ is not concentrated in a straight line.*

Proof: See [12]. □

In this way, approximation problems in the set of planes are transformed into approximation problems in the set of points in Euclidean 3-space, whose metric is based on d_μ.

The introduced metric depends on the *choice of the reference direction*, which we identified with the z-axis of the underlying coordinate system. One has to note that with decreasing angle between the considered planes and the reference direction, the geometric meaning of the distance is getting worse.

This motivates the following method for determining an appropriate reference direction.

Assuming for the moment an arbitrary coordinate system, the reference direction (z-axis of the new coordinate system), which shall be represented by the unit vector v, can be determined as follows: Let the planes U_i under consideration be given by $U_i : u_{i,0} + u_{i,1}x + u_{i,2}y + u_{i,3}z = 0$ with unit normal vectors $n_i = (u_{i,1}, u_{i,2}, u_{i,3})$. We want to minimize the angles α_i between the unknown vector v and the unit normal vectors n_i. For that we maximize the sum of squared cosines of the angles α_i. This results in the maximization of the quadratic form

$$Q(v) = \sum_i (n_i \cdot v)^2,$$

under the constraint $\|v\| = 1$. This is a well known eigenvalue problem.

It might even be necessary to use *different reference directions* in order to fully cover the space of planes appropriately. This results in different local mappings of the space of planes to affine 3-space and in different Euclidean metrics. For the application we are dealing with in the next section, this strategy is sufficient.

Remark: For the computation of an interpolating or approximating real valued function defined on all planes within some region of interest, one will construct partial solutions by local mappings to Euclidean 3-space and application of known techniques for scattered data interpolation or approximation in this point model. Those partial solutions have to merged later into a final interpolating/approximating function. A method how this can be handled follows from [10]. There, interpolating functions on lines are constructed by local mappings of line space into affine 4-space. We are currently investigating applications of scattered data fitting in the space of planes: One application deals with the detection of symmetries (with respect to planes) of an object or point cloud. Another application concerns the approximation of a surface or point cloud by a sweeping surface which is generated by the motion of a planar profile curve.

2.2 Properties of the metric in the space of planes

At first sight, there seems to be a wide variety of possible choices of the metric because of the choice of the domain of interest. However, it is clear that the introduction of a Euclidean metric in an affine space requires the choice of an ellipsoid which plays the role of the unit sphere. A uniform scaling of this ellipsoid implies a multiplication of all arising Euclidean distances with a constant factor, and thus it is not relevant. This amounts to 5 real parameters which fix a Euclidean metric; they can be seen as the ratio of the essential coefficients in the symmetric matrix in (4).

We now ask, to which special domains Γ of interest we may confine our considerations without losing the existing degrees of freedom. We will see that it is sufficient to work with rectangular domains, which simplifies the computation of the entries in (4).

For that we may introduce the *Poinsot central ellipse* $p(\Gamma)$ of a domain Γ. Let us assume that the coordinate system is chosen such that the origin is the barycenter of Γ and that the coordinate axes agree with the principal axes of Γ. The area of Γ and the quadratic moments M_x, M_y of Γ with respect to the axes, which also appear in (4), are

$$A = \int dxdy, \; M_x = \int y^2 dxdy, \; M_y = \int x^2 dxdy. \tag{6}$$

The Poinsot central ellipse $p(\Gamma)$ of Γ is only dependent on the principal axes and the moments M_x, M_y. With respect to the chosen coordinate system its equation is

$$p(\Gamma): \; M_x x^2 + M_y y^2 = 1. \tag{7}$$

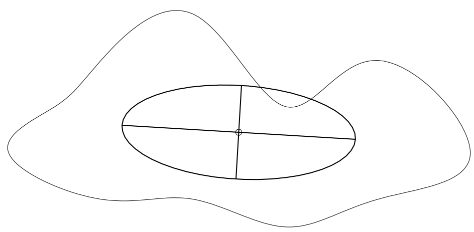


Figure 2: Poinsot central ellipse of a planar domain.

Theorem 2 *The Euclidean metric (4) in the space of planes to a given domain of interest Γ depends only on the area of Γ and on the Poinsot central ellipse of Γ. Up to an unimportant uniform scaling, all possible distance functions can be achieved by working with a rectangle Γ.*

Proof: The definition of the metric in the space of planes is purely geometric and thus it is independent of the choice of the coordinate system. We therefore associate a special Cartesian system with the domain Γ. Its origin is the barycenter of Γ, and its axes agree with the principal axes of the domain Γ. These are those lines L for which the moment of Γ with respect to L assumes an extremal value. Under these assumptions, we have

$$\int xdxdy = \int ydxdy = \int xydxdy = 0.$$

Now the matrix in (4) is a diagonal matrix. The three nontrivial entries are the area A of Γ, and the moments M_x, M_y with respect to the principal (i.e., coordinate) axes.

Thus $p(\Gamma)$ and the area completely determine the distance function (4). Geometrically, only the ratio of the three entries in the matrix is important. Picking a rectangle with the same barycenter and principal axes, i.e., $[-a, a] \times [-b, b]$, we obtain

$$A : M_y : M_x = 3 : a^2 : b^2.$$

Since the moments are positive in any case, we clearly obtain all ratios for $A : M_y : M_x$ by working with rectangles only. This completes the proof. □

To further understand the metric we investigate the *spheres of planes* in the original space. Naturally, a sphere of planes with central plane C is the set of all planes which possess a constant distance from C. This set of planes appears in the affine dual space A^* as ellipsoid. Since a duality is determined by a linear transformation in the vector spaces, the planes possessing constant distance from C will envelope a quadric in the original space E^3. More precisely, this quadric has the following properties (see Fig. 3).

Theorem 3 *All planes having constant distance to a given plane C, envelope a two-sheeted hyperboloid Ω. Intersecting Ω with planes parallel to C results in a family of ellipses. Their orthogonal projections onto the reference plane form a family of ellipses which are homothetic to the Poinsot central ellipse $p(\Gamma)$.*

Proof: Consider an affine map α, which maps the central plane $C: \ z = c_0 + c_1x + c_2y$ onto the reference plane $P : z = 0$,

$$\alpha : \ x' = x, \ y' = y, \ z' = z - c_0 - c_1x - c_2y.$$

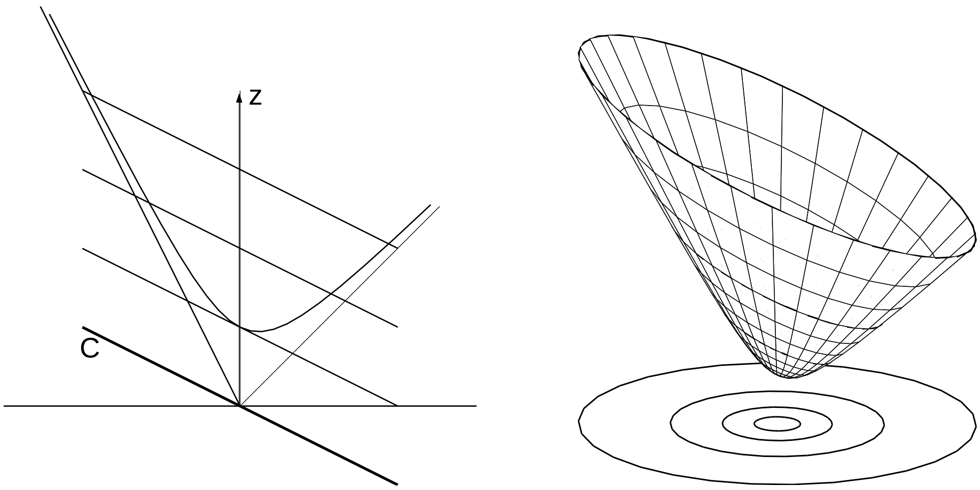


Figure 3: Left: Front view of a sphere of planes and sections with planes parallel to C. Right: Axonometric view and top view of the section curves.

Under this affine map, the distance between two planes is mapped to the distance between the image planes, since the differences of distances in z-direction remain the same. Thus, the set of planes with constant distance d to C is mapped under α to the set of planes with the same constant distance d to P. Using the notation from formula (6), the latter planes $z = u_0 + u_1 x + u_2 y$ satisfy the equation

$$Au_0^2 + M_y u_1^2 + M_x u_2^2 - d^2 = 0, \tag{8}$$

which represents a quadric Ω in (affine) plane coordinates. To obtain a representation of Ω in affine point coordinates x, y, z we have to form the inverse of the matrix of the quadratic form on the left hand side. This leads to

$$\frac{1}{A} + \frac{x^2}{M_y} + \frac{y^2}{M_x} - \frac{z^2}{d^2} = 0, \tag{9}$$

which is the equation of a two-sheeted hyperboloid. Intersecting Ω with a plane V parallel to P, i.e., a plane $z = c$, and projecting onto P yields

$$\frac{x^2}{M_y} + \frac{y^2}{M_x} = -\frac{1}{A} + \frac{c^2}{d^2} = C. \tag{10}$$

For any constant c, for which $C > 0$ holds, one obtains an ellipse which is just a scaled version of the Poinsot central ellipse (7). Clearly, planes V with $C = 0$ are tangent to the hyperboloid and for planes V with $C < 0$, the intersection is not real.

Applying α^{-1}, the projection onto P does not change, and thus the theorem is proved even in the general case. $\square$

2.3 A convenient coordinate transformation in A^*

Since the distance between two planes is derived from the positive definite quadratic form (11), A^* becomes a Euclidean space itself. But unfortunately the scalar product matrix in (4) usually differs from the canonical Euclidean scalar product matrix. Thus, the reference surface for measuring distances between two points in A^* is *not* the unit sphere with respect to the canonical Euclidean metric but an ellipsoid, defined by the scalar product matrix in (4).

In order to visualize the images of planes in A^* in a more convenient way one may apply a coordinate transformation to the image points (u_0, u_1, u_2) of planes $z = u_0 + u_1 x + u_2 y$ such that the scalar product matrix becomes the identity matrix. Although our eyes are no measurement devices, measuring with a reference sphere has visual advantages to measuring with a reference ellipsoid. This is especially apparent if we are using orthogonal projections.

But also for computational reasons like computation of a lot of distances between points in A^* or computation of Euclidean invariants and applying algorithms like curve fitting (see section 4.3) it is advantageous to apply a coordinate transformation in A^* such that the scalar product matrix becomes the unity matrix.

So, let D be the scalar product matrix with respect to formula (11). With help of eigenvectors and eigenvalues of D we define the mentioned transformation. Let S be the matrix containing the eigenvectors v_i of D (normalized with $v_i^T \cdot v_i = 1$) as column vectors and let s_1, s_2, s_3 be the corresponding eigenvalues. The matrix

$$T = \left[\frac{1}{\sqrt{s_1}} v_1, \frac{1}{\sqrt{s_2}} v_2, \frac{1}{\sqrt{s_3}} v_3\right]$$

has the property that $I = T^T \cdot D \cdot T$ and it is the inverse of the transformation matrix we are looking for. Since T possesses orthogonal columns the transformation matrix T', the inverse to T, is simply

$$T' = \begin{bmatrix} \sqrt{s_1} v_1^T \\ \sqrt{s_2} v_2^T \\ \sqrt{s_3} v_3^T \end{bmatrix},$$

where v_i^T are row vectors. It follows that the distances between points $T' \cdot x$ in A^* can be measured by the canonical Euclidean distance function (based on the identity matrix).

Remark: The eigenvectors v_i define the axes of the reference ellipsoid, the eigenvalues are the squares of the inverse lengths of these axes.

3 Detection and reconstruction of planar faces in point clouds

Assume that we are given a cloud of measurement points, which might be scanned data from an object. Our goal is the automatic construction of a CAD model of that object. This *reverse engineering* problem has a variety of remarkable applications [16].

In the following we are interested in the detection and reconstruction of planar faces in point clouds. A known solution to this problem uses the Gaussian sphere [15]. For each data point, one computes a plane of regression to the point and its nearest neighbors. The unit normal vectors of those planes describe points on the Gaussian unit sphere. Points from a planar face will give rise to nearly identical local regression planes and thus to a point cluster on the Gaussian sphere. The detection of planar faces is reduced to the detection of point clusters on the Gaussian sphere. An obvious disadvantage of this approach is that we loose information when going from the regression plane to the Gaussian sphere. Parallel planar faces cannot be separated directly on the Gaussian sphere. Moreover, the loss of information is critical in case of noisy data and complicated objects.

From the previous discussion, the readers will already observe, that mapping the planes to points in the dual space and using the Euclidean distance introduced there to detect point clusters will result in an algorithm which eliminates the problems of the Gaussian sphere method. We will now describe the algorithm in more detail.

3.1 An algorithm to detect planar faces

We are given a cloud of data points obtained by measuring an object, which is composed of planar faces, with a laser scanner. For simplicity we will restrict ourselves to those planar faces, which are visible from one (far) viewpoint. Additionally we assume that a coordinate system has been chosen such that the laser rays enclose small angles with the chosen z-axis. The angles between the faces and the z-axis should not be too small since faces of the object which enclose small angles with the laser rays are clearly not well represented by the data obtained by scanning from one viewpoint.

Let $p_i = (p_{i1}, p_{i2}, p_{i3})$, $i = 1, \ldots, N$, be unorganized points (measurements) of the object. According to the assumptions, all faces are contained in planes which are graphs $z = u_0 + u_1 x + u_2 y$ over the plane $P : z = 0$.

We give a brief outline of the algorithm of finding a single face of a piecewise planar object.

1. Compute local planes of regression (local planar fits) ϵ_i for all data points p_i.

2. Compute the distances $d_\Gamma(.,.)$ between each pair of local planes of regression ϵ_i, ϵ_j.

3. Determine the plane ϵ (point E in A^*) which possesses a maximal number of neighbors (planes ϵ_k) in a spherical neighborhood in A^*. This set of neighboring planes defines the maximal cluster of local regression planes.

4. Compute a plane of regression φ with respect to all data points p_k determined by the maximal cluster (defined by local planes of regression ϵ_k).

Each step of above outlined algorithm shall be described in more detail.

3.1.1 Computing local planes of regression

Step 1: The starting point in recognizing planar faces is to compute local planes of regression with respect to data points in a neighborhood of each data point p_i. The neighborhood U_i is chosen as symmetric domain in $P : z = 0$ in the way that it contains sufficiently many data points to compute a local regression plane. The size of U_i depends on the distribution of the data points and the application.

The computation of a regression plane is a well-studied subject. A least squares formulation which incorporates the orthogonal distances to the solution plane leads to an eigenvalue problem (see e.g. [15]).

Since the objects under consideration possess edges (intersection of faces), the data points q_j in a neighborhood U_i might belong to one, two, or even more faces. In such cases the least squares solution which minimizes the l_2-norm or the errors might be a bad estimate of the carrier plane of the face we are looking for. A convenient method to find heavy outliers is to compute the local regression plane as minimizer of the l_1-norm of the errors. After having classified the heavy outliers we can compute the local regression plane as minimizer of the l_2-norm, but only with respect to the 'good' data points in the neighborhood of p_i.

After having computed robust regression planes for all data points p_i, we obtain N planes of regression ϵ_i which approximate the faces containing data points p_i. In particular, all data points which are contained in a fixed face F should possess local planes of regression which are close to the plane describing the face F.

For practical reasons we apply a small adjustment to the definition of the distance $d_\Gamma(A,B)^2$ of two planes A and B in formula (3) by normalizing by the area of the domain of interest Γ,

$$d_\Gamma(A,B)^2 = \frac{1}{\mathrm{area}(\Gamma)} \int_\Gamma ((a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)y)^2 dxdy. \qquad (11)$$

Usually, the accuracy of the measurement device is known and one can estimate standard deviations of the data points p_i (their coordinates), in particular a standard deviation τ of the z-coordinate.

Practical tests have shown that the distance between two local planes of regression $\epsilon_{i1}, \epsilon_{i2}$ to data points p_{i1}, p_{i2} belonging to the same face F is usually lower than τ (except for heavy outliers and points at edges).

Step 2: We compute all distances $d_\Gamma(\epsilon_i, \epsilon_j)$ between all local planes of regression. Thus we have to allocate space for an array of size $N(N-1)/2$ and we assume that this is possible. How to proceed if this fails will be discussed below.

Remark concerning the choice of Γ: We are given a piecewise planar object O in space all whose faces are not parallel to z. Its projection onto the xy-plane shall be denoted by O'. The coordinate system in the xy-plane is chosen such that the origin is the barycenter of O' and the axes x and y agree with the principal axes of O'. We compute the quadratic moments M_x, M_y and the area A of O' with equation (6) and choose Γ to be a rectangular domain $[-a,a] \times [-b,b]$ with $A : M_y : M_x = 3 : a^2 : b^2$. This is in accordance with the proof of theorem 2. Dependent on the application, other choices of Γ might be more natural and more suitable.

3.1.2 Clustering in the space of planes

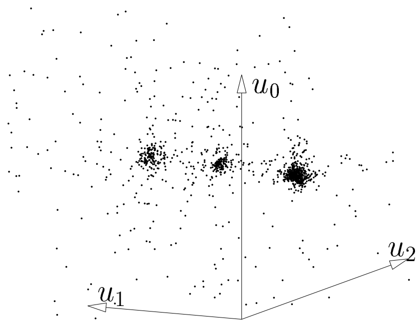
Step 3: We will determine the maximal cluster of image points E_k in A^* of the local regression planes ϵ_k. The maximal cluster shall be defined in the following way: We look for the point E possessing a maximal number of points E_k in a *spherical neighborhood* $S(\tau)$ of radius τ. All those points E_k form the *maximal cluster* in A^* and correspond to local planes of regression ϵ_k in Euclidean 3-space which are close to each other. Further, such a cluster of planes corresponds to points p_k which are contained in a planar face F of the object.

Remark: We assumed implicitly that each plane carries only a single face of the object. This is in general only true for convex objects. But if the object possesses planes which carry more than one face one has to classify the data points p_k found in step three according to the faces they are belonging to. This is again some sort of clustering since the faces belonging to a fixed plane have to be spatially separated.

Step 4: The plane φ containing the face F is computed as plane of regression with respect to the data points p_k, which are determined by the planes ϵ_k of the cluster. Having computed the plane φ containing the face F one eliminates all planes ϵ_k belonging to the maximal cluster from all further computations.

Remark: If the data set representing the object is very large, the allocation of an array of size $N(N-1)/2$ for storing distances is not possible. We want to give some hints on how to manage such a problem.

1. If all data points have to be taken into account we may partition the problem in A^* with help of an octree structure. Let the bounding box of the data E_k in A^* be the root of the tree. If the number of data points contained in its leaves is suitable we run the algorithm on the sub-boxes, otherwise we partition again.



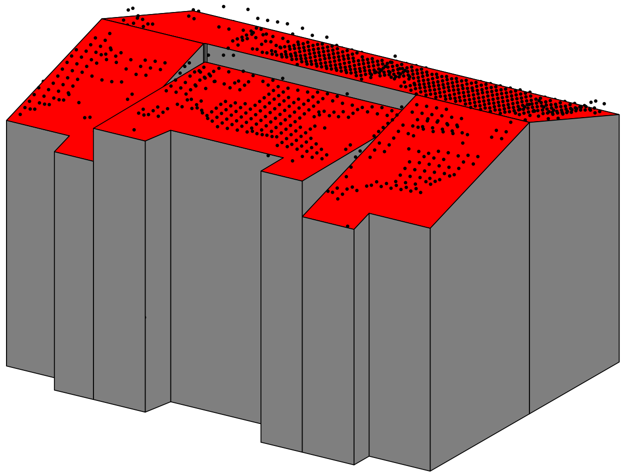


Figure 4: Left: Image points of planes of regression in A^*. Right: CAD model of a building.

2. One can apply a presegmentation on the Gaussian sphere. We define a covering of the Gaussian image (unit normals) of all local planes of regression by spherical caps. Then, above defined algorithm is applied to any of those subsets defined by the spherical caps (under the assumption that this method yields a significant data reduction). But since the caps in the Gaussian sphere have to overlap we have to merge the solutions (clusters) obtained with respect to the defined subsets.

3.1.3 Finding all faces of an object

Finally we discuss how one can find all faces belonging to an object. As in section 3.1 we assume that all faces are visible from one far viewpoint such that we can work with one space A^* for all planes of regression computed from the data points p_i.

Applying the above described algorithm iteratively until no or just a thin scattered set of local regression planes is left will serve finding all clusters and thus all faces of the object.

Two examples which shall illustrate the algorithm are displayed in figures 4 and 5.

Remark: If an object is scanned from different positions to avoid hidden or badly represented faces we would have to partition the problem so that each subset of data points we are working with describes points of faces which are visible from one far viewpoint. This is necessary in particular for the reconstruction of convex objects with planar faces. This partitioning is reached by the following procedure:

1. Compute local regression planes to all data points. The unit normal vectors of these local regression planes are points on the Gaussian sphere.

2. Compute a covering of the Gaussian sphere by regions U_j, for instance spherical caps. Determine the membership of unit normals vectors of the regression planes with respect to the regions U_j.

3. Define an affine space A_j^* (of local regression planes) for each region U_j and associate a unit vector z_j to each region.

   The direction z_j will be used as $z-$axis of a local coordinate system, such that all local regression planes whose unit normals belong to U_j are graphs of linear functions over the xy-plane P, i.e. $z = u_0 + u_1 x + u_2 y$.

4. Apply all necessary computations, for instance, *finding clusters of points*, in the space A_j^*.

5. Go back to initial Euclidean space and merge the results obtained by the local computations in the spaces A_j^*.

This procedure can be applied to different problems. The last step and step before last have to be defined according to the actual task. Especially, merging the partial solutions with respect to the spaces ${A^*}_j$ may be quite different and possibly difficult in various applications.

3.2 Reconstruction of buildings from airborne laser scanner data

The reconstruction of 3D urban models from airborne laser scanner data is an important research topic in geodesy and photogrammetry. We do not review the literature on this problem here, but just point to a very recent paper by Vosselman and Dijkman [17]. There, an extension of the Hough transform to 3 dimensions is used to recognize planar faces of the buildings'

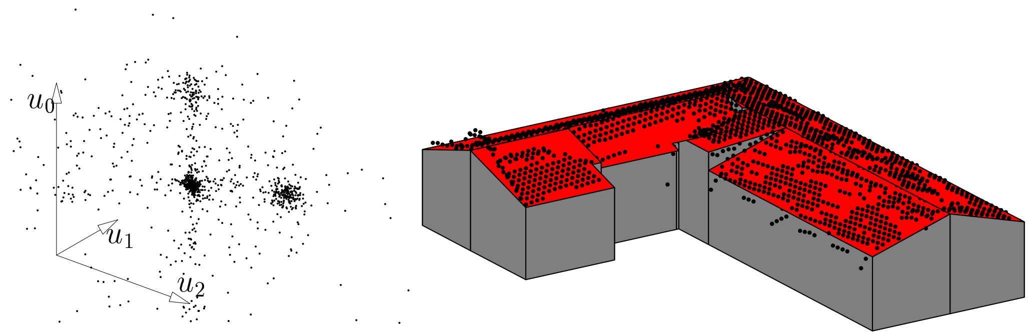


Figure 5: Left: Image points of planes of regression in A^*. Right: CAD model of a building.

roofs. This approach has some similarity to ours, since the Hough transform is also a special duality. However, in our method the *metric* in the space of planes plays a crucial role and thus it can be expected that the present approach is more reliable.

The examples displayed in figures 4 and 5 show in the right hand side CAD-models, which represent the simplified geometry of the buildings. The data are not original measurements but are resampled to a regular grid. Additionally, the ground plans of the buildings were used to generate these CAD-models. The left hand side shows in both cases the image points in A^* of the local planes of regression to all data points.

4 Approximation with developable surfaces

4.1 Some essentials on developable surfaces

Developable surfaces can be isometrically mapped (*developed*) into the plane, at least locally. When sufficient differentiability is assumed, they are characterized by vanishing Gaussian curvature. A non-flat developable surface is the envelope of its one parameter family of tangent planes. Such a developable surface locally is either a conical surface, a cylindrical surface, or the tangent surface of a twisted curve. Globally, of course, it can be a rather complicated composition of these three surface types. Thus, developable surfaces are ruled surfaces, but with the special property that they possess the same tangent plane at all points of the same generator (=*ruling*).

Because in all points of a generator line the tangent plane is the same, we can identify a developable surface with the one-parameter family of its tangent planes $U(t)$. Applying a duality in projective space, $U(t)$ is a curve in dual projective space. If this curve is a NURBS curve

$$U(t) = \sum_{i=0}^{n} U_i N_i^k(t), \tag{12}$$

where U_i are homogeneous coordinates of the control points and $N_i^k(t)$ are normalized B-splines over a certain knot vector, the original surface is a *developable NURBS surface*, since a duality is a linear (rational) mapping. Methods for computing a parameterization in standard NURBS tensor product form have been developed. For a detailed study of developable surfaces from the computational point of view we refer the interested reader to the textbook [13].

4.2 A brief summary of algorithms for developable surface approximation

Consider the following approximation problem. Given m planes $V_1, \ldots, V_m$ and corresponding parameter values t_i, approximate these planes by a developable surface $U(t)$, such that $U(t_i)$ is close to the given plane V_i. Here close is meant in an appropriately chosen Euclidean metric in the space of planes as outlined in section 2. The given planes are points in the resulting Euclidean space. Fitting a B-spline curve in A^* to these points, we obtain in the original space the tangent planes of a NURBS developable surface which solves the given problem [12]. Using a special class of developable NURBS surfaces with a particularly simple representation in the dual model, Pottmann and Wallner [12] also showed how to deal with given rulings of developable surfaces and how to approximate given points. Moreover, they dealt with the important issue of avoiding surface singularities within the domain of interest.

A class of developable surfaces useful for practical applications is the class of surfaces composed of smoothly joined cones of revolution. These *cone spline surfaces* are of degree 2, the development can be computed in an elementary way and the offsets are of the same type. Various interpolation and approximation algorithms for these surfaces have been developed

[2, 7, 8]. They work mainly with elementary geometric considerations instead of the dual representation. An important advantage of cone splines over arbitrary developable NURBS surfaces is the simple control over the singularities, which are the vertices of the cone segments.

4.3 A new algorithm based on curve-like point arrangements in the dual model

To approximate a nearly developable surface Φ with a developable surface we will use the dual model of the given surface.

If Φ is given in analytic form (parametrization or implicit representation) we start with a set of sample points p_i on Φ and compute the tangent planes U_i there.

If the surface Φ is given by measurement points p_i, estimates of the tangent planes U_i at p_i can be computed with help of neighboring data points of p_i.

At first we compute a reference direction and an appropriate domain Γ as explained in the previous sections. In this way, the tangent planes U_i are mapped to a point cloud in A^*. If the tangent planes came from a developable surface, the corresponding points u_i in A^* should be arranged along a curve. Because of the assumed deviation from a developable surface, this will no longer be the case. However, we can expect that the points u_i form a curve-like point cloud. Now, we may compute a best fit of this point cloud by a B-spline curve $u(t)$. Practically it is convenient to apply a coordinate transformation in A^*, such that the scalar product matrix becomes the unity matrix, as described in section 2.3. We estimate the parameter values for B-spline fitting with projections onto adjusting lines to data points in local neighborhoods of selected data points. There are also more advanced algorithms, see for instance [6]. The resulting B-spline curve $u(t)$ determines in the original space the set of tangent planes $U(t)$ whose envelope is a developable approximation surface of the given surface Φ.

This approach may be refined as follows. Small deviations of the given surface may result in large changes of the tangent planes. Thus, it is important to associate *weights* to the tangent planes U_i. A large weight w_i of U_i indicates that a large part of the surface Φ lies in an ε neighborhood of U_i.

The first test-example which is given in figure 6 shows a nearly developable surface Φ and the developable approximation in the right hand side.

The left hand side displays the image points u_i of local estimates of the tangent planes of Φ. The sampled data points p_i are obtained by the parametrization with respect to a regular grid in the uv plane. Φ is chosen to be the sum of a ruled surface of nearly constant slope s and a deviation function $(\epsilon_x, \epsilon_y, \epsilon_z)$,

$$p(u,v) = \begin{pmatrix} a\cos(u) \\ b\sin(u) \\ 0 \end{pmatrix} + v \begin{pmatrix} -b\cos(u) \\ -a\sin(u) \\ s\sqrt{a^2+b^2} \end{pmatrix} + \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{pmatrix}.$$

Φ is displayed as mesh and the developable surface $U(t)$ is shaded. The boundary curves of the developable $U(t)$ are lying in horizontal planes passing through the lowest and highest point of Φ. This is the reason why the boundary curves of Φ and $U(t)$ do not fit together. The left hand side of figure 6 also contains the approximating curve $u(t)$ to the curve–like region of scattered points u_i in A^*. The curve $u(t)$ is the dual image curve in A^* of the determined developable surface in the original space.

If the surface Φ is not fully visible from a single direction, it might be necessary to work with several local mappings to dual spaces A^*. The resulting partial solutions then have to be merged to a final approximating developable surface. The approximation by developable surfaces is current research work and some practically important details are not yet considered. They will be discussed in a further contribution.

Acknowledgement

This work has been partially carried out within the K plus Competence Center ADVANCED COMPUTER VISION. This work was funded from the K plus Program. In addition we would like to thank the unknown referees for their advice and corrections.

References

[1] Bodduluri, R. M. C., Ravani, B.: Design of developable surfaces using duality between plane and point geometries. Comp. Aided Design **25** (1993), 621–632.

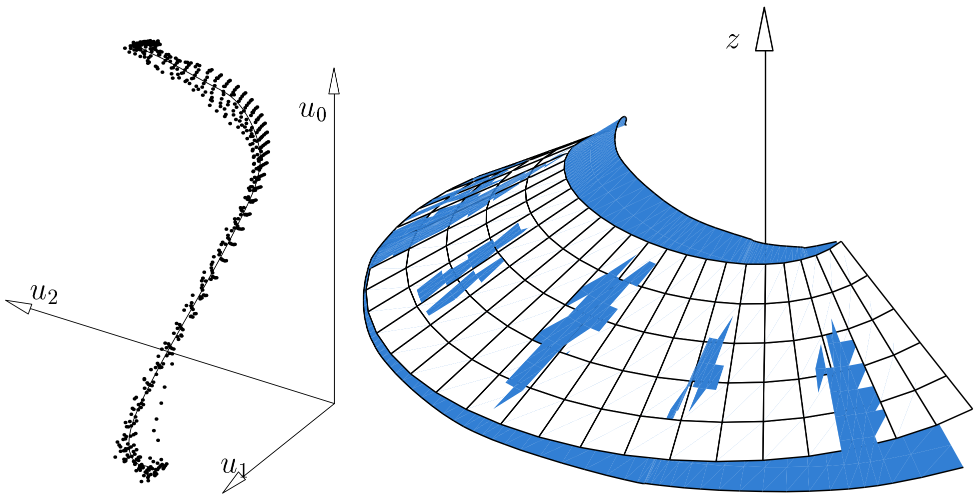


Figure 6: Left: Image points of planes of regression and approximating curve. Right: Nearly developable surface (mesh) and developable approximation (shaded).

[2] Chen, H.-Y., Lee, I.-K., Leopoldseder, S., Pottmann, H., Randrup, T., Wallner, J.: On surface approximation using developable surfaces. Graphical Models and Image Processing **61** (1999), 110–124.

[3] do Carmo, M. P.: *Differential Geometry of Curves and Surfaces*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[4] Hoschek, J., Pottmann, H.: Interpolation and approximation with developable B-spline surfaces. In: Mathematical Methods for Curves and Surfaces (M. Daehlen, T. Lyche, L.L. Schumaker, eds). Vanderbilt Univ. Press, 1995, pp. 255–264.

[5] Hoschek, J., Schneider, M.: Interpolation and approximation with developable surfaces. In: Curves and Surfaces with Applications in CAGD (A. Le Méhauté, C. Rabut, L. L. Schumaker, eds). Vanderbilt Univ. Press, 1997, pp. 185–202.

[6] Lee, I.-K.: Curve reconstruction from unorganized points, Comp. Aided Geom. Design **17** (2000), 161–177.

[7] Leopoldseder, S.: Algorithms on cone spline surfaces and spatial osculating arc splines, Comp. Aided Geom. Design **18** (2001), 505–530.

[8] Leopoldseder, S., Pottmann, H.: Approximation of developable surfaces with cone spline surfaces. Comp. Aided Design **30** (1998), 571–582.

[9] Piegl, L., Tiller, W.: *The NURBS Book*, Springer, Berlin, 1995.

[10] Peternell, M., Pottmann, H.: Interpolating functions on lines in 3-space In: Curve and Surface Fitting: Saint Malo 1999 (P.J. Laurent, P. Sablonnière and L.L. Schumaker, eds.), Vanderbilt University Press, Nashville, TN, 2000, pp. 351–358.

[11] Pottmann, H., Peternell, M.: On approximation in spaces of geometric objects, In: The Mathematics of Surfaces IX (R. Cipolla, R. Martin, eds), Springer, 2000, 438–458.

[12] Pottmann, H., Wallner, J.: Approximation algorithms for developable surfaces. Comp. Aided Geom. Design **16** (1999), 539–556.

[13] Pottmann, H., Wallner, J.: *Computational Line Geometry*, Springer, Berlin - Heidelberg - New York, 2001.

[14] Rousseeuw, P.J., Leroy, A.M.: *Robust Regression and Outlier Detection*, Wiley&Sons, New York, 1987.

[15] Várady, T., Benkö, P., Kós, G.: Reverse engineering regular objects: simple segmentation and surface fitting procedures, Int. J. Shape Modeling **4** (1998), 127–141.

[16] Várady, T., Martin, R. R., Cox, J.: Reverse engineering of geometric models — an introduction. Comp. Aided Design **29** (1997) 255–268.

[17] Vosselman, G., Dijkman, S.: 3D building model reconstruction from point clouds and ground plans, Intl. Archives of Photogrammetry and Remote Sensing, Vol. XXXIV-3/W4, Annapolis, MD, 2001, pp. 37–43.