Euclidean Invariant Descriptions for Matching of Curves Extracted from Range Images

Tomáš Pajdla, Luc Van Gool
Katholieke Universiteit Leuven
ESAT, Kardinaal Mercierlaan 94, B-3001 Leuven
Technical Report Nr. KUL/ESAT/MI2/9417

April 20, 1995

Abstract

A method for surface registration via the matching of feature curves is presented.

First, various differential and semi-differential invariants of 3-D curves on surfaces under Euclidean motions are introduced. New semi-differential invariants requiring only first derivatives and one reference point are presented for both 3-D curves and 3-D curves on surfaces. By the use of semi-differential descriptions, the computation of high order derivatives is avoided. We demonstrate that curve registration can be successfully performed even on data from a simple and cheap 3-D sensor. Since high order continuity is not necessary there is no need to perform computationally intensive surface and curve reconstruction. Simple "first-order" techniques apply instead. The solution to reference point selection through "sliding pairs" is proposed. It demonstrates that simple semi-differential invariants can successfully replace curvature and torsion.

Second, a new similarity measure (PSC) for partially overlapping curves based on ϵ-reciprocity is defined. It is shown that combining ϵ-reciprocity with the robust Least Median of Squares motion estimation allows to use Besl's Iterative Closest Point Algorithm to register partially overlapping curves.

Finally, the experiment involving 3-D data acquisition, visible surface reconstruction, C^2 continuous curve reconstruction, curve matching via geometrical hashing, curve similarity evaluation, and robust Euclidean motion estimation is presented.

1 Introduction

Model based 3-D vision relies on the existence of a database containing explicit models. Geometric models, among them, are undoubtedly needed to recover object orientation from a single view [Low87], to perform recognition by alignment [Hut93], and to link vision based shape perception with CAD shape manipulation [HH89]. This work addresses surface matching as a step towards automatic CAD model reconstruction from a sequence of unregistered range images.

A large number of depth perception algorithms have recently appeared as the outcome of the intensive research carried out in the area of active and passive range sensing [Jar93][MA93] [Rio84][VO90]. Laser based active range finders provide highly accurate data but often need a controlled environment and, generally, the scanning process takes time. On the other hand, systems utilizing structured light provide 3-D data faster but of lower quality and resolution.

With no exception, all 3-D sensing systems provide viewer oriented data i.e. only a small portion of a scene is measured at once. Moreover, the sampling rate is in general unevenly distributed over a region of measurement.

The complexity of geometric modeling from range data is severely influenced by the a priory knowledge used. When nothing is known about viewer motions between particular measurements, the task involves the following steps:

1. 3-D surface measurement.
2. Visual surface reconstruction.
3. Surface matching.
4. Fine surface registration.
5. Fusion of several views into a consistent representation.

We concentrate on matching of moderate quality 3-D surfaces with the aid of curves detected on the surfaces. We assume Euclidean reconstructions and therefore the transformations between neighboring views are pure Euclidean motions. Classical approaches use differential invariants of curves, curvature and torsion, [KHW91][GA94] or differential invariants of surfaces [LT90]. Our method utilizes a semi-differential invariant description of curves. Basically, one difficult task, the stable computation of high derivatives is exchanged for another one, the detection of a stable reference point in unregistered views. Nevertheless, semi-differential descriptions are particularly useful when:

1. the surface either *has not* enough complex differential structure (e.g. cylinder, sphere) or the measurements are of low accuracy avoiding thus robust computation of 2nd and higher order differential characteristics,
2. the surface *has* other features e.g. curves appearing in the range images (C^1 discontinuities) or in gray level images such as surface markings.

In the second case we assume to have a registered gray level image of an underlying 3-D surface. The whole class of range finders based on structured light delivers such aligned 3-D and 2-D images of a surface. Laser based scanners having this capability have already appeared. Cyberware 3030 PS Scanner provides both 3-D surface measurements and its 24 bit color image.

The rest of the paper is organized as follows. In the next section we give a short overview of approaches related to surface matching. In section 3 various differential and semi-differential invariants of curves are introduced. Finally, in section 4 we present an experiment with matching two objects in a real environment. Here we define a new similarity measure (PSC) for partially overlapping curves based on ϵ-reciprocity and propose a robust Least Median of Squares motion estimation algorithm.

2 Related work

For an overview of surface registration methods before 1989 see [FMN89]. Most of the systems utilized features like angles, points, normals and their combinations to describe curves and surfaces. Another excellent survey can be found in [Bes90].

In this section we give a survey of some recent contributions closely related either to the model reconstruction from unregistered range images or to the surface and/or 3-D curve matching based on geometrical hashing.

Fan, Medioni, and Nevatia [FMN89] use a surface patch approach to 3-D surface matching. Their objective is to recognize CAD object models in range images. The key idea of their algorithms is a surface segmentation followed by an attributed graph description capturing adjacency of regions. The machinery of graph matching is then used to generate candidates for registration. Nodes in the graph description are in one to one correspondence with surface patches. Every node carries nodal attributes like area, orientation, and average of principal curvature over the whole patch. The edges of the graph are attributed with the type of adjacency like jump edge, concave or convex crease edge etc. Basically, both the first and second order differential description of a surface is used.

Bergevin, Larendeau, and Poussart [BLP92] use an exhaustive search space method to register complex surfaces with partial overlap. They randomly generate a large number of initial candidate transformations. The correct transformation is selected in the verification process where pairs of adjacent triangles from different views are matched to check the validity of the transformation. The approach relies on the heuristics that corresponding regions in an overlapping part of the range image are covered with similarly oriented triangles having similar areas. (Delaunay triangulations on different resolutions form surface description.)

Stein and Medioni [SM93] have developed the TOSS system for the recognition of multiple three dimensional object models from range data. The idea is to use characteristic curves as well as surface patches. In their case curves on a surface correspond to depth and orientation discontinuities. A polygonal approximation is used to represent curves and a discrete version of curvature and torsion gives an invariant description. It is computed on different levels of resolution forming a curve description unit called the *super segment*. The surface is described using features called the *super splash*. The splash at a point P with radius ρ is a collection of sampled normals on geodesic circle of radius ρ. The splash constitutes periodic curve over the circle and is encoded using super segment representation. The collection of splashes computed for different radii ρ constitute the super splash at the point P. Super splashes and super segments are used to index a database of precomputed models in order to generate matching candidates. Correct matching is elected via checking rigidity of the transformation between objects.

Higuichi, Delingette, Hebert, and Ikeuchi [DHI92][DHI93] [Del94][HHI93] propose the approach based on a mesh representation of a surface. A mesh of points is fit to an input data set from each particular view, and a discrete curvature measure - *simplex angle* - is computed at every node of the mesh. Then, a canonical mapping between the mesh and a standard spherical mesh is established forming, altogether with the curvature index, the Spherical Attribute Image (SAI). The transformation between views is computed by comparing their SAIs over all possible rotations of their supporting spheres.

Kishon, Hastie, and Wolfson [KHW91] presented an algorithm to find the longest common subcurve of two 3-D curves. They combine edges extracted in a gray level image with an underlying dense range image to obtain 3-D curves. The curves are then smoothed by splines fitted to the curve data points. Shape signatures based on a multiresolution representation of curvature and torsion are used as invariant curve descriptions. They are matched using a geometrical hashing technique which produces a number of matching candidates. The matching candidates are later verified by a robust, least squares, 3-D curve matching technique [SS87].

Guéziec and Ayache [GA94] have extended the curve matching approach presented by Kishon et al. using an iterative B-spline fitting method to obtain stable curvature and torsion measurements. Optimal transformations between different views are obtained using a Kalman filter based estimation over a number of transformation hypotheses resulting from matched Frenet frames attached to curves. The authors present real experiments with crest lines extracted from CT-scan isosurfaces [Gue93]. They also compared the performance of differential descriptions (curvature and torsion) and mixed 2nd order and 1st order semi-differential descriptions (curvature, length of the vector between a reference and a curve point, and angles between a reference direction and a curve tangent vector). The mixed set of invariants is reported to be more stable than the set of differential invariants alone.

What is the most related work and how our work differs. Our work has been inspired mainly by works [KHW91] and [GA94]. Guéziec and Ayache [GA94] aim at solving 3-D curve matching through differential invariants. They develop elaborate technique to obtain stable 3rd derivatives but finally combine curvature with a semi-differential invariant. We explicitly introduce the framework of semi-differential invariance avoiding the combination of difficult problems of both approaches - stable curvature computation and reference point selection. It allows us to avoid the computation of the 2nd derivative. We also replace Kalman filter based Euclidean motion estimation by a more robust method.

3 Invariant descriptions of space curves

This section describes a number of invariants that can be used for the recognition of space curves (i.e. non-planar curves). It is assumed that the three-dimensional coordinates of the curve points are available, e.g. through the use of an active range scanner. The invariant description of the curve should be such that it is the same, irrespective of the viewpoint under which it is observed. This implies that recognition ought to work under arbitrary relative rotations and translations of the curve and the acquisition system. Therefore, invariants under the group $E(3)$ of three-dimensional (3-D) Euclidean motions are called for. As discussed elsewhere [VGMEO94], the use of coordinate derivatives introduces the additional issue of invariance under reparameterization.

$\mathbf{X}$ denotes the coordinates $(X, Y, Z)^T$ of an arbitrary point on the curve. $\mathbf{X}^{(i)}$ represents the i-th coordinate derivatives with respect to the curve parameter t: $\left(\frac{d^iX}{dt^i}, \frac{d^iY}{dt^i}, \frac{d^iZ}{dt^i}\right)^T$. Finally, an expression like $\mathbf{X}_j$ represents some fixed reference point and the subscript j is used as an identification tag.

We will discuss differential and semi-differential invariants focusing on descriptions based on first order derivatives. Since 3-D curves are also assumed to lie on a surface the description is extended behind the scope of curves incorporating surface structure as well.

3.1 Basic definitions

By a curve we understand a piecewise regular curve, i.e. regular everywhere but in a finite number of points, usually denoted as a "broken curve" [doC76, page 244]. It means that cusps, loops, and selfintersections are allowed though, from a practical point of view, self intersecting curve can always be treated as two separate curves touching in a "special" point.

By surface we understand a piecewise regular orientable surface [doC76, page 52], where piecewise means that the surface is partitioned into a set of regular domains [doC76, page 97] "glued" together by their boundaries called *discontinuity curves.* C^0 discontinuity curves are called *jump* edges, C^1 discontinuity curves are called *crease* or *ridge* edges, and C^2 discontinuity curves will be called *curvature edges.* Notice that isolated point discontinuities (e.g. the apex of a cone) or connected regions of point discontinuities are excluded. Moreover, discontinuity curves are piecewise regular as follows from the definition of a regular domain.

3.2 Differential Invariants

3.2.1 Curvature - k, torsion - τ

Curvature and torsion as a function of arc length uniquely determine a regular curve up to an Euclidean motion [doC76, page 19]. This fact has been previously used to match spatial curves in the work of Kishon, Hastie, and Wolfson [KHW91] followed by the work of Guéziec and Ayache [GA94]. Let $\mathbf{X}(t) : I \rightarrow R^3$ be a regular curve parameterized by t. Then, curvature resp. torsion are explicitly given by:

$$k(t) = \frac{\|\mathbf{X}^{(1)} \wedge \mathbf{X}^{(2)}\|}{\|\mathbf{X}^{(1)}\|^3}, \quad \text{resp.} \quad \tau(t) = -\frac{(\mathbf{X}^{(1)} \wedge \mathbf{X}^{(2)}) \cdot \mathbf{X}^{(3)}}{\|\mathbf{X}^{(1)} \wedge \mathbf{X}^{(2)}\|^2} \ .$$

where the dot stands for the inner and $\wedge$ for the outer product. We can see that in order to compute curvature resp. torsion at a point second resp. third derivatives of the point coordinates with respect to the parameter are required. The robust estimation of high order derivatives is questionable in noisy or quantized data [HPS94]. It has been also mentioned by Guziec and Ayache that particularly torsion can be highly unstable. A combination of curvature and first order semi-differential invariants has been proposed to avoid computation of a third derivative.

3.2.2 Normal curvature - k_n, geodesic curvature - k_g

When a curve is contained on a surface more can be done. Two basic situations can be considered. First, the whole curve is identified with some discontinuity curve and second, the whole curve is contained on a regular domain.

If a curve is on a regular domain, local surface structure can help to define invariant signatures of the curve. Imagine the task to match two instances of the same curve in the same surface differing by an Euclidean motion. If the surface were a plane, simple signed curvature together with arc length would be enough to completely characterize curves. When the surface is not planar the more general concepts of geodesic curvature and normal curvature have to be applied [doC76, section 4-4.], but similar results are obtained. If the ordinary curvature is replaced by geodesic curvature and if normal curvature is added, a new set of invariants is established. The normal curvature k_n and the geodesic curvature k_g of a regular curve $\mathbf{X}(s)$, parameterized by the arc length s and that is contained on a regular surface having unit normal $\mathbf{N}(u(s), v(s))$, are given by:

$$k_n(s) = \mathbf{N} \cdot \mathbf{X}^{(2)} \ , \quad \text{resp.} \quad k_g(s) = \mathbf{X}^{(2)} \cdot (\mathbf{N} \wedge \mathbf{X}^{(1)}) \ .$$

By knowing the curve tangent, normal, and the surface normal at every point, both the geodesic and normal curvature are readily computed and therefore only second derivatives of a curve and first derivatives of a surface are needed. While curvature and torsion are invariant to a change of curve orientation, normal curvature changes its sign when surface orientation is reversed. Geodesic curvature, then, changes its sign with a change of both surface and curve orientations. Orientation of the surface does not pose any problem since it can be always set so that normal vectors point to a camera. On the other hand, orientation of a curve can not be generally fixed.

3.3 Semi-differential Invariants

In this section semi-differential invariants for different choices of measurements are derived. Unlike differential invariants, semi-differential ones combine measurements from several points to surpass the dimension of group orbit [VGMEO94]. The group $E(3)$ of 3-D Euclidean motions has 6 independent parameters (3 for rotation and 3 for translation). This low number already suggests that simple invariants exist to replace curvature and torsion. In order to safely describe and recognize space curves, three independent invariants ought to be combined. For Euclidean group two points and first derivatives there sufficess to find three independent invariants.

3.3.1 $\mathbf{X}$, $\mathbf{X}^{(1)}$, $\mathbf{X}_1$

In this case, we choose one reference point ($\mathbf{X}_1$), and consider points $\mathbf{X}$ along the curve, together with their first derivatives $\mathbf{X}^{(1)}$. According to the counting argument [VGMEO94], 9 measurements are used and 7 (6+1) parameters have to be surpassed. Hence, one would arrive at two independent invariants. In order to safely describe and recognize space curves, three independent invariants ought to be combined. Therefore, we can expect the resulting invariant set to fall short of one invariant feature. Since only one invariant is missing, one might consider the use of relative invariants instead. In that case, three independent relative invariants can be expected to exist.

Indeed, the following set of invariants is obtained:

$$\begin{aligned} i_1 &= \|\mathbf{X} - \mathbf{X}_1\| \ , \\ i_2 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{X}^{(1)}}{\|\mathbf{X}^{(1)}\|} \ , \\ i_3 &= \|\mathbf{X}^{(1)}\| \ . \end{aligned}$$

The double bars $\|$ represent the Euclidean norm and the dot represents the Euclidean inner product. The first and second expressions are absolute invariants, but the third is merely relative invariant, as expected. Note, however, that the factor that determines the change only depends on the reparameterization parameter at $\mathbf{X}$. As explained elsewhere [VGMEO94], this factor can be eliminated through integration. This method is applied to the third expression. The set of (absolute) invariants then becomes

$$
\begin{aligned}
i_1 &= \|\mathbf{X} - \mathbf{X}_1\| \ , \\
i_2 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{X}^{(1)}}{\|\mathbf{X}^{(1)}\|} \ , \\
i_3 &= \textstyle\int_{t_0}^{t} \|\mathbf{X}^{(1)}\| \ dt \ ,
\end{aligned}
$$

where the integration boundaries t_0 and t are supposed to correspond to the parameter values for the starting point of the integration and for $\mathbf{X}$, resp. These invariants have simple interpretations. The first is the distance between the variable curve point $\mathbf{X}$ and the reference point $\mathbf{X}_1$. The second is this distance times the cosine of the angle between the vectors $(\mathbf{X} - \mathbf{X}_1)$ and $\mathbf{X}^{(1)}$, the tangent to the curve at $\mathbf{X}$. The reader will recognize the third expression as Euclidean arc length.

Note that this set of expressions only represents an invariant set as long as the starting point of the integration corresponds between views. Thus, the starting point of the integration has silently been introduced as an additional reference point. However, if the reference point $\mathbf{X}_1$ can be made to lie on the curve, then it can play this role as well and the above set of expressions yields a set of absolute invariants requiring only coordinates and their first derivatives along the curve plus the coordinates of a single reference point.

3.3.2 $\mathbf{X}$, $\mathbf{X}^{(1)}$, $\mathbf{X}_1$, $\mathbf{X}_1^{(1)}$

$\mathbf{X}$ is an arbitrary point of the curve to be described, whereas $\mathbf{X}_1$, is a fixed reference point that doesn't have to reside on the curve yet has to be in a rigid position with respect to it. In this case 12 (4×3) measurements are introduced, and 8 $(6+2)$ independent parameters have to be dealt with. The counting argument therefore predicts that finding three independent absolute invariants is no problem. Indeed, the following set of absolute invariants can be used:

$$
\begin{aligned}
i_1 &= \|\mathbf{X} - \mathbf{X}_1\| \ , \\
i_2 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{X}^{(1)}}{\|\mathbf{X}^{(1)}\| \ \|\mathbf{X} - \mathbf{X}_1\|} \ , \\
i_3 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{X}_1^{(1)}}{\|\mathbf{X}_1^{(1)}\| \ \|\mathbf{X} - \mathbf{X}_1\|} \ .
\end{aligned}
\tag{1}
$$

The problem of determining the starting point of integration doesn't pose itself here. Yet, it may still be interesting to replace the second or third invariant by Euclidean arc length, in order to express the other two as a *function* of a parameter, thereby yielding a better support for matching.

Note that $\mathbf{X}_1$ should be found on a geometric object such as a curve to allow the extraction of the tangent vector $\mathbf{X}_1^{(1)}$.

3.3.3 $\mathbf{X}$, $\mathbf{X}_1$, $\mathbf{X}_2$

This combination offers 9 (3×3) measurements in order to beat a problem governed by 6 independent parameters. The counting argument predicts that precisely 3 independent absolute invariants can be found. Such a set is given by

$$\begin{aligned} i_1 &= \|\mathbf{X} - \mathbf{X}_1\| \ , \\ i_2 &= \|\mathbf{X} - \mathbf{X}_2\| \ , \\ i_3 &= \|\mathbf{X}_1 - \mathbf{X}_2\| \ . \end{aligned}$$

The third invariant is trivial from a practical point of view, however. It doesn't change with $\mathbf{X}$. Hence, if only point coordinates are to be used, a third reference point is called for.

3.3.4 $\mathbf{X}$, $\mathbf{X}^{(1)}$, $\mathbf{X}_1$, $\mathbf{N}$

In this case we have 12 measurements with 10 degrees of freedom because two degrees of freedom are lost due to the unity of the surface normal $\mathbf{N}$ and the perpendicularity of $\mathbf{X}^{(1)}$ and $\mathbf{N}$. The counting argument predicts $10 - 7 = 3$ independent invariants. Indeed, the following set of invariants can be found:

$$\begin{aligned} i_1 &= \|\mathbf{X} - \mathbf{X}_1\| \ , \\ i_2 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{X}^{(1)}}{\|\mathbf{X}^{(1)}\| \ \|\mathbf{X} - \mathbf{X}_1\|} \ , \\ i_3 &= \frac{(\mathbf{X} - \mathbf{X}_1) \ \cdot \ \mathbf{N}}{\|\mathbf{X} - \mathbf{X}_1\|} \ . \end{aligned}$$

Notice that no reference direction is needed. This can be advantageous because an erroneous reference direction biases signatures of all points on a curve. This set of invariants can be applied only to curves contained on a C^1 continuous domain where the surface normal is well defined.

3.3.5 Finding reference points

Finding good reference points from single views is a hard problem. The best examples that come to mind are singularities like sharp changes in orientation or points where curves drawn on surfaces cross. The former type is only safe because the curves considered are markings on an opaque surface. Such singularities therefore cannot be generated by non-generic viewpoints.

Different reference point selection strategies can be considered for which we introduce some terminology. The most trivial *One-Reference-point* strategy selects one reference point and one reference direction common to all curves. The *N-reference-points* selection strategy uses one reference point on every curve. The reference directions are identified with tangent vectors at corresponding points. More expensive but practically important is the *All-reference-points* strategy which uses all points of a curve as reference points, Figure 1. Finally, the method that avoids difficult search for reference points the *No-reference-point* selection strategy, which builds on the invariance of arc length under Euclidean motions. Every point on a curve has at least one "neighbor" at some fixed arc length distance along that curve. Thus, for a fixed arc length distance, the set of "sliding" pairs of points $(\mathbf{X}_1, \mathbf{X}_2)$ is easy to generate, Figure 2. Note, that the third and fourth strategies are the only ones which make sense in the case of Euclidean motions. The other two approaches use already matched points what reduces

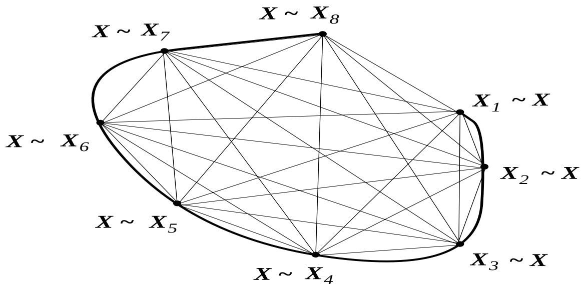


Figure 1: All-reference-points selection strategy. All points act as points as well as reference points resulting in $O(n^2)$ description.

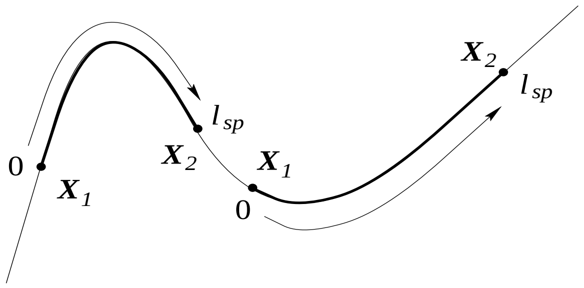


Figure 2: Constant Arc-length Sliding Pairs. The point and its reference point are fixed by a constant arc-length distance.

the motion estimation to a very simple task. We present results mainly to investigate the stability of our description.

Of course, once a reference point has been found, other points usually are easy to extract as well. It suffices to look at the zero-crossings and extrema of the invariants that are produced in this section.

4 Experiment

In this section we present an experiment. The goal is to bring two cylindrical soft drink cans into registration. To simplify the data manipulation we have taken only one frame with two cans in different positions instead of two different frames of the same can related by a motion.

The sets of invariants introduced above give us a rich collection to select from. Nevertheless, not all invariants are equally usefull. In order to demonstrate the use of semi-differential descriptions and considering the following requirements the set of invariants defined by the equation (1) has been chosen:

- **Locality**. In order to cope with occlusion signatures have to be local.
- **Robustness**. High order derivatives are excluded because our 3-D scanner does not provide sufficiently precise nor dense data for their calculation.
- **Uniqueness**. No two different curves, i.e. related by a non-Euclidean transformation, should have the same description.
- **One reference point**. Combining two points reduces hard general matching to a simple matching through a oneparameter group of rotations.

4.1 Visible surface reconstruction

The one-shot range finder provides a set of 3-D point samples of a surface. The visible surfaces have to be reconstructed in order to find curves on them. Usually, at least C^1-continuous surface reconstructions are generated [Ter88][MS92] [HDD+92][STT93][SD92] [SS92][Fra82] [WW92][FRC92]. In the reported experiment only C^0 continuity is imposed to the surface and C^1 continuity is imposed in a later stage on curves extracted from the surface. We can

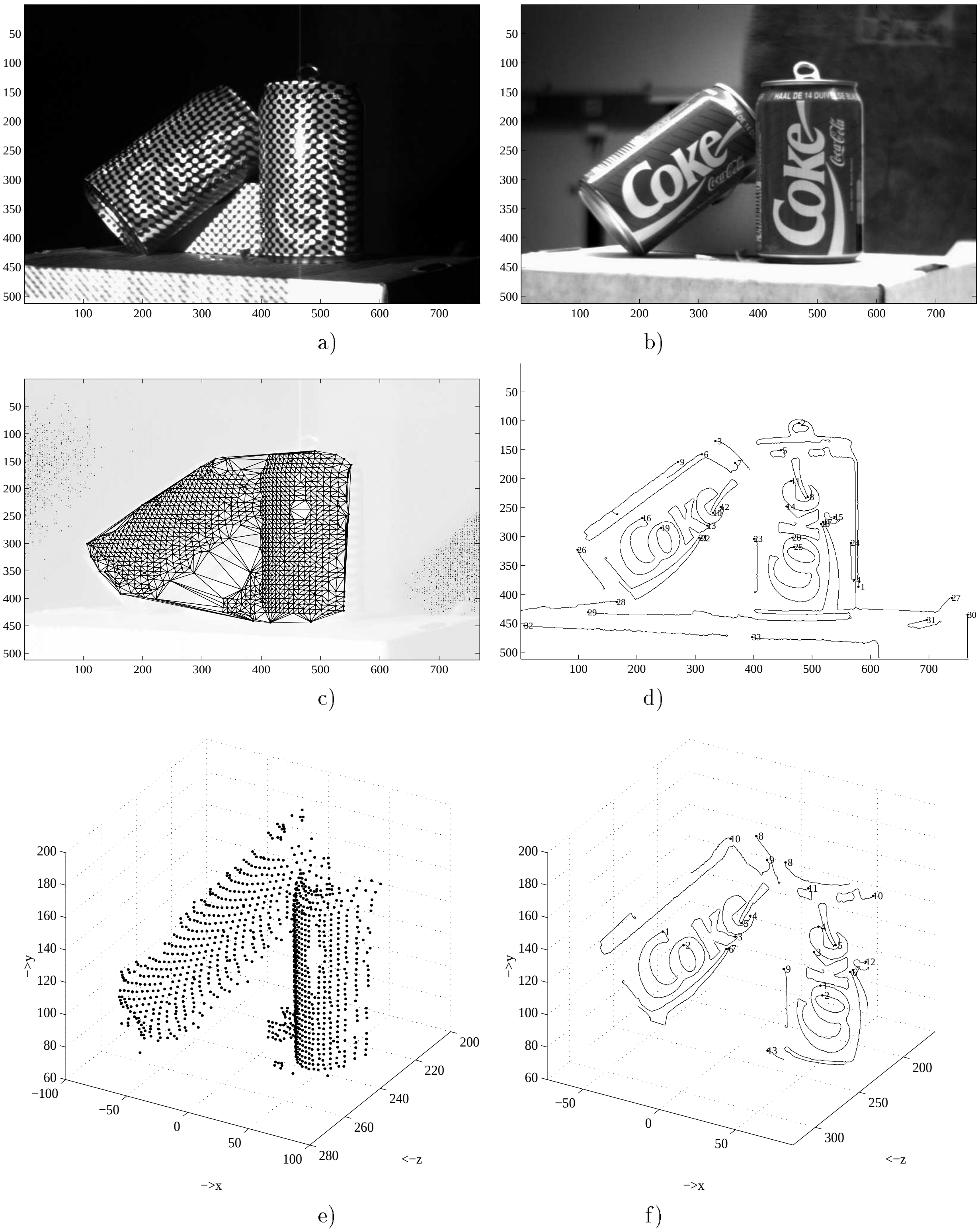


Figure 3: 3-D curves on the surfaces of CocaCola cans. a) Illumination-coded image, b) Gray level image, c) Delaunay triangulation, d) Edges from the gray level image, e) 3-D points measured on the surfaces, f) 3-D curves extracted from the surfaces.

skip the computationally intensive process of C^1-continuous surface reconstruction since the curves are primarily detected as gray level edges in a high resolution image.

A set of 3-D point samples of cans were measured using the one-shot Illumination Coded Range Finder developed previously at ESAT and based on a single structured light pattern [Paj94][VO90] and a few outliers falling far away from the "reasonable" domain of measurement were removed by simple plane clipping, see Fig. 3a) and Fig. 3e).

To reconstruct a free-form surface from a set of 3-D points a boundary of the points in 3-D space has to be found [Vel93] [Boi84]. It can be defined as *a simple closed polyhedron of triangular faces through all data points of a given set* [Vel93, page C-118]. Generally, boundary computation is not a trivial task. See [Vel93] for 3-D point set boundary computation based on γ-Graphs, [CSYL88] for 3-D point set triangulation using a smoothness criterion, and [Boi84] for Delaunay triangulation based boundary computation.

In our case a 2-D Delaunay (sometimes called Dirichlet or Thiessen) triangulation [PS85] [DFP85] can be used since the 3-D point set is in one to one correspondence with its projection to the image. In such a case the boundary of 3-D points can be defined in terms of a 2-D triangulation of their images. Delaunay triangulation can be a good choice because it provides the most equiangular triangulation in the plane and the "least rough" triangulation of the surface [Rip90]. Besides that, it can be computed with a $O(NlogN)$ algorithm. The least rough means that among all possible triangulations in the plane the integral $\int_S (\bigtriangledown z(x,y))^2 ds$ is minimal, where the gradient is computed on a linear interpolation over triangles expressed as a function over the domain of triangulation [1]. Of course, other planar triangulations might be used as well. The result of triangulation is shown in Fig. 3c). Once some triangulation is obtained a C^0 surface can be constructed by simple linear interpolation over the triangles.

4.2 3-D curve reconstruction

The reconstruction of curves from point samples, similarly to surface reconstruction, is a difficult task [MS93] [CG91][DHI91][CW94] [GA94]. As simple a scheme as possible is applied. Since the description requires first order derivatives, at least C^1 continuity of a curve is needed. It has already been mentioned in [MS93] that it is useful to minimize the $(n+1)$-st derivative in order to achieve stable estimates of n-th derivative. Hence, we use 3rd order B-spline approximations [Far90] that possess C^2 continuity. To arrive at a coordinate free parameterization, a polygonal approximation is generated first, the arc length of which is then used to parameterize the smoothing B-spline. If the polygonal approximation is fine enough, the fitted B-spline should approximately minimize

$$\int k^2 + \lambda \sum {d_i}^2 \,, \tag{2}$$

where k is curvature, d_i is Euclidean distance between data points and spline vertices, and λ denotes a smoothing parameter.

A gray level image of the scene, Fig. 3b), was captured with the same camera orientation as used for the acquisition of the 3-D data. We have switched off the projector and opened the iris to get a brighter image. Edges, Fig. 3d, were obtained by the following sequence of operations:

1. Gaussian smoothing ($\sigma = 1.5$),

[1] We have applied the algorithm by Fortune [For87] available via ftp at netlib2.cs.utk.edu

2. convolution with a Sobell edge operator,

3. edge tracking (minimal edge length = 50 pixels, minimum threshold = 40, maximum threshold = 80, tolerance = 5) [2].

3-D curves, Fig. 3f), were computed by projecting edges extracted in the image onto the reconstructed surface. Only those edge points which have a surface support are used. The polygonal approximation that naturally allocates more vertices in high curvature regions was computed using the algorithm by Duda and Hart [DH73] providing C^0 continuous curves. The maximal error allowed has to be known a priory or estimated from data and supplied to the algorithm. Having the polygonal approximations C^2 continuous curves are constructed as a B-spline approximation through the vertices of the polygons.

The parameterization of a curve is constructed in order to approximate its arc length (note, that it generally cannot reach the arc length as straight line is the only one spline curve having coordinates expressed as a polynomials of the arc length). Spline knots are distributed over parameter intervals according to the length of the polygonal segments utilizing the idea of *chord length* parameterization often used for spline interpolation [Far90, page 130]. The implementation is based on algorithms by DeBoor [de 78] and the smoothing parameter was set experimentally. Finally, the curves are uniformly resampled following their arc length with a constant step ds what assures that no two strings of samples will be systematically shifted by more than $ds/2$. The arc length of a curve is computed via numerical integration using the simplest quadrature

$$s_i = s(t_i) = \sum_{i=0}^{n} \sqrt{\mathbf{x}'(t_i) \cdot \mathbf{x}'(t_i)} \Delta t \ ,$$

where $\Delta t \ = t_{i+1} - t_i$ and $\mathbf{x}'(t_i)$ stands for the tangent vector at a point $\mathbf{x}(t_i)$. Δt is set to ds since t is already an approximation of s. Being arc length, $s(t)$ is invertible and a function $t(s_i)$ can be easily constructed via inverting linear interpolation through sampled s_i. Having $t(s_i)$, resampled smoothed curve, $\mathbf{x}_i = \mathbf{x}(t(s_i))$, can be generated. Similarly, tangent vectors $\mathbf{x}'_i = \mathbf{x}'(t(s_i))$ are computed.

4.3 Curve description

The most delicate step in the approach is the selection of reference points and/or directions one needs to find at least one reference point and one reference direction. In general, they need not be related but in practice to look for directions in reference points is natural if they lie on curves (or surfaces if one were to use the direction of the normal).

The experiments were performed for three different reference point selection methods i.e. **One-reference-point**, **N-reference-points**, and **All-reference-points**. Curves detected on the left and right are called *model* curves and *test* curves respectively. In the case of **All-reference-points** selection strategy all points (together with their tangent vectors) on each model curve are taken as reference points of the curve. The descriptions of the model curves according to the eqs. (1) are then computed with respect to every reference point. The description of a test curve is computed only for one reference point and its reference direction

[2] The program "itest" developed by Dorin Ungureanu (Dorin.Ungureanu@esat.kuleuven.ac.be) was used to process images and extract edges.

on each curve. Having reference points, directions, and sampled points on curves, signature computation is merely substitution into the set of equations (1).

Since invariants i_2 and i_3 from the set (1) change sign with a change of curve orientation either absolute values or values computed for both orientations ought to be used. To simplify the task we have changed the orientation of some curves to make them compatible.

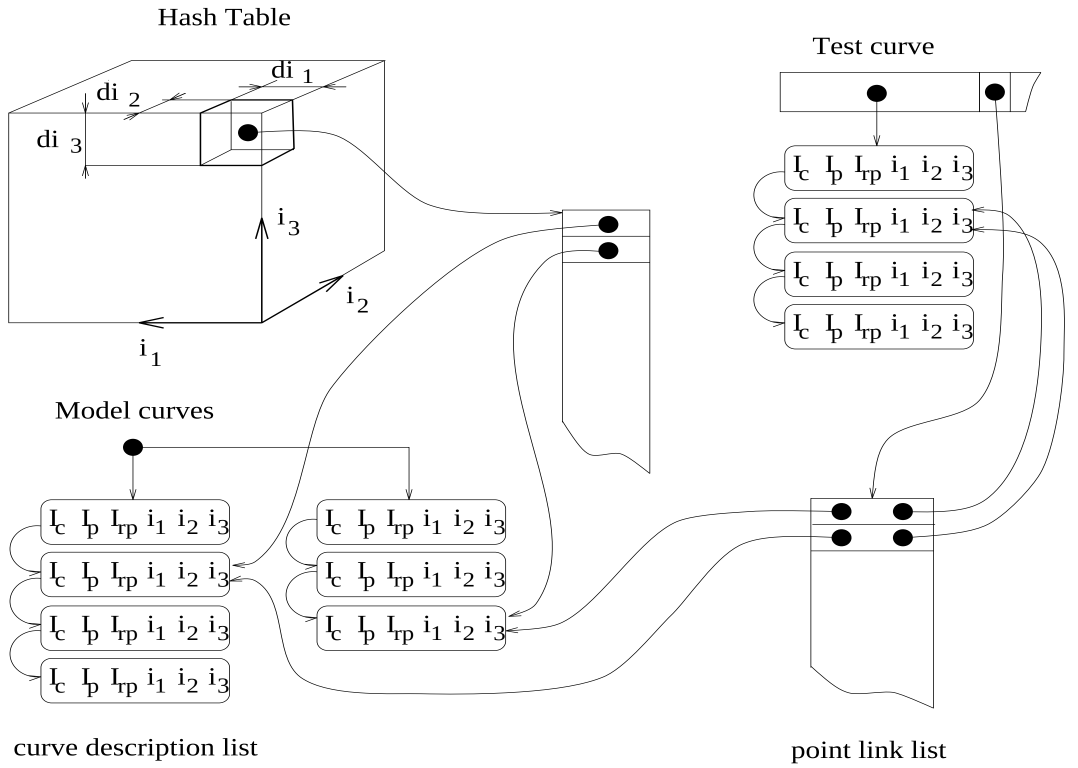


Figure 4: Hash table and curve description for matching.

4.4 Matching candidates via geometrical hashing

Efficient matching technique based on indexing into a hash table has been previously used by Guéziec and Ayache [GA94], and Stein and Medioni [SM93] to match 3-D curves. The error analysis of geometrical hashing is studied by Lamdan and Wolfson in [LH89].

Here, the scheme described in [KHW91] was followed. For every pair, point X - reference point X_1, we compute the corresponding invariants and store them into the *description entry* consisting of (I_c,I_p, I_{rp}, i_1, i_2, i_3), where I_c is the identification number of a curve and I_p resp. I_{rp} denote the index of the point resp. the index of the reference point on the curve. Thus, every curve has associated a list of description entries of size $m \cdot n$, where n is a number of points and m is a number of reference points of the curve [3], see Fig. 4.

Then, the three dimensional hash table with cells having sizes equal to quantization intervals is built. Every cell at a position (qi_1, qi_2, qi_3) has attached the list of pointers establishing links to all description entries for which holds (qi_1, qi_2, qi_3)=$quantize$(i_1, i_2, i_3).

[3] The implementation is based on the C list library written by Radim Šára, sara@vision.felk.cvut.cz

Test curves are matched against model curves by indexing the hash table. In order to cope with uncertainty and quantization, all cells within the *uncertainty ellipsoid* centered at (qi_1, qi_2, qi_3) are accesed. When any nonempty cell is entered a *point link entry* for every model curve description entry, attached to the cell, is created. After all test curves are processed, each curve has attached the list of point link entries establishing correspondences between description entries on a test curve and description entries on model curves.

In general, one can have entries from one model curve but related to different reference points. On the other hand, there is only one reference point for every test curve. If curves are in correspondence, reference points are in correspondence as well. Note that all model description entries pointing to a single test curve have to have the same reference point used to compute their signatures. So, for every pair of curves we find the reference point on the model curve, say I_{rp}, as the one geting the maximum number of votes from the test curve. Only those point link entries having their reference point indexes equal to I_{rp} form the set of matching candidates between curves.

The numbers of matching candidates for every pair of curves can be arranged to form a mismatch table. Rows of the table indexed by i correspond to model curves while columns indexed by j correspond to test curves. An element on the position i, j holds the number of votes from the test curve j for the model curve i.

4.5 Robust motion estimation

Geometrical hashing is an efficient method to reduce the number of possible matching candidates. However, it can not be expected that no false matches will appear. On the other hand, the estimation of Euclidean motions can be naturally formulated as minimizing the sum of squared distances between corresponding points. Straightforward application of the Least Squares estimator (LSQ) to the whole set of matching candidates would certainly fail due to outliers resulting from mismatched points. To remedy the situation we introduce a two step robust motion estimation algorithm.

The scheme for object recognition as described in [LH89] solves the problem using verification process. It means that Euclidean transformation is computed for every pair of curves having enough votes in the mismatch table or even for every triple of points. The correct transformation would be the one which minimizes the overall LSQ criterion. This process resembles Monte Carlo robust regression methods.

In this section, we shall shortly recapitulate LSQ estimation of the Euclidean motion, robust regression method based on Least Median of Squares and finally, we will describe a two step robust motion estimation algorithm.

4.5.1 LSQ motion estimation

Euclidean motion estimation from a set of registered points, say $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$, $i = 1, 2, \cdots, n$ based on the minimization of squared distances aims at obtaining a matrix of rotation, R, and a vector of translation, $\mathbf{t}$, so that

$$e^2(R, \mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{y}_i - (R\mathbf{x}_i + \mathbf{t})\|^2$$

the overall mean squared error is minimized. See [SA91] for a survey of methods published before 1991 and [Kan94] for some methods which have appeared later.

Basically, all methods decompose the motion to a translation and a rotation using the following fact. If two sets of points are related by Euclidean motion, $\mathbf{y}_i = R\mathbf{x}_i + \mathbf{t}$, then the quantities $\mathbf{p}_i = \mathbf{x}_i - \mu_x$ and $\mathbf{q}_i = \mathbf{y}_i - \mu_y$, where μ_x resp. μ_y are the centroids of the sets $\mathbf{x}_i$ resp. $\mathbf{y}_i$, are related by a pure rotation, $\mathbf{q}_i = R\ \mathbf{p}_i$. For details see [AHB87][Ume91][Kan94]. The statement is true only if the sets are really in exact correspondence what in noisy environments never happens or if the number of corresponding points is high and samples are drawn from one random population. Once R is computed $\mathbf{t}$ is obtained using $\mu_y = R\mu_x + \mathbf{t}$. Various approaches were used to handle optimal estimation of R from $\mathbf{p}_i$ and $\mathbf{q}_i$ mostly differing in the parameterization of $SO(3)$. The exception is the method by Walker and Shao [WS91] who use *dual number quaternion* representation of Euclidean motions and do not explicitly compute the decomposition based on centroids.

We have implemented the method based on the following theorem given by Umeyama in [Ume91].

Theorem 1 *Let* $X = \{\mathbf{x}_i\}$ *and* $Y = \{\mathbf{y}_i\}$, $i = 1, 2, \cdots, n$ *be corresponding point patterns in m-dimensional space. The minimum value of* ϵ^2 *of the mean squared error*

$$e^2(R, \mathbf{t}, c) = \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{y}_i - (cR \cdot \mathbf{x}_i + \mathbf{t})||^2$$

of these two patterns (R: rotation, t: translation, c: scaling) is given as follows:

$$\epsilon^2 = {\sigma_y}^2 - \frac{tr(DS)}{{\sigma_x}^2}$$

where

$$\begin{aligned}
\mu_x &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \\
\mu_y &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i \\
{\sigma_x}^2 &= \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{x}_i - \mu_x||^2 \\
{\sigma_y}^2 &= \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{y}_i - \mu_y||^2 \\
\Sigma_{xy} &= \frac{1}{n} \sum_{i=1}^{n} (\mathbf{y}_i - \mu_y)(\mathbf{x}_i - \mu_x)^T
\end{aligned}$$

and let a singular value decomposition of Σ_{xy} *be* UDV^T *(*$D = diag(d_i)$*,* $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$*). When rank(*$\Sigma_{xy}$*)* $\geq m - 1$*, the optimum transformation parameters are determined as follows:*

$$\begin{aligned}
R &= USV^T \\
\mathbf{t} &= \mu_y - cR\mu_x \\
c &= \frac{1}{{\sigma_x}^2} \mathrm{tr}(DS)
\end{aligned}$$

where

$$S = \begin{cases} I & \text{if } \det(UV) = 1 \\ \mathrm{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(UV) = -1 \ . \end{cases}$$

We folow Umeyama's approach since it can properly handle rotations versus reflections. Moreover, this SVD formulation is advantageous since data weighting can be easily incorporated and the covariance matrix of estimated parameters is readily available. Moreover, stable algorithms for SVD decomposition are widely available.

4.5.2 Least Median of Squares motion estimation

The traditional LSQ estimator builds on the strong assumption that random errors come from a single distribution If the distribution is close to a Gaussian distribution, the method turns to be MAP estimator. However, it becomes unreliable in the presence of outliers or if errors come from a mixture of distributions having different means.

When the Euclidean motion is computed, outliers due to large distances between mismatched pairs might disturb the LSQ estimation. From this point of view mismatches can be viewed as outliers. Various robust regression techniques like Robust Sequential Estimators [MB93] or the Least Median of Squares (LMSq) can be used to cope with outliers. See [MMR91] for a survey of robust regression algorithms in computer vision and [WZX91] for a robust motion estimation algorithm using continuation.

The LMSq estimator as proposed by Rousseeuw and Leroy in [RL87] is used [RW94]. The trick is to replace the standard least squares criterion $min \sum_{i=1}^{n} {r_i}^2$ by its robust version

$$\min \ \text{median} \ {r_i}^2 \ , \quad i = 1 \cdots n \ .$$

Such proposed estimator can cope with up to 50% of outliers and is immune even to leverage points. Unfortunately, the objective function is not convex and has numerous local minima. It is not possible to obtain a closed form solution as in the case of ordinary Least Squares nor can it be implemented using a sequential reweighted LSQ estimator as in the case of M-estimators. However, the approximate solution can be obtained using the resampling algorithm of Rousseeuw. The amount of computation of the basic algorithm is almost prohibitive since a fit through all p-tuples, where p is the number of parameters involved in the fit, should be performed. A Monte Carlo technique has been proposed in [RL87] to speed up the computation for larger data sets. It can be shown that the probability that all m randomly selected p-tuples will contain at least one outlier is

$$P = [1 - (1 - \epsilon)^p]^m,$$

where ϵ denotes the fraction of data contaminated by outliers. Note that P does not depend on the size of the data set. In our case when $p = 6$, $m > 50$ is needed to have $P < 0.002$ for 30% contamination.

In the presence of Gaussian noise the LMSq estimator is considerably less efficient than the LSQ estimator but almost the same efficiency can be achieved applying the LSQ estimation to a trimmed data set. Binary weights w_i can be allocated to every data point according to the following scheme:

$$w_i = \begin{cases} 1 & \frac{|r_i|}{\hat{\sigma}} \le 2.5 \\ 0 & \frac{|r_i|}{\hat{\sigma}} > 2.5 \end{cases} \quad , \quad \text{where} \quad \hat{\sigma} = 1.4826 \left(1 + \frac{5}{n-5}\right) \sqrt{\operatorname{med} {r_i}^2}$$

is the robust estimation of scale.

The factor 1.4826 is used to obtain a consistent estimation in the case of Gaussian noise. The term $\frac{5}{n-5}$ is a correction for small finite samples. The scale estimation is undoubtedly necessary if data have unitless residuals or if it is difficult to find their physical interpretation. However, if residuals are obtained as squared Euclidean distances between points, the outlier cut off threshold can be set according to the expected level of errors in a sample.

The LMSq motion estimation algorithm consists of two steps:

1. **Robust LMSq fit and outlier detection.** All correspondences entering the algorithm are considered. From this population m random triples of corresponding points are drawn and the LSQ optimal Euclidean motion is generated for each triple. The median of squared errors with respect to *all* points is stored together with the corresponding transformation. When a certain number of samples is exceeded, the transformation minimizing the median criterion among all drawn transformations is used to transform points into the a single coordinate frame. The residuals are used to eliminate outliers.

2. **Trimmed LSQ fit.** After outliers are removed an ordinary LSQ fit is performed for all inliers providing the final optimal transformation.

4.6 Matching hypotheses verification

Geometrical hashing provides candidate point matches between curves albeit does not completely solve the matching. False candidates may be generated if curves are locally similar. To discriminate between correct and false matches a verification must be used. Basically, we distinguish two problems, the *recognition* of unrelated pairs curves and the *registration* of frames attached to families of curves in a rigid configuration.

Curve recognition can be viewed as a matching of sub-curves from the test database to the complete curves in the model database. Hence, one is looking for the mapping $j = \lambda(i) : I \longrightarrow J$ assigning the test curve $i \in I$ to the model curve $j \in J$, where I and J are the index sets of test and model curves respectively. Since the curves in the test database are mutually independent every curve can be matched independently and therefore the matching can be carried out by $\mathrm{O}(M \times N)$ complexity algorithm, where M and N denote the number of curves in the model and in the test database respectively. Notice that different test curves can be matched to one model curve.

Considering registration we have two equivalent test databases rather than one model and one test database. The registration poses itself as a matching of parts of curves from the first database to parts of curves from the second database under the constraint of a common transformation for all matching sub-curves. The fact that curves are supposed to lie on rigid surfaces introduces the global constraint. The main difference, compared to the recognition, is that a global correspondence rather than equivalence classes of curves has to be found. It means that $\lambda(i)$ need not to be a mapping anymore because its support need not to cover the whole of the set I but, on the other hand, it has to be a one-to-one function since the incidence is preserved under Euclidean motions. Algorithmically we are facing much more difficult problem akin to a sub-graph isomorphism which is known to be a NP hard problem [BC82].

Even though above mentioned tasks are not equivalent they both require to evaluate a similarity of corresponding curves. In other words a curve similarity measure has to be defined. Besl in [Bes90] proposes the L_2 curve matching metric for the j-th model curve $\mathbf{y}_j(s)$ and the i-th test curve $\mathbf{x}_i(s+s_0)$ both parameterized by their arc length with the reparameterization offset s_0:

$$L_2(\mathbf{x}_i, \mathbf{y}_j) = \inf_{s_0} \inf_{R,\, T} \int_0^l \|\mathbf{y}_j(s) - R\ \mathbf{x}_i(s+s_0) - T\|^2 ds\ . \tag{3}$$

There are two major drawbacks of the proposed metric. First, prior to carrying out the integration the reparameterization offset has to be found and second, the metric is susceptible to occlusions since local errors in the arc length estimation accumulates and finally can completely flaw the similarity measure.

To remedy the situation we introduce two similarity measures using the L_2 metric which do not rely on a global arc length estimation. Let us assume that the optimal R and T were found and all what remains to do is to evaluate the similarity.

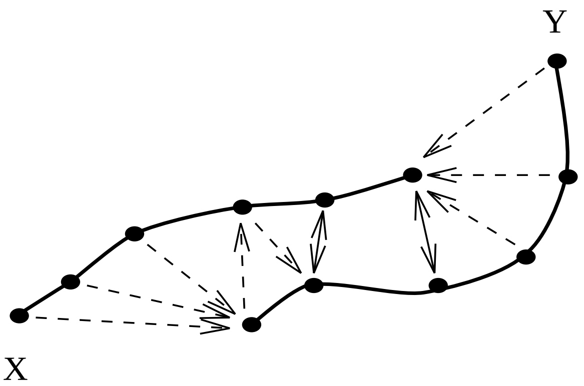


Figure 5: The correspondence established using the closest point relation between two curves X and Y. Notice that the curves are treated as discrete point sets. Reciprocal pairs are related by full lines with arrows on both sides.

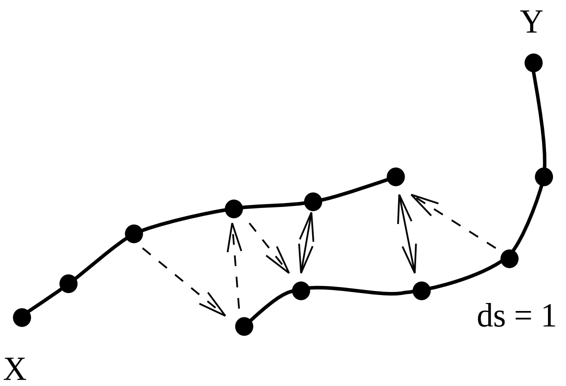


Figure 6: The correspondence established using the ϵ-reciprocity relation for ϵ equal to the sampling interval ds.

4.6.1 A similarity for matching complete curves

The first measure evaluates the global similarity of curves (GSC) taking into account the local error in the registration as well as the difference in lengths of the curves. Curves $\mathbf{x_i}(s)$ and $\mathbf{y_j}(s)$ are sampled following their arc lengths in equidistant intervals ds resulting in two point sets $X = \{x_i\}$ and $Y = \{y_j\}$. Then, the following formula for the similarity between two point sets defines GSC measure:

$$GSC(X,Y) = \left(\frac{\sum_{i=1}^{m} d(x_i, Y) + \sum_{j=1}^{n} d(y_j, X)}{2 \times \min(m,n)} \right)^{-1} \quad , \quad d(x_i, Y) = \min_{\forall y_j \in Y} \|x_i - y_j\| \; , \qquad (4)$$

where m and n are the numbers of points in the sets X and Y respectively. The function $d(x_i, Y)$ assigns to every point x_i from the set X the distance to its closest point in the set Y. Notice that the relation "closest point" does not obey, in general, symmetry, see Fig. 7. No knowledge about continuity is used and therefore the measure applies to raw clouds of points without any change.

4.6.2 A similarity for matching parts of curves

The second measure tests similarity for partially overlapping curves. The fact that the correspondence between matched curves has to be locally continuous is used to establish correspondences.

Clearly, if the point sets X and Y are exactly registered the points in corresponding parts of curves coincide. It implies that first, distances between coincident points are zero and second the closest point relation is symmetrical for those points. Let us call the set of point pairs in coincidence the set of reciprocal points (RP).

For registered continuous curves $\mathbf{x}_i(s_x)$ and $\mathbf{y}_j(s_y)$ the closest point correspondence, $s_y = \lambda_{RP}(s_x) = s_x + s_0$, relating reciprocal pairs is trivially piecewise continuous, and one-to-one. If a small Euclidean motion is applied to one of the curves, λ_{RP} splits into two continuous correspondences $s_y = \lambda_{CPX}(s_x)$ and $s_x = \lambda_{CPY}(s_y)$. Let the function

$$\triangle s_x(s_x, R, T) = s_x - s_{x_y} = s_x - \lambda_{CPY}(\lambda_{CPX}(s_x))$$

measures the arc length distance between the point $\mathbf{x}_i(s_x)$ and the point $\mathbf{x}_i(s_{x_y})$ which is the closest point to the point $\mathbf{y}$ which is, again, the closest point of the point $\mathbf{x}_i(s_x)$, see Fig. 7. It can be shown that

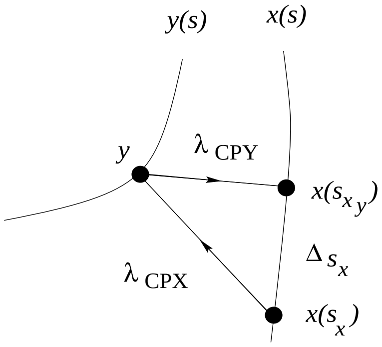


Figure 7: *Arc length distance $\triangle s_x(s_x, R, T)$ of nonreciprocal points related by the closest point relation. The point $\mathbf{x}_i(s_{x_y})$ is the closest point of the point $\mathbf{y}$ which is the closest point of the point $\mathbf{x}_i(s_x)$.*

$$\lim_{R \to E,\, T \to 0} \triangle s_x(s_x, R, T) = 0$$

using the following argument. Since for

$$R \to E,\ T \to 0 \ : \ \lambda_{CPY}(s_y) \to \lambda_{RP}^{-1}(s_y) \text{ and } \lambda_{CPX}(s_x) \to \lambda_{RP}(s_x) \; ,$$

$$\lim_{R \to E,\, T \to 0} \triangle s_x(s_x, R, T) = s_x - \lambda_{RP}^{-1}(\lambda_{RP}(s_x)) = s_x - s_x = 0 \; .$$

Similar relation holds for the arc length difference $\triangle s_y(s_y, R, T)$ on the curve $\mathbf{y}_j$. The local continuity of $\triangle s_x(s_x, R, T)$ and $\triangle s_y(s_y, R, T)$ assures that arc length distances have to be small for small perturbations of R and T, hence yielding the following rule of thumb: *locally consistent correspondences are only ϵ-reciprocal points i.e. points having $\triangle s_x(s_x, R, T) \leq \epsilon$ and $\triangle s_y(s_y, R, T) \leq \epsilon$.* For correct R, T the size of ϵ depends on the level of noise and on the size of sampling interval ds and it should be higher than the maximal sampling phase shift $ds/2$. Similarly to GSC, the curves $\mathbf{x_i}(s)$ and $\mathbf{y_j}(s)$ are sampled following their arc lengths in equidistant intervals ds resulting in two point sets $X = \{x_i\}$ and $Y = \{y_j\}$. The similarity measure for partially overlapping curves (PSC) is then given by

$$
\begin{aligned}
PSC(X,Y) &= \left(\frac{\sum_{x_i \in RP_\epsilon} d(x_i, Y \cap RP_\epsilon) + \sum_{y_j \in RP_\epsilon} d(y_j, X \cap RP_\epsilon)}{m+n} \right)^{-1}, \qquad (5) \\
RP_\epsilon &= \{x_i \mid x_i \in X \wedge \triangle s_x(s_x(x_i)) \leq \epsilon\} \bigcup \{y_i \mid y_i \in Y \wedge \triangle s_y(s_y(y_i)) \leq \epsilon\} \ ,
\end{aligned}
$$

where m and n stand for the number of ϵ-reciprocal points in the X and Y set respectively. The RP_ϵ denotes the set of all ϵ-reciprocal points and $d(\,\cdot\,,\,\cdot\,)$ has the same meaning as in the case of GSC similarity measure.

4.7 The Iterative Closest Point Algorithm

The iterative closest point algorithm (ICP) has been developed by Besl and McKay in [BM92]. The algorithm solves curve registration provided that a good initial estimate of transformation is available. We have applied a modified version of the ICP algorithm to obtain fine curve registration. The modification consist of replacing the LSQ estimator with LMSq and using only the ϵ-reciprocal point correspondence instead of a complete correspondence. Our algorithm is stated as follows:

1. Find the initial transformation by applying LMSq motion estimation algorithm to the set of matching candidates generated by hashing.

2. Compute the correspondence between two curves by finding ϵ-reciprocal points.

3. Compute LMSq optimal Euclidean motion between corresponding points.

4. Terminate the iteration when the mean square error difference drops below a preset threshold or if the maximal number of iterations is exceeded.

4.8 Global frame registration

We already mentioned that the global frame registration is a NP complete problem. Different strategies like tree search [SH81], relaxation [Dav79], maximal clique detection [HS89], etc. were used to constrain exhaustive search in order to reach a solution in a polynomial time. See also [Bes90] for other references. Here we present two simple alternatives aiming at demonstrating above mentioned methods rather than giving a complete solution to the global registration problem.

It has been shown how the Least Median of Squares based algorithm can handle motion estimation up to 50% of mismatches. The first idea that immediately comes to mind is to use the same algorithm for the global frame registration assuming that the majority of candidate correspondences is correct. Indeed, we can see that the global frame transformation has been correctly found in the experiment with **One-reference-point**, Figure 11, where the fraction of outliers does not exceed 11%, Fig. 11. On the other hand, the registration is not satisfactory if the large number of incorrect candidate matches is present as it shows up in the experiment with **All-reference-points**, see Figure 19.

The second simple strategy builds on the fact that every three corresponding points in a general position fix an Euclidean motion. The idea is to use the three best matching curves found in the verification step. Figure 26 shows the result of the LMSq motion estimation applied to the ϵ-reciprocal points of the curves "C", "o", and "e" which obtained the highest GSC similarity measures. Notice that the residuals, Fig. 24, are considerably lower than in the previous cases and no outliers are detected.

5 Results

5.1 Global frame registration via the LMSq algorithm using all candidate matches from hashing

We have carried out three experiments consisting of generating candidate matches and computing the global frame registration using the LMSq algorithm from the candidate matches. In this experiments no matching validation is involved. We hope that at one least half of matches is correct. The same set of invariants (1) for different reference point selection strategies as discussed in the section 3.3.5 was used. The results are grouped into the three sets of figures, each corresponding to one experiment. Every set consists of:

Figure 8 shows the corresponding pairs of points in 3-D for the first six pairs of corresponding curves. For each test curve, the model curve maximizing the number of votes in the match table is selected to constitute the pair of curves. The full list of corresponding curves is shown in Figure 10c). Figure 9 shows two orthogonal projections of the description space. Corresponding points are marked with a star and linked by a line. Again, only the first six pairs of curves are shown.

Two match tables generated by hashing are shown in figures Figure 10a) and Figure 10b). The first one gives the absolute number of votes while the second one gives the relative number of votes with respect to the lengths of test curves.The corresponding curves were equally numbered and therefore the tables should be diagonally dominant when curves are correctly matched.

Final transformation, Figure 10d), bringing curves into the registration is given in the form of 4×4 homogeneous matrix $\mathbf{M}$, where $\mathbf{M} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\,\mathbf{T} \\ 0 & 1 \end{bmatrix}$.

The plot of global frame registration residuals is shown in Figure 10d). The full horizontal line on a level **LMSqCutOff** delineates inliers (below the line) from the outliers. Dashed line shows the level proposed by Rousseeuw and Leroy using the robust scale estimate denoted by **rs**. The number of points entering the Trimmed LSQ estimation is denoted by **# of inliers** and it can be found below the figure altogether with the number of outliers detected by the algorithm denoted as **# of outliers**.

Values of important parameters involved in the matching algorithm can be found in Figure 10f). The following parameters are shown:

polygApxErr	maximum error in polygonal approximation
smoothLambda	spline smoothing parameter
dsResampling	curve resampling step
hTdI_1, hTdI_2, hTdI_3	hash table quantization intervals
confEllAx_1,...	principal axes of the confidence ellipsoid
LMSqSamplNum	number of random samples in the LMSq algorithm
LMSqCutOff	absolute outlier cut off level
LMSqRSLevel	relative outlier cut off level, multiplies $\hat{\sigma}$

Finally, two plots of registered 3-D curves are shown in Figure 11.

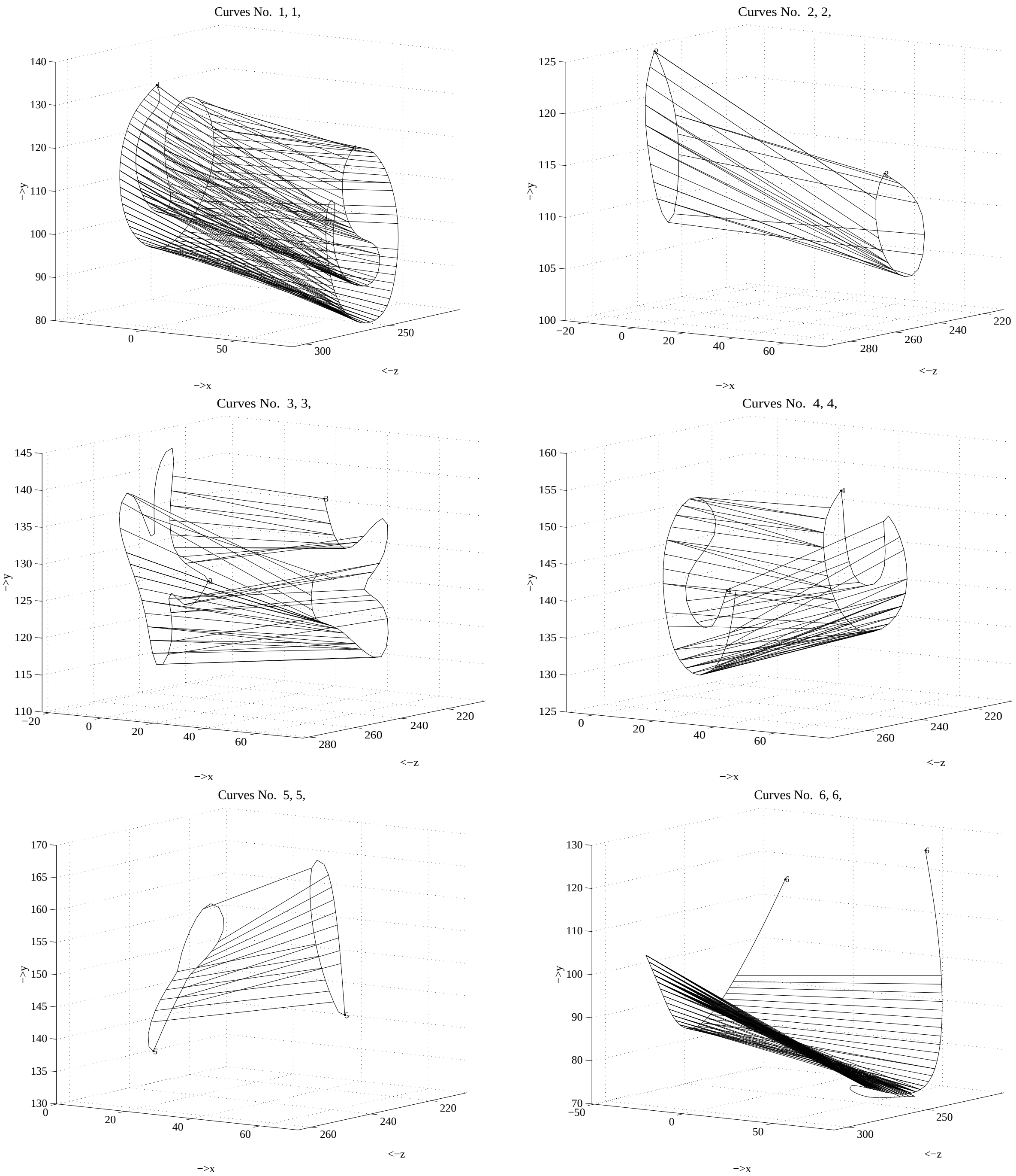


Figure 8: **One-reference-point**: Candidate point matches produced by hashing shown in 3-D (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

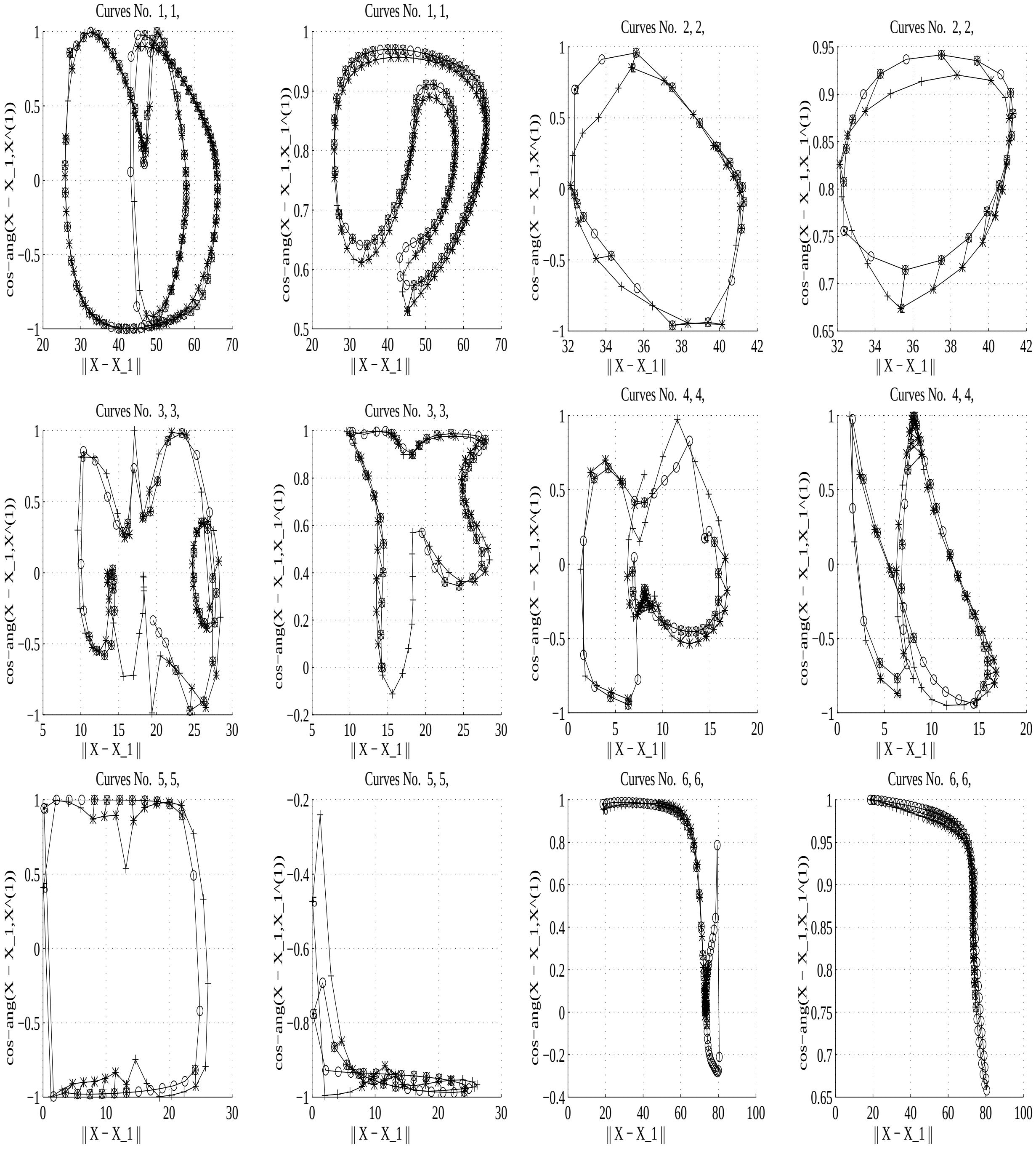


Figure 9: **One-reference-point**: Candidate point matches produced by hashing in the description space (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

Mismatch Table (number of votes)

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	127	4	4	0	0	2	0	0	3	0	0	0	0
2	5	22	0	0	0	0	0	0	0	0	0	0	0
3	5	0	47	0	0	2	2	0	0	0	0	5	0
4	0	0	0	51	3	0	0	0	0	0	0	0	0
5	0	0	0	2	15	0	0	0	0	0	0	0	0
6	1	0	1	0	0	83	0	0	0	0	0	2	0
7	0	0	1	0	0	44	0	0	0	0	0	2	0
8	0	0	0	0	0	0	0	20	0	0	0	0	0
9	2	0	0	0	0	5	0	0	17	0	4	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0

a)

Number of votes/testCurveLen [%]

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	141	19	9	0	0	3	0	0	15	0	0	0	0
2	6	105	0	0	0	0	0	0	0	0	0	0	0
3	6	0	107	0	0	3	15	0	0	0	0	45	0
4	0	0	0	121	11	0	0	0	0	0	0	0	0
5	0	0	0	5	54	0	0	0	0	0	0	0	0
6	1	0	2	0	0	117	0	0	0	0	0	18	0
7	0	0	2	0	0	62	0	0	0	0	0	18	0
8	0	0	0	0	0	0	0	77	0	0	0	0	0
9	2	0	0	0	0	7	0	0	85	0	29	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0

b)

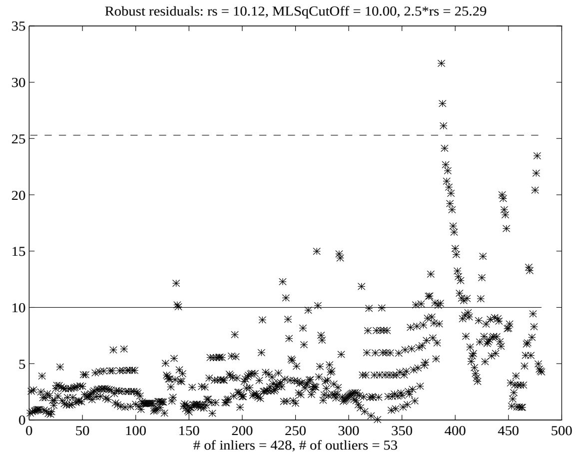
Maximum vote number matching

test	1	2	3	4	5	6	7	8	9	11	12
model	1	2	3	4	5	6	3	8	9	9	3

c)

0.6448	0.6121	0.4579	−248.5518
−0.7537	0.6088	0.2476	38.3325
−0.1272	−0.5048	0.8538	103.6518
0.0000	0.0000	0.0000	1.0000

d)



e)

dsResampling	2.0000
polygApxErr	2.0000
smoothLambda	−3.0000
hTdI_1	2.0000
hTdI_2	0.0417
hTdI_3	0.0417
confEllAx_1	2.0000
confEllAx_2	0.1500
confEllAx_3	0.1500
MLSqSamplNum	1000.0000
MLSqCutOff	10.0000
MLSqRSLevel	2.5000

f)

Figure 10: **One-reference-point**: a) Absolute match table. b) Relative match table. c) Maximum vote number matching. d) Transformation matrix e) Residuals of the global frame motion estimation using the LMSq algorithm applied to all matching candidates produced by hashing.

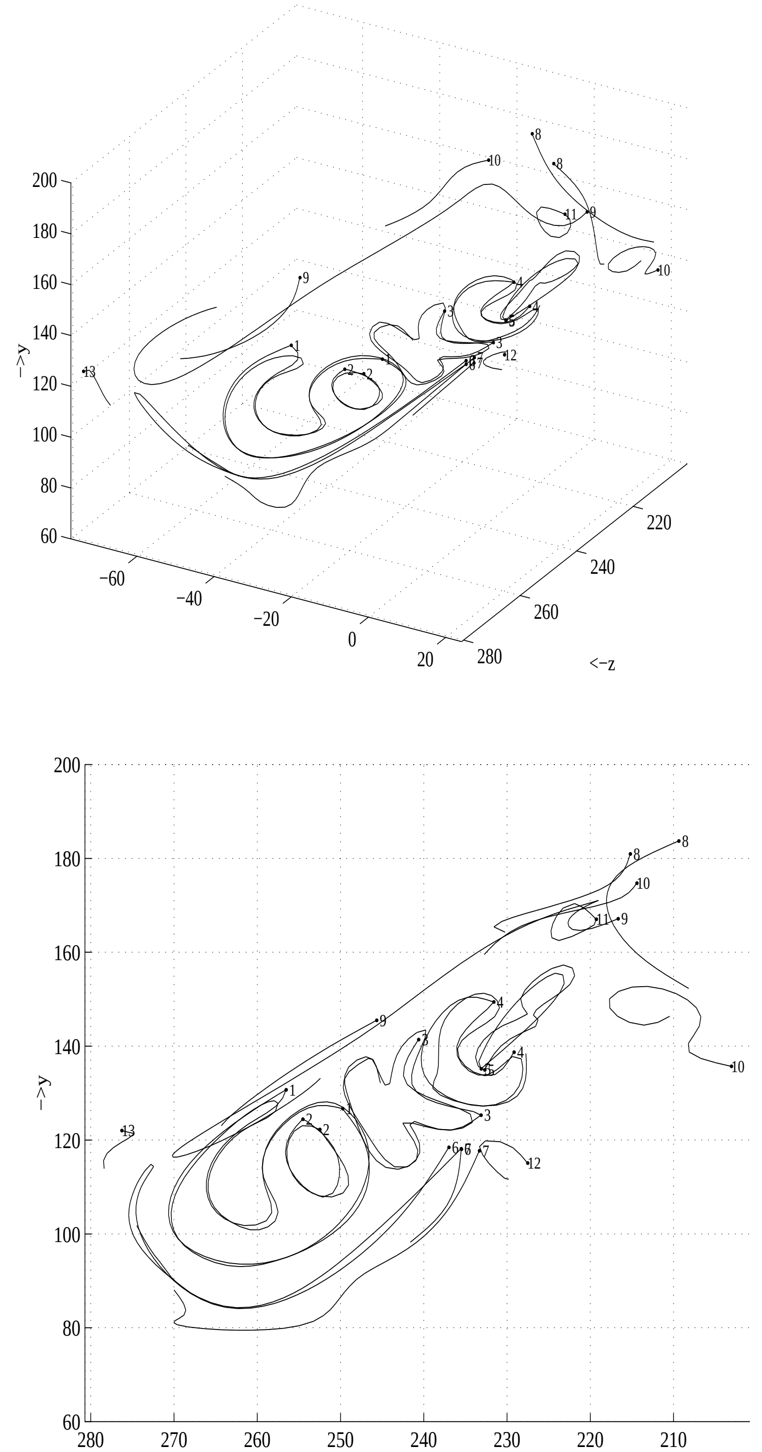


Figure 11: **One-reference-point**: Global frame LMSq registration using matching candidates from the hashing (No validation is involved. The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

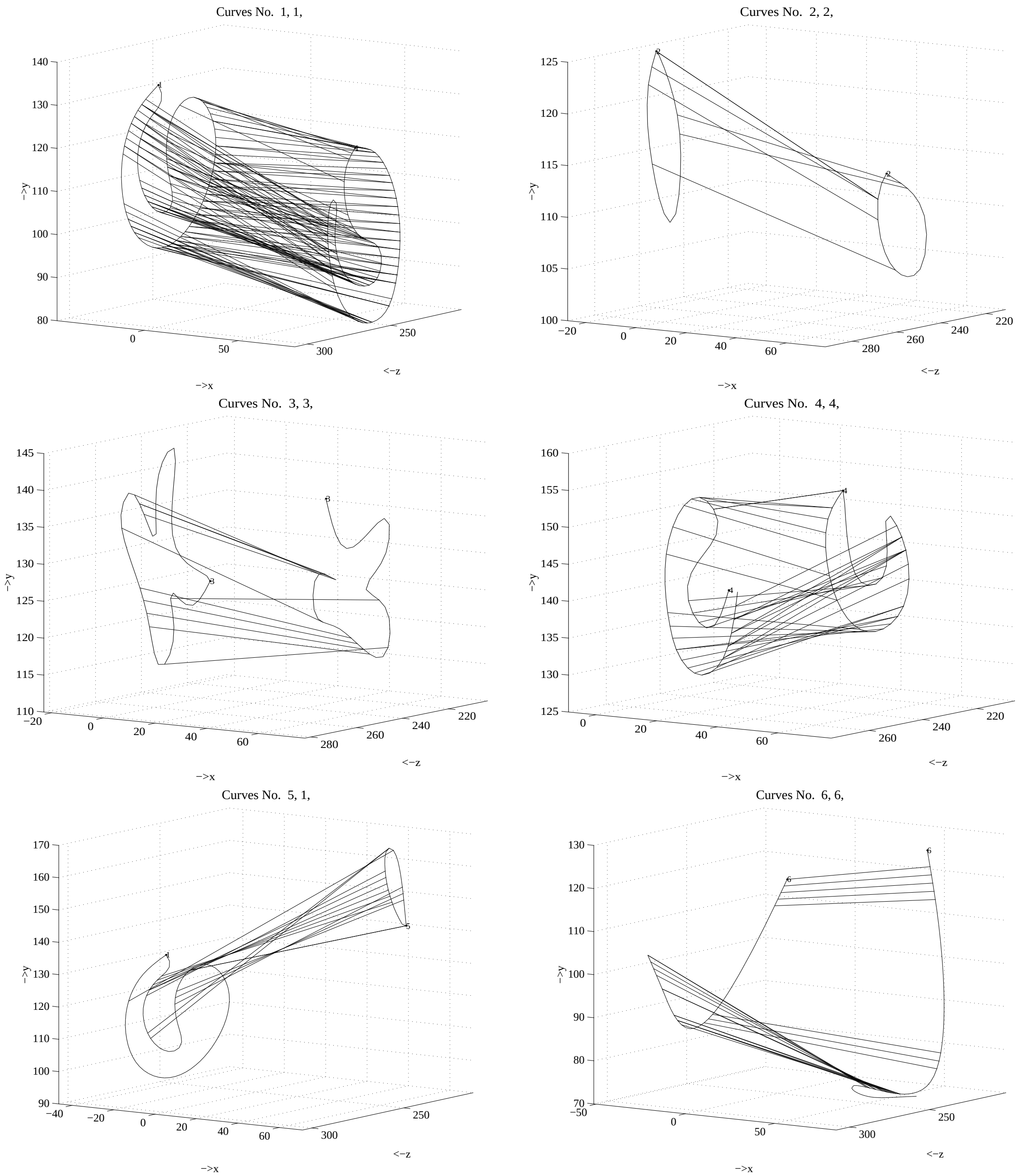


Figure 12: **N-reference-points**: Candidate point matches produced by hashing shown in 3-D (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

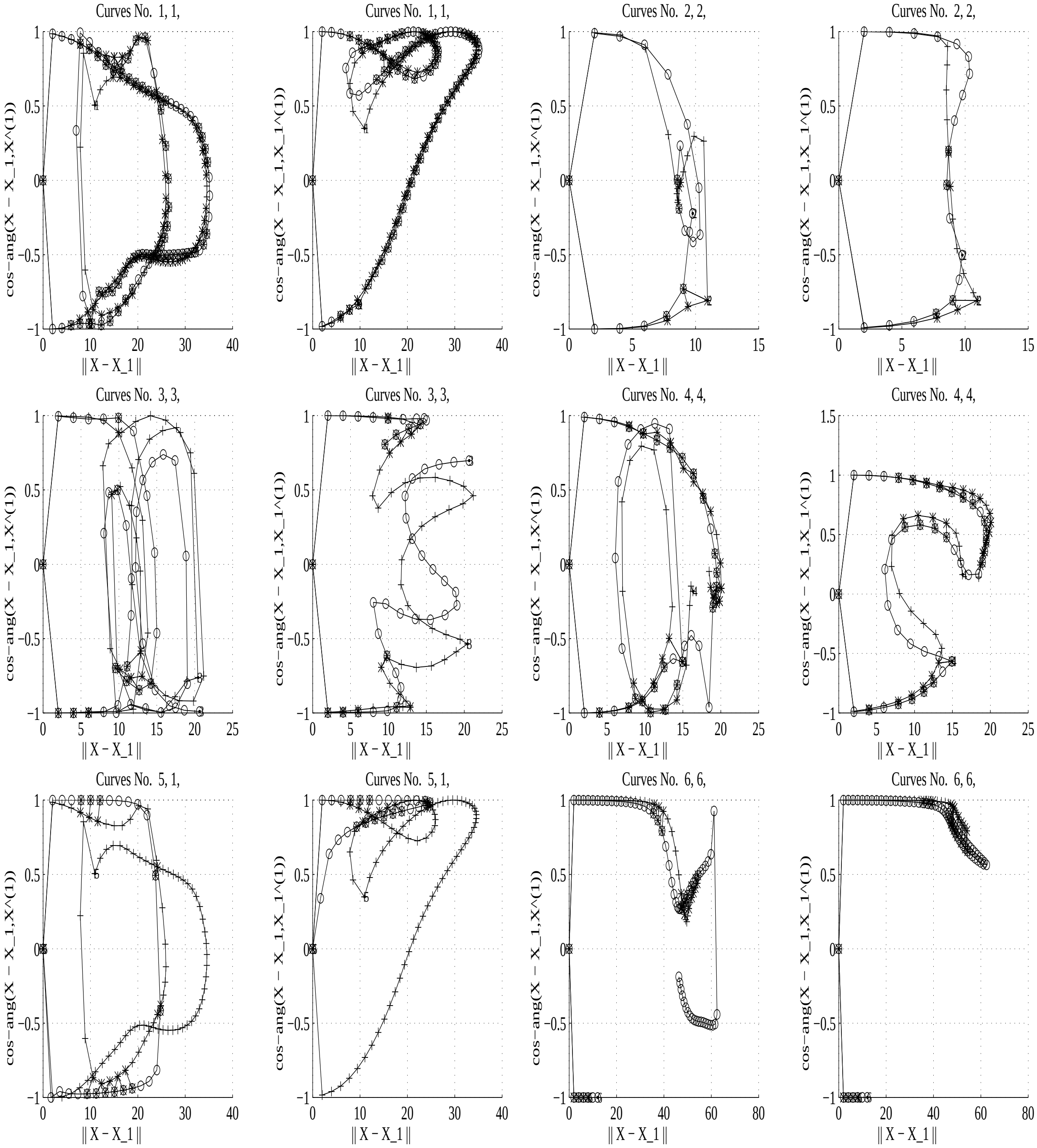


Figure 13: **N-reference-points**: Candidate point matches produced by hashing in the description space (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

Mismatch Table (number of votes)

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	109	4	10	16	14	7	4	8	8	2	3	2	4
2	5	7	8	3	2	5	1	4	4	2	2	2	3
3	8	2	11	4	2	7	1	6	4	2	2	2	2
4	14	3	10	32	5	8	4	9	8	2	2	2	4
5	8	3	9	6	10	4	5	7	5	2	2	2	3
6	1	1	4	3	2	21	1	3	3	2	2	2	1
7	1	1	2	2	2	4	3	2	2	1	1	1	1
8	1	1	3	2	6	9	6	9	2	1	1	1	1
9	3	1	4	8	3	7	1	3	4	3	2	2	1
10	5	2	5	4	2	4	1	4	9	1	2	1	2

a)

Number of votes/testCurveLen [%]

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	121	19	23	38	50	10	31	31	40	9	21	18	50
2	6	33	18	7	7	7	8	15	20	9	14	18	38
3	9	10	25	10	7	10	8	23	20	9	14	18	25
4	16	14	23	76	18	11	31	35	40	9	14	18	50
5	9	14	20	14	36	6	38	27	25	9	14	18	38
6	1	5	9	7	7	30	8	12	15	9	14	18	12
7	1	5	5	5	7	6	23	8	10	5	7	9	12
8	1	5	7	5	21	13	46	35	10	5	7	9	12
9	3	5	9	19	11	10	8	12	20	14	14	18	12
10	6	10	11	10	7	6	8	15	45	5	14	9	25

b)

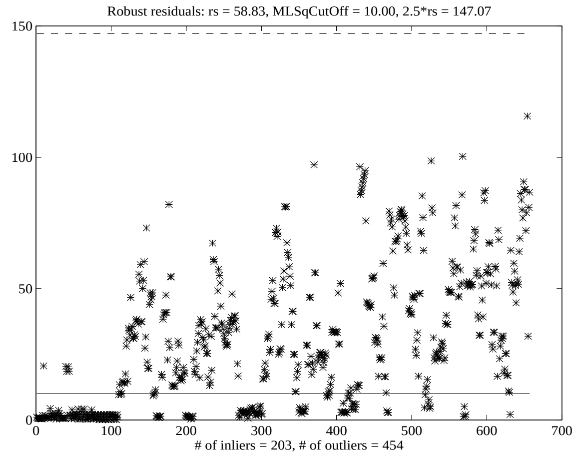
Maximum vote number matching

test	1	2	3	4	5	6	7	8	9	10	11	12	13
model	1	2	3	4	1	6	8	4	10	9	1	1	1

c)

0.6697	0.5890	0.4524	−246.4120
−0.7288	0.6385	0.2475	33.9663
−0.1431	−0.4954	0.8568	102.9802
0.0000	0.0000	0.0000	1.0000

d)

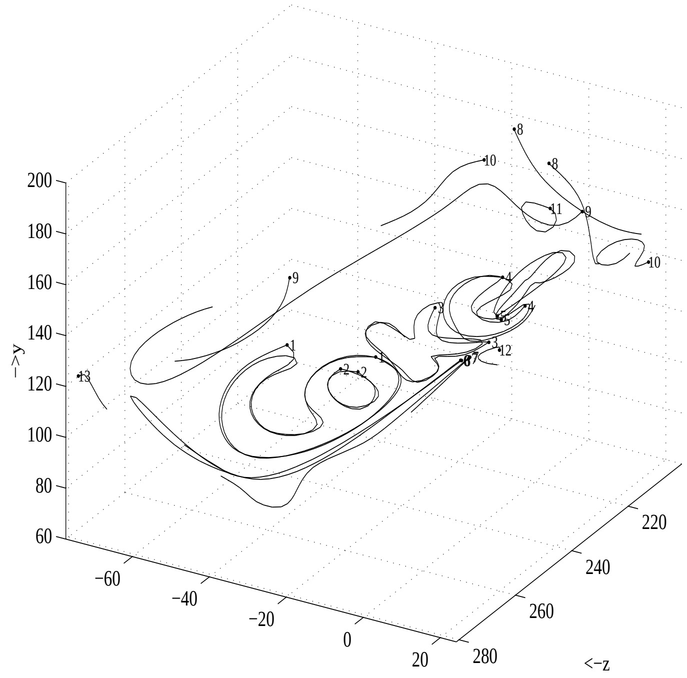


e)

dsResampling	2.0000
polygApxErr	2.0000
smoothLambda	−3.0000
hTdI_1	2.0000
hTdI_2	0.0417
hTdI_3	0.0417
confEllAx_1	2.0000
confEllAx_2	0.1500
confEllAx_3	0.1500
MLSqSamplNum	1000.0000
MLSqCutOff	10.0000
MLSqRSLevel	2.5000

f)

Figure 14: **N-reference-points**: a) Absolute match table. b) Relative match table. c) Maximum vote number matching. d) Transformation matrix e) Residuals of the global frame motion estimation using the LMSq algorithm applied to all matching candidates produced by hashing.



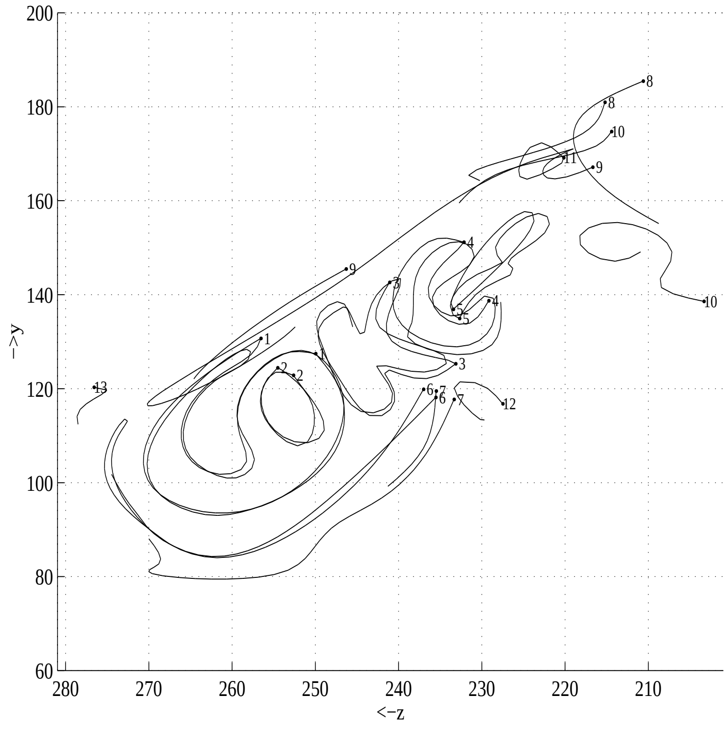


Figure 15: **N-reference-points**: Global frame LMSq registration using matching candidates from the hashing (No validation is involved. The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

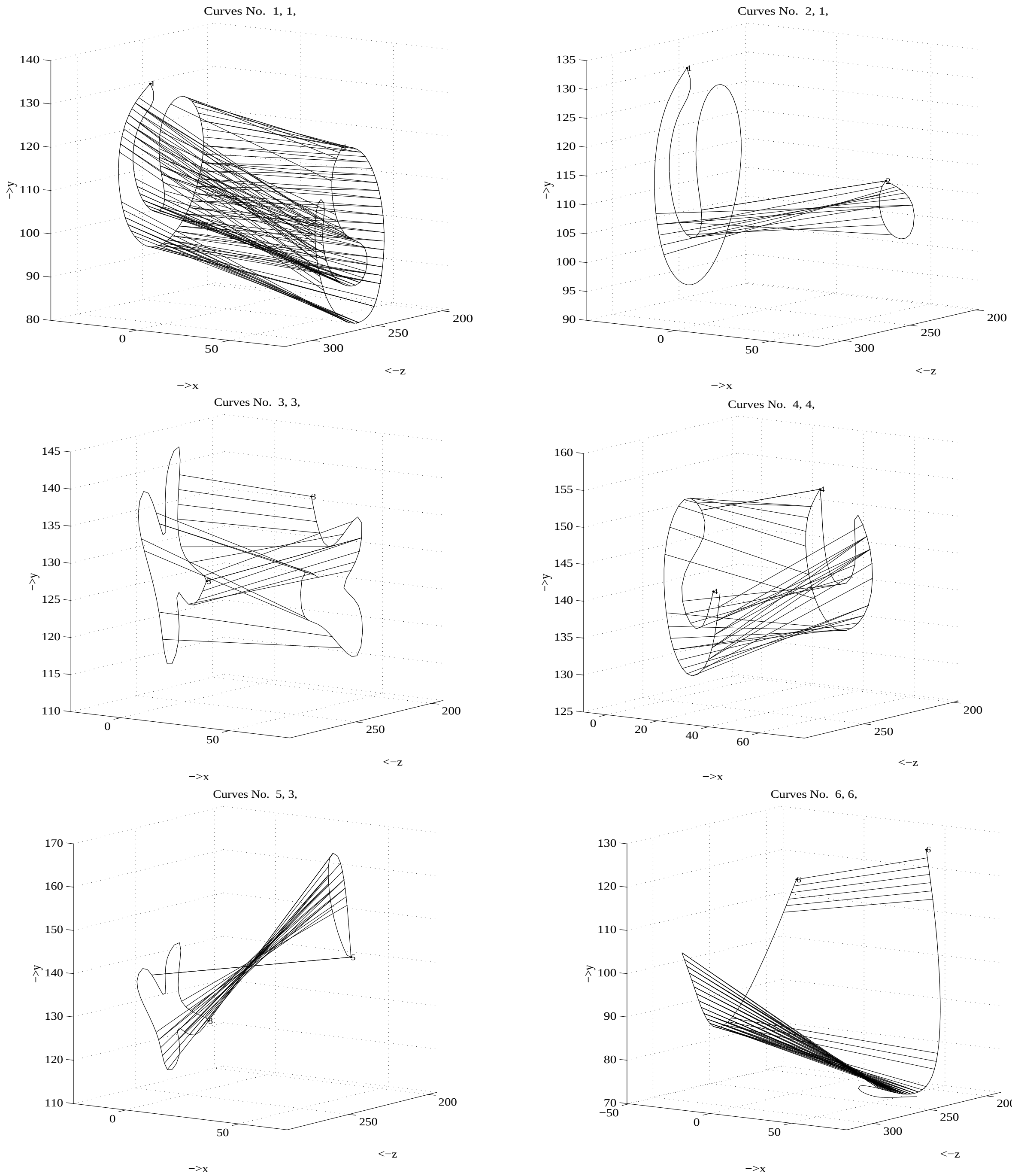


Figure 16: **All-reference-points**: Candidate point matches produced by hashing shown in 3-D (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

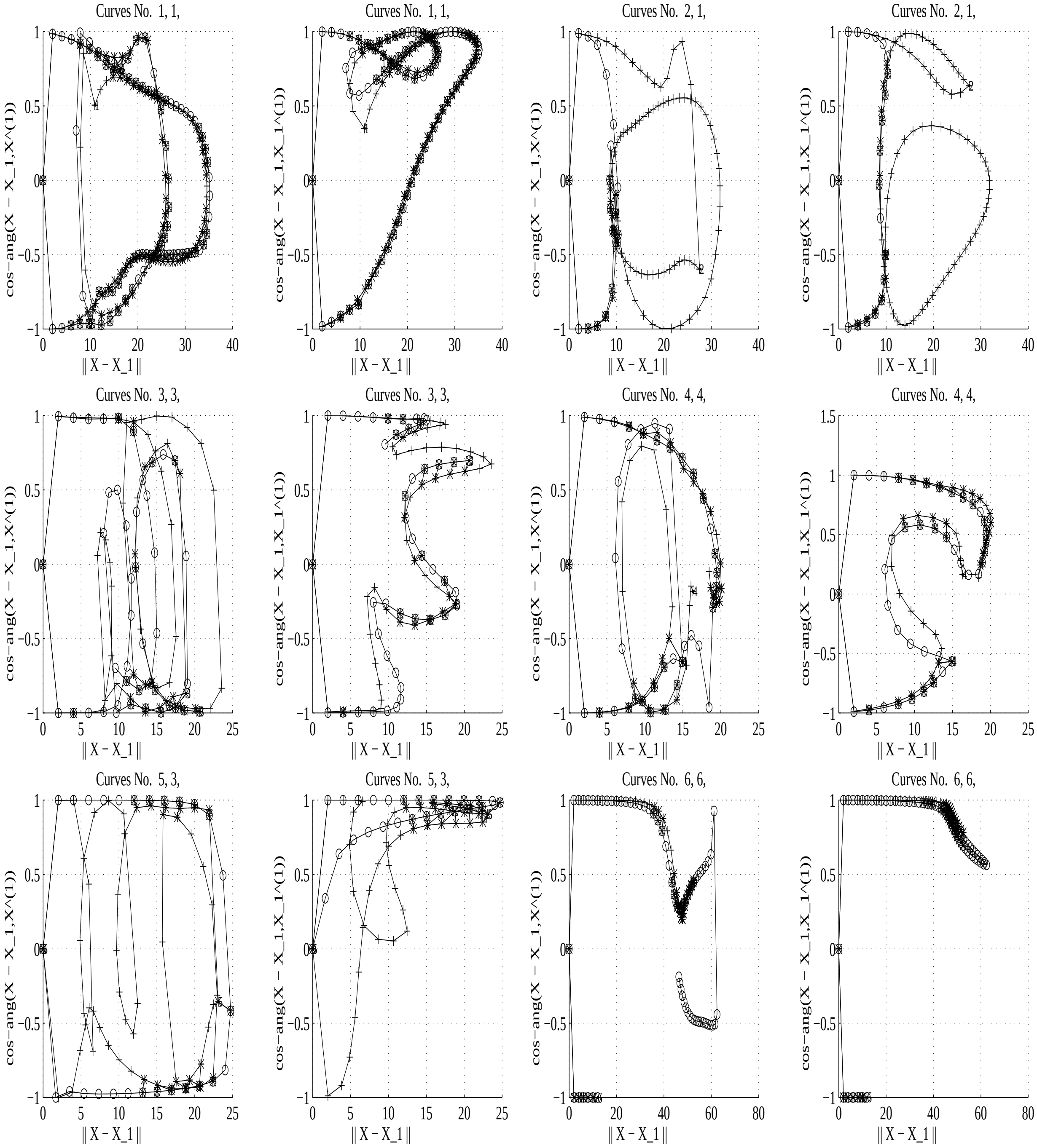


Figure 17: **All-reference-points**: Candidate point matches produced by hashing in the description space (The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

Mismatch Table (number of votes)

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	109	14	16	21	14	14	10	23	17	7	5	4	5
2	11	11	11	10	6	7	5	7	7	9	5	3	5
3	22	6	20	15	22	11	9	11	13	7	6	4	6
4	21	8	10	32	8	8	5	9	8	9	6	5	5
5	17	7	13	11	12	7	6	10	10	6	8	6	6
6	20	5	9	11	10	47	10	20	16	3	3	2	5
7	20	5	9	8	7	29	6	9	9	6	5	3	5
8	5	3	4	4	8	11	7	10	7	4	2	3	3
9	17	7	8	11	10	11	9	12	9	9	5	3	5
10	15	5	9	11	7	11	6	10	13	5	4	2	5

a)

Number of votes/testCurveLen [%]

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	121	67	36	50	50	20	77	88	85	32	36	36	62
2	12	52	25	24	21	10	38	27	35	41	36	27	62
3	24	29	45	36	79	15	69	42	65	32	43	36	75
4	23	38	23	76	29	11	38	35	40	41	43	45	62
5	19	33	30	26	43	10	46	38	50	27	57	55	75
6	22	24	20	26	36	66	77	77	80	14	21	18	62
7	22	24	20	19	25	41	46	35	45	27	36	27	62
8	6	14	9	10	29	15	54	38	35	18	14	27	38
9	19	33	18	26	36	15	69	46	45	41	36	27	62
10	17	24	20	26	25	15	46	38	65	23	29	18	62

b)

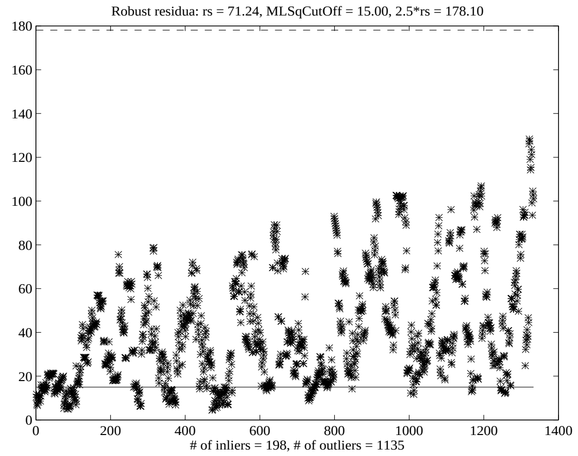
Maximum vote number matching

test	1	2	3	4	5	6	7	8	9	10	11	12	13
model	1	1	3	4	3	6	1	1	1	2	5	5	3

c)

0.7844	0.5148	0.3460	−220.4552
−0.6188	0.6879	0.3795	−13.0179
−0.0426	−0.5117	0.8581	98.8996
0.0000	0.0000	0.0000	1.0000

d)



e)

dsResampling	2.0000
polygApxErr	2.0000
smoothLambda	−3.0000
hTdI_1	2.0000
hTdI_2	0.0417
hTdI_3	0.0417
confEllAx_1	2.0000
confEllAx_2	0.1500
confEllAx_3	0.1500
MLSqSamplNum	1000.0000
MLSqCutOff	10.0000
MLSqRSLevel	2.5000

f)

Figure 18: **All-reference-points**: a) Absolute match table. b) Relative match table. c) Maximum vote number matching. d) Transformation matrix e) Residuals of the global frame motion estimation using the LMSq algorithm applied to all matching candidates produced by hashing.

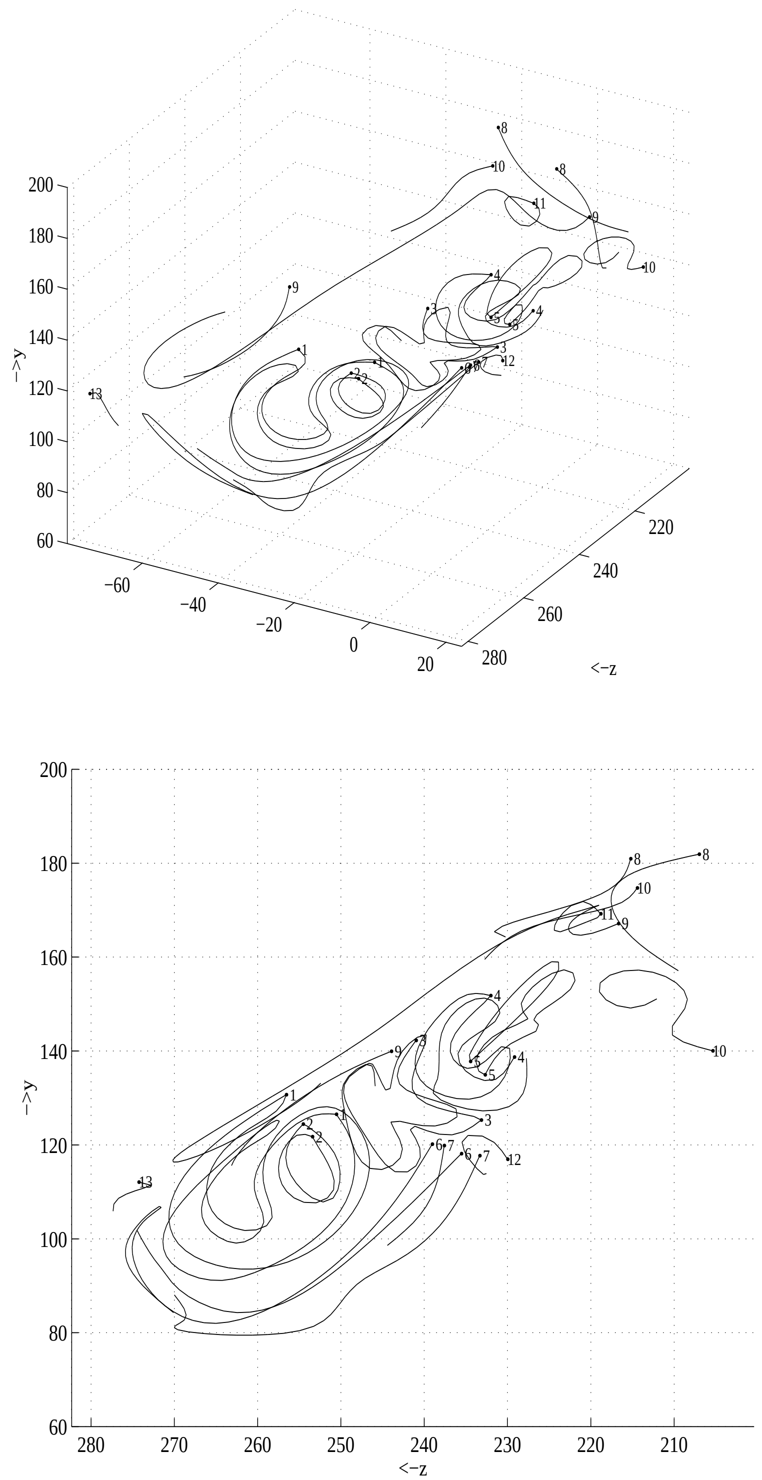


Figure 19: **All-reference-points**: Global frame LMSq registration using matching candidates from the hashing (No validation is involved. The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

5.2 Curve and global frame registration using the best matching curves from the validation

Two experiments with matching validation were carried out. In both cases we have used the candidate matches produced in the All-reference-points experiment.

Figure 20 shows the corresponding pairs of points in 3-D for the first six pairs of corresponding curves. For each test curve, the model curve maximizing the GSC measure is selected to constitute the pair of curves. The full list of corresponding curves is shown in Figure 24c). Figure 21 shows two orthogonal projections of the description space.

Two plots of registered 3-D curves using the LMSq algorithm applied to all candidate matches produced by hashing are shown in Figure 25.

Figure 22 shows the first six best GSC matching curves in 3-D after 10 iterations of the ICP algorithm. The numerical values of the GSC measure are shown in the table 24a). Notice that completely corresponding curves like "C" and "o" have a mean square error lower that 0.5 mm (the similarity measures are defined as the reciprocal values of the mean square errors). Figure 23 shows the first six best PSC matching curves in 3-D after 10 iterations of the ICP algorithm. The numerical values of the PSC measure are shown in the table 24b). Notice also how the PSC similarity increased for the curve "k" which has a substantial part missing. Of course, the similarity also increased for other curves since it is easier to find partial matches than global ones.

The global transformation between frames was computed using the three best GSC matching curves i.e. "C", "o", and "e" and the final transformation matrix is shown in Figure 24d). The plot of robust residuals is shown in Figure 24d) and, finally, two plots of registered curves are shown in Figure 26.

Values of all important parameters involved in the verification can be found in Figure 24f). Following parameters are shown:

MatchSelectionMethod	The method used to generate the matching tables e.g. Figure 24c. $0 \rightarrow$ maximum vote number matching, $1 \rightarrow$ maximum similarity matching.
VeriLMSqSamplNum	number of random samples for the LMSq algorithm used in the ICP algorithm.
VeriLMSqCutOff	absolute outlier cut off level used in the ICP algorithm.
VeriLMSqCutOffLevel	relative outlier cut off level used in the ICP algorithm, multiplies $\hat{\sigma}$.
VeriMatchIterNum	Number of iterations used in the ICP algorithm.
VeriMatchTermEps	Error tolerance used to terminate the ICP algorithm.
CCDistSumMethod	Selects the method for computing denominators of equations (4) and (6) used in the ICP algorithm. $0 \rightarrow\ m + n$, $2 \rightarrow\ 2 \times \min(m, n)$.

CCDistNNbrMethod Selects the method to compute nominators of equations (4) and (6) used in the ICP algorithm. $0 \rightarrow$ GSC, $1 \rightarrow$ PSC.

CCDistNNbrEps ϵ parameter in ds units for ϵ-reciprocal points establishment used in the ICP algorithm.

GlobRegistrCurvNum The number of best matching curves used for the global frame registration.

GRDistSumMethod Selects the method for computing denominators of equations (4) and (6) used in the global frame registration.

GRDistNNbrMethod Selects the method to compute nominators of equations (4) and (6) used in the global frame registration. $0 \rightarrow$ GSC, $1 \rightarrow$ PSC.

GRDistNNbrEps ϵ parameter in ds units for ϵ-reciprocal points establishment used in the global frame registration.

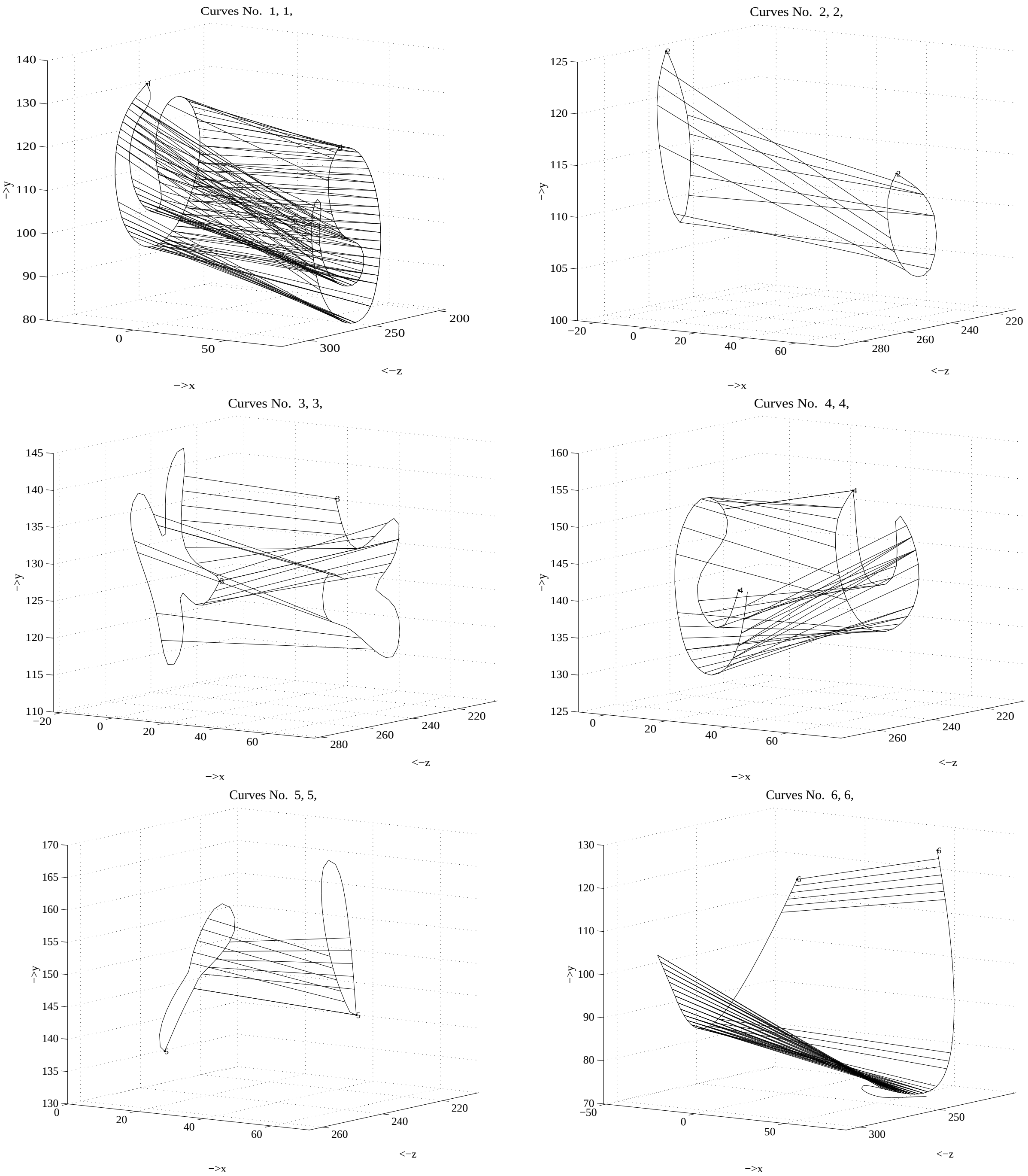


Figure 20: **All-reference-points** Candidate point matches produced by hashing shown in 3-D (The best matching curves are selected as those having the maximum GCS measure resulting from the validation.).

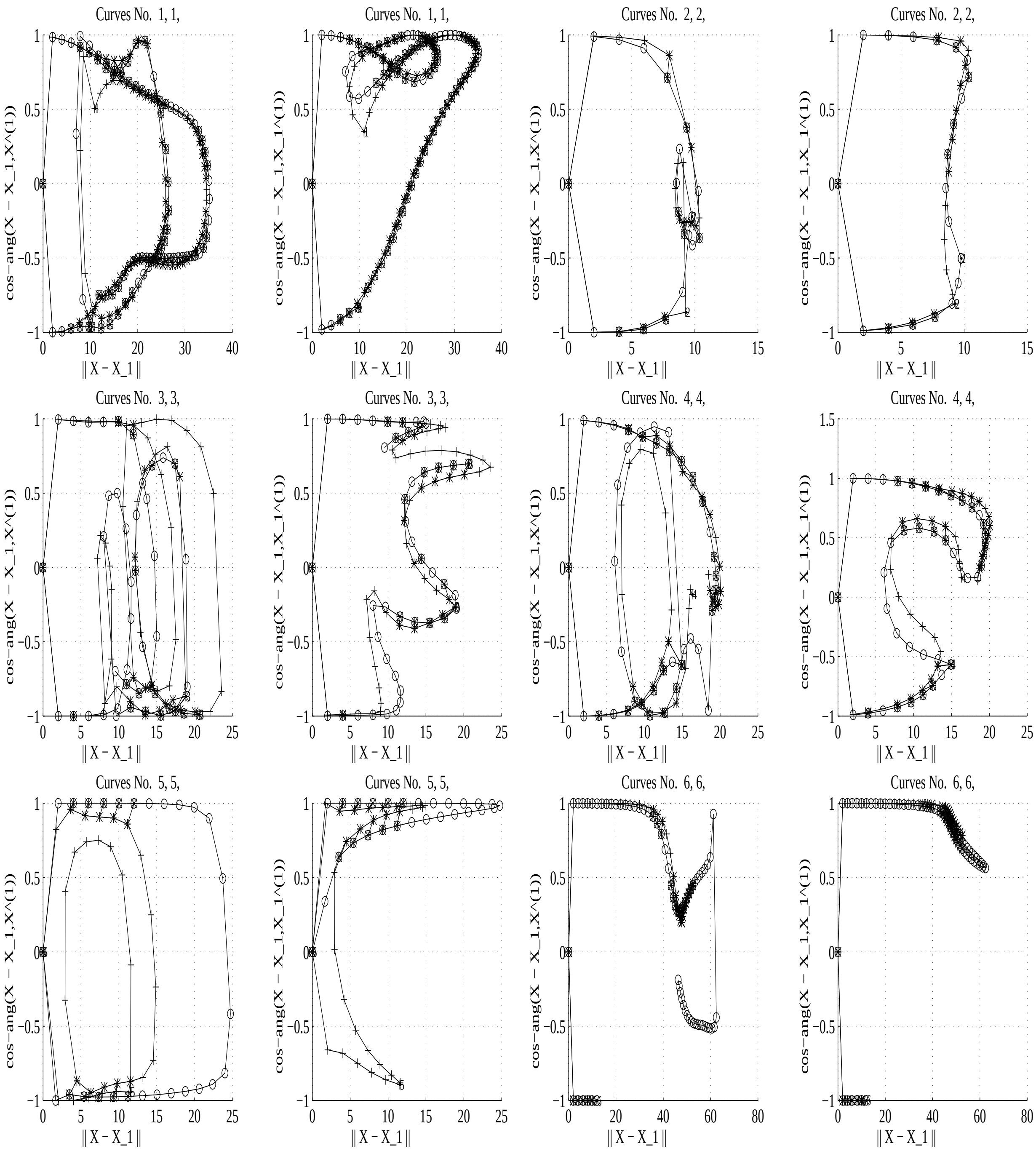


Figure 21: **All-reference-points** Candidate point matches produced by hashing in the description space (The best matching curves are selected as those having the maximum GCS measure resulting from the validation.).

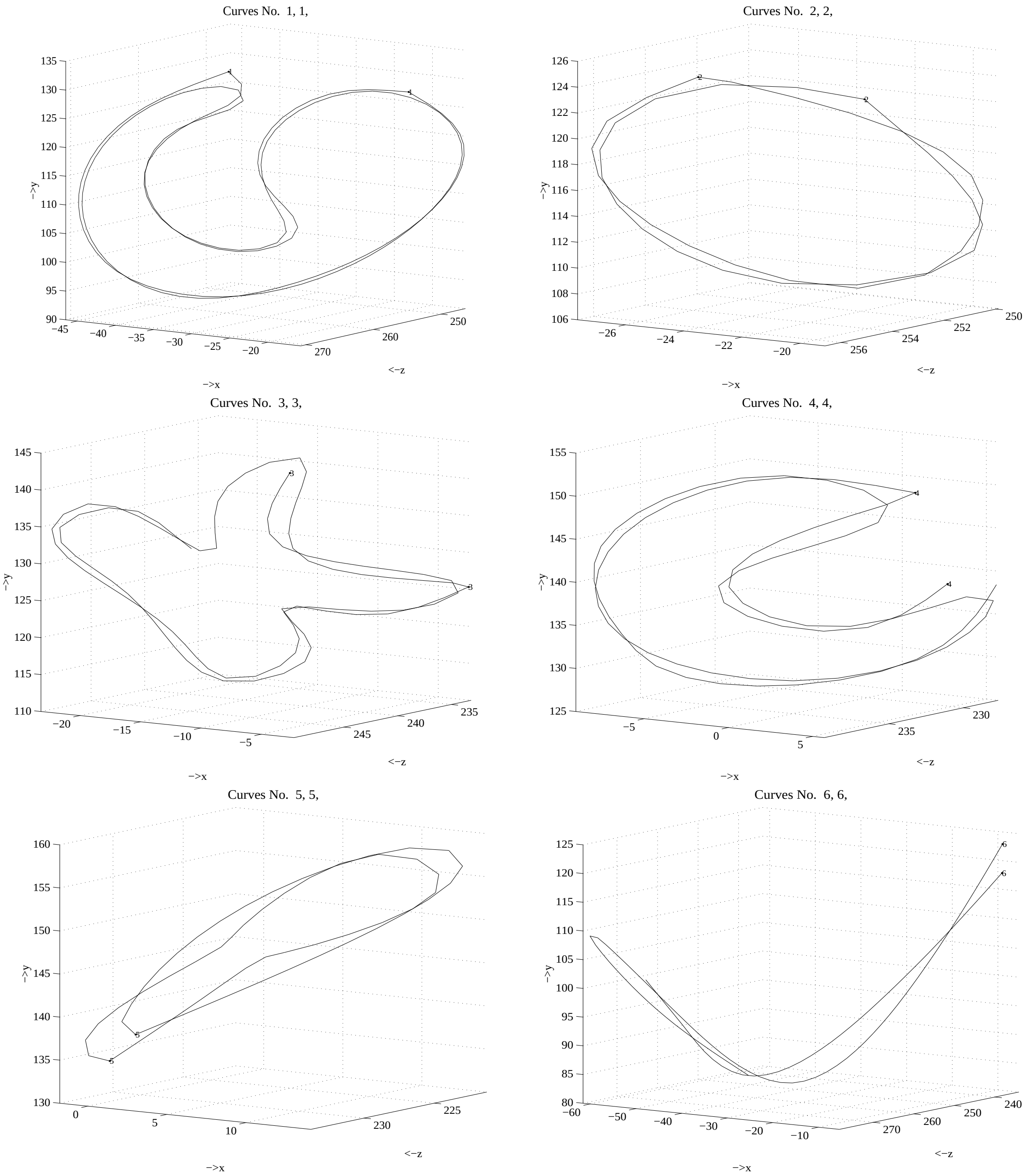


Figure 22: **All-reference-points**: The best GSC matching curves after 10 iterations of the ICP algorithm.

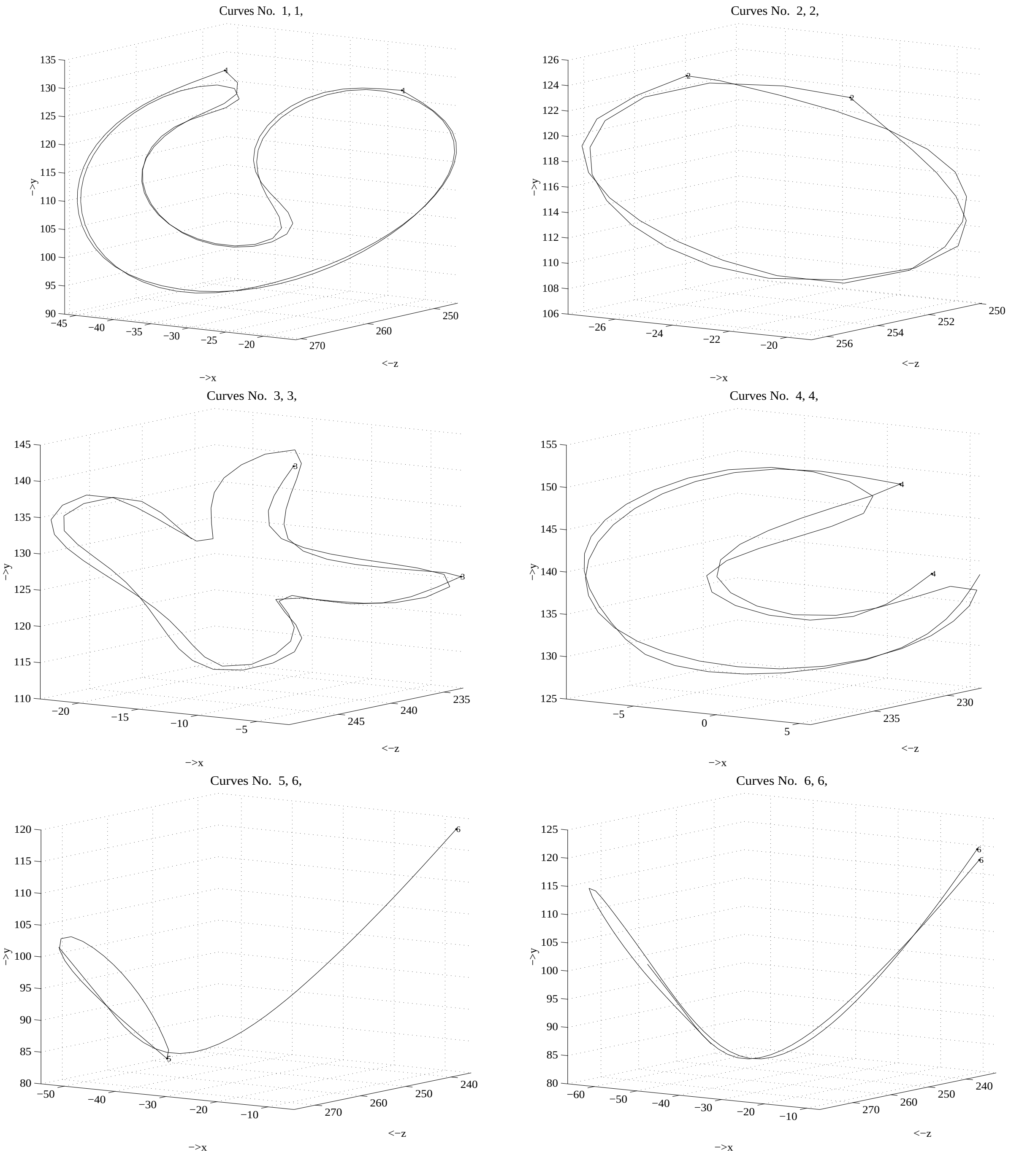


Figure 23: **All-reference-points**: The best PSC matching curves after 10 iterations of the ICP registration algorithm.

Curve−Curve Similarity Table

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.22	0.00	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	3.53	0.04	0.06	0.26	0.00	0.08	0.02	0.06	0.32	0.15	0.08	0.05
3	0.02	0.03	0.68	0.14	0.04	0.01	0.01	0.02	0.02	0.03	0.01	0.01	0.01
4	0.02	0.05	0.15	0.85	0.08	0.00	0.02	0.03	0.03	0.05	0.02	0.01	0.01
5	0.01	0.10	0.06	0.08	0.40	0.00	0.12	0.06	0.13	0.15	0.04	0.03	0.03
6	0.01	0.00	0.01	0.01	0.01	0.06	0.00	0.04	0.01	0.01	0.00	0.00	0.00
7	0.01	0.00	0.01	0.01	0.01	0.03	0.00	0.04	0.01	0.01	0.00	0.00	0.00
8	0.00	0.05	0.02	0.03	0.13	0.00	0.16	0.12	0.62	0.10	0.00	0.02	0.02
9	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.07	0.02	0.03	0.12	0.00	0.46	0.05	0.16	0.09	0.07	0.00	0.06

a)

Curve−Curve Similarity Table

model/test	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.22	0.20	0.22	0.20	0.17	0.15	0.66	0.36	0.32	0.12	0.20	0.24	0.43
2	0.32	3.61	0.14	0.30	0.32	0.25	0.44	0.25	0.31	0.43	0.26	0.29	0.51
3	0.15	0.24	1.27	0.26	0.26	0.29	0.27	0.33	0.37	0.43	0.31	0.31	0.78
4	0.18	0.49	0.22	0.86	0.25	0.14	0.22	0.13	0.20	0.27	0.22	0.48	0.43
5	0.21	0.28	0.59	0.26	0.32	0.39	0.31	0.31	0.44	0.18	0.42	0.44	0.42
6	0.20	0.23	0.18	0.18	0.72	0.80	1.23	0.20	0.31	0.27	0.27	0.00	0.45
7	0.13	0.26	0.24	0.09	0.33	0.25	0.49	0.16	0.19	0.29	0.24	0.24	0.43
8	0.20	0.23	0.38	0.19	0.52	0.56	1.14	0.44	0.63	0.22	0.00	0.39	0.68
9	0.24	0.23	0.28	0.24	0.09	0.25	0.13	0.16	0.17	0.29	0.24	0.37	0.48
10	0.21	0.26	0.30	0.29	0.50	0.35	0.83	0.34	0.29	0.28	0.24	0.00	0.45

b)

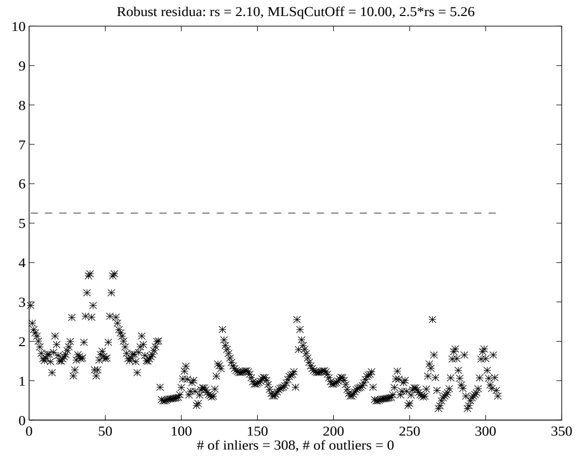
Maximum curve−curve similarity matching

test	1	2	3	4	5	6	7	8	9	10	11	12	13
model	1	2	3	4	5	6	10	8	8	2	2	2	10
similarity	2.22	3.53	0.68	0.85	0.40	0.06	0.46	0.12	0.62	0.32	0.15	0.08	0.06

c)

0.6632	0.5972	0.4512	−246.5262
−0.7406	0.6105	0.2807	29.5449
−0.1078	−0.5203	0.8471	105.6865
0.0000	0.0000	0.0000	1.0000

d)



e)

MatchSelectionMethod	1
VeriMLSqSamplNum	50
VeriMLSqCutOff	3.0000
VeriMLSqRSLevel	2.5000
VeriMatchIterNum	10
VeriMatchTermEps	0.0000
CCDistSumMethod	2
CCDistNNbrMethod	0
CCDistNNbrEps	2.0000
GlobRegistrCurvNum	3
GRDistSumMethod	0
GRDistNNbrMethod	1
GRDistNNbrEps	2.0000

f)

Figure 24: **All-reference-points**: a) Global curve similarity measure (GSC). b) Curve similarity measure for partially overlapping curves (PSC). c) Maximum GSC similarity matching. d) Transformation matrix. e) Residuals of the global frame motion estimation using ϵ-reciprocal points of the three best GSC matched curves. f) Parameters of GSC, PSC and global frame matching computation.

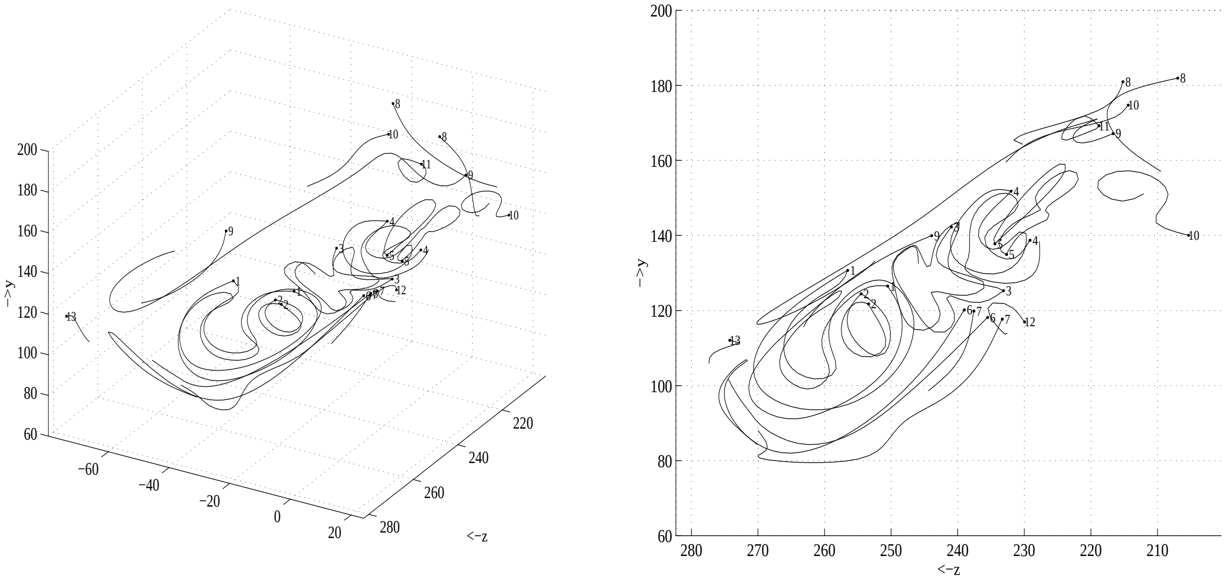


Figure 25: **All-reference-points** Global frame LMSq registration using matching candidates from the hashing (No validation is involved. The best matching curves are selected as those having the maximum number of candidate matches – votes – after hashing.).

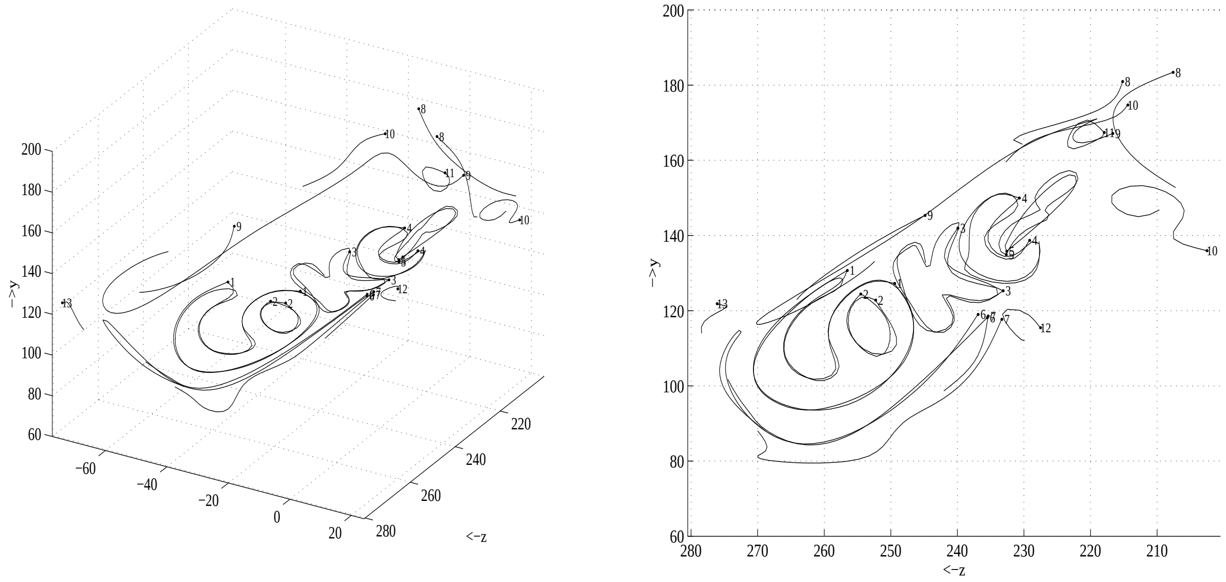


Figure 26: **All-reference-points** Global frame LMSq registration using ϵ-reciprocal points of the three best GSC matching curves. Only "C", "o", and "e" curves were used to register the frames.

6 Conclusion

We have shown that simple semidifferential invariants can successfully replace the traditional set of differential invariants, i.e. curvature and torsion. Hence, only the 1st derivative of the curves has to be evaluated instead of the 2nd and 3rd order derivatives. On the other hand, we have opened the problem of reference point selection which has been overcome by a brute force "try-all" like algorithm.

We have proposed a new curve similarity measure (PSC) based on ϵ-reciprocity. The PSC measure combined with the robust LMSq based motion estimation allow to solve one of the most important drawbacks of Besl's ICP algorithm - the inability to match two geometrical entities related by a partial overlap.

We have seen that almost perfect global frame registrations were obtained for candidate matching generation using One-reference-point and N-reference-points reference point selection strategies followed by LMSq motion estimation. It suggests that the LMSq algorithm can be a good solution to the global registration problem if majority of hypothetical matches is correct. The registration obtained in the case of All-reference-points selection strategy has been biased by a large number of outliers but it still provides a good stating point for the ICP algorithm or for the alternative approach given by Szeliski in [Sze88]. It has been shown how matching validation based on curve similarity measure can help to select correctly matched curves.

The most important problems which remain opened are finding good reference point selection strategies and effective registration under global constraints. Future work will extend the experiments to other sets of invariants and improve both visible surface reconstruction and curve reconstruction. Particularly, the fact that curves are contained on a surface might be useful.

The use of a one-shot method for the extraction of range data (i.e. based on a single projection pattern) in combination with comparably robust curve descriptions allow the propounded system to be used with dynamic and relatively uncontrolled scenes.

Acknowledgment

We would like to thank Radim Šára from the Czech Technical University, Prague for providing the general C list library that was used to implement geometrical hashing. Support by Esprit Basic Research Action 'VIVA' and the Belgian project IUAP-50 (Inter-Universitaire Attractie-Pool) on Robotics and Industrial Automation is gratefully acknowledged.

References

[AHB87] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares fitting of two 3-D point sets. *IEEE Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

[BC82] D.H. Ballard and Brown C.M. *Computer Vision.* Prentice Hall, Inc., Engelwood Cliffs, New Jersey 07632, 1982.

[Bes90] P.J. Besl. *The Free-Form Surface Matching Problem*, pages 25–71. Academic Press, Inc., 1990.

[BLP92] R. Bergevin, D. Larendeau, and D. Poussart. Estimating the rigid transformation between two range views of a complex object. In *11th IAPR International Conference on Pattern Recognition*, volume 1, pages 478–582, The Hague, The Netherlands, September 1992. IEEE Computer Society Press, Los Alamitos, CA.

[BM92] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[Boi84] J.D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transaction on Graphics*, 3(4):266–286, October 1984.

[CG91] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape and design. *Computer Graphics*, 25(4):257–266, July 1991.

[CSYL88] B.K. Choi, H.Y. Shin, Y.I. Yoon, and J.W. Lee. Triangulation of scattered data in 3-D space. *Computer Aided Design*, 20(5):239–248, June 1988.

[CW94] F.S. Cohen and J.Y. Wang. Modeling image curves using invariant 3-d object curve models–a path to 3-D recognition and shape estimation from image contours. *IEEE Pattern Analysis and Machine Intelligence*, 16(1):1–23, January 1994.

[Dav79] L.S. Davis. Shape matching using relaxation techniques. *IEEE Pattern Analysis and Machine Intelligence*, 1(1):60–72, January 1979.

[de 78] C. de Boor. *A Practical Guide to Splines*. Springer, 1978.

[Del94] H. Delingette. Simplex meshes: a general representation for 3-D shape reconstruction. Research Report 2214, INRIA Sophia-Antipolis, France, March 1994. Available by ftp at ftp.zenon.fr.

[DFP85] L. De Floriani, B. Falcidieno, and C. Pienovi. Delaunay-based representation of surfaces defined over arbitrarily shaped domains. *Computer Vision, Graphics, and Image Processing*, 32:127–140, 1985.

[DH73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, 1973.

[DHI91] H. Delingette, M. Hebert, and K. Ikeuchi. Energy functions for regularization algorithms. In *SPIE: Geometric Methods in Computer Vision*, volume 1570, pages 104–115, 1991.

[DHI92] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–144, April 1992.

[DHI93] H. Delingette, M. Hebert, and K. Ikeuchi. A spherical representation of the recognition of curved objects. In *Fourth International Conference on Computer Vision*, pages 103–112. IEEE Computer Society Press, Los Alamitos, CA, May 1993.

[doC76] M.P. doCarmo. *Differential Geometry of Curves and Surfaces.* Engelwood Cliffs, New Jersey, 1976.

[Far90] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide.* Computer Science and Scientific computing. Accademic Press, Inc., 1990.

[FMN89] T. Fan, G. Medioni, and R. Nevatia. Recognizing 3-D objects using surface description. *IEEE Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, November 1989.

[For87] S.J. Fortune. A sweepline algotithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987. Software available on netlib2.cs.utk.edu.

[Fra82] R. Franke. Scattered data interpolation: Test of some methods. *Mathematics of Computation*, 38(157):181–200, January 1982.

[FRC92] H. Ferguson, A. Rockwood, and J. Cox. Topological design of sculptured surfaces. *Computer Graphics*, 26(2):149–156, July 1992.

[GA94] A. Gueziec and N. Ayache. Smoothing and matching of 3-D spatial curves. *International Journal of Computer Vision*, 12(1):79–104, 1994.

[Gue93] A. Gueziec. Large deformable splines, crest lines and matching. In *Fourth International Conference on Computer Vision*, pages 650–657, Berlin, May 1993. IEEE Computer Society Press, Los Alamitos, CA.

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[HH89] Charles Hansen and Thomas C. Handerson. CAGD-Based computer vision. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1181–93, November 1989.

[HHI93] K. Higuichi, M. Hebert, and K. Ikeuchi. Building 3-D models from unregistered range images. Research Report CMU-CS-93-215, Robotics Institute - Carnegie Mellon University, November 1993. Available by http at http://www.cs.cmu.edu.

[HPS94] V. Hlavac, T. Pajdla, and M. Sommer. Improvement of the curvature computation. In *12th International Conference on Pattern Recognition*, Jerulem, Israel, October 1994. IAPR.

[HS89] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, November 1989.

[Hut93] D.P. Huttenlocher. Recognition by alignment. In A.K. Jain and P.J. Flyn, editors, *Three Dimensional Object Recognition Systems*, volume 1 of *Advances in Image Communication*, pages 311–326. Elsvier Science Publishers B.V., 1993.

[Jar93] R. Jarvis. Range sensing for computer vision. In A.K. Jain and P.J. Flyn, editors, *Three Dimensional Object Recognition Systems*, volume 1 of *Advances in Image Communication*, pages 17–56. Elsvier Science Publishers B.V., 1993.

[Kan94] K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Pattern Analysis and Machine Intelligence*, 16(5):543–549, May 1994.

[KHW91] Eyal Kishon, Trevor Hastie, and Haim Wolfson. 3-D curve matching using splines. *Journal of Robotic Systems*, 8(6):723–743, 1991.

[LH89] Y. Lambdan and Wolfson H.J. On the error analysis of geometrical hashing. Technical Report 476, Robotics Research Laboratory, New York University, October 1989.

[Low87] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–295, 1987.

[LT90] P. Liahg and J.S. Todhunter. Representation and recognition of surface shapes in range images: A differential geometry approach. *CVGIP:IP*, 52:78–109, 1990.

[MA93] M. Maruyama and S. Abe. Range sensing by projecting multiple slits with random cuts. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):647–651, June 1993.

[MB93] M. Mirza and K. Boyer. Performance evaluation of a class of m-estimators for surface parameter estimation in noisy range data. *IEEE Transactions on Robotics and Automation*, 9(1):75–85, February 1993.

[MMR91] P. Meer, D. Mintz, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 64:59–70, 1991.

[MS92] H.P. Moreton and C.H. Sequin. Functional optimization for fair surface design. *Computer Graphics*, 26(2):167–176, July 1992.

[MS93] H.P. Moreton and C.H. Sequin. Scale-invariant minimum-cost curves: Fair and robust design implements. In R.J. Hubbold and R. Juan, editors, *EUROGRAPHICS'93*, volume 12, pages C–473–C–484, 1993.

[Paj94] T. Pajdla. Binary illumination coded range finder: Reimplementation and improvements. Personal notes, to appear as a technical report. Tomas.Pajdla@esat.kuleuven.ac.be, August 1994.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry - an Introduction*. Texts and Monograph in Computer Science. Springer-Verlag, New York, 1985.

[Rio84] M. Rioux. Laser range finder based on synchronzed scanners. *Applied Optics*, 23(21):3837–3844, November 1984.

[Rip90] S.D. Rippa. Minimal roughness property of the Dealunay triangulation. *Computer Aided Geometrical Design*, 7:489–497, 1990.

[RL87] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[RW94] P.J. Rousseeuw and J. Wagner. Robust regression with a distributed intercept using Least Median of Squares. *Computational Statistics & Data Analysis*, 17:65–75, 1994.

[SA91] B. Sabata and J.K. Aggarwal. Estimation of motion from pair of range images: A review. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(3):309–324, November 1991.

[SD92] R.L. Stevenson and E.J. Delp. Viewpoint invariant recovery of visual surfaces from sparse data. *IEEE Pattern Analysis and Machine Intelligence*, 14(9):897–909, September 1992.

[SH81] L.G. Shapiro and R.M. Haralick. Structural descriptions and inexact matching. *IEEE Pattern Analysis and Machine Intelligence*, 3(5):504–519, September 1981.

[SM93] F. Stein and G. Medioni. Structural indexing: Efficient three dimensional object recognition. In A.K. Jain and P.J. Flyn, editors, *Three Dimensional Object Recognition Systems*, volume 1 of *Advances in Image Communication*, pages 353–373. Elsvier Science Publishers B.V., 1993.

[SS87] J.T. Schwartz and M. Sharir. Identification of partially obscured objects in in two or three dimensions by matching of noisy characteristic curves. *The International Journal of Robotics Research*, 6:29–44, 1987.

[SS92] S. Sinha and B.G. Schunck. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Pattern Analysis and Machine Intelligence*, 14(1):37–55, January 1992.

[STT93] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Curvature and continuity control in particle-based surface model. In *SPIE: Geometric Methods in Computer Vision*, San Diego, CA, July 1993.

[Sze88] R. Szeliski. Estimating motion from sparse range data without correspondence. *IEEE-CH266-1/88*, pages 207–216, 1988.

[Ter88] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, July 1988.

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

[Vel93] R.C. Veltkamp. 3-D computational morphology. In R.J. Hubbold and R. Juan, editors, *EUROGRAPHICS '93*, volume 12, pages C–115–C–127, 1993.

[VGMEO94] L. Van Gool, T. Moons, Pauwels E., and A. Oosterlinck. Vision and lie's approach to invariance. ESAT MI2 Technical Report Nr. KUL/ESAT/MI2/9407, Katholieke Universiteit Leuven, 1994.

[VO90] P. Vuylsteke and A. Oosterlinck. Range image acquisition with single binary-encoded light pattern. *IEEE Pattern Analysis and Machine Intelligence*, 12(2):143–164, 1990.

[WS91] M.W. Walker and L. Shao. Estimating of 3-d location prameters using dual number quaternions. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(3):358–367, November 1991.

[WW92] W. Welch and A. Witkin. Variational surface modeling. *Computer Graphics*, 26(2):157–166, July 1992.

[WZX91] T. Wang, X. Zhuang, and X. Xing. Robust 3-d motion estimation without point correspondences. *International Journal of Robotics and automation*, 7(2):64–69, 1991.