periodica polytechnica

Electrical Engineering
53/3-4 (2009) 163–178
doi: 10.3311/pp.ee.2009-3-4.08
web: http://www.pp.bme.hu/ee
© Periodica Polytechnica 2009

RESEARCH ARTICLE

Extracting geometric information from images with the novel Self Affine Feature Transform

Zoltán Prohászka / *Béla* Lantos

Received 2009-12-01, accepted 2010-02-10

Abstract

Based on our research, the Self Affine Feature Transform (SAFT) was introduced as it extracts quantities which hold information of the edges in the investigated image region. This paper gives details on algorithms which extract various geometric information from the SAFT matrix. As different image types should be analysed differently, a classification procedure must be performed first. The main contribution of this paper is to describe this classification in details. Information extraction is applied for solving different 2-dimensional image processing tasks, amongst them the detection of convergent lines, circles, ellipses, parabolae and hiperbolae or localizing corners of calibration grids in a robust and accurate manner.

Keywords

SAFT · feature extraction · affine invariance · optical flow · classification

Acknowledgement

The research in this article was supported by the Hungarian National Research Program under grant No. OTKA K 71762.

Zoltán Prohászka
Department of Control Engineering and Information Technology, BME, H-1117 Budapest Magyar Tudósok krt. 2., Hungary
e-mail: prohaszka@iit.bme.hu

Béla Lantos
Department of Control Engineering and Information Technology, BME, H-1117 Budapest Magyar Tudósok krt. 2., Hungary
e-mail: lantos@iit.bme.hu

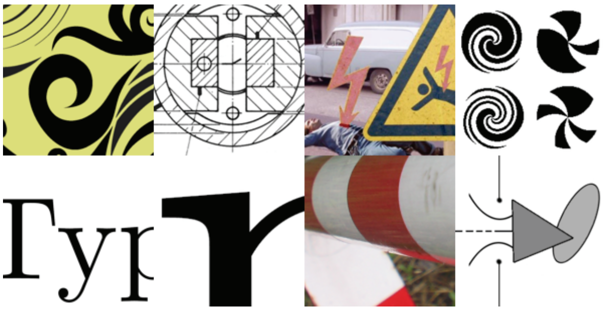

Fig. 1. Typical input images.

1 Introduction

This article investigates the operation of the Self Affine Feature Transform (SAFT) on geometric images. It focuses on the extraction of exact geometric quantities rather than analysing how similar details of digital photographs are handled. It is assumed, that the input image is geometric, it has a content which can be described as colored areas bounded by (analytical) curves. Such images are: drawings, blue-prints, signs, markers, typographic letters, high resolution vector graphics, or photographs of geometric scenes, etc., see Fig. 1.

SAFT was introduced in [13]. It was proposed, that SAFT can classify certain image types, and then analyse each type in a corresponding way. The main contribution of this paper is to describe this classification procedure in details. The classification is done invariantly against affine transformations, but the sensitivity of this process against noise is increased in certain geometric situations. For each class, affine invariant quantities are extracted by the means of a class-specific normalisation.

The introduced methods are robust, although the sensitivity against geometric transformations and noise cannot be neglected.

The operation of the SAFT detector on photographs requires a different starting point. SAFT can be effectively used to match similar details in a transformation invariant way. This property is out of the scope of the current article, it is described in [14], where further comparison is made with the widely used detectors as SIFT [7], MSER [9], GLOH [12] and SURF [2].

1.1 Positioning SAFT in the Image Processing Era

The Self Affine Feature Detector shows an interesting view when compared to other IP methods. It can be positioned among Transformation Invariant Features, since it has many similar properties, thus it can be used to solve similar tasks. However, SAFT can be used to extract exact geometric information too, thus it can be positioned for example among classic corner detectors, etc. These abilities of SAFT are due to the fact, that the calculation of the feature is based on geometrical definitions and is done by the tools of calculus and linear algebra. In certain situations, it can also be used to substitute the Hough transform. Due to the above written, the SAFT cannot be positioned unambiguously among existing image processing algorithms, it has a lot of application areas.

1.2 Sections overview

Section 2 lists related image processing methods. Section 3 gives basic introduction to the emerging entities: affine flows, their invariant subspaces, the formulation of the affine LK detector, the formulation of SAFT is highlighted. The interaction of affine flows and the SAFT features with affine coordinate transformations are shown. A very important point of our contribution is the extraction of affine invariant properties from affine flows and creating the SAFT feature. This makes the introduced algorithms behave invariantly against affine coordinate transformations. The calculation of transformations which normalises flow shape or image content is the other pillar of our methods. Section 4 describes the above mentioned classification and normalisation of affine flows. Section 5 shows useful relations for velocity distribution of invariant flows. Section 6 describes the classification and normalisation of the SAFT matrix. Section 7 gives examples how the previously shown results can be applied for solving emerging problems in vision scenario.

2 Related work

Many existing techniques (for example Edge detectors, the standard and generalised Hough transform, interest point detectors, and feature descriptors) can solve only a subset of the image processing tasks that SAFT can solve, and vice versa. Many similarities can be found in the formulations how the usual methods and SAFT works. Each of these similarities gives us an other viewing point of the SAFT matrix **M**, and the information enclosed in it, see (10).

2.1 Relation to classic detectors

SAFT is related to classic corner detectors. The 2 × 2 symmetric matrix $\mathbf{C} = \mathbf{g}\mathbf{g}^\mathsf{T}$ ($\mathbf{g}$ is the image gradient) used by the Harris [5] and by the Shi and Tomasi [16] corner detectors also appears in the 6 × 6 SAFT matrix **M**, see (10). This will be explained later in sec.3.5.1.

There is also a similarity in the formulation of SAFT and affine shape adaptation [1], [11]. SAFT contains more information and changes during affine transformations in a more complex way. The philosophy of affine shape adaptation can be combined with SAFT, but one has to take into account that the computational requirement of SAFT is higher than that of simpler methods, thus considerable less iterative steps can be afforded.

Different versions of the Hough transform can detect lines [10], vanishing points [18], circles and ellipses by using original 2-D, cascade-cubic, 3-D and 5-D parameterizing respectively. 5-D parameter space is too large for practical applications.

SAFT is able to detect and extract the parameters of these geometric objects, but only if the investigated region contains only one such an object. SAFT is sensitive to the presence of additional edges when solving these tasks, but have far less computational complexity than the Hough transform.

2.2 Transformation invariant features and SAFT

Transformation invariant feature descriptors are also related to SAFT. We have to highlight SIFT [7], MSER [9], GLOH [12] and SURF [2]. They are used rather to match details of photographs via the comparison of the distance of feature vectors, than extracting crisp geometric information from drawings. SAFT can be used for this purpose also, it is described in [14] together with a longer comparison to the above methods.

SAFT is also related to Affine Moment Invariants [4], [15] and to Color Image Moments [17]. The highest order moment utilized in SAFT is the 2$^{\text{nd}}$, while the others use higher degree moments also. Affine Moment Invariants use only moments calculated over the intensity image, as Color Image Moments over the multichannel image:

$$\mathbf{M}_{ij}^{klm} = \int x^i y^j R^k G^l B^m \,\mathrm{d}A,$$

where $R(x, y)$ is the red channel of the image, for example. In contrast, SAFT operates on the gradient images, see (15).

2.3 SAFT and the Lucas-Kanade detector

Relation of SAFT to the Lucas-Kanade (LK) detector is essential, since both the formulation of SAFT and its basic idea descends from the affine LK detector. Already the first paper of Lucas and Kanade [8] have suggested the affine extension of their method, to determine the infinitesimal affine transformation between two similar portion of images.

It is infrequently described or applied in related papers and implementations, that the real, reliable output of any LK type detector is not an n-dimensional parameter vector, rather the output consists of multiple linear constraints against the parameter vector, see sec.3.3.1 and [13]. This problem generally affects computer vision implementations. Most algorithms lack the ability to handle the reliability/uncertainty of information propagated between different stages of the image processing pipeline. This results that such an implementation cannot detect at all, if the input contains a singular set-up. The basic idea of SAFT is the recognition that the shape of the error function is important, and the locus of its minimum is not enough to be known.

Nomenclature

To distinguish from scalar variables, bold lower-case letters are used for vectors ($\mathbf{v}$) and bold capitals for matrices ($\mathbf{M}$). Vectors are column vectors by default, row vectors appear as transposed columns. Kronecker product is notated by $\otimes$. Homogeneous quantities are denoted by subscript $_\mathsf{H}$, to aid differntiating between homogeneous and standard representations. $\mathbf{q} \bowtie \mathbf{T}_\mathsf{H}$ denotes that $\mathbf{q}$ is invariant under transformation $\mathbf{T}_\mathsf{H}$. C_α and S_α will refer $\cos(\alpha)$ and $\sin(\alpha)$ respectively, where α is an arbitrary rotation or angle.

Let us consider the function $\mathbf{f}(\mathbf{x})$ over the domain S, $\mathbf{x} \in S$ (both $\mathbf{f}$ and $\mathbf{x}$ can be column vectors or scalars). The term $\mathbf{I}$ *is the homogeneous* 2^{nd} *range-, domain- or range-domain-moment of* $\mathbf{f}$ is defined by:

$$\mathbf{I} = \int \begin{bmatrix} \mathbf{v}(\mathbf{x}) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}(\mathbf{x})^\mathsf{T} & 1 \end{bmatrix} \otimes g(\mathbf{x})\, \mathrm{d}S,$$

where $\{\mathbf{v}(\mathbf{x}); g(\mathbf{x})\}$ can be one of the following: $\{[\mathbf{f}(\mathbf{x})]; 1\}$, $\{[\mathbf{x}]; \mathbf{f}(\mathbf{x})\}$ or $\{[\mathbf{f}(x)^\mathsf{T}\mathbf{x}^\mathsf{T}]^\mathsf{T}; 1\}$ respectively. Notice that the Kronecker product ($\otimes$) usually simplifies to multiplication by scalar, except for domain-moments if $\mathbf{f}$ is vector valued. Range-moments represent function value histograms, while domain-moments represent the distribution of functions on the input space.

The image $I(x, y)$ is integrated according to the weighting function $w_A(x, y)$ on the domain A. Image derivatives are in vector $\mathbf{g}(x, y)$. The : (colon) operator appearing in matrix subscripts denotes MATLAB®, OCTAVE® style multiple indexing.

There is a quick reference in the appendix about the frequently used variables.

3 Basic relations of the SAFT detector

SAFT was introduced by describing the invariance of an image to different affine flows [13]. This invariance can be computed by the affine LK detector. This section investigates the properties of affine flows and transformations.

3.1 Affine flows: definition

The basic formulation of the SAFT feature comes from an image's invariance against infinitesimal affine transformations. *Infinitesimal affine transformations will be referred as affine flows, which satisfy that the local velocity depends linearly both on homogeneous position and on flow parameters:*

$$\mathbf{v} = \mathbf{Q} \cdot \mathbf{p}_\mathsf{H} = [{\mathbf{p}_\mathsf{H}}^\mathsf{T} \otimes \mathbf{I}_{2\times 2}] \cdot \mathbf{q} \tag{1}$$

where $\mathbf{v}$ is 2×1 column vector of local velocity, $\mathbf{Q}$ is 6 DoF 2×3 parameter matrix, $\mathbf{p}_\mathsf{H} = [x \;\; y \;\; 1]^\mathsf{T}$ is 3×1 homogeneous position and $\mathbf{q}$ contains the elements of $\mathbf{Q}$ in column-major order, thus $\mathbf{q}$ is the parameter vector. It is possible to give an alternative definition:

$$\mathbf{v} = [\mathbf{I}_{2\times 2} \otimes \mathbf{p}_\mathsf{H}^\mathsf{T}] \cdot \hat{\mathbf{q}}, \tag{2}$$

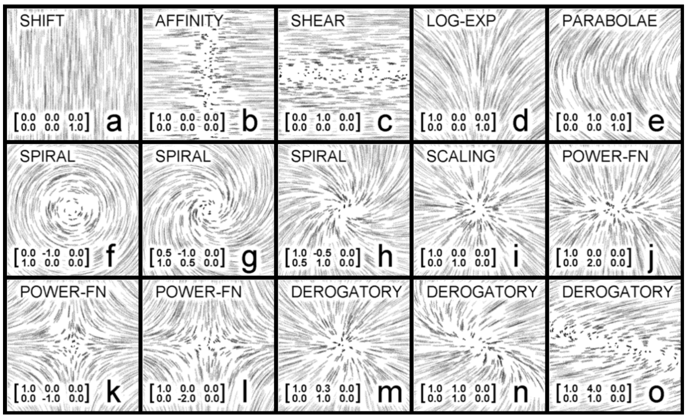
SHIFT
$\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$ a
AFFINITY
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$ b
SHEAR
$\begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}$ c
LOG-EXP
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$ d
PARABOLAE
$\begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$ e
SPIRAL
$\begin{bmatrix} 0.0 & -1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$ f
SPIRAL
$\begin{bmatrix} 0.5 & -1.0 & 0.0 \\ 1.0 & 0.5 & 0.0 \end{bmatrix}$ g
SPIRAL
$\begin{bmatrix} 1.0 & -0.5 & 0.0 \\ 0.5 & 1.0 & 0.0 \end{bmatrix}$ h
SCALING
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$ i
POWER-FN
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 \end{bmatrix}$ j
POWER-FN
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 0.0 \end{bmatrix}$ k
POWER-FN
$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -2.0 & 0.0 \end{bmatrix}$ l
DEROGATORY
$\begin{bmatrix} 1.0 & 0.3 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$ m
DEROGATORY
$\begin{bmatrix} 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$ n
DEROGATORY
$\begin{bmatrix} 1.0 & 4.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$ o

Fig. 2. Streamlines of normalised affine flows, flow names and $\mathbf{Q}$ matrices

with reversed order of Kronecker product, where $\hat{\mathbf{q}}$ contains the element of $\mathbf{Q}$ in row-major order. All matrices descending from this alternative definition will be notated by hats ($\hat{\ }$). $\mathbf{Q}$ will be decomposed as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{F} & \mathbf{t} \end{bmatrix} = \begin{bmatrix} s_x & r_x & t_x \\ r_y & s_y & t_y \end{bmatrix}, \;\; \mathbf{q} = \begin{bmatrix} \mathbf{f} \\ \mathbf{t} \end{bmatrix}, \;\; \mathbf{Q}_\mathsf{H} = \begin{bmatrix} & \mathbf{Q} & \\ 0 & 0 & 0 \end{bmatrix}. \tag{3}$$

The elements of $\mathbf{F}$ is collected to $\mathbf{f}$ in column-major order ($\mathbf{f} = [s_x \;\; r_y \;\; r_x \;\; s_y]^\mathsf{T}$).

Affine flows can be characterized by the streamlines of the flows. These streamlines are conic sections (if $\text{trace}(\mathbf{F}) = 0$), but exponential spirals or power functions can also occur. Fig. 2 shows examples for affine flows.

These streamlines characterise the flow better (not regarding flow strength) than algebraic formulations, since streamlines do not need a coordinate frame to be chosen.

3.1.1 Coordinate transformations on affine flows

We can describe the same flow in coordinate frames A and B, which have the relation:

$$\mathbf{p}_{\mathsf{H}B} = \mathbf{T}_{\mathsf{H}BA}\mathbf{p}_{\mathsf{H}A}, \;\; \mathbf{T}_{\mathsf{H}BA} = \begin{bmatrix} \mathbf{R}_{BA} & -\mathbf{c}_{BA} \\ 0 \;\; 0 & 1 \end{bmatrix} \tag{4}$$

where $\mathbf{R}_{BA}$ is regular, but not necessarily orthogonal. The transformation rules for $\mathbf{Q}$ and $\mathbf{q}$, $\hat{\mathbf{q}}$ are

$$\begin{aligned} \mathbf{v}_A &= \mathbf{Q}_A\mathbf{p}_{\mathsf{H}A}, \mathbf{v}_B = \mathbf{Q}_B\mathbf{p}_{\mathsf{H}B} \\ \mathbf{v}_B &= \mathbf{R}_{BA}\mathbf{v}_A, \mathbf{v}_A = \mathbf{R}^{-1}_{BA}\mathbf{v}_B \\ \mathbf{v}_A &= \mathbf{R}^{-1}_{BA}\mathbf{Q}_B\mathbf{T}_{\mathsf{H}BA}\mathbf{p}_{\mathsf{H}A} \\ \mathbf{Q}_A &= \mathbf{R}^{-1}_{BA}\mathbf{Q}_B\mathbf{T}_{\mathsf{H}BA} \\ \mathbf{Q}_B &= \mathbf{R}_{BA}\mathbf{Q}_A\mathbf{T}^{-1}_{\mathsf{H}BA}. \end{aligned} \tag{5}$$

From (1):

$$\mathbf{v}_B = \mathbf{R}_{BA}\mathbf{v}_A = (1 \otimes \mathbf{R}_{BA})((\mathbf{p}^\mathsf{T}_{\mathsf{H}B}\mathbf{T}^{-\mathsf{T}}_{\mathsf{H}BA}) \otimes \mathbf{I}_{2\times 2})\mathbf{q}_A$$

$$\mathbf{v}_B = ((\mathbf{p}^\mathsf{T}_{\mathsf{H}B}\mathbf{T}^{-\mathsf{T}}_{\mathsf{H}BA}) \otimes \mathbf{R}_{BA})\mathbf{q}_A$$

$$\mathbf{v}_B = (\mathbf{p}^\mathsf{T}_{\mathsf{H}B} \otimes \mathbf{I}_{2\times 2})(\mathbf{T}^{-\mathsf{T}}_{\mathsf{H}BA} \otimes \mathbf{R}_{BA})\mathbf{q}_A$$

Using (1) again:

$$\mathbf{v}_B = (\mathbf{p}_{\mathsf{H}}{}_B^{\mathsf{T}} \otimes \mathbf{I}_{2\times 2})\mathbf{q}_B, \quad \mathbf{q}_B = \mathbf{S}_{BA}\mathbf{q}_A$$

$$\mathbf{S}_{BA} = \mathbf{T}_{\mathsf{H}}{}_{BA}^{-\mathsf{T}} \otimes \mathbf{R}_{BA} \tag{6}$$

$$\mathbf{q}_A = \mathbf{S}_{AB}\mathbf{q}_B, \quad \mathbf{S}_{AB} = \mathbf{S}_{BA}^{-1}$$

$$\hat{\mathbf{S}}_{BA} = \mathbf{R}_{BA} \otimes \mathbf{T}_{\mathsf{H}}{}_{BA}^{-\mathsf{T}}, \hat{\mathbf{q}}_B = \hat{\mathbf{S}}_{BA}\hat{\mathbf{q}}_A \tag{7}$$

The transformation $\mathbf{S}$ (and $\hat{\mathbf{S}}$) are unitary, if and only if the affine transformation encodes rotation and scaling around the origin.

If $\mathbf{q} = c \cdot [1 \ \ 0 \ \ 0 \ \ 1 \ \ 0 \ \ 0]^{\mathsf{T}}$, where c is any non-zero real parameter, then any coordinate transformation that leaves the origin in place, does not change the parameters of this flow, except the magnitude of it. The coordinate transformation of pure scaling leaves any flow parameter vector with $\mathbf{t} = \mathbf{0}$ unchanged.

3.1.2 Flow invariance against global planar transformations

In this subsection we investigate, if the plane together with affine streamlines are distorted by affine transformations (for example axonometric photographing), then how the flow parameters response to it. The results presented here are rather geometric than coordinate-geometric relations, thus they also could be proved without using coordinates.

A trivial, but important property of affine flows is that the affine transformation of affine flow streamlines results in affine flows. The affine transformation of pure scaling leaves any flow with $\mathbf{t} = \mathbf{0}$ in place. As the streamlines of the flow move together with the image plane, the new parameters of the flow can be computed by (5), if the applied transformation is $\mathbf{T}_{\mathsf{H}}$.

It is trivial that if $\mathbf{Q}$ is invariant to transformations of $\mathbf{T}_{\mathsf{H}}$ ($\mathbf{Q} \bowtie \mathbf{T}_{\mathsf{H}}$), then $c_1\mathbf{Q} \bowtie (\mathbf{T}_{\mathsf{H}})^{c_2}$ for any real (non-zero) $c_1, c_2 \in \mathbb{R}$. This means, that the 'magnitude' of $\mathbf{Q}$ and $\mathbf{T}_{\mathsf{H}}$ does not play role in flow invariance. We do not prove here the following duality conjuncture: If $\mathbf{Q}_1 \bowtie (\mathbf{Q}_{2\mathsf{H}} + \mathbf{I})$ then $\mathbf{Q}_2 \bowtie (\mathbf{Q}_{1\mathsf{H}} + \mathbf{I})$. It is obvious, that all affine flows are invariant against the transformation which represents itself. Applying transformation rules:

$$\mathbf{p}_{\mathsf{H}}{}' = (\mathbf{I}_{3\times 3} + c\mathbf{Q}_{\mathsf{H}})\, \mathbf{p}_{\mathsf{H}}, c \in \mathbb{R}$$

$$\mathbf{v}' = (\mathbf{I} + c\mathbf{F})\mathbf{v} = (\mathbf{Q} + c\mathbf{F}\mathbf{Q})\mathbf{p}_{\mathsf{H}} = \mathbf{Q}(\mathbf{I}_{3\times 3} + c\mathbf{Q}_{\mathsf{H}})\, \mathbf{p}_{\mathsf{H}} \Rightarrow$$

$$\mathbf{v}' = \mathbf{Q}\mathbf{p}_{\mathsf{H}}{}' \Rightarrow \mathbf{Q}' = \mathbf{Q}$$

3.2 Invariant subspaces of affine flows

It can be recognised, that many affine flows contain places with zero flow velocity, so we might investigate this numerically by solving the following equation:

$$\mathbf{0} = \mathbf{v} = \mathbf{Q}\mathbf{p}_{\mathsf{H}}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{F} & \mathbf{t} \end{bmatrix}, \mathbf{p}_{\mathsf{H}} = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

$$\mathbf{p} = -\mathbf{F}^{-1}\, \mathbf{t} \tag{8}$$

Applying the properties of the solutions of linear equations we get:

- If $\mathbf{F}$ is regular (nonsingular) then the invariant region is a point, called the fixed-point. See Fig. 2/f...o.
- If $\mathbf{F}$ has a defect of 1 and t is in the range of $\mathbf{F}$ then the invariant region is a straight line. See Fig. 2/b,c.
- If $\mathbf{F}$ has a defect of 1 and t is not in the range of $\mathbf{F}$ then there is no invariant region. See Fig. 2/d,e.
- If $\mathbf{F} = \mathbf{0}$ and $\mathbf{t} = \mathbf{0}$, then the invariant region is the entire plane, but there is no flow at all (zero flow).
- if $\mathbf{F} = \mathbf{0}$ and $t \neq \mathbf{0}$, there is no fixed-point or invariant region, since the flow is uniform shift. See Fig. 2/a.

3.3 Algebraic Background of the SAFT Descriptor

The formulation of SAFT originates from the affine extension of the LK detector, thus we describe the essential equations of the LK detector (with the notations of this paper) and then give the definition of SAFT.

3.3.1 Formulation of the Lucas-Kanade detector

The generalised LK detector can determine the linear parameters of the optical flow between two images. The gradients and difference of these images are used in the calculations. The detector integrates quadratic error functions of flow parameters arising from the squared equation error of linear constraints against local flow velocities. Thus, the affine extension determines a quadratic cost function on the parameter vector $\mathbf{q}$. The primary output of any LK detector is the homogeneous quadratic form of the reprojection error's dependence from flow parameters.

Usually, the extensions of the LK detector assume that $\mathbf{v}(\mathbf{p})$, the local flow velocity depends on planar position $\mathbf{p}$ and linearly on flow parameters $\mathbf{q}$.

$$\mathbf{v}(\mathbf{p}) = \mathbf{L}(\mathbf{p})\mathbf{q} \tag{9}$$

In the standard detector $\mathbf{L}_{STD}(p) = \mathbf{I}_{2\times 2}$, $\mathbf{q}_{STD} = [t_x \ \ t_y]^{\mathsf{T}}$. The affine extension can use for example $\mathbf{L} = \mathbf{p}_{\mathsf{H}}{}^{\mathsf{T}} \otimes \mathbf{I}_{2\times 2}$ and $\mathbf{q} = [s_x \ \ r_y \ \ r_x \ \ s_y \ \ t_x \ \ t_y]^{\mathsf{T}}$. The LK detector integrates squared reprojection error at image points:

$$e^2 = \int (\mathbf{v}(\mathbf{p})^{\mathsf{T}}\mathbf{g}(\mathbf{p}) - \Delta I(\mathbf{p}))^2 w(\mathbf{p})\, \mathrm{d}A$$

where $\Delta I(\mathbf{p})$ is the difference between the two images at point $\mathbf{p}$, $\mathbf{g} = \nabla I(\mathbf{p})$ is the (averaged) gradient of the image(s), and $w(\mathbf{p})$ is the applied windowing function. This results

$$e^2 = [\mathbf{q}^{\mathsf{T}} 1] \begin{bmatrix} \mathbf{M} & \mathbf{n} \\ \mathbf{n}^{\mathsf{T}} & h \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{M} & \mathbf{n} \\ \mathbf{n}^{\mathsf{T}} & h \end{bmatrix} = \int \begin{bmatrix} \mathbf{L}(\mathbf{p})^{\mathsf{T}}\mathbf{g}(\mathbf{p}) \\ -\Delta I(\mathbf{p}) \end{bmatrix} \begin{bmatrix} \mathbf{g}(\mathbf{p})^{\mathsf{T}}\mathbf{L}(\mathbf{p}) & -\Delta I(\mathbf{p}) \end{bmatrix} w\, \mathrm{d}A.$$

Minimising this error leads the relation $\mathbf{q}_{opt} = \arg\min(e^2(\mathbf{q}))$, $\mathbf{q}_{opt} = -\mathbf{M}^{-1}\mathbf{n}$, but the full information extracted from the images is carried by $\mathbf{M}$ and $\mathbf{n}$. Usually only the solution of the above equation is returned by the LK implementations. As described above, we are interested in the linear constraints against this solution, so we will use the full information encapsulated in matrix $\mathbf{M}$.

3.3.2 Calculating the SAFT descriptor

The LK detector assumes that two, slightly different image will be analyzed to determine the optimal optical flow between them. However, all equations and assumptions remain valid if we feed the same picture to the detector instead of two different images. Why would one do so, as it is obvious that the optimal solution is the zero flow, as $\mathbf{n} = \mathbf{0} \Rightarrow \mathbf{q}_{opt} = \mathbf{0}$ in this case? The reason is, that we want to investigate, how the reprojection error depends if we disturb flow parameters around the optimum $\mathbf{q}_{opt} = \mathbf{0}$.

$$\mathbf{M} = \int (\mathbf{p}_\mathsf{H} \otimes \mathbf{g})(\mathbf{p}_\mathsf{H} \otimes \mathbf{g})^\mathsf{T} w \, \mathrm{d}A \tag{10}$$

$$\hat{\mathbf{M}} = \int (\mathbf{g} \otimes \mathbf{p}_\mathsf{H})(\mathbf{g} \otimes \mathbf{p}_\mathsf{H})^\mathsf{T} w \, \mathrm{d}A$$

The matrix $\mathbf{M}$ (and $\hat{\mathbf{M}}$) will be block symmetric with 18 independent elements and positive semi-definite. The squared total error can be expressed as:

$$e^2 = \int e(\mathbf{p})^2 w \, \mathrm{d}A = \mathbf{q}^\mathsf{T}\mathbf{M}\mathbf{q} = \hat{\mathbf{q}}^\mathsf{T}\hat{\mathbf{M}}\hat{\mathbf{q}}.$$

3.3.3 Coordinate transformations

The behavior of the algorithm depends on the chosen coordinate frame. The dependence is the following:

$$\mathbf{M}_B = \mathbf{S}_{AB}^\mathsf{T}\mathbf{M}_A\mathbf{S}_{AB}, \;\; \hat{\mathbf{M}}_B = \hat{\mathbf{S}}_{AB}^\mathsf{T}\hat{\mathbf{M}}_A\hat{\mathbf{S}}_{AB} \tag{11}$$

where $\mathbf{S}_{AB}$ and $\hat{\mathbf{S}}_{AB}$ are the same as in (6) and (7).

3.4 Normalising Flow Strength

In order to compare the resultant error for affine transformations of the image, a flow strength measure has to be defined.

Having defined the measure the question arises: 'Which are the affine flows, which deform the image minimally?' Obviously, if the flow strength is halved, the resultant squared error is quartered, so we have to append: '...among affine flows with normalised strength.'

$$\text{minimize } \mathbf{q}^\mathsf{T}\mathbf{M}\mathbf{q}, \text{ such that } \mathbf{q}^\mathsf{T}\Sigma\mathbf{q} = 1.$$

where the 6×6 symmetric matrix Σ describes quadratic flow strength measure. A good selection for Σ is to ensure uniform flow energy in the window of interest. The flow velocity's range-moment can be calculated as

$$\tilde{\mathbf{W}} = \int \mathbf{v}\mathbf{v}^\mathsf{T} w \, \mathrm{d}A = \int \mathbf{Q}\mathbf{p}_\mathsf{H}\mathbf{p}_\mathsf{H}{}^\mathsf{T}\mathbf{Q}^\mathsf{T} w \, \mathrm{d}A \tag{12}$$

$$\tilde{\mathbf{W}} = \mathbf{Q} \int \mathbf{p}_\mathsf{H}\mathbf{p}_\mathsf{H}{}^\mathsf{T} w \, \mathrm{d}A \mathbf{Q}^\mathsf{T}$$

Let us denote the area of the window as $A_w = \int w \, \mathrm{d}A$ and $\Theta_{\mathrm{H}w} = \int \mathbf{p}_\mathsf{H}\mathbf{p}_\mathsf{H}{}^\mathsf{T} w \, \mathrm{d}A$. Then $\tilde{\mathbf{W}} = \mathbf{Q}\mathbf{Q}^\mathsf{T} A_w$, if $\Theta_{\mathrm{H}w} = A_w \cdot \mathbf{I}_{3\times 3}$, which means that the inertia matrix of the processing window is $A_w \cdot \mathbf{I}_{2\times 2}$ (the inertia around the origin (axis Z) is $2 \cdot A_w$, mean radius of inertia is $\sqrt{2}$) and the CoG. of the window is at the origin. Such windows are:

- disk with $r_{\max} = 2$
- Gaussian bell with $\sigma = \sqrt{2}$, $r_{\max} \gg 1$
- square with sides $2\sqrt{3}$
- Hann(ing) window $w = (1 + \cos(\pi \sqrt{x^2 + y^2}/r_{\max})/2$ with $r_{\max} \approx 2 \cdot 1.4655$

Choosing the window and the coordinate frame according to the above ensures that the normalisation will be $\Sigma = \mathbf{I}_{6\times 6}$.

Remark I: This condition is assumed in the rest of this article.

The results above determine the optimal selection of the coordinate frame versus the investigation window. Therefore, the coordinate unit is defined relative to the size of the window.

Remark II: In the following, the term *unit (length)* will refer the coordinate unit resulting from the deductions above.

3.5 Invariant flows of M

If $\mathbf{M}$ is singular, then the null space contains the parameters of affine flows, the linear combination of which does not change the investigated image at all. These will be referred later as invariant flows. The term quasi-invariant flow will refer a flow which changes the image minimally.

The determination of invariant flows is coordinate frame independent (in the case of non-singular transformations). This means, that if we transform $\mathbf{M}_A$ into frame B, calculate the null space and transform it back to frame A, then we get the same space as directly calculating from $\mathbf{M}_A$.

If $\mathbf{N}$ denotes the null space of $\mathbf{M}$ then applying (6) for vectors in the flow parameter space results

$$\mathbf{N}_A^\mathsf{T}\mathbf{M}_A = \mathbf{0}, \;\; \mathbf{N}_B^\mathsf{T}\mathbf{S}_{AB}^\mathsf{T}\mathbf{M}_A\mathbf{S}_{AB} = \mathbf{0}$$

$$\tilde{\mathbf{N}}_A = \mathbf{S}_{AB}\mathbf{N}_B \;\Rightarrow\; \tilde{\mathbf{N}}_A^\mathsf{T}\mathbf{M}_A = \mathbf{0}$$

Since $\operatorname{rank}(\mathbf{N}_A) = \operatorname{rank}(\tilde{\mathbf{N}}_A) = \operatorname{defects}(\mathbf{M}_A)$, $\mathbf{N}_A$ and $\tilde{\mathbf{N}}_A$ describe the same subspace, the null space of $\mathbf{M}_A$.

This property ensures the processing of rank 5 $\mathbf{M}$ matrices in an affine invariant way. However, if the image detail and thus the calculated $\mathbf{M}$ descriptor are sensible to the exact size and shape of the accumulation window, as image transformations are affecting them, the affine invariance is weakened, see sec.8.

The determination of quasi-invariant flows (flows with small, but non-zero eigenvalues) is coordinate frame dependent, but it is rotation and scaling invariant.

3.5.1 Decomposition of M

Sometimes, the fixed-points of the flows are determined on the image plane by the image processing task. In this case, we want to search among those flows only, which leave the corresponding point in place. See details in [13]. The following decomposition of $\mathbf{M}$ is used frequently for different purposes.

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{bmatrix} \tag{13}$$

where $\mathbf{A}$ is 4×4 symmetric matrix, B is 4×2, and $\mathbf{C}$ is 2×2 symmetric. $\mathbf{C}$ is frequently used in keypoint detectors, for example in the Harris [5] and in the Shi and Tomasi [16] corner detectors. It is also the fundamental output of the standard LK problem, as $e^2 = [t_x \;\; t_y]\mathbf{C}[t_x \;\; t_y]^\mathsf{T}$ describes how sensitive is the image detail against different translations. The sum of squared errors of two uniform shifts in any two perpendicular directions can get by:

$$E_{AC} = \text{trace}(\mathbf{C}) = e_{t_x}^2 + e_{t_y}^2, \tag{14}$$

which is the accumulated energy of the 'AC' component in the image.

$\mathbf{C}$ can also be interpreted as the 2nd order component of the gradient histogram of the image ($\int \mathbf{g}\mathbf{g}^\mathsf{T} w \, \mathrm{d}A$). E_{AC} will be used to normalise the eigenvalues of $\mathbf{M}$ and to determine the rank of $\mathbf{M}$.

3.5.2 Practical relations

During the implementation the reader must pay attention, that images looking singular do not produce singular $\mathbf{M}$ even if only a small noise is present on the image. As experienced, smoothing the image enhances noise invariance. It is advantageous to apply a considerable amount of blur. This is especially important when processing low resolution images with few distinct color values. Blur decreases the scatter of measured gradients, but enhances scatter in position. This effect is similar to the uncertainty principle. Fortunately, we can eliminate it, by measuring gradient strength on a slightly blurred, and gradient direction on a highly blurred image. In the case of analysing drawings we suggest to apply grayscale morphology.

3.6 Alternative interpretations

This subsection gives alternative definitions of the introduced entities and quantities. $\mathbf{M}$ also can be interpreted as the homogeneous 2nd domain-moment of $\mathbf{g}\mathbf{g}^\mathsf{T}$.

$$\mathbf{M} = \int \begin{bmatrix} x^2 & xy & x \\ xy & y^2 & y \\ x & y & 1 \end{bmatrix} \otimes \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w \, \mathrm{d}A \tag{15}$$

$$\mathbf{M} = \int [\mathbf{p}_\mathsf{H}\mathbf{p}_\mathsf{H}{}^\mathsf{T}] \otimes [\mathbf{g}\mathbf{g}^\mathsf{T}] \cdot w \, \mathrm{d}A,$$

Similarly $\hat{\mathbf{M}} = \int [\mathbf{g}\mathbf{g}^\mathsf{T}] \otimes [\mathbf{p}_\mathsf{H}\mathbf{p}_\mathsf{H}{}^\mathsf{T}] w \, \mathrm{d}A$. The block symmetric property is clearly seen on the above formulae. Although $\mathbf{M}$ (and $\hat{\mathbf{M}}$) is the sum of Kronecker products, the spectral decomposition theorem of Kronecker products [6] cannot be utilized, since the required characteristics is lost during integration. Only the block symmetric property is preserved.

A 9×9 extension of $\mathbf{M}$ can be defined:

$$\mathbf{M}^\star = \int (\mathbf{p}_\mathsf{H} \cdot \mathbf{p}_\mathsf{H}{}^\mathsf{T}) \otimes (\mathbf{g}_\mathsf{H} \cdot \mathbf{g}_\mathsf{H}^\mathsf{T}) w \, \mathrm{d}A$$

where $\mathbf{g}_\mathsf{H} = [g_x \;\; g_y \;\; I]^\mathsf{T}$ and I is the image intensity. $\mathbf{g}$ can also be extended as $\mathbf{g}_\mathsf{H} = [g_x \;\; g_y \;\; 1]^\mathsf{T}$. In this case the lower right symmetric 3×3 block of $\mathbf{M}^\star$ describes only the shape of the interest window itself, not its contents.

4 Classification of affine flows

This section exploits the effects of affine transformations on affine flows.

Definition: *An affine flow is said to be nonsingular if* $\mathbf{F}$ *is nonsingular.*

The invariant subspace of such a flow always exists and is a point. The fixed-point's location is affected by the shift component ($\mathbf{t}$) of the flow, see (8). This component can be eliminated by shifting the origin to this point.

Definition: *Any flow with parameter vector* $\mathbf{t} = \mathbf{0}$ *will be called Translation Free Affine flow and will be referred as TFA flow. Moreover, Translation Free Affine transformation (*$\mathbf{c}_{BA} = \mathbf{0}$*) will be referred as TFA transformation.*

In the following we investigate TFA flows only, since applying (8) we can transform any nonsingular flow to TFA flow. Any TFA flow is invariant against pure scaling, and the flow describing pure scaling is invariant against any TFA transformation, see sec.3.1.2.

Since the origin is moved in a well determined place, we want to investigate, how $\mathbf{F}$ is changed during TFA transformations. From (5) $\mathbf{F}$ is transformed by

$$\mathbf{F}_A = \mathbf{R}_{BA}^{-1} \mathbf{F}_B \mathbf{R}_{BA}$$

Thus, $\mathbf{F}$ can be diagonalised by a TFA transformation retrieved by spectral decomposition. However, in the case when $\mathbf{F}$ is not diagonalisable the normalising transformation is calculated in a slightly different way. As flows with different strength have the same shape, the magnitude of the eigenvalues does not play role, only the ratio of them.

We have to warn MATLAB® users, that function $eig()$ does not give correct results nor throw a warning, if diagonalisation is not possible. If the eigenvalues are equal, one should check whether $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$ or $\text{rank}(\mathbf{U}) = n$. We recommend to use $\mathbf{U}_0, \mathbf{D}_0, \mathbf{V} = svd(\mathbf{F} - \text{trace}(\mathbf{F})\mathbf{I}/n)$; $\mathbf{D} = \mathbf{D}_0 + \text{trace}(\mathbf{F})\mathbf{I}/n$; $\mathbf{U} = \mathbf{U}_0^\mathsf{T}$ or similar solution in this case. The eigenvalues are calculated correctly in any situation.

Let us denote the discriminant of the second order characteristic polynomial of $\mathbf{F}$ by:

$$\kappa = \text{trace}(\mathbf{F})^2 - 4|\mathbf{F}| \tag{16}$$

κ describes the location of the eigenvalues in the complex plane, for example $\kappa = 0 \Leftrightarrow \lambda_1 = \lambda_2$. Moreover, let us denote

$$\tau = |r_x - r_y|^2 = \|\mathbf{F}\|_\mathbf{F}^2 - \text{trace}(\mathbf{F})^2 + 2|\mathbf{F}|$$

which measures the non-diagonalisable behavior of $\mathbf{F}$ if $\kappa = 0$. Keep in mind, that κ is transformation invariant, but τ is only rotation and scaling invariant, as the Frobenius norm ($\| \|_{\mathbf{F}}$) depends on affine transformations. However, $\mathrm{sgn}(\tau)$ is invariant against non-singular TFA transformations. Neither κ nor τ is invariant against flow strength changes, they are proportional to the squared strength.

The flow normalisation algorithm can be extended to return the uncertainty of the calculated transformation, (see sec.2.3), via a 7 × 7 homogeneous symmetric matrix of the parameters of the normalising affine transformation.

4.1 Constrained diagonalisation

Uncertain situations during normalisation can be avoided conceptually if we formalise our goals and use the following definition: Given the flow $\mathbf{q}$, with normalised strength (in the origin-centered 4 unit diameter window), we ask if there is any affine transformation, which has no scaling, but does have any rotation, a constrained translation and a constrained affinity component, which would transform $\mathbf{Q}_{\mathsf{H}}$ to have similar non-zero entries in the diagonal, or right above the diagonal. This can be referred as constrained diagonalisation. If it fails due to violating constraints, it indicates that the normalisation would be numerically uncertain.

The constraint against translation should allow translation strength identical to the case where the fixed-point of a non-singular flow is no further from the origin than 4..7 units. The constraint against affinity should allow affinity factors up to 4..7.

4.2 Classification of nonsingular flows

Flow classification can be performed transformation invariantly based on the eigenvalues of $\mathbf{F}$. As these values can be complex conjugate pairs, it is better to use combinations of them as $\lambda_1 + \lambda_2$ and $\lambda_1\lambda_2$, or κ and τ. Crisp flow classification has the problem, that two very close eigenvalues can be treated differently depending on additional noise. Applications requiring robust behavior might use rather a hypothesis-verification-like method than crisp classification.

Problems arise during normalisation also. The normalising transformation is only one particular solution, however, infinite such transformations can exist, since additional scaling usually does not change the flow. In most cases, additional unambiguities arise.

Based on $\mathrm{sgn}(\kappa)$ and $\mathrm{sgn}(\tau)$, flows with nonsingular $\mathbf{F}$ can be divided into groups with the following properties:

'SPIRAL' Flows:

- $\kappa < 0$
- the eigenvalues of F are conjugate complex numbers
- the streamlines are exponential spirals or ellipses (Fig. 2/f,g,h)
- the class has one real parameter $\beta = \arg(\lambda)$ describing the ratio of scaling and rotation. It has 180° ambiguity
- the normalised flow has uniform squared velocity distribution ($\tilde{\mathbf{W}}$)
- the normalised (isotropic) flow is rotation (and scaling) invariant, thus the normalising transformation is ambiguous up to any combination of rotation and scaling.

'POWER FN' (Power Function) Flows:

- $\kappa > 0$
- the eigenvalues of F are real numbers
- the streamlines consist of power functions (Fig. 2/j,k,l)
- the class has one real parameter describing the exponent $\gamma = \lambda_2/\lambda_1$. Notice, that the inverse of this parameter results a flow rotated by 90°, so $\mathrm{sgn}(\gamma)|\log(|\gamma|)|$ identifies the exponent better.
- $\tilde{\mathbf{W}}$, the velocity histogram of the normalised flow depends on γ. It is uniform if $\gamma = \pm 1$
- The normalised flow is invariant against affinities parallel to the axes (and to scaling). Therefore, the normalising transformation is ambiguous up to any combination of scaling and axis parallel affinity.

'SCALING' Flows:

- $\kappa = 0$ and $\tau = 0$
- the eigenvalues of F are equal real numbers and F is a similarity matrix
- the streamlines are a fan of straight lines (Fig. 2/i)
- the flow cannot be normalised, as it is invariant against any transformation.

'DEROGATORY' Flows:

- $\kappa = 0$ and $\tau > 0$
- the eigenvalues of F are equal real numbers and F is a non-diagonalisable matrix
- the streamlines are combined exponential and power functions (Fig. 2/m,n,o). The streamlines of the normalised flows satisfy:
$$\begin{bmatrix} x \\ y \end{bmatrix} = e^t \begin{bmatrix} 1 \\ t + c_6 \end{bmatrix}, \ t, c_6 \in \mathbb{R}$$
- the normalising transformation of the flow depends on flow strength up to an axis perpendicular affinity. Any two derogatory TFA flows can be transformed to cover each other. The normalisation should consist of a rotation which eliminates r_y, and an axis perpendicular affinity which makes the normalised $\mathbf{F}$ look like $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. See sec.4.4 for more detail.

These groups are closed to the addition of pure scaling, which is identical to add the same real number to the eigenvalues.

Remark: It can be recognized, that trace($\mathbf{F}$), the scaling component of $\mathbf{F}$ does not play role in classification, it is orthogonal to the above discussed problem.

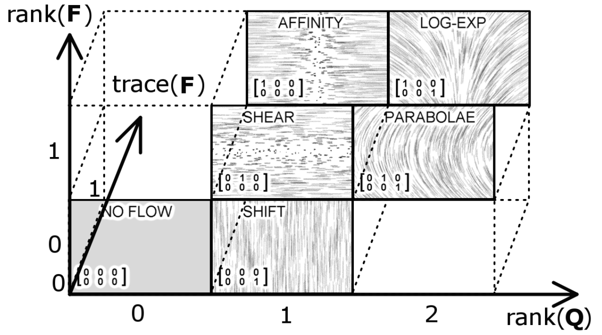
rank(**F**)
trace(**F**)
AFFINITY
$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$
LOG-EXP
$\begin{bmatrix}1&0&0\\0&0&1\end{bmatrix}$
SHEAR
$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$
PARABOLAE
$\begin{bmatrix}0&1&0\\0&0&1\end{bmatrix}$
NO FLOW
$\begin{bmatrix}0&0&0\\0&0&0\end{bmatrix}$
SHIFT
$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$
1
1
0
0
0
1
2
rank(**Q**)

Fig. 3. Classification of singular flows. Names, matrix **Q** and streamlines of normalized flows are shown.

4.3 Classification of singular flows

The classification of singular flows (rank(**F**) < 2) is relatively simple, if we assume that flow strength has been normalised and the flow was transformed to its natural basis frame. The six emerging cases are well distinguishable based on the following factors: rank(**F**), rank(**Q**), trace(**F**). As rank calculation is ambiguous (see later sec.6.1), the result of classification is uncertain for flow with very small, but non-zero elements in **Q**. The six possible flow can be seen in Fig **??**, together with the matrix of the normalised flow.

4.4 Problems of algebraic diagonalization

The normalisation code of 'SPIRAL' flows does not use $eig()$, since it results complex **D** and **U**. The implemented algorithm does not diagonalise **F**, it finds a transformation which results

$$\mathbf{F}' = |\lambda| \begin{bmatrix} C & -S \\ S & C \end{bmatrix},\ C^2 + S^2 = 1,\ C, S \in \mathbb{R}.$$

Strength change of flows with diagonalisable **F** matrix scales λ_{12}, but leaves the diagonalising basis transformation unchanged, and coordinate transformation of the flow does not change the eigenvalues. Neither is true for derogatory flows. For 'DEROGATORY' flows, we can adjust nearly any λ and any ratio of λ and the upper right element by applying affinity to the image.

By other words, the measure of derogatoriness is subject to point of view. The almost straight streamlines of a slightly derogatory flow (Fig. 2/m) can be curved by applying considerable amount of affinity (Fig.2/n,o).

The matrix $\mathbf{F} = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$ can be changed via $\begin{bmatrix} 1 & 0 \\ 0 & e \end{bmatrix}$ axis parallel affine transformation and c flow strength amplification:

$$c \begin{bmatrix} 1 & 0 \\ 0 & 1/e \end{bmatrix} \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e \end{bmatrix} = \begin{bmatrix} c\lambda & ce \\ 0 & c\lambda \end{bmatrix}$$

Therefore any two derogatory flow's streamlines can be transformed to cover each other.

Let us consider a 'SCALING' flow with two similar eigenvalues and an additional noise resulting in $\mathbf{F} = \begin{bmatrix} \lambda & \delta \\ 0 & \lambda \end{bmatrix}$ where δ is small. The calculated normalising transformation and its inverse is very sensitive to δ.

A flow with very small, infinitesimal τ can be transformed to a general derogatory flow with a large amount of affinity applied.

Therefore, flow parameters calculated in the primary measurement's coordinate system with small τ should be treated as 'SCALING' flows, others should be normalised according to the following:

- the flow needs to be quasi-diagonalised by a robust method
- the matrix of the flow should be made looking like $\begin{bmatrix} a & \pm a \\ 0 & a \end{bmatrix}$ (by an y-affinity $e = \lambda$)
- the flow strength needs to be normalised (by $\mathbf{q}' = \mathbf{q}/|\mathbf{q}|$)

Strictly speaking, we did not do anything else in this section than re-implement the diagonalisation of the 3×3 matrix $\mathbf{Q}_\mathsf{H}$. It always has at least one zero eigenvalue, but the structure of the Jordan-blocks (elementary divisors) of this matrix describes different flow classes. This viewpoint is the homogeneous view of matrix **Q**, it is opposed to the inhomogeneous dissection $\mathbf{Q} = [\mathbf{F}\ \ \mathbf{t}]$. Moreover, we showed clear examples, when standard, complex-valued diagonalisation procedures are not suitable to our application. We know, at least from the above described problems, that the normalising transformation of Jordan-blocks are extremely sensitive to additional noise. Vision applications are required to treat small elements as zero and similar elements as equal based on estimated SNR, in contrast to standard algebraic applications. This is the other reason, why we re-implemented the diagonalisation of this 3×3 block upper-triangular matrix. We find the two viewpoints of this classification procedure similarly important both during understanding the theoretical background of affine flow normalisation and during implementing applications. The right null-space of $\mathbf{Q}_\mathsf{H}$ is the homogeneous representation of the flow's fixed-point. It can handle 'PARABOLAE' flow's infinite far fixed-point $\mathbf{p} = [x\ \ y\ \ 0]^\mathsf{T}$, see sec.6.5.3. If we change **Q** a little bit, this fixed-point becomes finite, it will refer to the distant center of concentric ellipses or the center of large hyperbolae in the opposite direction. This case illustrates many, above described computer vision and matrix calculus related phenomena, it might help deeper understanding of them.

If constrained diagonalisation of a near-elliptic or near-hyperbolic flow fails, then instead of returning 'FAIL', we should assume that the flow is of 'PARABOLAE' type. The reconstructed flow's parameters will differ from the original. We can choose this method, if the error of modeling the flow as 'PARABOLAE' is smaller than modeling it as a flow of its original type and by a normalising transformation which satisfies the constraints. Similarly, slightly derogatory flows might be modeled as 'SCALING', 'POWER FN' flows with distant fixed-point and large/infinitesimal $|\gamma|$ as 'LOG-EXP'.

5 Velocity distribution of invariant flows

Let us consider the case if the input image looks like Fig. 7/1a. **M** will have two defects, the null space consists of two 6-D vectors. One can choose a coordinate frame (see sec.6.5.3 how to perform it), such that the null space can be decomposed to:

$$\begin{bmatrix}1&0&0&0&0&0\\0&0&0&1&0&0\end{bmatrix} \quad or \quad \frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&1&0&0\\1&0&0&-1&0&0\end{bmatrix}$$

The flows represented by the base vectors of the null space can be seen in Fig. 7/1c,1d. The invariant subspaces of the nullspace vectors of **M** depend on the exact chosen base. In the following, we investigate the effect of rotating these basis vectors by the $n \times n$ unitary matrix **U**:

$$\begin{bmatrix}\mathbf{q}_1'^{\mathsf{T}}\\ \vdots \\ \mathbf{q}_n'^{\mathsf{T}}\end{bmatrix} = Null'(\mathbf{M}) = \mathbf{U}Null(\mathbf{M}) = \mathbf{U}\begin{bmatrix}\mathbf{q}_1^{\mathsf{T}}\\ \vdots \\ \mathbf{q}_n^{\mathsf{T}}\end{bmatrix}$$

where $n \in \mathbb{N}$ is the defect of **M**, in the above example

$$n = 2,\ [\mathbf{q}_1\ \mathbf{q}_2]^{\mathsf{T}} = \begin{bmatrix}1&0&0&0&0&0\\0&0&0&1&0&0\end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix}C_\alpha & -S_\alpha\\ S_\alpha & C_\alpha\end{bmatrix}.$$

Since the invariant subspace of a flow is defined as the locations where flow velocity is zero, hence we investigate the distribution map of the flow velocity for nullspace vectors $\mathbf{q}_i$, $i = 1..n$ as a function of $\mathbf{p}_\mathsf{H}$.

$$|\mathbf{v}_i(\mathbf{p}_\mathsf{H})|^2 = \mathbf{v}_i(\mathbf{p}_\mathsf{H})^{\mathsf{T}}\mathbf{v}_i(\mathbf{p}_\mathsf{H}) = \mathbf{p}_\mathsf{H}{}^{\mathsf{T}}\mathbf{Q}_i^{\mathsf{T}}\mathbf{Q}_i\mathbf{p}_\mathsf{H} \tag{17}$$

$$|\mathbf{v}_i(\mathbf{p}_\mathsf{H})|^2 = \mathbf{p}_\mathsf{H}{}^{\mathsf{T}}\mathbf{W}_{\mathsf{H}i}\mathbf{p}_\mathsf{H},\quad \mathbf{W}_{\mathsf{H}i} = \mathbf{Q}_i^{\mathsf{T}}\mathbf{Q}_i$$

As we can see, the squared velocity is a homogeneous quadratic function of planar position. Transforming by **U** yields:

$$\mathbf{q}' = \mathbf{U}\mathbf{q} \equiv \mathbf{q}_k' = \sum_{i=1}^{n} U_{ki}\mathbf{q}_i \Rightarrow \mathbf{Q}_k' = \sum_{i=1}^{n} U_{ki}\mathbf{Q}_i,\quad k = 1..n$$

Let us investigate $\mathbf{W}_\mathsf{H} = \sum_k \mathbf{W}_{\mathsf{H}k}$, the accumulated velocity map, and its dependence from **U**:

$$\mathbf{W}_\mathsf{H}' = \sum_k \mathbf{Q}_k'^{\mathsf{T}}\mathbf{Q}_k'$$

$$\mathbf{W}_\mathsf{H}' = \sum_k \sum_i U_{ki}\mathbf{Q}_i^{\mathsf{T}} \sum_j U_{kj}\mathbf{Q}_j \tag{18}$$

Let $\tilde{U}_{ij}$ denote the total contribution of $\mathbf{Q}_i{}^{\mathsf{T}}\mathbf{Q}_j$ to $\mathbf{W}_\mathsf{H}'$:

$$\mathbf{W}_\mathsf{H}' = \sum_i \sum_j \tilde{U}_{ij}\mathbf{Q}_i{}^{\mathsf{T}}\mathbf{Q}_j,\quad \tilde{U}_{ij} = \sum_k U_{ki}U_{kj} \Rightarrow \tag{19}$$

$$\hat{\mathbf{U}} = \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}_{n\times n} \Rightarrow \tilde{U}_{ij} = \delta(i,j)$$

Thus

$$\mathbf{W}_\mathsf{H} = \mathbf{W}_\mathsf{H}' = \sum_i \mathbf{Q}_i{}^{\mathsf{T}}\mathbf{Q}_i \tag{20}$$

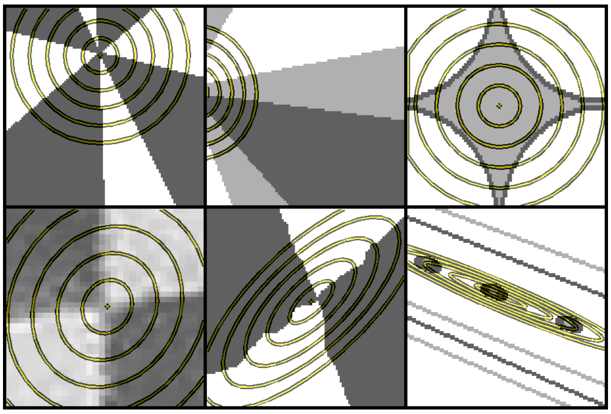

Fig. 4. Potential lines of squared flow velocity distribution ($\mathbf{W}_\mathsf{H}$) overpainted on real images

As it has been shown, $\mathbf{W}_\mathsf{H}$, the velocity map of possible image movement is independent of nullspace rotations. The deduction above can be extended to all cases, even if **M** has no defect at all. All we have to do is to multiply the eigenvectors of **M** by the inverse square root of the corresponding eigenvalue:

$$\mathbf{q}_i = \mathbf{V}_{:,i}\mathbf{D}_{ii}^{-1/2},\ \mathbf{V}\mathbf{D}\mathbf{V}^{\mathsf{T}} = \mathbf{M} \tag{21}$$

where : is the MATLAB® colon operator used in indexing. Such flow parameters have the property, that any linear combination of them yield to image reprojection error proportional to the magnitude of vector **r** describing the linear combination:

$$\mathbf{q}^* = \begin{bmatrix}\mathbf{q}_1 & . & . & . & \mathbf{q}_6\end{bmatrix}\mathbf{r},\ e^2 = \mathbf{q}^{*\mathsf{T}}\mathbf{M}\mathbf{q}^* = \mathbf{r}^{\mathsf{T}}\mathbf{r}$$

In other words, (orthogonal) vectors $\mathbf{q}_i$ describe a hyper-ellipsoid in the affine flow parameter space. Any flow parameter on the surface of this ellipsoid results unit reprojection error. Let us choose any six parameter vector, which represents the same ellipsoid. All such a basis can be described as **UV**, where **V** comes from (21), **U** is arbitrary and $\mathbf{U}\mathbf{U}^{\mathsf{T}} = \mathbf{I}_{6\times6} = \mathbf{V}\mathbf{V}^{\mathsf{T}}$. Equations from (17) to (20) can be applied in this case also, since they do not suppose anything about $\mathbf{q}_i$, they utilize only the orthogonality of **U**. Therefore, $\mathbf{W}_\mathsf{H}$ is independent of **U**. It is easy to prove, that $\mathbf{W}_\mathsf{H}$ can be calculated as

$$\mathbf{W}_\mathsf{H} = (\hat{\mathbf{M}}^{-1})_{1:3,1:3} + (\hat{\mathbf{M}}^{-1})_{4:6,4:6} \tag{22}$$

An implementation can decide which kind of pseudo inverse to use in (22), we found $\tilde{\lambda}_i = \lambda_{min}/\lambda_i$ and $\tilde{\lambda}_i = \mathrm{trace}(\mathbf{C})/\lambda_i$ to be a good solution.

One can also investigate (numerically) for each point $\mathbf{p}_\mathsf{H}$, which unit-strength flow results the smallest squared error among those having zero velocity at $\mathbf{p}_\mathsf{H}$. The shape of this error function is also interesting to investigate.

The application and abilities of the above results can be clearly seen on Fig. 4. Images have been blurred prior to the calculation of **M**. Then $\mathbf{W}_\mathsf{H}$, the quadratic velocity distribution was calculated and was overpainted on the original input images.

6 Classifying shapes with SAFT

The extraction of geometric information from the SAFT descriptor of a region depends on the content of that region. Analyzing parallel lines or concentric ellipses needs different algorithms, since the extractable information is fundamentally different. To resolve this problem, classes of shapes are defined. Shapes belonging to the same class should behave similarly when analyzed with SAFT. First we have to decide the class of the investigated region, then we can extract relevant information with the corresponding algorithms.

6.1 The rank of $\mathbf{M}$

The rank of **M** plays important role in the classification. The calculation of the rank is usually done by the investigation of singular values. Since M is symmetric and positive semidefinite, the eigenvalues and the singular values are identical. Standard algorithms would declare an eigenvalue as a defect if it is close to zero compared to the working precision of the CPU (or the GPU). We suggest to compare this eigenvalues by E_{AC}, the averaged AC information in the image, see sec.3.5.1.

$$\lambda_i' = \left\{ \begin{array}{c|l} \lambda_i & \lambda_i > c'E_{AC}, \quad c' \in (0.01\ ,\ 0.1) \\ 0 & \text{otherwise} \end{array} \right.$$

For a certain image, different c' values can result different ranks. Therefore, some images can be located in more groups. No matter what the ranks are, a proper implementation should identify the class based on invariant flows.

Let us investigate the following problem. The image contains only a circular disc in the origin with unit intensity change in a distance d. Its radius will be denoted by r.

$$\mathbf{g}_\alpha = \begin{bmatrix} C_\alpha & S_\alpha \end{bmatrix}^\mathsf{T} /d;$$

$$\mathbf{M} = \int_0^{2\pi} \begin{bmatrix} rC_\alpha & rS_\alpha & 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} rC_\alpha & rS_\alpha & 1 \end{bmatrix} \otimes \mathbf{g}_\alpha \mathbf{g}_\alpha^\mathsf{T} d \cdot r \cdot \mathrm{d}\alpha$$

Decomposing **M** by (13) yields

$$\mathbf{A} = \frac{\pi}{4d} \begin{bmatrix} 3r^3 & 0 & 0 & r^3 \\ 0 & r^3 & r^3 & 0 \\ 0 & r^3 & r^3 & 0 \\ r^3 & 0 & 0 & 3r^3 \end{bmatrix}, \quad \mathbf{B} = 0 \ \ \mathbf{C} = \frac{\pi}{4d} \begin{bmatrix} 4r & 0 \\ 0 & 4r \end{bmatrix}.$$

The eigenvalues are

$$\hat{\lambda} = \begin{bmatrix} 2 & 2 & 2r^2 & r^2 & r^2 & 0 \end{bmatrix} E_{AC}/4, \quad E_{AC} = 2\pi r/d.$$

Depending on the exact values of r and c', the calculated rank of such an image can be 2, 3 or 5. Theoretically it is always 5.

For general image content, one can make a table showing theoretically possible combinations of rank(**C**) and rank(**M**), but it does not apply in practice, since the calculated rank(**M**) can always be less than the theoretical value. It must not be less than rank(**C**). If and only if the image contains gradients only in one direction, then **C** in (13) is singular and **M** has multiple defects, at least 3.

$$\mathbf{C} = \int \mathbf{g}\mathbf{g}^\mathsf{T} w \,\mathrm{d}A \quad \text{and} \quad \mathrm{rank}(C) < 2 \Rightarrow$$

$$\mathbf{g}(x, y) = \mathbf{g}_0 f(x, y), \quad \mathbf{C} = \int f^2 w \,\mathrm{d}A \mathbf{g}_0 \mathbf{g}_0^\mathsf{T}.$$

Thus, the gradients on the image are parallel. Moreover

$$\mathbf{M} = \mathbf{G}_K \otimes \mathbf{g}_0 \mathbf{g}_0^\mathsf{T}, \mathbf{G}_K = \int f^2(x, y) \begin{bmatrix} x & y & 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} x & y & 1 \end{bmatrix} w \,\mathrm{d}A. \tag{23}$$

Applying the theorem of the spectral decomposition of Kronecker products [6] , rank(**M**) $\leq$ 3 (theoretically). Based on these, we can set up the practical rule:

$$\mathrm{rank}(\mathbf{C}) \leq \mathrm{rank}(\mathbf{M}) \leq 3 \cdot \mathrm{rank}(\mathbf{C}) \tag{24}$$

Applications can enforce these inequalities after rank calculations.

6.2 The eigenvalues of $\mathbf{M}$

The eigenvalues of **M** do not behave as ideal in practice as in theory. We highlight five important effects which alter them.

- **Random Noise** increases the eigenvalues uniformly.
- **Rasterisation** increases the scatter of gradient direction very much (consider 45^o line), and the width on infinitesimal line with one pixel. It is obvious that the image must be blurred.
- **Blur** decreases the eigenvalues of translation-like flows, which relate to **C**, the gradient value histogram, but increases eigenvalues which relate to the gradient position histogram. One can apply hybrid gradient evaluation shown in sec.3.5.2.
- **The distance of the fixed-point** of the invariant flow affects the curvature of the flow within the image. If it is decreased, then the eigenvalues related to flows with far fixed-points (high translational components) are decreased also.
- If the image consists only of one streamline (which case is very important), then **the area of useful measurements** (which is indicated by $\mathbf{G}_K$) can be considerably smaller (or flatter), than the investigation window. This effect decreases eigenvalues of flows, the invariant subspace of which coincides with the area of useful image content.

Certain geometric arrangements theoretically imply two defects, see sec.6.5.3. These situations can hardly be distinguished in practice from the situation of a single streamline of a general flow based only on the eigenvalues. Regarding the flows associated with them gives us more chance to do so.

Implementing rank calculation based on the ratio of neighbouring eigenvalues was investigated in our research too. One might determine the defects of **M** where the ratio of neighbouring eigenvalues are the highest. However, it is better to return an 'UNCERTAIN' flag if this ratio is less than 3...4 and to investigate other candidates also, where the ratio is higher than 7...10.

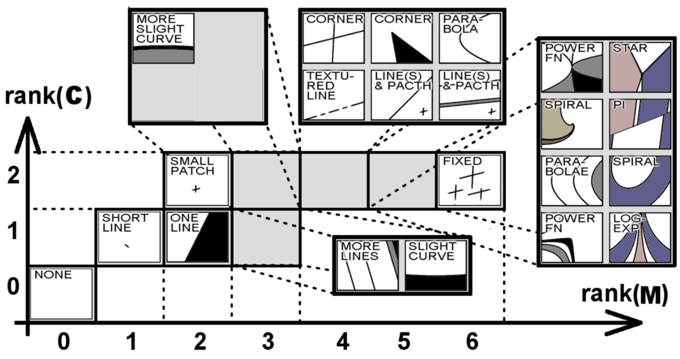

Fig. 5. Typical SAFT class members, arranged by rank(**C**) and rank(**M**).

6.3 Normalising and classifying in the practice

Normalising the SAFT matrix requires the same viewpoint as used by flow normalisation in sec.4.1. We investigate if there is a constrained coordinate transformation, which makes null as many elements of **M** as possible. However, in case of images from real measurements, this criterion needs to be widened, as the searched transformation should minimize the maximum of the small elements in **M**.

Despite the definitions of constrained normalisation is described in this paper, in our research we used heuristic methods also to avoid uncertain situations during normalisation.

Determining image class can be ambiguous in certain situations. An alternative, but computationally more complex solution is, that for every candidate class we calculate the parameters of the image with a robust, class specific method, and determine the error of our hypothesis. For several classes there are simple formulae which can be used for this purpose. Otherwise, the error can be calculated by constructing the SAFT matrix of the hypothesis. The error of original SAFT matrix against eigenflows of the hypothesis must be evaluated.

6.4 Brief enumeration of classes

Fig. 5 shows images of typical class members, and the corresponding ranks which are theoretically possible. Similar image contents can evaluate to other ranks due to effects listed in sec.6.2.

The classes can be constructed as follows: Pure geometric classes can be found by theoretical reasoning and via experiments. Small details might then be added to these classes (a small portion of a curve deleted, or a very short line or small patch (noise) added to a clear area). These added details change the rank of **M**.

Fig. 6 is covered with multi-scale interest windows, which are analysed and classified. Windows are eliminated if the rank of **M** equals 6 or 0. Remaining ones are considered interesting. Windows are referred to be stable if neighbouring windows give similar results. They are of maximal size, if no bigger, covering window with similar results can be found.

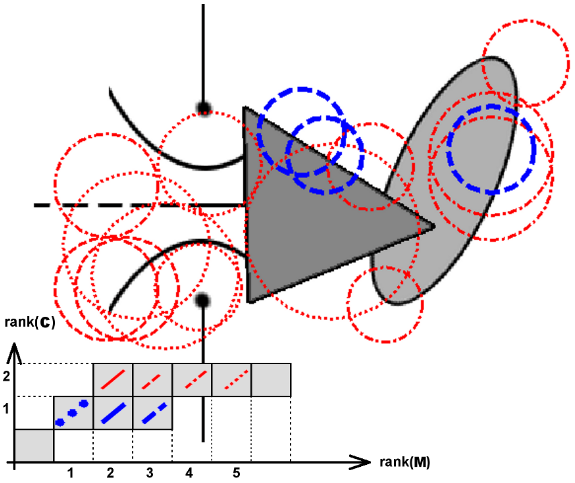

Fig. 6. Classified stable windows of maximal size. Line style refers rank(**M**), width indicates rank(**C**). See Fig. 5 and (24) also.

6.5 rank($\mathbf{C}$) $= 2$

6.5.1 Group rank($\mathbf{M}$) $= 6$

Once we diagnosed rank($\mathbf{M}$) $= 6$, there is nothing more to do. The image does not contain information which can be used extracted based on invariant affine flows. However, we can still determine the transformation which normalises the quadratic function $\mathbf{W}_\mathsf{H}$.

6.5.2 Group rank($\mathbf{M}$) $= 5$

Classification and normalisation of this group is done by classifying and normalising its singular flow. However, if the fixed-point is very far, the normalising transformation will contain too much affine distortion. We suggest to neglect normalisation if the fixed-point is far from the center, etc. See sec.6.3.

Note, that not all flow types can occur, since flows with rank($\mathbf{Q}$) $= 1$ have parallel gradients. The SAFT matrices of such flows have multiple defects. A rank($\mathbf{M}$) $= 5$ image has no gradient component parallel to the streamlines of its singular flow.

6.5.3 Group rank($\mathbf{M}$) $= 4$

Images of this group has two independent invariant flows, which are encapsulated in the null space of **M**. Multiplying this null space by a 2 × 2 rotation matrix $\mathbf{U}_\alpha$, we will have another pair of independent invariant flows. Thus, these shapes cannot be characterized by two affine flows, two fixed-points, etc, since they are changing and moving with the 2 × 2 rotation.

$$\begin{bmatrix}\hat{\mathbf{q}}_1' & \hat{\mathbf{q}}_2'\end{bmatrix} = \mathbf{U}_\alpha \begin{bmatrix}\hat{\mathbf{q}}_1 & \hat{\mathbf{q}}_2\end{bmatrix}, \hat{\mathbf{q}}_i = \begin{bmatrix}\mathbf{q}_{i_x} \\ \mathbf{q}_{i_y}\end{bmatrix}, \mathbf{Q}_i = \begin{bmatrix}\mathbf{q}_{i_x}^\mathsf{T} \\ \mathbf{q}_{i_y}^\mathsf{T}\end{bmatrix}$$

Two intersecting lines (or half lines) lie on a degenerate hyperbola. This detail is very frequent in human-built environment. Photographing calibration grids also yield to this kind of details [19], [3]. Parabolae are also very frequent in artificial environment: As every circle appears as a non-degenerate conic, the apex of this conic is often undistinguishable from a perfect parabola.

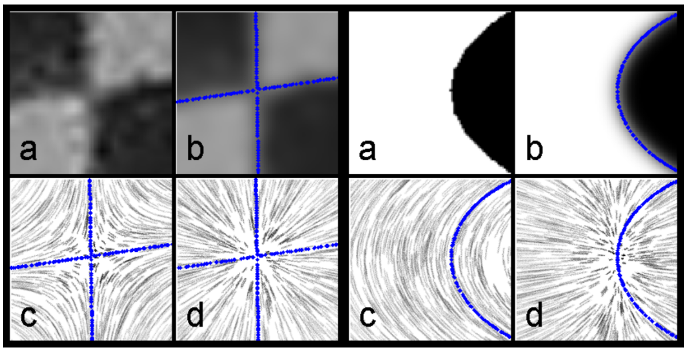

Fig. 7. rank($\mathbf{M}$) $=$ 4 images, 'CORNER' and 'PARABOLA'. Original (a) and blurred (b) image, flow with zero (c) and maximal (d) scale component. b,c,d have conic $\mathbf{K}'$ overpainted.

Theoretically there are only two non 1-dimensional conics, the degenerated hyperbola ('CORNER') and the single 'PARABOLA' which cause rank($\mathbf{M}$) $=$ 4, see Fig. 7. The 'TEXTURED LINE' belongs to this class also.

Obviously, in practice, image contents similar to these can also yield to rank($\mathbf{M}$) $=$ 4. Analysing apexes of ellipses or hyperbolae, which are similar to a parabola returns the equation of the investigated conic with small, often negligible error.

We expect that any flow lying in the null space will have flow direction parallel to the elements of the gradient image. We are interested in the locations, where the two flow components of the null space are parallel to each other. If the velocities of the two flows are parallel at a given location, then a linear combination of the two flows can be found which results zero velocity at the given point. As all linear combinations are described with null-space rotations, the locations of parallel flow velocities are identical to the locations of the travel of the invariant subspace as the basis of the null space is rotated.

We can approximate the homogeneous coordinates of the fixed-point by calculating the right null-space of $\mathbf{Q}$ by vector product.

$$\begin{aligned}{\mathbf{p}_\mathsf{H}}'_{fix} = \mathbf{q}'_{1_x} \times \mathbf{q}'_{1_y} = {C_\alpha}^2\mathbf{q}_{1_x} \times \mathbf{q}_{1_y} + \\ {S_\alpha}^2\mathbf{q}_{2_x} \times \mathbf{q}_{2_y} + C_\alpha S_\alpha(\mathbf{q}_{2_x} \times \mathbf{q}_{1_y} + \mathbf{q}_{1_x} \times \mathbf{q}_{2_y})\end{aligned} \qquad (25)$$

$${\mathbf{p}_\mathsf{H}}'_{fix} = ({C_\alpha}^2 + {S_\alpha}^2)\mathbf{p}_{\mathsf{H}1} + ({C_\alpha}^2 - {S_\alpha}^2)\mathbf{p}_{\mathsf{H}2} + (2C_\alpha S_\alpha)\mathbf{p}_{\mathsf{H}3},$$

which is the parametric equation of a general conic section ($\mathbf{p}_{\mathsf{H}_i}$ can be calculated with ease.) We can conclude, that the fixed-point travels on a conic. Note, that it can reach infinity, which is the case for 'PARABOLAE' flows. The above approximation is correct only if $\mathbf{Q}$ is non-singular. $\mathbf{Q}$ will be singular for example, if the image contains only a degenerated hyperbola.

The properties of these conics can be get by alternative methods:

- A: Determine the locations, where the two invariant flows are parallel to each other. These locations are independent to the rotation of the null space's basis, and are characterized by matrix $\mathbf{K}'$:
  $$\mathbf{p}_\mathsf{H}{}^\mathsf{T}\mathbf{K}'\mathbf{p}_\mathsf{H} = 0 \qquad (26)$$
  $$\mathbf{K} = \mathbf{Q}_1^{\mathsf{T}\prime}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{Q}'_2, \mathbf{K}' = (\mathbf{K} + \mathbf{K}^\mathsf{T})$$
  The locations where any two affine flows are parallel to each other lie on a conic, but usually the flows are not parallel to this conic. Thus, the error of the hypothesis can be measured as the flow strength of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ through the conic $\mathbf{K}' = 0$. To evaluate this integral $\mathbf{K}' = 0$ should be parametrized first.
- B: Investigate the travel of the fixed-point through null-space rotations. One can numerically determine the path of the fixed-point or approximate it by (25). The fixed-point travels along the image curve for 'PARABOLA' and 'TEXTURED LINE', but our approximation is not suitable for 'CORNER'-s.

If the null space calculation is done correctly instead of the vector-product approximation, then method B is identical to method A. Further classification and (constrained) normalisation can be done by the diagonalisation of $\mathbf{K}'$.

If we would like to detect 2nd order conics, we want to utilise that flows having conic shaped streamlines do not have any scaling component. The strength of the 'SCALING' flow component should be nullified:

$$\mathrm{trace}({\mathbf{F}_1}'_\alpha) = C_\alpha \mathrm{trace}(\mathbf{F}_1) + S_\alpha \mathrm{trace}(\mathbf{F}_2) = 0.$$

For 'CORNER' type images, if trace($\mathbf{F}_1{}'$) $= 0$, then $\mathbf{q}'_1(\alpha)$ represents a 'POWER-FN' flow with $\lambda_{12} = \pm 1$. If $\mathbf{M}$ is affected by noise, etc, $\mathbf{K}'$ will describe a true hyperbola instead of a degenerate one, but flow normalisation is hardly affected by this effect. For 'PARABOLA' type images, trace($\mathbf{F}_1{}'$) $= 0$ can happen only if both lambda's are 0. This referred by $\mathbf{q}'_1$ describing a 'PARABOLAE'-type non-diagonalisable singular flow. The fixed-point of this flow lies at infinity. If the image is a little bit distorted, or noisy, etc, $\mathbf{K}'$ will not describe a parabola, but the apex of an ellipse or a hyperbola with a distant fixed-point. Constrained normalisation should fail in these cases due to the distance of the fixed-point. The best choice is to enforce parabola detection, which can be done by seeking the flow, which has trace($\mathbf{F}$) $= 0$ and $|\mathbf{F}| = 0$. The flows corresponding to the three smallest eigenvalues can be used to find the one belonging to minimum self-affine error. See sec.7.6 for a different perspective of the same problem.

rank($\mathbf{M}$) $=$ 4 can also occur by imaging a small patch and one line.

6.5.4 rank($\mathbf{M}$) $<$ 4

rank($\mathbf{M}$) $=$ 3 occurs in the practice if the image contains the SAFT class of 'ONE LINE', or 'MORE LINES' and a small gradient element which makes $\mathbf{C}$, the gradient range histogram regular. This can be caused by a small, 2-D image detail, or if the 'lines' are slightly curved. The first case is often very sensitive to the investigation window, and can be suppressed by Gaussian or Hann(ing) windowing. The second case can be analyzed by the curvature method, see sec.7.4.

rank($\mathbf{M}$) $= 2$ occurs rarely in the practice, we experienced it when analyzing a short, slightly curved 'line', or an even shorter straight line, or a very small patch. $\mathbf{G}_K$, the gradient domain-moment localises these details with ease.

6.6 rank($\mathbf{C}$) = 1

The rank($\mathbf{C}$) $= 1$ case occurs only if the image is one dimensional. The null-vector of $\mathbf{C}$ describes the orientation of the image. The image consists of parallel lines in this direction. The intensity of the lines may slowly vary along them.

The following questions emerge:

- Are there one, or more lines present on the image?
- Perpendicular to the lines, where is the centroid of the line(s)?
- What is the approximate span of the lines, perpendicular to them?

Notice, that normalisation can be done only in the 1-dimensional direction perpendicular to the lines. This can be calculated based on $\mathbf{G}_K$, see (23).

In the practice, a single, slightly curved 'line' can cause rank($\mathbf{C}$) $= 1$ and rank($\mathbf{M}$) $= 3$ also. Therefore, it is better to answer all the above questions based on $\mathbf{G}_K$.

This kind of image details was not investigated so deeply, as image content is very simple in these cases. The only useful applications found are the ability of the SAFT descriptor to differentiate between a single line and multiple parallel lines, and to determine the position of a single line beside its orientation.

7 Applications

Information extraction from $\mathbf{M}$ needs to decide what is the SAFT class of the image. It can be done either by the numerically sensitive classification or can be chosen based on preliminary hypothesis. Once the class is selected, specific information extraction and normalisation can be used to determine class specific quantities.

Other class independent measures can also be used, for example the domain-moment of squared gradients $\mathbf{G}_K$, or the accumulated velocity map $\mathbf{W}_\mathrm{H}$.

This section summarizes very briefly several applications, most of them illustrating vision scenario when emerging problems are solved by utilizing the relations described in the previous sections. The sensitivity/robustness of the developed algorithms against different effects are notated.

7.1 Vanishing point of lines

Finding the vanishing point of lines can be done by calculating the minimum of $\mathbf{W}_\mathrm{H}$, see sec.5 and sec.3.2. This method is robust against blur, transformations and noise. See sec.7.5 also.

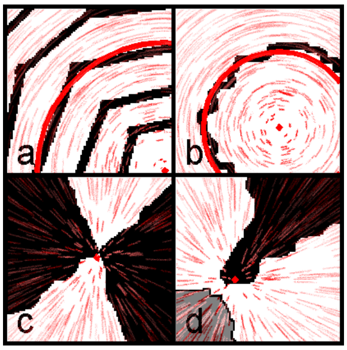

Fig. 8. Enforcing curvature (a,b) and scaling center (c,d) detection.

7.2 Number of lines

Classic edge detectors can find the main direction of lines in the investigation window, but cannot distinguish whether a single line or multiple parallel lines are present. As these fall into different SAFT classes, differentiating could be done by checking the rank of $\mathbf{M}$, see sec.6.6. This method is sensitive to certain effects (blur, curvature, etc.) It is more robust to use rank($\mathbf{G}_K$) for this purpose.

7.3 Parameters of one line

Once it is detected and classified, SAFT is able to determine the position and angle of one line. The classification itself is sensitive to various effects, but the extraction of the parameters can be considered absolutely robust.

7.4 Curvature

If we expect, that the image consists only of concentric circular arcs, then we need to investigate the null space of the following 3 × 3 symmetric matrix:

$$\mathbf{T}_r^\mathrm{T}\mathbf{M}\mathbf{T}_r, \tag{27}$$

where

$$\mathbf{T}_r^\mathrm{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

The smallest singular value describes the error of our assumption. If our assumption holds, then the null space exists and encodes $\mathbf{p}_{fix_r}$, the location of concentricity. If the gradient domain moment is evaluated with this point as the center, then the average radius of the arcs trace($\mathbf{G}'_K$) can be found by:

$$\mathbf{G}'_K = \mathbf{G}_K/\mathrm{trace}(\mathbf{C}) - \mathbf{p}_g\mathbf{p}_g + \mathbf{p}_{fix_r}\mathbf{p}^\mathrm{T}_{fix_r}$$

where $\mathbf{p}_g$ is the center of 'mass' of the gradient domain moment $\mathbf{G}_K$. In case of one single arc it returns the exact radius. This method is similarity invariant and robust against noise.

7.5 Scaling center

A method similar to sec.7.4 can be used to find the center of scaling, and image resistance against it. $\mathbf{M}$ has to be transformed by $\mathbf{T}_s$,

$$\mathbf{T}_s^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

This method is affine invariant and robust against noise.

7.6 Detection of conics

If we want to enforce conic detection, we can analyse the eigensystem of 5 × 5 matrix $\mathbf{T}_c^{\mathsf{T}}\mathbf{M}\mathbf{T}_c$, where $\mathbf{T}_c$ is the right null space of [1 0 0 1 0 0]. We utilized the fact that flows with conic streamlines do not have any scaling components. The error of our hypothesis is given by the smallest eigenvalue.

7.7 Normalisation of ellipses

When photographing circular markers from an angle, it is expected to find the transformation which circularises the ellipse and moves the origin to the center of the normalised circle. We have to test if the image class is the rank($\mathbf{M}$) $=$ 5 Ą 'SPIRAL' class, with zero spiral parameter, then the flow normalising transformation does the job, see sec.4.2.

7.8 Classifying image details

Classifying image details is one of the main focus of this paper. It is useful, for example, when a mobile robot has to search for markers which were designed to be easily distinguishable with SAFT. Notice, that several classes can identify a flow parameter in an affine invariant manner, for example the 'SPIRAL' and 'POWER FN' classes, sec.6. Image classification is the most sensitive method described in this article. It should be used on high resolution details (100 pixels was found to be enough) of geometrical content. For practical cases, calculating a characteristic measure showing that the image belongs to a certain class is superior solution to determining the exact class. Threshold levels used in classification can be determined by soft techniques based on typical examples.

7.9 Normalising details

The other main contribution of this article is a new method for normalisation. We highlight the scenario how it can be applied. A marker can be designed to have a well distinguishable SAFT-class type part. Based on the projection of this part the rectifying transformation can be found. When this is applied to the other parts of the marker, further processing can be carried out on the rectified image. This process is robust, if the SAFT marker is chosen properly.

7.10 Vectorization

Since SAFT can distinguish the types of common curves used in science and engineering (as circles, ellipses, parabolae, hyperbolae, exponential spirals), hence it can be used to determine the (largest) window in the close neighbourhood for which $\int |\mathbf{g}|^2 \, \mathrm{d}A$ is maximum and contains only one curve.

Once the type of the curve is determined via classification, several parameters of this curve can be easily extracted. This kind of application has not to deal with image noise. It can be sensitive to other details falling into the window of interest. Once the curve is identified, another SAFT matrix should be calculated in the vicinity of the curve. It is a good choice to select investigated area based on the Harris corner detector's output. See sec.6.4 and Fig. **??**.

7.11 Grid point Extraction

Computer vision applications frequently require camera calibration. This calibration is usually done by photographing a chequered texture. The usage of SAFT offers an alternative method for sub-pixel accurate grid point extraction, which localises grid corners much better than well-spread gradient, Laplacian or model-fitting methods. The location of the intersection can be get by the minimum locus of $\mathbf{W}_{\mathsf{H}}$. We should use the largest window resulting 'CORNER' class. This method was found to be robust, but blur has to be applied according to image noise level (high frequency image content).

If the directions of the intersecting lines have also to be determined, we can utilize that the vicinity of grid points belongs to the 'CORNER' SAFT class and apply the normalising algorithm described in sec.6.5.3.

7.12 Accelerating computations

Several real-life applications of the above methods can be sped up by the same method seen in sec.7.4. If $\mathbf{M}$ is processed by an application in a very special manner (for example extracting one real number), then simplifications can be made. These simplifications can even affect the computation of $\mathbf{M}$, as in sec.7.4. In these cases, the computational speed of the simplified algorithm can be considerably higher than the speed approximated at first glance. This has to be highlighted, as the numeric 6 × 6 symmetric eigenvalue problem can be considered too slow for many applications.

8 The Influence of pre-processing on affine invariance

The investigation of the same image content in a different coordinate frame can be done in several ways. One can suppose either if the image and the integration window is fixed, only the coordinate frame is changing, or the image is transformed regarding to the fixed domain and coordinate frame. These two situations are geometrically different.

When similar image content is analysed (for example the photograph of the same spatial detail from different angles), the relative transformation of the interest window affects the resultant $\mathbf{M}$ in various ways:

- What kind of image content falls inside/outside the integration window?
- What is the kernel for the initial low-pass filter? (It is usually the integration window's kernel scaled down.)

- What is the coordinate frame of the primary calculation of $\mathbf{M}$? (This can be corrected and modified afterward, whenever needed. It correctly processes the strength of gradients in different directions due to affinities. It cannot handle the effect of different low pass filtering.)
- What is the region, that is incorporated when normalising flow strengths (sec.3.4)?

If the interest window remains the same, but the image is distorted, then this corresponds to the simultaneous effect of all phenomenon listed above.

Applying non-singular affine transformations to the image results the same as we apply the inverse transformation to the processing (and normalising) window, and transform $\mathbf{M}$ by S encoding the forward transformation, except for the effect of low-pass filtering.

In the following, we assume that the neighbourhood of the investigation window contains the same geometric information as the window itself. This results that changing the integration area or windowing function, but keeping the coordinate frame in place, changes $\mathbf{M}$ minimally.

In these cases, the following algorithms behave affine invariant (in the order of robustness against other effect: noise, blur)

- finding the accumulated fixed-point of images with singular $\mathbf{M}$
- determine the flow parameter if there is any
- classification of affine flows
- verifying preliminary hypotheses about image context
- finding the normalised image and parameters of images with rank($\mathbf{M}$) $= 5$ and 4.
- locating conic sections
- finding the normalised flow and parameters of affine flows.

The following methods are affine invariant, but very sensitive to other effects:

- determination of the null space of $\mathbf{M}$ (finding invariant flows)
- classification of $\mathbf{M}$

Constrained normalisation is not absolutely affine invariant by definition, but is a good choice to increase the invariance against noise, blur, rasterisation, etc. It is affine invariant only if the constraints are not violated.

If the condition above does not hold, then the change in the integration domain yields different results. This can be thought as a problem partially related to occlusion artifacts.

The technique of affine shape adaptation can be applied to ensure higher affine invariant operation, but requires considerable more computational power [1].

9 Color channels

Until this point, we supposed that the image consists of one (grayscale) channel. As for the LK detector, the simple addition of the resultant $\mathbf{M}$ matrix of every channel results the correct propagation of sum of squared pixel errors, measured channel-wise.

$$\mathbf{M}_{RGB} = \mathbf{M}_R + \mathbf{M}_G + \mathbf{M}_B$$

The resultant SAFT matrix $\mathbf{M}_{RGB}$ encodes more information than $\mathbf{M}_{gray}$ of the grayscale version of the multichannel image. Generally, the above formulation fits to geometric information extraction from SAFT (for example the calculation of $\mathbf{W}_{\mathsf{H}}$). When SAFT is used as feature detector, one might want to compare the feature vector (or matrix) of each color channel separately.

10 Conclusions and future plans

This paper described the classification of geometric images with the SAFT detector. The calculations are processed affine invariantly. A normalising affine transformation is also gained, which is underdetermined and should satisfy constraints. Class-dependent affine invariant quantities can be easily extracted from the normalised descriptor matrix. This extraction process was briefly described in the paper.

The present article has been concentrated on the SAFT based image information extraction and classification details. The developed algorithms can be applied for solving different 2-dimensional image processing tasks, amongst them detection of convergent lines, circles, ellipses, parabolae and hiperbolae or localizing corners of calibration grids in a robust and accurate manner. The invariance of the introduced methods against planar transformation, noise, blur and rasterization was demonstrated in the article.

There are further results of this research, which have been not published, articles about these results are expected to be contributed in the near future.

References

1 **Baumberg A**, *Reliable feature matching across widely separated views*, IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 1774–1781.

2 **Bay H, Ess A, Tuytelaars T, Van Gool L.**, *SURF: Speeded Up Robust Features*, Computer Vision and Image Understanding (CVIU) **110** (2008), no. 3, 346–359.

3 **Bouguet J Y**, *Camera Calibration Toolbox for Matlab.* http://www.vision.caltech.edu/bouguetj/calib_doc/.

4 **Flusser J, Suk T**, *Pattern recognition by affine moment invariants*, Pattern recognition **26** (1993), no. 1, 167–174.

5 **Harris C, Stephens M**, *A combined corner and edge detector*, 4th Alvey Vision Conference, 1988, pp. 147–151.

6 **Laub Alan J**, *Matrix analysis for scientists and engineers*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.

7 **Lowe D**, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision **60** (2004), no. 2, 91–110.

8 **Lucas B D, Kanade T**, *An iterative image registration technique with an application to stereo vision*, Imaging Understanding Workshop, 1981, pp. 121–130.

9 **Matas J, Chum O, Urban M, Pajdla T**, *Robust wide-baseline stereo from maximally stable extremal regions*, British Machine Vision Conference (Cardiff, UK), 2002, pp. 384–393.

10 **El Mejdani S, Egli R, Dubeau F**, *Old and new straight-line detectors: Description and comparison*, Pattern Recogn. **41** (2008), no. 6, 1845–1866.

11 **Mikolajcyk K, Schmid C.**, *Scale & affine invariant interest point detectors*, International Journal on Computer Vision **60** (2004), no. 1, 63–86.

12 **Mikolajczyk K, Schmid C**, *A performance evaluation of local descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (2005), no. 27, 1615–1630.

13 **Prohaszka Z**, *Affine Invariant Features from Self-Flow*, RAAD 17th International Workshop on Robotics in Alpe-Adria-Danube Region (Ancona, Italy), 2008, pp. 1–10.
`http://mycite.omikk.bme.hu/doc/40788.pdf`.

14 ______, *Matching Image Details with the Self Affine Feature Transform*, V. Magyar Szamítógépes Grafika és Geometria konferencia (Budapest, Hungary), 2010, pp. 206–213.
`http://mycite.omikk.bme.hu/doc/83793.pdf`.

15 **Rahtu Esa, Salo Mikko, Heikkil Janne, Flusser Jan**, *Generalized affine moment invariants for object recogn*, 18th International Conference on Pattern Recognition (ICPR'06), 2006, pp. 634–637.

16 **Shi J, Tomasi C**, *Good Features to Track*, 9th IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.

17 **Shih J L, Chen L H**, *Color Image Retrieval Based on Primitives of Color Moments*, Recent Advances in Visual Information Systems (2002), 19–27, `http://dx.doi.org/10.1007/3-540-45925-1_8`.

18 **Tuytelaars T, Proesmans M, Gool L V**, *The cascade Hough transform*, ICIP, 1998, pp. 736–739.

19 **Zhang Z**, *A flexible new technique for camera calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 11, 1330–1334.

11 Appendix

11.1 Frequently used variables

	size	description	Eq.
$\mathbf{C}$	2×2	lower-right part of $\mathbf{M}$	(13)
$\mathbf{D}$	$n \times n$	general diagonal matrix	
$\mathbf{F}$	2×2	linear part of $\mathbf{Q}$	(3)
$\mathbf{f}$	4×1	translation free parameters of affine flow, elements of $\mathbf{F}$	(3)
$\mathbf{g}(x, y)$	2×1	gradient of the image at x,y	
$\mathbf{G}_K$	3×3	domain-moment of gradient strength	(23)
$\mathbf{I}$	$n \times n$	identity matrix	
$\mathbf{M}, \hat{\mathbf{M}}$	6×6	SAFT descriptor matrix	(10)
$\mathbf{p}_\mathrm{H}$	3×1	Homog. planar position in current frame $[x \;\; y \;\; 1]^\mathsf{T}$	(1)
$\mathbf{Q}$	2×3	Matrix of affine flows	(1)
$\mathbf{Q}_\mathrm{H}$	3×3	Homog. extension of $\mathbf{Q}$	(3)
$\mathbf{q}, \hat{\mathbf{q}}$	6×1	affine flow parameters, elements of $\mathbf{Q}$	(1)
$\mathbf{R}$	2×2	Linear component of $\mathbf{T}_\mathrm{H}$, not necessarily orthogonal!	
$\mathbf{S}, \hat{\mathbf{S}}$	6×6	Matrices transforming $\mathbf{q}, \mathbf{M}$ and $\hat{\mathbf{q}}, \hat{\mathbf{M}}$	(6),(7)
$\mathbf{T}_\mathrm{H}$	3×3	Homog. affine coordinate transformation matrix	(4)
$\mathbf{t}$	2×1	constant shift in $\mathbf{Q}$	(3)
$\mathbf{U}$	$n \times n$	general Unitary matrix (of column-eigenvectors)	
$\mathbf{v}$	2×1	local flow velocity	(1)
$\tilde{\mathbf{W}}$	2×2	2^{nd} range-moment of flow velocity	(12)
$\mathbf{W}_\mathrm{H}$	3×3	Homogeneous domain-moment of sqared flow velocity	(20)
$w(x, y)$	1×1	windowing function of the image at x,y	