Formulation of 3D and Projective Extensions of the Self Affine Feature Transform

Zoltan Prohaszka[1]

Department of Control Engineering and Information Technology,
Budapest University of Technology
prohaszka@iit.bme.hu

Abstract. The Self Affine Feature Transform (SAFT) can analyze both geometric and photographic 2D image details. It is capable to identify analytical curves and extract their parameters. This is achieved by investigating the invariance of the 2D image against planar affine transformations. This paper presents the spatial extension of this model, which can process vectorized or volumetric 3D information and detect analytical surfaces in space. Projective extensions are introduced also. The goal of the article is to show theoretical equivalence of the described planar and spatial methods. Only very brief test results are given.

1 Introduction

The Self Affine Feature Transform (SAFT) was introduced by Prohaszka in 2008 [1]. It can extract geometric information from 2D grayscale images invariantly from regular affine transformations. It can be used also to compress image detail information into a 54-dimensional feature vector. This enables it to be used for photograph registration also. It's purely analytical, continuous formulation enables a wide variety of post-processing algorithms to be used on the descriptor matrix [2], and easy implementation on GPUs. These properties make SAFT be different from standard feature descriptors, which are mainly defined by algorithms, often having branches and discrete tables. SAFT performs average in photograph registration but outstanding in geometric image analysis. SAFT relates to several moment extracting methods, for example RGB moments used in [3], however SAFT calculates moments of image gradients.

The SAFT method must be differentiated from the Self-Affine Mapping System (SAMS or SMS) [4, 5]., which uses self-similarity transformations of discrete, finite value, while SAFT uses continuous range of infinitessimal transformations.

This article extends the SAFT image descriptor into the 3rd dimension, yielding to SAFT-3D, which can analyze and register spatial data similarly to the planar SAFT descriptor. It is able to indicate geometric surfaces, and retrieve their parameters. 3D input data can be either volumetric (produced by medical diagnostic tools) or vectorized scanned 3D data. This kind of data is analyzed for the purpose of reverse engineering of shapes. The field of 3D reverse engineering has a multitude of algorithms to solve emerging task. They can be organized

into five main phases, according to [6]. Our method can be utilized during the last phase: Surface Fitting. The Translational Vector Field method described in [6] is equivalent to the 3D extension of the Harris corner detector, except normal vectors has to be used instead of gradients. As the matrix of the 3D-Harris corner detector is embedded in our formulations, the Translational Vector Field is a special case of our general algorithm. Our method can also be interpreted as the generalization of [7]. It analyzes the invariance of surfaces against euclidean transformations only (6 DoF), but neither describe planar, projective nor affine connections of the proposed method. However, these methods have been widely used and tested, and these tests can be considered as a proof that the formulation presented here works in the practice, at least some special case of it. The purpose of our article is to show the geometric identity of these methods, highlighting that any filtering, pre- and post-processing or interpretation working well in the planar case might be tested for volumetric data processing and vice versa.

This paper also describes the formulation of a projective planar extension, which is demonstrated briefly.

2 The SAFT Detector

This section briefly summarizes previous results about the SAFT detector, which are essential to understand this article. More details can be found in [1] and [8].

2.1 Notations and Symbols

$\otimes$ refers Kronecker product of matrices. Its precedence is between $*$ and $+$. Readers not familiar with $\otimes$ need only a very short reading about it. According to the notations of [8], subscript $_{\mathsf{H}}$ denotes homogeneous quantities. Projective equivalence of homogeneous representations is notated by $\cong$.

The standard, 2D SAFT descriptor will be referred in this paper as SAFT-2D also, when it is necessary to distinguish form the spatial extension. We keep the notations used in the basic SAFT articles [8], even the matrices of spatial extension will be notated with the same letters as those used on the plane. This should not be confusing, since there are no formulations using both versions. To help seeking matrix sizes, a quick variable index can be found in Table 1.Two columns are used to describe the sizes for the planar and spatial interpretations. Image and spatial density derivatives are located in vector $\mathbf{g}^{\mathsf{T}}(\mathbf{p}) = \nabla I(\mathbf{p})$. Despite gradients are contravariant quantities, $\mathbf{g}$ itself is a column vector to avoid confusion.

2.2 2D Affine Flows

Affine flows (see Figure 1 for planar examples) in any dimension can be described as local velocity $\mathbf{v}$ is a linear function of homogeneous position $\mathbf{p}_{\mathsf{H}}$:

$$\mathbf{v} = \mathbf{Q}\mathbf{p}_{\mathsf{H}} \ , \tag{1}$$

Table 1. Frequently Used Variables

	Size-2D	Size-3D	Description	Eq.
$\mathbf{A}$	4×4	9×9	Upper-left part of $\mathbf{M}$	(11)
$\mathbf{C}$	2×2	3×3	Harris detector's matrix, lower-right part of $\mathbf{M}$	(11)
$\mathbf{g}$	2×1	3×1	Gradient or surface-normal vector	
$\mathbf{L}$	2×6	3×12	Linear relation between local velocity and flow parameters	(3)
$\mathbf{M}$	6×6	12×12	SAFT matrix	(7)
$\mathbf{M}^*$	9×9	16×16	Self Projective matrix	(16)
n	1×1	1×1	Number of Dimensions, 2 or 3 here.	
$\mathbf{p}_\mathsf{H}$	3×1	4×1	Homogeneous position $[x\ y\ 1]^\mathsf{T}$ or $[x\ y\ z\ 1]^\mathsf{T}$	(1)
$\mathbf{Q}$	2×3	3×4	Affine flow matrix	(1)
$\mathbf{Q}_\mathsf{H}$	3×3	4×4	Homogeneous, or projective extension of $\mathbf{Q}$	(9)
$\mathbf{q}$	6×1	12×1	Affine flow parameters, elements of $\mathbf{Q}$	(2)
$\mathbf{v}$	2×1	3×1	Local flow velocity	(1)

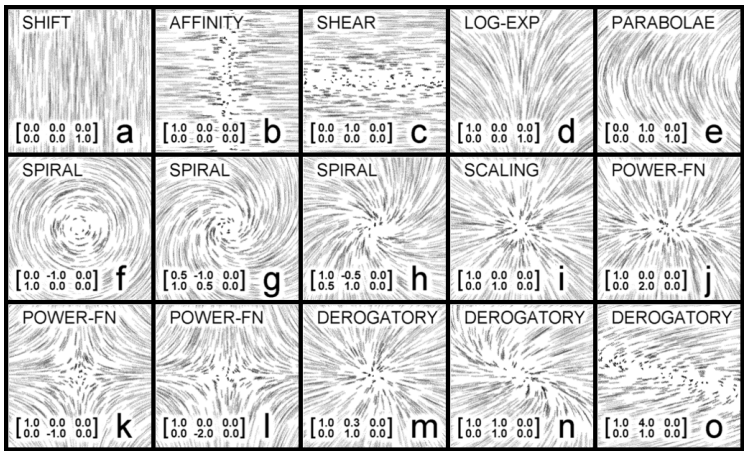

Fig. 1. Typical affine flows, and corresponding $\mathbf{Q}$ matrices.

where $\mathbf{Q}$ is a $n \times (n+1)$ matrix, and n is the dimensionality of the investigated space (2 or 3 in this article). The elements of $\mathbf{Q}$ are the linear parameters of the flow. They can be collected into the parameter vector $\mathbf{q}$ in column major order:

$$\mathbf{v} = \mathbf{L}_n(\mathbf{p}_\mathsf{H})\mathbf{q},\ \mathbf{L}_n(\mathbf{p}_\mathsf{H}) = \mathbf{p}_\mathsf{H}^\mathsf{T} \otimes \mathbf{I}_{n\times n}\ , \tag{2}$$

$$\mathbf{L}_2 = \begin{bmatrix} x\ 0\ y\ 0\ 1\ 0 \\ 0\ x\ 0\ y\ 0\ 1 \end{bmatrix}\ ,\quad \mathbf{L}_3 = \begin{bmatrix} x\cdot\mathbf{I}_{3\times3}\ y\cdot\mathbf{I}_{3\times3}\ z\cdot\mathbf{I}_{3\times3}\ \mathbf{I}_{3\times3} \end{bmatrix}\ . \tag{3}$$

The properties of affine flows are preserved during regular affine transformations.

Affine flows of small strength are identical to infinitesimal affine transformations.

2.3 Image Invariance against Affine Flows

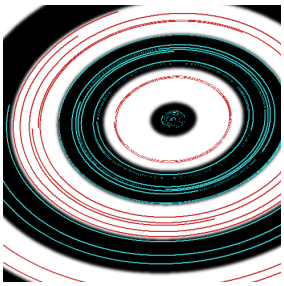

Fig. 2. Concentric ellipses and streamlines of their invariant flow.

The affine Lucas-Kanade (L-K) detector estimates the optimal affine transformation between two images I and I' [9]. If this affine registration succeeds, the dependence of the registration error against small disturbances of the optimal transformation will be the quadratic function of affine flow parameters. This quadratic function is hold by the local working variables of the L-K detector, and can be managed to be returned also. The investigated region is measured at sample locations which are indexed by $k \in \mathbb{N}$. Each measurement has a local error e_k, which is a linear function of flow parameters. The square of these errors are summed symbolically resulting the quadratic error function:

$$e_k = \mathbf{g}_k^\mathsf{T}\mathbf{v}_k - (I(\mathbf{p}_k) - I'(\mathbf{p}_k)),\ e^2(\mathbf{q}) = \sum_k e_k^2(\mathbf{q})\ , \tag{4}$$

$$e^2(\mathbf{q}) = \begin{bmatrix}\mathbf{q}^\mathsf{T}\ 1\end{bmatrix}\begin{bmatrix}\mathbf{M}\ \mathbf{n} \\ \mathbf{n}^\mathsf{T}\ o\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ 1\end{bmatrix},\ \nabla e^2 = 2\mathbf{q}^\mathsf{T}\mathbf{M} + 2\mathbf{n}^\mathsf{T} \Rightarrow \mathbf{q}_{opt} = -\mathbf{M}^{-1}\mathbf{n}\ , \tag{5}$$

$$e(\mathbf{q})^2 - e_{opt}^2 = \Delta\mathbf{q}^\mathsf{T}\mathbf{M}\Delta\mathbf{q},\ \Delta\mathbf{q} = \mathbf{q} - \mathbf{q}_{opt}. \tag{6}$$

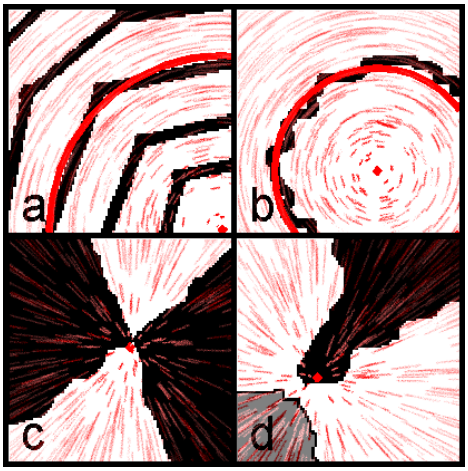

Fig. 3. SAFT can be used to find optimal center of rotation (a,b) or scaling (c,d) in closed form.

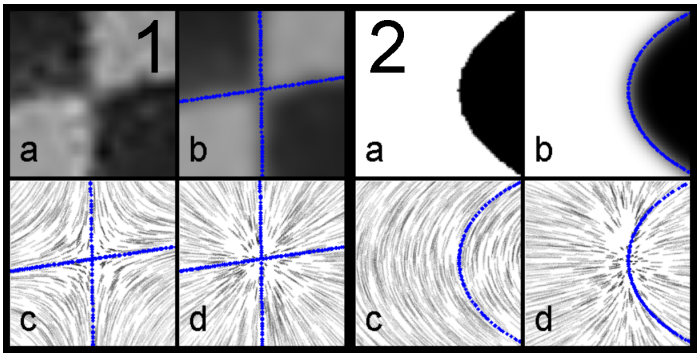

Fig. 4. Conics resulting rank 4 **M** matrices are precisely reconstructed.

When the two images are identical ($I = I'$), this quadratic function describes the sensitivity of this single image against infinitesimal affine transformations. $\mathbf{M}$, the quadratic matrix of this error function is called the Self-Affine Feature Transform (SAFT):

$$\mathbf{M} = \sum_k \mathbf{M}_k, \;\; \mathbf{M}_k = \mathbf{L}_k^\mathsf{T} \mathbf{g}_k \mathbf{g}_k^\mathsf{T} \mathbf{L}_k = \mathbf{p}_{\mathsf{H}k} \mathbf{p}_{\mathsf{H}k}^\mathsf{T} \otimes \mathbf{g}_k \mathbf{g}_k^\mathsf{T} \; . \tag{7}$$

Due to this summation, $\mathbf{M}$ preserves positive semi-definite and block-symmetric properties of $\mathbf{M}_k$, therefore it has 18 independent elements.

For example, images of concentric circles, polygon vertices and lines are invariant against rotation, scaling around the vertex and against translation respectively. See Figure 2. This can be expressed by algebra as the parameter vector of these flows is in the null-space of the corresponding $\mathbf{M}$ matrix. See [1] for more details, where several methods are given. Figure 3 demonstrates localization of mean circle on images and localization of scaling center.

Regular and degenerate conics can be identified in closed formulation, Figure 4 shows cases when $\mathrm{rank}(\mathbf{M}) = 4$.

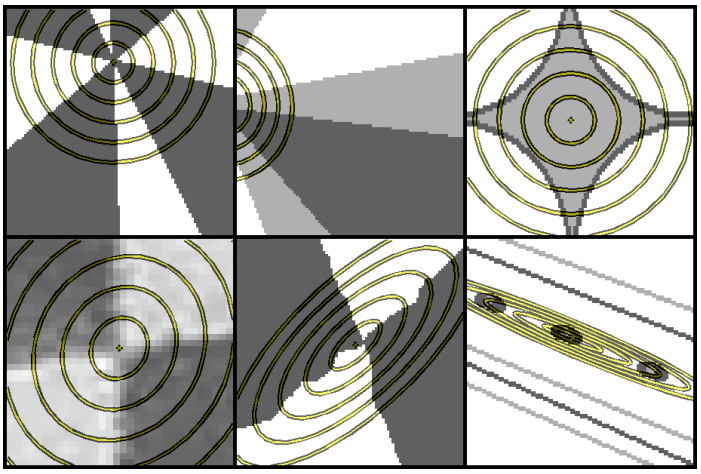

Fig. 5. The minimum of W_H identifies the accumulated fixpoint of invariant flows. Potential lines reflect it's uncertainty in different directions. 2D case is illustrated.

3 3D Extension

The spatial extension of the SAFT detector and it's formulation is a straightforward generalization of the planar case. Before giving definition of SAFT-3D, the properties of 3D affine flows need to be investigated.

3.1 3D Affine Flows

3D affine flows have 12 scalar parameters, which can be organized into the 4×3 matrix $\mathbf{Q}$:

$$\mathbf{v} = \mathbf{Q}\mathbf{p}_\mathsf{H} \ . \tag{8}$$

Examples to these flows are:

- (Elliptic) rotation around an axis.
- (Elliptic) rotation around and translation along an axis: screw flow.
- Any 2D affine flow in a 2D subspace and with
  - 0 velocity
  - constant velocity
  - proportional velocity (affinity)

  along the 3rd direction.

Similarly to the planar case [2], 3D flows can be normalized and classified by the eigen-decomposition of flow parameter matrix. It is convenient to use $\mathbf{Q}_\mathsf{H}$, the homogeneous extension of $\mathbf{Q}$, since an $(n+1) \times (n+1)$ matrix can be decomposed with classic methods.

$$\mathbf{Q}_\mathsf{H} = \begin{bmatrix} \mathbf{Q} \\ 0\ 0\ 0\ 0 \end{bmatrix}, \quad \mathbf{v}_\mathsf{H} = \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \mathbf{Q}_\mathsf{H}\mathbf{p}_\mathsf{H}, \quad \mathbf{Q}_\mathsf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1} \ . \tag{9}$$

It can be seen that λ_4, the eigenvalue corresponding to the direction $[0\ 0\ 0\ 1]^\mathsf{T}$, is always 0. $\mathbf{U}$ is the normalizing affine transformation, while $\mathbf{D}$ contains class-specific data, affine invariant informations of the flow. The enumeration of classes can be done by listing possible combinations of elementary divisors, together with the analysis whether a complex conjugate pair is, or is not present among the eigenvalues.

- Complex conjugate roots present:
  - One non-zero root.
  - Two zero roots, 2 cases: whether Jordan block present or absent.
- Only real roots:
  - Four different roots.
  - Three similar non-zero roots, 3 cases:
    - normal 3×3 block,
    - 3×3 Jordan block,
    - 2×2 Jordan block.
  - Three similar zero roots, 3 cases as above.
  - Two pairs of identical roots, 4 cases: whether Jordan block present or absent in each.
  - Two different and two identical roots, 4 cases: identical roots are zero or not, Jordan block present or absent.

These cases add up to 18 possible classes of 3D affine flows.

3.2 The SAFT-3D Descriptor

The formulation of SAFT-3D can be obtained by direct extension of the standard, 2D SAFT formulation on 3D affine flows. It is straightforward, only the size of the matrices (and vectors) differs from the planar formulation, but all equations hold. For convenience, we repeat (7):

$$\mathbf{M} = \sum_k \mathbf{M}_k,\ \mathbf{M}_k = \mathbf{L}_k^\mathsf{T}\mathbf{g}_k\mathbf{g}_k^\mathsf{T}\mathbf{L}_k = \mathbf{p}_{\mathsf{H}k}\mathbf{p}_{\mathsf{H}k}^\mathsf{T} \otimes \mathbf{g}_k\mathbf{g}_k^\mathsf{T}\ . \tag{10}$$

In 3D, $\mathbf{M}$ has 60 independent elements, namely 10 different 3×3 symmetric blocks with 6 independent elements in each. Similarly to 2D, $\mathbf{M}$ can be partitioned as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{bmatrix}\ , \tag{11}$$

where the 3×3 matrix $\mathbf{C} = \sum_k \mathbf{g}_k\mathbf{g}_k^\mathsf{T}$ is the matrix of the 3D Harris corner detector. The 9×9 matrix $\mathbf{A}$ describes the examined object's invariance against translation-free affine flows (flows leaving the origin in place).

The minimum of the quadratic function $\mathbf{p}_\mathsf{H}^\mathsf{T}\mathbf{W}_\mathsf{H}\mathbf{p}_\mathsf{H}$ is a reliable, robust representation of the cumulated fixed-point of the invariant flows, see [2] and Figure 5 for illustration the planar case. For n dimensions:

$$\mathbf{W}_{\mathsf{H}i,j} = \sum_{l=1}^{n} \mathbf{M}^\dagger_{(i-1)n+l,(j-1)n+l}\ ,\quad 1 <= i,j \in \mathbb{N} <= n+1\ , \tag{12}$$

where $\mathbf{M}^\dagger = \mathbf{U}\mathbf{D}^\dagger\mathbf{U}^\mathsf{T}$. $\mathbf{D}^\dagger$ can be calculated from the eigenvalues of $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ as $\lambda_i^\dagger = \lambda_{min}/\lambda_i$.

One might be interested about the invariance of analyzed 3D data against rigid affine flows (which has only rotational and translational components). These flows lie in a subspace of $\mathbf{q}$:

$$\mathbf{q} = \mathbf{T}_{rot}\mathbf{q}_{rot}\ ,\quad \mathbf{T}_{rot} = \begin{bmatrix} \mathbf{S}_x & \mathbf{S}_y & \mathbf{S}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3\times3} \end{bmatrix}^\mathsf{T}\ , \tag{13}$$

where $\mathbf{S}_x$ is the cross-product matrix of the first unit-vector $\hat{\mathbf{x}} = [1\ 0\ 0]^\mathsf{T}$, etc. Invariant rigid flows can be obtained by taking the null-space of the matrix $\mathbf{M}_{rot} = \mathbf{T}_{rot}^\mathsf{T}\mathbf{M}\mathbf{T}_{rot}$. This method is the 3D extension of the planar version described in eq. (27), section 7.4 of [2], which is demonstrated in Figure 3/a,b. The above formulation is also equivalent to the results presented in [7], where practical examples can be found which show the abilities of this method.

3.3 Application of SAFT-3D

It is out of the scope of the current article to give a detailed, exhaustive description of the problems that can be solved by SAFT-3D, together with numerical test results. Only a few problems are presented, and brief test results are given only for the most important application.

SAFT-3D can be either applied on vectorized or volumetric 3D data. For vectorized data, for example 3D scanned 2.5D 'landscape' surfaces or triangle grids, we can measure accurate position. Unit-length normals of faces are used instead of gradients. 'Grayscale' volumetric data can also be used, which are mainly collected during medical diagnostics and contain natural, non-geometric shapes. However, we expect that ribs, skulls, long bones, cylindrical and spherical joints can easily be identified by SAFT-3D, since their boundary has a density gradient above average, and their shape is somewhere between natural and geometric. Volumetric data of engineering objects can be considered 'binary' or having only distinct density values. For volumetric data, the dataset should be blurred before evaluating the gradient operator. Some inconvenience emerged during tests, which could be eliminated by measuring gradient strength and direction based on different filters, as [2] suggests.

SAFT-3D can be used to reverse-engineer datasets of the above types. For any interest region of different sizes, the SAFT-3D matrix has to be summed. This multi-summation can be sped up by extending the area-integral technique used in SURF [10] to 3D. After gaining $\mathbf{M}$ matrices, either the eigen-system has to be taken, or other post-processing techniques used for SAFT-2D can be applied after the straightforward modification of them. Geometric shapes yield rank-loss of the SAFT-3D matrix.

Definition 1. *Defect of a matrix: The difference between the number of elements in the diagonal of a matrix and the rank of the matrix (rank-loss) will be called as the defect of the matrix.*

Let us investigate the analysis of a torus. Only one defect will be observed, a rotation around the axis of the torus. Then, any plane can be selected, which contains the axis, or multiple planes of this type has to be averaged along rotational streamlines. The emerging 2D cross-section might be analyzed with SAFT-2D. we will see that the cross section is rotationally symmetric, since we analyze a torus.

The averaging and projecting to the actual hyperplane can be repeated again. If the torus is a toroidal tube or made of tubes with different densities, we can get the thickness of each component analyzing the resulting 1D averaged image, which was projected according to the streamlines of the single 2D invariant flow of rotation. This last step can be applied during standard, 2D SAFT analysis also, if the 2D SAFT matrix has one single defect.

As SAFT descriptors behave affine invariant, elliptically distorted multiple-walled toroidal tubes, or elliptical tubes bent to hyperbolic shape can be analyzed and parametrized efficiently with the above described method. Single or multiple cylindrical or cone surfaces are sharply identified by SAFT-3D.

It is clear, that decomposing a 12×12 matrix is a little bit slow, especially, if this has to be repeated more thousand times. There are much faster methods, which can retrieve the parameters of a cylindrical surface, for example. However, these methods can not handle general surfaces of translation or rotation, at least not uniformly together with simple spheres, cylinders, planes and quadric surfaces.

Extending the method in [7], region growing algorithms can try to link simple primitives together until the unified shape's SAFT-3D matrix still have some defects. When no compound shape can be grown to preserve any invariance, then the given area can be treated as a 3D primitive. These parts can be linked together into a tree describing the analyzed surface.

3.4 Enumeration of Basic Types

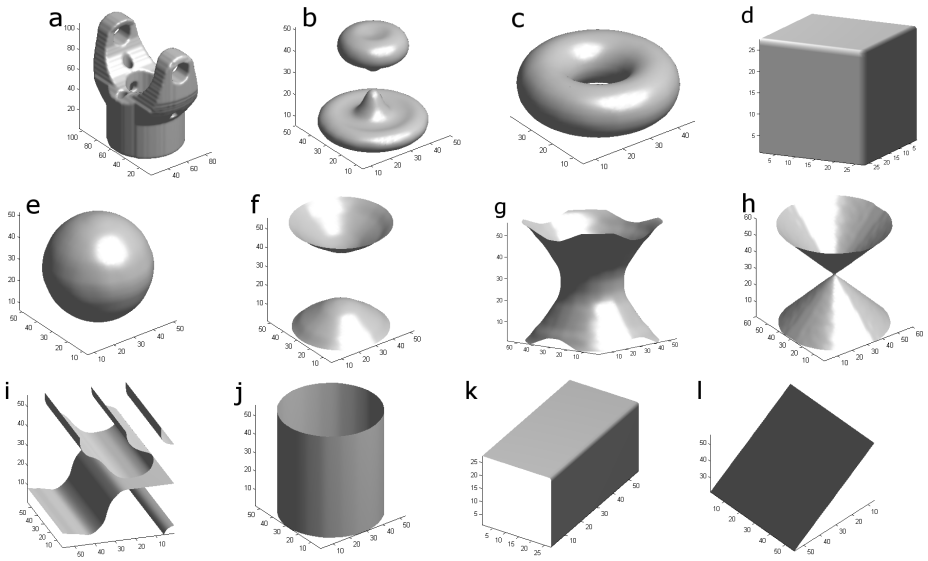

Fig. 6. Typical shapes which result different defects of $\mathbf{M}$ and $\mathbf{C}$.

To test basic properties of SAFT-3D, various volumetric objects were generated and tested. Figure 6 shows the boundary surfaces of these objects. A wider enumeration of typical surfaces follows below, which uses the following notation at the end of the description of each item: $\{\text{rank}(\mathbf{M}); \text{rank}(\mathbf{C}); \text{rank}(\mathbf{M}_{rot})\}$. Note that $\text{rank}(\mathbf{M}_{rot})$ is not affine-, but only similarity-invariant.

General objects do not have any singular flow, see Figure 6/a:$\{12; 3; 6\}$.

Examples to surfaces of single defects are:

- Elliptical revolution of a general curve, Figure 6/b,c:$\{11; 3; 5\}$.
- General curve projected from one point, $\{11; 3; 6\}$.

Examples to surfaces of multiple defects are:

- 3D corner or vertex: a triangle projected from one point, this is the vicinity of a 3-vertex of a polyhedron. It has 3 invariant flows, 3 (skew) affinities with respect to the vertex, along faces, Figure 6/d:$\{9; 3; 6\}$,

– Extruded curve: 4 invariant flows, each is parallel to the line of extrusion, but depends differently on $x, y, z, 1$, Figure 6/i:$\{8; 2; 5\}$.
– regular quadrics
  • ellipsoids: 3 defects: 3 elliptical rotations Figure 6/e:$\{9; 3; 3...6\}$
  • elliptical revolution of straight lines (hyperbolae Figure 6/f,g:$\{9; 3; 5\}$ and cones, Figure 6/h:$\{8; 3; 5\}$).
– Degenerate quadrics:
  • Skew prismatic surfaces of conic sections, they have generally 5 invariant flows: 4 flows parallel to the axis and a conic flow and one in the plane of the conic section, Figure 6/j:$\{7; 2; 4$ *or* $5\}$. A single extruded parabola, or extrusion of a broken line, or two intersecting (half-) lines (Figure 6/k:$\{6; 2; 5\}$) has 2 additional freedoms also, yielding a total 6.
  • Planar surface, it has 9 defects, 3 flows are fixed only: translation of the plane and tilting the plane. Figure 6/l:$\{3; 1; 3\}$

3.5 Test Results

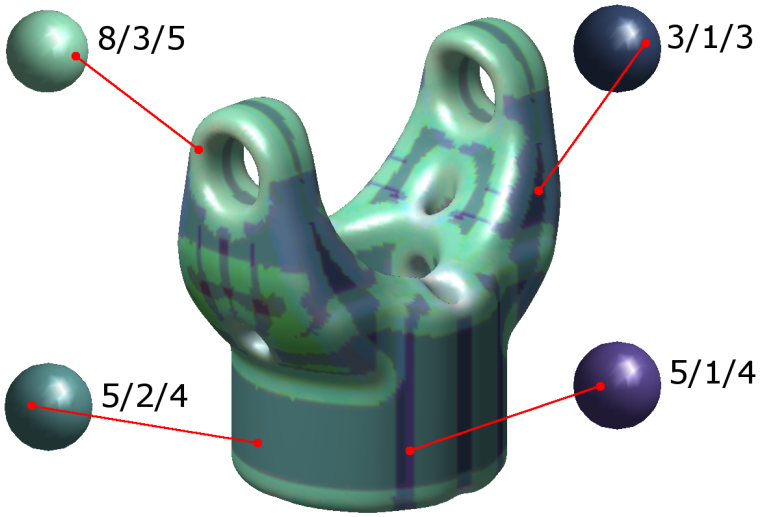

Fig. 7. Automatic detection of different types of surfaces. Virtual surface of the part of a universal joint was analyzed. Numbers next to color guide balls show rank of $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{M}_{rot}$ respectively. Red-Green-Blue colors are assigned also by these ranks. (Eigenvalues less than $0.001 \cdot E_{AC}$ are considered to be zero.)

The test performed to show basic abilities of SAFT-3D was intended to simulate reverse-engineering applications. The surface of a mechanical part (part

of a universal joint) was analyzed. The surface data was obtained from design data, not by 3D scanning. It consists of approximately 52000 vertices. SAFT-3D analysis was performed for the vicinity of each vertex. Such an analysis included 40 vertices in average. The color of the central vertex was assigned based on the rank of $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{M}_{rot}$ respectively, see Figure 7. During rank calculation, eigenvalues less than $0.001 \cdot E_{AC}$ were considered to be zero. The darker the surface is, the more invariant flow it has. The darkest patches are flat, which has a signature of 3/1/3. Differences of measured ranks from theoretical values (for example, cylindrical surfaces would yield 5/2/4 vs. 7/2/4) are due to the small size of the investigated volume (relative to the cylinder radius) and can change by rank-threshold parameter. It can be seen that surfaces of the same type can be effectively merged together.

4 Projective 2D Extension of SAFT

Since infinitesimal affine transformations generate affine flows, we can define projective flows on the unit sphere whose streamlines are obtained by repeated infinitesimal projective transformations. Projective flows obtained by this construction can be used to define the Self-Projective Feature (SPF), the projective extension of SAFT.

In projective geometry, we cannot simply express the velocity at a given point, but can express the direction of it:

$$\mathbf{v}_{\mathsf{H}} \cong \mathbf{Q}_{\mathsf{H}}\mathbf{p}_{\mathsf{H}} \ . \tag{14}$$

However, this relation is enough to trace a streamline on a unit (hemi-) sphere. It is easy to prove that the projection of streamlines of translation free 3D affine flows on the surface of the unit sphere gives the streamlines of projective flows.

Any image drawn on the surface of the unit (hemi-) sphere must be extruded along beams going through the origin. This 3D volumetric data can be analyzed (theoretically) by SAFT-3D. We are not interested in 3D translations, so only $\mathbf{M}^* = \mathbf{A}$, the upper-left 9×9 portion of $\mathbf{M}$ is interesting to us. It is trivial that the volume will be invariant against the scaling 3D flow $\mathbf{q}^* = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^{\mathsf{T}}$. The error function in the null-space of this flow parameter vector describes the Self-Projective nature of the image.

Alternatively, we can express the resistance of an image against projective flows:

$$e_k = \mathbf{g}_{\mathsf{H}k}^{\mathsf{T}}\mathbf{v}_{\mathsf{H}k} \ . \tag{15}$$

$$\mathbf{M}_k^* = \mathbf{p}_{\mathsf{H}k}\mathbf{p}_{\mathsf{H}k}^{\mathsf{T}} \otimes \mathbf{g}_{\mathsf{H}k}\mathbf{g}_{\mathsf{H}k} \ , \tag{16}$$

where $\mathbf{g}_{\mathsf{H}k}$ is perpendicular to $\mathbf{p}_{\mathsf{H}k}$, so it is calculated on the surface of the unit sphere, not on the plane of $z = 1$.

General perspective projections of affine flows yield projective flows:

$$\epsilon \cdot \mathbf{Q}_{\mathsf{H}projective} + \mathbf{I}_{3\times3} \cong \mathbf{H}_{pa}(\mathbf{I}_{3\times3} + \epsilon \cdot \mathbf{Q}_{\mathsf{H}affine})\mathbf{H}_{pa}^{-1}, \quad |\epsilon \in \mathbb{R}| << 1 \ , \tag{17}$$

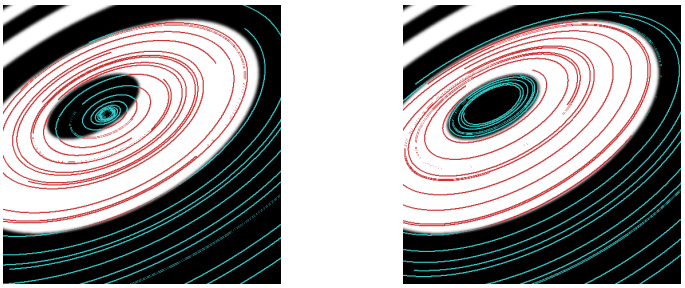

Fig. 8. Projection of concentric ellipses, best fitting invariant affine and invariant projective flow.

where $\mathbf{H}_{pa}$ is the affine-to-projective transformation ($\mathbf{p}_{\mathsf{H}p} \cong \mathbf{H}_{pa}\mathbf{p}_{\mathsf{H}a}$). This highlights a typical advantage of them over affine flows. If we photograph concentric circles with a highly perspective arrangement, the resulting image's edges fit to a projective flow, but not to any affine flow. Consider that the CoG of the resulting ellipses do not cover each other, as in Figure 8. The above statements hold for the perspective projections of other image details which are invariant to one or more affine flows.

During basic implementation, I found that results depend on whether the investigated detail is projected to the 'arctic' areas of the unit hemisphere, or nearly to the whole hemisphere. I achieved best results if the detail was scaled as if it projected to $\pm 45°$ from the z axis. The following Matlab code calculates $\mathbf{M}^*$:

```
%InG: n*m image giving gradient strength  (sharp)
%InD: n*m image giving gradient direction (blurred)
%T_pu: unit-to-pixel transformation
T_up=inv(T_pu);
M=zeros(9,9);
  for y=1:n-1
    for x=1:m-1
      pu=T_up*[x+0.5;y+0.5;1];
      pu_=pu/norm(pu);
%Matrix which flattens anything at pu:
      Flatten_u=eye(3)-pu_*pu_.';
      RG_=InG(y:y+1,x:x+1);
      RD_=InD(y:y+1,x:x+1);

%derivatives:
      dxG=[ 1 1]*(RG_)*[-1;1]*0.25;
      dyG=[-1 1]*(RG_)*[ 1;1]*0.25;
      dxD=[ 1 1]*(RD_)*[-1;1]*0.25;
      dyD=[-1 1]*(RD_)*[ 1;1]*0.25;
```

```
      gpG=[dxG,dyG,0];
      gpD=[dxD,dyD,0];
      lG=norm(gpG);
      lD=norm(gpD);
      if(lD*lG>0)

        gp=gpD/lD*lG;
        gu=gp*T_pu;%!!!! gradient is contravariant!
%it is transforming by inverse transformation
%with respect to locations.

%Projection on sphere:
        gu_=Flatten_u*gu.';
        M=M+kron(pu_*pu_.',gu_*gu_.');

      end
    end
  end
```

The projective 3D extension can be obtained by straightforward combination of Section 3.2 and Section 4. The **A** sub-matrix of the 4D extension has to be analyzed. However, Euclidean 3D transformations are likely to apply during 3D data acquisition, while 3D projective distortions do not occur. Beyond the mathematical interest, no practical importance is seen. Therefore, the detailed formulation is left to the reader interested in this topic.

5 Conclusions

We introduced SAFT-3D, the spatial extension of the planar SAFT descriptor in this paper. It can be used to process 3D data and detect surfaces, which are very common in engineering. Results analyzing typical surfaces were shown. Since the formulation of SAFT-3D is almost identical to the planar version, hence many post-processing algorithms of the planar versions can be applied with minor modification only. Pre-processing methods suggested to SAFT-2D are useful for SAFT-3D also. As main contribution, the article showed, that the discussed image and volume processing methods are geometrically and algebraically equivalent, thus any connecting method used in the planar case might be applied in 3D and vice versa.

The paper also formalizes the Self-Projective Feature, which can handle perspective projected image contents which are originally self-affine invariant.

Acknowledgements

The research in this article was supported by the Hungarian National Research Program grant No. OTKA K 71762 and. This work is connected to the scien-

tific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project, (Project ID: TAMOP-4.2.1/B-09/1/KMR-2010-0002).

References

1. Prohaszka, Z.: Affine invariant features from self-flow. In: RAAD 17th International Workshop on Robotics in Alpe-Adria-Danube Region. (2008) 1–10
2. Prohaszka, Z., Lantos, B.: Extracting geometric information from images with the novel self affine feature transform. Accepted to: Periodica Polytechnica, Electrical Engineering (2010)
3. Tuytelaars, T., Van Gool, L.: Wide baseline stereo matching based on local, affinely invariant regions. In: Proc. of BMVC-00. (2000) 412–425
4. Ida, T., Sambonsugi, Y.: Self-affine mapping system and its application to object contour extraction. IEEE Transactions on Image Processing **9** (2000) 1926–1936
5. Kawanaka, H., Kato, H., Matsubara, F., Iwahori, Y., Woodham, R.J.: Contour Based on Edge Background Subtraction for Self-affine Mapping System. Springer Berlin / Heidelberg (2007)
6. Várady, T., Facello, M.A., Terék, Z.: Automatic extraction of surface structures in digital shape reconstruction. Comput. Aided Des. **39** (2007) 379–388
7. Gelfand, N., Guibas, L.J.: Shape segmentation using local slippage analysis. In: SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, New York, NY, USA, ACM (2004) 214–223
8. Prohaszka, Z.: Matching image details with the self affine feature transform. In: V. Magyar Szamitogepes Grafika es Geometria Konferencia. (2010) 206–213
9. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Imaging Understanding Workshop. (1981) 121–130
10. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Computer Vision and Image Understanding (CVIU) **110** (2008) 346–359