

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK
Lehrstuhl für Mustererkennung und Bildverarbeitung

Equivariant Holomorphic Filters -
Theory and Applications

Internal Report 3/07

Marco Reisert

April 2007

Equivariant Holomorphic Filters - Theory and Applications

Marco Reisert
Computer Science Department
Albert-Ludwigs-University Freiburg
79110 Freiburg, Germany
reisert@informatik.uni-freiburg.de

April, 2007

Abstract

This article presents a new class of nonlinear image filters, which can be used in various application domains in early vision. The filters are equivariant with respect to the Euclidean motion. To obtain equivariance we use the tool of group integration. Group integration offers a constructive way to compute invariant features in various spaces. In this work it is used to define a projection that maps any image transformation on its equivariant part. Applying this concept on an image transformation based on a holomorphic kernel we are able to derive a class of filters which can be computed very efficiently. We show relations of our approach with Volterra filters, steerable filters and complex moments. In the context of object detection the filter may be interpreted as some kind of generalized Hough transform.

The proposed algorithm is used for enhancing noisy contours and rapid object detection in microscopical images. The obtained results are compared to state-of-art approaches.

1 Introduction

In image processing the term 'filter' is always related to a special class of image transformation, which is characterized by the fact that its members are equivariant with respect to the group of translations. If \mathcal{F} is an image transformation, then it is said to be equivariant with respect to some group \mathcal{G} , if $g\mathcal{F}(\mathbf{x}) = \mathcal{F}(g\mathbf{x})$ holds for all images

\mathbf{x} and all $g \in \mathcal{G}$. Here the expression $g\mathbf{x}$ denotes the action of the group on the image \mathbf{x} . If \mathcal{F} is linear in \mathbf{x} and \mathcal{G} is the group of translations, it is just a convolution of the image with some kernel function known as the impulse response. For nonlinear image transformation this concept is generalized by the so called Volterra filters.

In this work we develop image transformations that are not only equivariant with respect to translations but also with respect to rotations in the image plane. That is, we consider the special Euclidean group of motion $SE(2)$ as the equivariance group. For linear filters the generalization is straight-forward, the only further restriction to an ordinary linear filter is that the impulse response has to be rotationally symmetric. For nonlinear transformation the answer is not quite as simple. We need some kind of generalization of Volterra's principle to $SE(2)$. It will turn out that the concept of group integration gives us such a tool by hand. The so called Reynolds operator $\mathbf{R}_{\mathcal{G}}$, which is based on the group integral, provides a projection on a certain linear subspace, which will be associated with the space of \mathcal{G} -equivariant image transformations.

If one has to deal with two dimensional rotations, complex analysis is known to be a powerful mathematical tool. For a fast and cheap computation of our filter we propose a special type of kernel function, which has its origin in complex analysis. Basically, it is a gaussian-windowed, holomorphic function. The gaussian controls the locality and scale of the filter, while the holomorphic part determines the filter characteristics. Applying the projection operator $\mathbf{R}_{\mathcal{G}}$ on such a holomorphic image transformation yields the desired $SE(2)$ -equivariant filter.

The paper is organized as follows: in the preceding subsection we give a few references to prior work which is related to ours. In the following we shortly give an introduction to the principles of group integration and how to obtain best invariant approximations. In Section 3 we present the holomorphic filter and show an efficient way to implement it. Section 4 proposes some simple variants and modifications of the framework and gives more insights into the relation of the proposed class of filters to Volterra filters and steerable filters. Then we propose how to train such a filter by an ordinary regression scheme. And then finally we present basically two types of experiments: the enhancement of noisy contours and the detection and analysis in microscopical images.

1.1 Related Work

The idea of group integration (GI) to obtain invariant representation has its origin in classical invariant theory. In pattern recognition it is widely used to obtain invariant representation that can be used for indexing large image or shape databases for fast retrieval. For an introduction to GI in the field of pattern recognition see [2, 3]. Application for shape retrieval can be found in [4] or [5]. We will use GI to project a

nonlinear image transformation on a rotation- and translation equivariant transformation (an introduction into the mathematical details can be found in [1]).

Volterra filters are the canonical generalization of the linear convolution to a nonlinear mapping. They are widely used in the signal processing community and also find applications in image processing tasks. In [6] a general framework for edge enhancement is proposed which is based on the idea of Volterra filters. In [7, 8] several image processing applications are discussed, like image sharpening, texture discrimination and nonlinear interpolation. In [9] Volterra filters are used to detect noisy images of curves. The filter proposed in this work might be interpreted as a kind of 'joint' Volterra filter, for translation and rotation.

Steerable filters, introduced in [10], are a common tool in early vision and image analysis. A generalization for non-group like deformations was proposed in [11] using an approximative scheme. Applications of steerable filters are widespread, e.g. in local orientation analysis [12], 2D rotation invariant object recognition [13, 14] and feature detection [15]. For 2D rotations steerable filters get a very simple form in complex notation and are closely related to complex filters [16] and complex moments [17]. We will see that our filter computes a certain subset of gaussian-windowed complex moments in a first step.

The generalized Hough transform [18] is a major tool for the detection of arbitrary shapes. Many modern approaches [19, 20] for object detection and recognition are based on this idea that local parts of the object cast votes for the putative center of the object. If the proposed algorithm is used in the context of object detection, it may be interpreted as some kind of voting procedure for the object center. This interpretation will later help us to design the scale parameters of the filter.

2 Group Integration

Let H be some vector space. The objects $\mathbf{v} \in H$ living in this space need not to be interpreted as images. They are rather mappings from images to images. At this point we are not yet interested in the details, just consider H as some arbitrary linear space. Further let \mathcal{G} be some compact group acting on the vectors $\mathbf{v} \in H$ by some representation \mathbf{T}_g (for basics in group representation theory with application to image processing see [21]). We call the set of all points that stay unchanged during the action of the group

$$H^{\mathcal{G}} = \{\mathbf{v} \in H \mid \forall g \in \mathcal{G} : \mathbf{v} = \mathbf{T}_g \mathbf{v}\}. \quad (1)$$

For example, consider the three dimensional rotations acting on the \mathbb{R}^3 . If we rotate around some fixed axis through the origin, then the rotation axis itself is the space $H^{\mathcal{G}}$.

It is obviously a linear space, because the group acts linearly on H . Consider a vector \mathbf{v} that is not lying on the rotation axis. What is the best approximation of \mathbf{v} in $H^{\mathcal{G}}$, i.e. what is the solution of the optimization problem

$$\min_{\mathbf{w} \in H^{\mathcal{G}}} \|\mathbf{v} - \mathbf{w}\|^2 \quad (2)$$

For the rotation example it is a very simple question. It is just the orthogonal projection of \mathbf{v} on the rotation axis. But how does this projection look in general for arbitrary groups or representations, respectively? To find such a projection we have to introduce the principle of group integration. Let $f : \mathcal{G} \rightarrow \mathbb{C}$ be some function defined on the group \mathcal{G} . The left invariant Haar integral over f is denoted by $\int_{\mathcal{G}} f(g) dg$. The left invariance means that $\int_{\mathcal{G}} f(hg) dg = \int_{\mathcal{G}} f(g) dg$ for all $h \in \mathcal{G}$. For compact groups we can assume that the measure is normalized such that $\int_{\mathcal{G}} dg = 1$ (for more about Haar integrals we refer to [22]). The canonical projection on $H^{\mathcal{G}}$ is known as the Reynolds operator (see e.g. [1]). It just performs a simple averaging over the whole group,

$$\mathbf{R}_{\mathcal{G}} = \int_{\mathcal{G}} \mathbf{T}_g dg.$$

A result from classical invariant theory is that the best approximation of a vector \mathbf{v} with the fix point property $\mathbf{v} = \mathbf{T}_g \mathbf{v}$ is the projection via the above defined group integration operator.

Unfortunately, the above theory is only well developed for compact groups. For non-compact groups the Haar measure is not normalizable and $\mathbf{R}_{\mathcal{G}}$ is not bounded anymore and everything breaks down. Nevertheless, in the next section we transfer the introduced concept to the non-compact group $SE(2)$ and we will see that the projections of particular kind of objects are bounded and well-behaved.

3 Holomorphic Filters

The image function is represented by a square integrable complex function defined on the complex plane \mathbb{C} . It is denoted by \mathbf{x} , an element of $L_2(\mathbb{C})$. The 'pixels' of \mathbf{x} , i.e. its function values are written in unbold face $x(z) = (\mathbf{x})(z)$. The area measure in the complex plane is denoted by $dz\bar{z}$, which is in ordinary cartesian $dz\bar{z} = du dv$, where $z = u + \mathbf{i}v$. For further introduction in complex analysis see e.g. [23]. By calligraphic letters, e.g. \mathcal{A} , we denote image transformations, i.e. mappings from $L_2(\mathbb{C})$ into itself. The space of image transformations has to be associated with the space H from the last section.

The special Euclidean group usually acts on the image function by

$$(g\mathbf{x})(z) := x(e^{-i\phi}(z - t)), \quad (3)$$

where ϕ is a rotation angle and t a translational shift. We use a small g for denoting the group representation to distinguish it formally from the naturally induced group action on an image transformation \mathcal{A} which is given by

$$(\mathbf{T}_g\mathcal{A})[\mathbf{x}] := g\mathcal{A}[g^{-1}\mathbf{x}].$$

By this definition the fixpoint property $\mathbf{T}_g\mathcal{A} = \mathcal{A}$ from the last section is identical to the equivariance of the image transformation $\mathcal{A}[g\mathbf{x}] = g\mathcal{A}[\mathbf{x}]$.

As already pointed out we are interested in a nonlinear behavior of our filter. The natural extension of a linear image transformation to a nonlinear homogeneous of n th order can be written as follows

$$(\mathcal{A}[\mathbf{x}])(z_0) = \int_{\mathbb{C}^n} x(z_1) \dots x(z_n) a(z_0, z_1, \dots, z_n) dz_1 \bar{z}_1 \dots dz_n \bar{z}_n$$

The function a , the kernel, completely describes the transformation. We propose to use kernels of the following form

$$a(z_0, \dots, z_n) = h(z_0, \dots, z_n) e^{-\sum_{k=0}^n \lambda_k |z_k|^2}$$

where $\lambda_k \in \mathbb{R}$ and h is holomorphic in z_k for $k > 0$ and anti-holomorphic in z_0 , i.e. we can write h as follows

$$h(z_0, z_1, \dots, z_n) = \sum_{i_0, \dots, i_n} \alpha_{i_0, \dots, i_n} \bar{z}_0^{i_0} z_1^{i_1} \dots z_n^{i_n}, \quad (4)$$

where the sum is $(n+1)$ -fold with indices $i_k \geq 0$ that are bounded by some finite cutoff index. The $\alpha_{i_0, \dots, i_n} \in \mathbb{C}$ are some expansion coefficients which have to be learned. The form of the kernel might remind the reader of the well known complex moments (see [17] for a review) and complex filters (see [16]). The advantage of such filters is that the group acts irreducibly on the components, they do not mix during a rotation. The Gaussians control the scale behavior of the image transformation. For $k > 0$ the parameters λ_k determine at which size the input image is sampled. The parameter λ_0 controls the size of the output image. The function h is the mapping which assigns to each input image the corresponding output image. The proposed kernel has the advantage that it is separable, i.e. we can write the image transformation as follows,

$$(\mathcal{H}[\mathbf{x}])(z_0) = \sum_{i_0, \dots, i_n} \alpha_{i_0, \dots, i_n} \bar{z}_0^{i_0} e^{-\lambda_0 |z_0 - t|^2} \left(\prod_{k=1}^n \int_{\mathbb{C}} x(z_k) z_k^{i_k} e^{-\lambda_k |z_k|^2} dz_k \bar{z}_k \right),$$

which makes the computation tractable. The choice of this particular type of kernel is further driven by the following observation. In [11] Perona introduced the concept of computing optimal steerable approximations of certain image templates. He computed a bank of optimal filters for an elongated edge template, in fact, this bank is very similar to the kernels $z^i e^{-\lambda_0 |z_0 - t|^2}$ for $i = 0, \dots, n$. As edges are one of the most important image features it seems reasonable to choose such functions as the basis for our kernel function. And fortunately, there is a fast method to compute convolutions with $z^i e^{-\lambda_0 |z_0 - t|^2}$ -type of kernels. We will later show how they can be expressed by complex derivatives; these are much cheaper to compute than a full convolution or a approximation with separable filters as proposed in [11].

To make the image transformation \mathcal{A} equivariant with respect to the Euclidean motion we now want to use the proposed theory from the last section. Applying the Reynolds operator $\mathbf{R}_{\mathcal{G}}$ on \mathcal{A} yields

$$\mathcal{H}[\mathbf{x}] = \int_{SE(2)} (\mathbf{T}_g \mathcal{A})[\mathbf{x}] dg = \int_{SE(2)} g \mathcal{A}[g^{-1} \mathbf{x}] dg, \quad (5)$$

a $SE(2)$ -equivariant transformation. After inserting all the definitions from above we obtain

$$\begin{aligned} (\mathcal{H}[\mathbf{x}])(z_0) &= \frac{1}{2\pi} \int_0^{2\pi} \int_{\mathbb{C}} \sum_{i_0, \dots, i_n} \alpha_{i_0, \dots, i_n} (e^{i\phi} (\bar{z}_0 - \bar{t}))^{i_0} e^{-\lambda_0 |z_0 - t|^2} \\ &\quad \left(\prod_{k=1}^n \int_{\mathbb{C}} x(e^{i\phi} z_k + t) z_k^{i_k} e^{-\lambda_k |z_k|^2} dz_k \bar{z}_k \right) d\phi dt \bar{t} \end{aligned} \quad (6)$$

The most inner integral is a convolution of the input function with a $(\mathbf{g}_k^{(i)})(z) := z^i e^{-\lambda_k |z|^2}$. By reparametrizing the inner integral according to $z' = -ze^{i\phi}$ it can be rewritten as

$$\int_{\mathbb{C}} x(e^{i\phi} z_k + t) z_k^{i_k} e^{-\lambda_k |z_k|^2} dz_k \bar{z}_k = (-e^{-i\phi})^{i_k} (\mathbf{x} * \mathbf{g}_k^{(i_k)})(t),$$

where the asterisk $*$ denotes a convolution. Now we are able to compute the integral over ϕ independently. By collecting all the ϕ dependencies we have

$$\frac{1}{2\pi} \int_0^{2\pi} e^{i\phi(i_0 - \sum_{k>0} i_k)} d\phi = \delta_{i_0, i_1 + \dots + i_n} \quad (7)$$

This means that the sum in equation (6) gives only a contribution whenever $i_0 = i_1 + \dots + i_n$. This condition is very similar to the rotation invariance constraint known

in the context of complex moments (see e.g [17]). The outer integral in eq. (6) over t is also a convolution, but this time with the complex conjugate $\bar{\mathbf{g}}_0^{i_0}$. Concluding everything leads to

$$(\mathcal{H}[\mathbf{x}])(z_0) = \sum_{\substack{\sum i_l = i_0 \\ i_l \in \mathbb{N}}} \alpha_{i_0, \dots, i_n} (-1)^{i_0} \bar{\mathbf{g}}_0^{(i_0)} * \left(\prod_{k=1}^n (\mathbf{x} * \mathbf{g}_k^{(i_k)}) \right) \quad (8)$$

Let us summarize. At first several feature images $\mathbf{x}_k^{(i)} = \mathbf{x} * \mathbf{g}_k^{(i)}$ are computed. They act like some kind of neighborhood descriptors of each pixel. The larger the Gaussians in $\mathbf{g}_k^{(i)}$ the larger the corresponding neighborhood. All possible point-wise products of such descriptor images are computed. According to the 'degree' $i_1 + \dots + i_n = i_0$ of the product images they are convolved with the conjugates $\bar{\mathbf{g}}_0^{(i_0)}$. This approach achieves the rotation equivariance of the filter. Finally, everything is summed up weighted by the parameters α_{i_0, \dots, i_n} . The question remains how to design the filter parameters and how to choose the width of the Gaussians, which we will discuss later.

3.1 Differential Formulation

The convolutions with the functions $\mathbf{g}_k^{(i)}$ are computationally expensive, even if we use a Fast Fourier Transform (FFT) or a decomposition into separable filters to speed it up. We will use complex differential calculus to figure out a more efficient way. Actually the function $\mathbf{g}_k^{(i)}$ is proportional to the i th order complex derivatives of a gaussian $\mathbf{g}_k = \mathbf{g}_k^{(0)}$, that is

$$\frac{\partial \mathbf{g}_k^{(i-1)}}{\partial \bar{z}} = -\lambda_k \mathbf{g}_k^{(i)},$$

where the partial derivative with respect to \bar{z} is defined by $\frac{\partial}{\partial \bar{z}} = \frac{1}{2}(\frac{\partial}{\partial x} + \mathbf{i} \frac{\partial}{\partial y})$. The z -derivative is defined accordingly. Inserting this relation into equation (8) and using the fact that convolutions and derivations commute gives

$$(\mathcal{H}[\mathbf{x}])(z) = \sum_{\substack{\sum i_l = i_0 \\ i_l \in \mathbb{N}}} \beta_{i_0, \dots, i_n} \frac{\partial^{i_0}}{\partial z^{i_0}} \left(\mathbf{g}_0 * \prod_{k=1}^n \frac{\partial^{i_k} x_k}{\partial \bar{z}^{i_k}} \right),$$

where we used $\mathbf{g}_k := \mathbf{g}_k^{(0)}$ and further $x_k(z) := (\mathbf{x} * \mathbf{g}_k)(z)$ and

$$\beta_{i_0, \dots, i_n} := (-1)^{i_0} \alpha_{i_0, \dots, i_n} \prod_{k=0}^n (-\lambda_k)^{-i_k}.$$

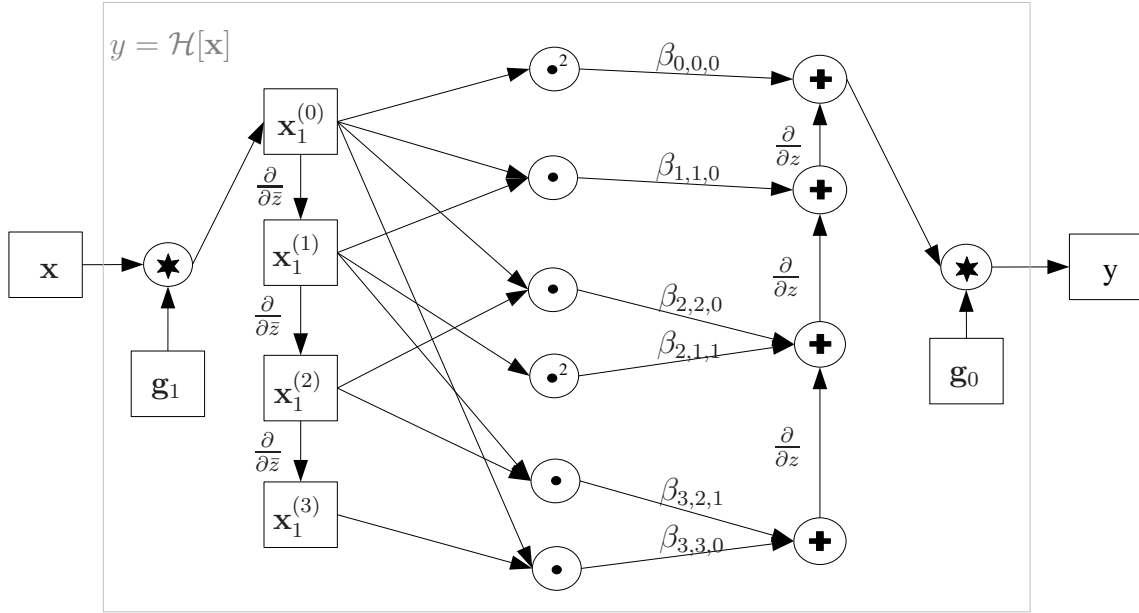


Figure 1: The workflow of a second order-filter ($n = 2$). The holomorphic function is expanded up to a degree of $m = 3$. The star '*' in the circle is indicating a convolution of the two incoming images. The dot '.' or the plus '+' indicate the point wise multiplications or addition of the incoming images, and a squared dot \cdot^2 a multiplication of the input with itself. The labels at the arrows indicate a multiplication or differentiation, respectively.

To speed up the algorithm we limit the number of convolutions by using just one size of Gaussians for the input, i.e. $\lambda_1 = \lambda_2 = \dots = \lambda_n$. So we only have to compute one convolution of the input image with $g_1^{(0)}$. And secondly, several index combinations which appear twice or more times can be discarded. For example, for a $n = 2$ (second order)-filter the monom for $\beta_{2,2,0}$ is the same as for $\beta_{2,0,2}$. In Figure 1 all index combination for a second-order filter with $m = 3$ are shown. If we now reformulate the filter by the use of the linearity of the derivative and linearity of the convolution we can obtain the following expression

$$(\mathcal{H}[x])(z) = g_0 * \left(\sum_{i_0=0}^m \frac{\partial^{i_0}}{\partial z^{i_0}} \sum_{\sum i_l = i_0} \beta_{i_0, \dots, i_n} \prod_{k=1}^n x_1^{(i_k)}(z) \right),$$

where we used the abbreviation $\frac{\partial^i x_1}{\partial \bar{z}^i} = x_1^{(i)}$. We only have to compute two convolutions with Gaussians, the remaining steps are computations of derivatives which can be performed quickly by the use of finite difference schemes. The whole computation

procedure is sketched in Algorithm 1.

Algorithm 1 Filter Algorithm $\mathbf{y} = \mathcal{H}[\mathbf{x}]$

Input: \mathbf{x}

Output: \mathbf{y}

- 1: Initialize filter output $\mathbf{y} := 0$
- 2: Convolve $\mathbf{x}_1 := \mathbf{g}_1 * \mathbf{x}$
- 3: Compute derivatives $(\mathbf{x}_1^{(i)})(z) := \frac{\partial^i \mathbf{x}_1}{\partial \bar{z}^i}$ for $i = 1, \dots, m$
- 4: **for** $i_0 = m : -1 : 1$ **do**
- 5:

$$\mathbf{y} := \mathbf{y} + \frac{\partial}{\partial z} \sum_{\sum i_l = i_0} \beta_{i_0, \dots, i_n} \prod_{k=1}^n \mathbf{x}_1^{(i_k)}$$

- 6: **end for**
 - 7: Let $\mathbf{y} := \mathbf{y} + \beta_{0, \dots, 0} \mathbf{x}_1$
 - 8: Convolve $\mathbf{y} := \mathbf{g}_0 * \mathbf{y}$
-

After a initial convolution with a gaussian we have to compute m derivatives with respect to \bar{z} . By an iterative scheme the number of outer z -derivatives can also be reduced to m . The number of multiplications is at most of order m^n . Of course there is much space for optimization by making use of intermediate results. In Figure 1 a workflow graph of the filter is shown. One can see that we only need to keep the m 'derivative'-images in memory. The rest can be computed in place.

4 Filter Variants and Related Filters

In this section we propose three modifications of the proposed filter and discuss relations to Volterra filters and steerable filters.

4.1 Mixed Holomorphic Filters

The condition arising from the rotation equivariance $i_1 + \dots + i_n = i_0$ can be quite restrictive for low order filters. But there is a easy way to generalize the filter such that already for low order filters the condition holds more often. We required the filter kernel $h(z_0, \dots, z_n)$ in equation (4) to be holomorphic in z_k for $k > 0$. We can also be less restrictive by just demanding that it is holomorphic or anti-holomorphic. As a consequence the indices in the equivariance condition may also give a negative contribution which leads to much more possibilities. If the input image $x(z)$ is real the

actual algorithm has to be modified only slightly, because we know that $\frac{\partial x}{\partial z} = \frac{\bar{\partial x}}{\partial z}$. So we do not only form products of the derivatives $x_1^{(i)}$ but also mixed products of $x_1^{(i)}$ and $\bar{x}_1^{(i)}$. With a little abuse of notation, namely $x_1^{(-i)} = \bar{x}_1^{(i)}$ this is gently incorporated in our formal framework.

4.2 Orientation Filters

Up to now we have only considered 'scalar' images, which meant that the transformation behavior of the input and output image is according to equation (3). It is no problem to extend the proposed filter to different transformation behaviors. As an example we want to design a filter which maps a 'scalar' input image on an 'orientation' image that describes some kind of directional quantity. Later in the experiments we use such a kind of filter to jointly detect a structure and also estimate its orientation in one step. If the orientation is encoded by the phase of a complex number, the 'orientation' image has to behave like

$$(gy)(z) := e^{i\phi}y(e^{-i\phi}(z-t)). \quad (9)$$

The corresponding algorithm for the filter is very similar to the original one. During the derivation of the filter in the Section 3 only equation (7) changes, i.e. the index combination has to fulfill now $i_0 + 1 = i_1 + \dots + i_n$.

Of course, also the input image can be an 'orientation' image, for example the gradient image. Generalization from 'orientation'-like images to higher order tensor-images with transformation behavior $(gy)(z) := e^{ir\phi}y(e^{-i\phi}(z-t))$ for $r > 1$ are also worth to consider and can be included in our framework without problems.

4.3 Non Homogeneous Filters

The proposed filter is homogeneous of order n , i.e. for any number $\gamma \in \mathbb{C}$ ($\lambda \in \mathbb{R}$ for a mixed holomorphic filter) it holds that $\mathcal{H}[\gamma\mathbf{x}] = \gamma^n\mathcal{H}[\mathbf{x}]$ (homogeneity is just equivariance with respect to the group of gray value scalings). In [24] advantages of homogeneous filters and construction principles are discussed. It depends on the application whether homogeneity is desired or not. But it is no problem to generalize the proposed filter to non homogeneous filters, which are sums of filters of different orders.

4.4 Relation to Volterra Filters

Volterra filters are well known in the field of image and signal processing. They are the nonlinear extension of the linear convolution. In our terms, they are translation-

equivariant image transformations. One can show that, if we restrict the integration range in equation (5) to be just the group of translations, we arrive at a particular kind of Volterra filter of order n , which arises from the holomorphic kernel. As we are also considering rotation equivariance, the filter may also be interpreted as some kind of Volterra filter with respect to the rotation group. And that is actually the case. Imagine the partial derivatives $\frac{\partial^i x_k}{\partial \bar{z}^i}$ as the frequency representation of a function defined on a circle. The function is representing the neighborhood of some point z in the image plane. The index i corresponds to the angular frequency number of the neighborhood, the index k can be associated with the radial part of the function. The mapping, which assigns to each neighborhood the expressions

$$\sum_{\sum i_k = i_0} \beta_{i_0, \dots, i_n} \prod_{k=1}^n x_k^{(i_k)}(z) \quad \text{for } i_0 \geq 0 \quad (10)$$

can be interpreted as a Volterra filter in a frequency domain representation. The condition $\sum i_k = i_0$ assures the rotation equivariance. It corresponds to the constraint that the filter kernel only depends on the relative quantities between input and output variables as it is known from ordinary Volterra filters.

4.5 Relation to Steerable Filters

Steerable filters are well known in the pattern recognition community. We show that in complex analysis the theory of steerable filters for 2D rotations gets much more simple and is directly related to our work. In fact, in complex theory the transformation behavior (the steering) has a canonical form in terms of the irreducible representations of the rotation group corresponding to the behavior known from Fourier transformations (see e.g. [21]).

In cartesian coordinates $(u, v) \in \mathbb{R}^2$ a steerable filter \mathbf{s} is a function of the form

$$s(u, v) = \sum_{k=1}^M \sum_{i=0}^k \eta_{k,i} \frac{\partial^{k-i}}{\partial u^{k-i}} \frac{\partial^i}{\partial v^i} g(u, v).$$

By this the convolution of an image function \mathbf{x} with a rotated version of \mathbf{s} can be expressed by a linear combination of the individual filter responses

$$\mathbf{x}^{(k,i)} = \mathbf{x} * \frac{\partial^{k-i}}{\partial u^{k-i}} \frac{\partial^i}{\partial v^i} g(u, v)$$

The function $g(u, v)$ is an arbitrary rotationally symmetric function. Usually the weights of the linear combination are a cumbersome expression depending on the

rotation angle. They are higher order tensor products of the representations of the two dimensional rotation group. The reduction of such higher order representation to irreducible ones is a well known task in group representation theory. For the two dimensional rotation group the irreducible representation is known to be $e^{i\phi i}$ for $i \in \mathbb{Z}$. By choosing the filter coefficients as

$$\eta_{k,i} = \xi_k \frac{1}{2^k} \binom{k}{i} (-\mathbf{i})^{n-k}$$

and using the binomial formula, we arrive at

$$s(u, v) = \sum_{k=1}^M \xi_k \underbrace{\left(\frac{1}{2} \left(\frac{\partial}{\partial u} - \mathbf{i} \frac{\partial}{\partial v} \right) \right)^k}_{(\mathbf{g}_0^{(k)})_{(u+iv)}} g(u, v).$$

If we choose $g(u, v)$ to be a Gaussian then we get the complex Gaussian derivatives $\mathbf{g}_0^{(k)}$ which we already know from our holomorphic filter. Now, the steering property becomes very simple

$$\mathbf{x} * (g_\phi \mathbf{s}) = \sum_{k=1}^M \xi_k e^{i k \phi} \mathbf{x}^{(k)},$$

where the individual filter responses are $\mathbf{x}^{(k)} = \mathbf{x} * \mathbf{g}_0^{(k)}$ and $(g_\phi \mathbf{s})(z) := s(e^{-i\phi} z)$ is defined as usual. This means that the first step in our filter algorithm, the computation of the derivatives, is nothing else than a particular filter response of a complex steerable filter.

5 Filter Design

The question arises how to adapt the filter parameters $\boldsymbol{\beta} = (\beta_{i_0, \dots, i_n})$ to a specific problem. In this work we follow the traditional approach of minimizing the mean square error. It is widely used in signal processing community in the context of adaptive nonlinear filters (see e.g. [25]). While in signal processing mostly iterative algorithms and stochastic gradient approaches are common, we use the closed form solution. For a given input image \mathbf{x} and a desired output image \mathbf{y} we want to minimize

$$J(\boldsymbol{\beta}) = \|\mathcal{H}_\beta[\mathbf{x}] - \mathbf{y}\|^2.$$

As \mathcal{H}_β is linear in $\boldsymbol{\beta}$ it is simple to find an explicit expression for the optimum. Let \mathbf{u}_j be the column vector containing the monoms for the j th pixel, that is the vector with

components

$$(\mathbf{u}_j)_I = \left(\frac{\partial^{i_0}}{\partial z^{i_0}} \prod_{k=1}^n x_1^{(i_k)} \right) (z_j),$$

where I is a multiindex ranging over all $(i_0, \dots, i_n) = I$ with $\sum i_k = i_0$ and z_j are the discrete pixel coordinates in complex notation. Further let $v_j = (\mathbf{y})(z_j)$ be the values of target image at position z_j , then the solution of the optimization problem can be written as follows

$$\boldsymbol{\beta} = \left(\sum_j \mathbf{u}_j \mathbf{u}_j^\dagger \right)^{-1} \sum_j v_j \mathbf{u}_j,$$

where j ranges over all discrete image sites.

As the number of filter parameters is usually much lower than the number of pixels in the image, the correlation matrix is in most cases well behaved and regular, such that the inversion makes no difficulties. But in some cases regularization is necessary. We found that ordinary regularization schemes by $J(\boldsymbol{\beta}) \mapsto J(\boldsymbol{\beta}) + \rho \|\boldsymbol{\beta}\|^2$ with the L_2 -norm do not work. The filter parameters are driven in ranges that are not reasonable, because by using the standard norm it is implicitly assumed that the monoms $(\mathbf{u})_I$ are uncorrelated, which is actually not the case. A more reasonable assumption is that the input pixels $x(z_j)$ are pairwise uncorrelated and Gaussian distributed. To find the corresponding norm for regularization we have to compute the expectation valued of $\mathbf{u}\mathbf{u}^\dagger$ under this assumption. But the computation of this matrix is quite cumbersome and involves computations of higher order cumulants of the gaussian distribution. To avoid this explicit computation we follow in practice a much more simple way; the input training image \mathbf{x} is artificially corrupted by adding gaussian distributed noise before training. This is done multiple times and the corresponding estimates for the correlation matrix are averaged. We found that this approach works very well in practice and theoretically converges to the desired regularization norm.

From an implementational viewpoint the training procedure itself is a little bit more costly than an application of the filter itself. Most of the time is spent on building the correlation matrix. For each valid multiindex I the feature image $(\mathbf{u}_j)_I$ has to be computed, which involves multiplications and multiple differentiations.

6 Experiments

Before starting with the experiments let us clarify some details of the implementation. For speed reasons, we use the FFT to perform the initial and final convolutions (line 2 and line 8 in Algorithm 1). As already pointed out we want to approximate the differentiation (line 3 and 4) by a finite difference scheme. Higher order derivatives are

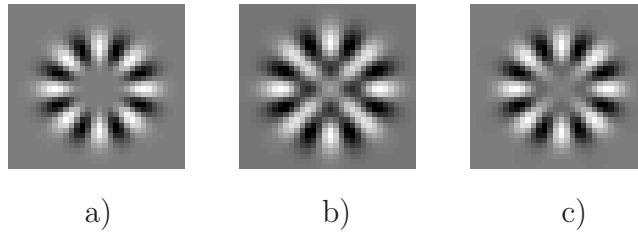


Figure 2: The real part of the function $\mathbf{g}_0^{(8)}$ in a 32×32 grid is shown. In image a) it is computed explicitly, in b) by a crude approximation with finite difference operator Δ_1 of first order. Image c) shows a approximation with second order finite differences by the use of Δ_2 .

obtained by multiple applications of the first order derivative. This approach is rather crude and inaccurate, because the approximation errors are accumulated by multiple applications of the rough approximation. But it helps to speed up the algorithm and for low orders the effect is not too hazardous. The important issue is that the errors behave 'isotropically', such that the rotation behavior and hence the rotation equivariance is not destroyed. In Figure 2 we try to illustrate the errors which occur when the expansion degree gets too high. We computed the function \mathbf{g}_0^8 in three different ways. First we computed it by the direct use of the formula $z^8 e^{-\lambda_0 |z|^2}$ in an accurate way. Then we iteratively apply a first order finite difference operator Δ_1 or alternatively a second order operator Δ_2 on the plain gaussian \mathbf{g}_0^0 . The difference operators are given by

$$\Delta_1 = \begin{pmatrix} 0 & \mathbf{i} & 0 \\ 1 & 0 & -1 \\ 0 & -\mathbf{i} & 0 \end{pmatrix} \quad \Delta_2 = \begin{pmatrix} 0 & 0 & \frac{-\mathbf{i}}{8} & 0 & 0 \\ 0 & 0 & \mathbf{i} & 0 & 0 \\ \frac{-1}{8} & 1 & 0 & -1 & \frac{1}{8} \\ 0 & 0 & -\mathbf{i} & 0 & 0 \\ 0 & 0 & \frac{\mathbf{i}}{8} & 0 & 0 \end{pmatrix}.$$

For details how to compute higher order approximations of derivatives see e.g. [26]. Figure 2 shows that the approximations obviously produce artefacts around the origin. These artefacts are not compliant with the original rotation behavior anymore. The accurate version in Figure 2 a) has a rotation symmetry of degree 8, the error introduced by the first order scheme has a rotation symmetry of degree 4, i.e. the rotation equivariance of the filter is partially destroyed. One can see that the second order scheme substantially reduces this error, while doubling the computationally load. In the experiments we exclusively used the first-order approximations. We found that, despite the errors even for low degrees, it makes no difference in practice whether we use Δ_1 or Δ_2 up to degrees of about 8.

Finally, some words about the handling of the boundary of the image domain. We


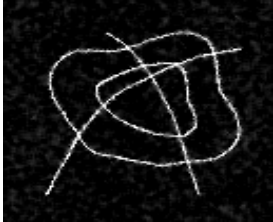


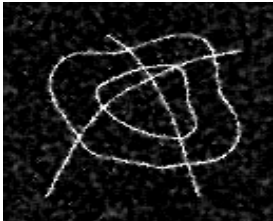
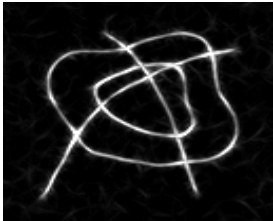
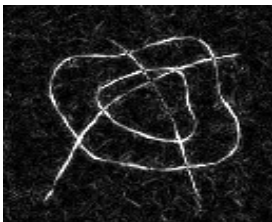
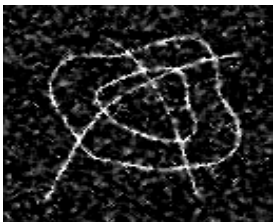
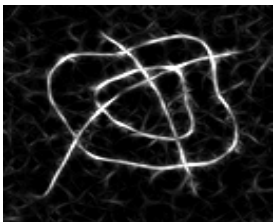
SNR	Holo. Filter	Wiener Filter	Steerable Filter
0.7			
0.4			
0.3			

Figure 3: Results for the contour denoising experiment. We compare our approach at different noise levels with an optimal Wiener filter and a steerable filter.

did not find a method which works fine in general. Any approach which makes use of modified finite differences at the borders of the image, makes an implicit assumption about the continuation of the image beyond the boundary. This leads, depending on the task and type of image, to undesired artefacts near the boundary. We found that a cyclic extension of derivatives, as one is used to get from FFT-based convolutions, is appropriate for most of the tasks.

All experiments are performed on a *Pentium 4, 2.8Ghz* with *MATLAB*. The time consuming parts are implemented in *C++* using the *MEX*-interface.

6.1 Enhancing Noisy Contours

Let us begin with a toy example. The filter should remove noise from contour images. It makes use of the prior knowledge that the original image was a contour before it was

distorted by additive noise. We use a non-homogeneous mixed-holomorphic second-order filter ($n = 2$) with a holomorphic expansion up to a degree of $m = 7$. For example, for $i_0 = 5$ we have the following monoms

$$x_1^{(5)}, x_1^{(0)} x_1^{(5)}, \bar{x}_1^{(1)} x_1^{(6)}, \bar{x}_1^{(2)} x_1^{(7)}.$$

These are all possible combinations such that $i_2 - i_1 = 5$, where $i_1, i_2 \in \{0, \dots, 7\}$. This means that we use a filter kernel $h(z_0, z_1, z_2)$ which is anti-holomorphic in z_1 and holomorphic in z_2 . Additionally we have the singleton $x_1^{(5)}$ belonging to the linear filter, which makes the filter non-homogeneous. Thus, the dimension of the parameter vector β is given by $(m + 1)(m + 2)/2 + (m + 1) = 44$.

We use gray scale images of size 128×128 . The gray value of the image varies between 0 and 1. The question arises how to choose the sizes of the Gaussians \mathbf{g}_0 and \mathbf{g}_1 . We used $\lambda_0 = \lambda_1 = 1$ measured in pixel units, i.e. the gaussian are rather small to preserve the sharpness of the contour. Also other values for λ_0, λ_1 work. The choice is always driven by the demands of the specific task.

For training we used an image with some painted white strokes on black background. The input training image is corrupted by uncorrelated additive gaussian noise with a signal to noise ratio of 0.3. The target image contains just the clean white strokes. Results on a toy image are shown in Figure 3. For comparison we show results obtained by an adaptive noise-removal Wiener filter (see [27], p.548, equations 9.44 – 9.46) and a steerable filter [10]. For the steerable filter we used an elongated Gabor-like line template in 56 orientations. The images were distorted by additive gaussian noise of different noise levels. The proposed holomorphic filter performs quite well. The output of the Wiener filter is more grainy and less impressive. The results of steerable filter are comparable to the holomorphic filter. However the response on the noise is somewhat different. The steerable filter tends to produce much more contour like fringes even in areas where no signal is present. Another observation is that the holomorphic filter is able to localize the contour a little bit more precisely than the steerable filter, which can be explained by the fact that for the steerable filter also slightly misplaced line templates can give contributions (of course, a non-maximum suppression would also make the response of the steerable filter more precise). As the holomorphic filter is explicitly trained to return a sharp contour such problems are avoided. Loosely spoken, the holomorphic filter computes implicitly a kind of deconvolution with a certain template to get this more precise response.

The steerable filter also provides a second information, the orientation of the detected line segments. This information can be extremely useful for further postprocessing. Until now the holomorphic filter only provides evidence for the presence of the contour but not its orientation. In Section 4.2 we explained how to design holomorphic filters that can return orientation-like images, where the orientation is encoded in the

complex phase of the response. We will now use this approach to build a holomorphic filter which also gives the direction of the detected line segments. First we have to determine according to which transformation law the output image has to be transformed. To find the right transformation behavior consider the rotational symmetry of the problem. A line template has obviously a rotational symmetry of degree $r = 2$, i.e. after a rotation with $2\pi/r = \pi$ it turns out to be the same. Hence, the filter will only be able to recover the orientation modulo π . So the output image has to obey the following transformation law

$$(g\mathbf{y})(z) := e^{i2\phi}y(e^{-i\phi}(z - t)),$$

where g is a rotation around the origin of angle ϕ . The implementation of the filter is the same as before, just the index combination are differently chosen, according to $i_0 + 2 = i_1 + i_2 - i_3$. The training procedure is also similar, but we need a different target image which also includes the orientation information. We constructed this information manually. The orientation is assigned to each pixel on the contour in the target image by multiplying it with a complex unit number representing the orientation. Additionally the target image gets a small smooth such that the width of the detected contour is similar to the one obtained by a steerable filter. The rest of the training procedure is exactly as before. Figure 4 shows the orientation responses for the holomorphic filter in comparison to the steerable filter at a sharp corner and a crossing. The holomorphic filter is able to handle the sharp corner a little bit better. The steerable filter produces some kind of overshoots due to the elongated line template. This can bring further postprocessing steps into difficulties, like a contour tracker. The effect can be attenuated with the use of a shorter line template. But then also the detection confidence for straight line segments drops. The holomorphic filter does not have such problems. It does not explicitly use a template but adaptively learns to detect contours that also can contain sharp corners. At the crossing the response of the holomorphic filter is a little bit confused and weak. The steerable filter does not have such problems. It is very unlikely to happen that the optimal orientation of the line template is not parallel to one of the dominant orientations.

6.2 Analyzing Pollen Grains

Analysis techniques for data acquired by microscopy typically demand for a rotation and translation invariant treatment. The microscopical images of particles like cells, pollen grains or spores have usually no predetermined orientation. In this experiment we use the holomorphic filter for the analysis of pollen grains.

Palynology, the study and analysis of pollen, is an interesting topic with very diverse applications like in Paleoclimatology or Forensics. An important feature of certain

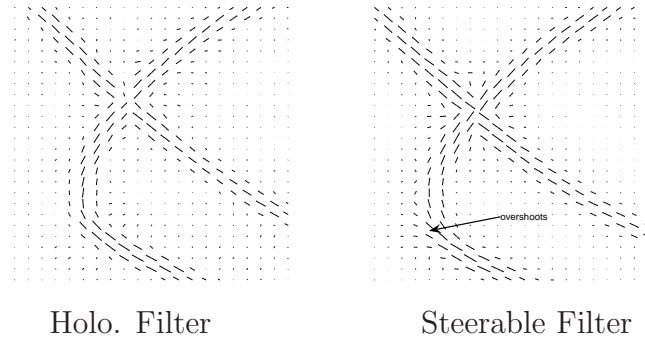


Figure 4: Comparison of the quality of the orientation estimates. The steerable filter shows overshoots at sharp corners. The holomorphic filter can overcome this problem. Therefore the holomorphic filter has small problems at the crossings.

types of pollen grain are the so called porates that are small pores on the surface of the grain. Their relative configuration is crucial for the determination of the species. We want to use the proposed filter to detect such porates. The input images are acquired by translucent microscopy, i.e. there may be varying illumination and contrast conditions. Such changes should not have an impact on the detection results. To make the filter invariant against global additive change of the gray values the filter must not depend on zero degree $x_1^{(0)}$ expressions because they carry the information about the local mean of the images. The contrast changes affect the images by a scaling of the gray values. We will not make the filter invariant against such a scaling but equivariant, i.e. we use a homogeneous filter. As the local maxima of the filter response serve as detection hypotheses a gray scale change will not affect the detection results if the filter is homogeneous.

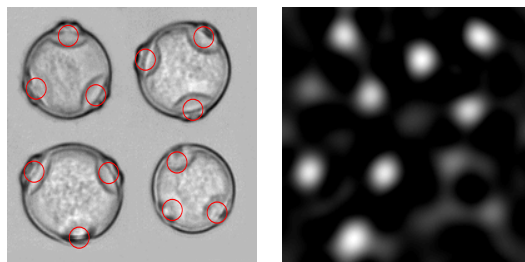


Figure 5: Example for the porate detection. On the left you see the original input image. On the right the output of the filter. For visualization filter responses below zero are set to zero. The local maxima of the filter output are marked in the left image by red circles.

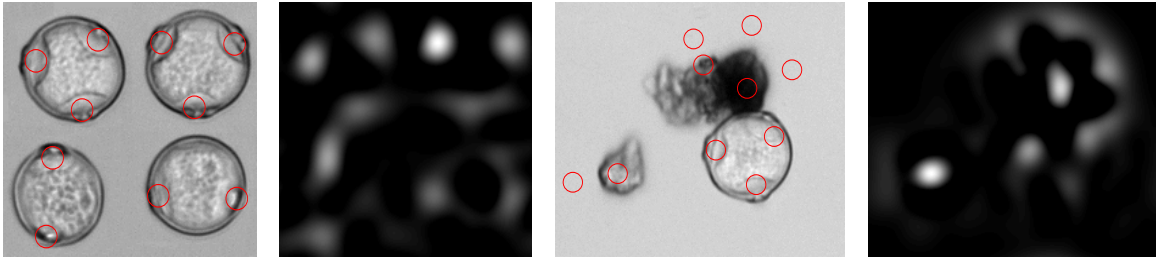


Figure 6: Results for the porate detection. Some of the birch pollen grains have only two porates because they were not deposited in equatorial pose. For very strong gradients, as in right image, the filter gets severe problems.

This time we use a third order filter ($n = 3$) with degree $m = 5$. The filter is again mixed holomorphic, two variables are chosen holomorphic, the third anti-holomorphic, i.e. we search for monomes that fulfill $i_0 = i_1 + i_2 - i_3$. We found 55 monoms fulfilling this selection rule, while not violating the gray value invariance constraint from above. The choice of λ_0 and λ_1 is motivated by interpreting the filter as some kind of generalized Hough transform [18]. Imagine that the object, in our case the porate, consists of several parts (just for imagination, the parts are actually the pixels of the object). Each part performs some kind of 'voting' for the putative center of the object. The parts are described by the derivatives $x_1^{(i)}$, which serve as local descriptors. The size of such hypothetical parts is determined by the width of the input gaussian \mathbf{g}_1 . The filter maps these local descriptions of the parts onto a 'voting' function for the object center. This mapping is actually the mapping described in equation (10). The size of the impact of the voting function depends on the parameter λ_0 . It has to be chosen, such that also the parts at the outer border has a influence on the decision for the object center. Hence, the width of the output Gaussian should be at least half of the diameter of the object. The images we use in this experiment are of size about 200×200 . A porate has an approximate length of 40 pixels (compare to Figure 5 and 6). So we used a output Gaussian with $\lambda_0 = \frac{1}{20^2}$. The input gaussian has a size of about $\lambda_1 = \frac{1}{2^2}$.

For the design of filter parameters β we used the eight birch pollen (four of them are shown in Figure 5). Each pollen grain possess 3 porates. The target output image is just an indicator image for the porate center. It contains everywhere zeros except at the object center location where the pixel is set to one. The object centers were manually labeled.

detect prec.	HOLO	SIFT	PCASIFT	INVFEAT
10 px	27%	48%	46%	49%
20 px	22%	26%	22%	48%

Table 1: Equal error rates for the porate dataset.

6.2.1 Results

The filter response for the training image is shown in Figure 5. The computation time for this filter is just under $200ms$. The local maxima of the filter response are marked in the original image by red circles. We only show up those local maxima that are above a certain threshold. Obviously, the filter performs very well for the test image. In Figure 6 we show some results for other images. In a clean environment (Figure 6 top) the porate are well detected (again with the same threshold as before). In the presence of dirt the filter gets problems. If the gradients are strong and well oriented along an edge (Figure 6 bottom) the problems get severe. At least for two of the three porates the local maxima are at the right position. The detection of the third upper right porate is shifted because of the strong influence of the responses from the neighboring gradients. Such problems are inherent if we use a homogeneous filter, because the filter response directly depends on the scale of the input gray values. If there are large gray values, or large gray value changes, the filter response gets also strong, regardless whether the shape or structure of the neighborhood is of interest or not. A possible way out would be a local normalization of the filter response or a preprocessing of the input images.

To measure the performance of our system we collected a test set of 150 segmented pollen grains with about 500 porates at all. The segmentation was done with a circular Hough transform. It may still happen that the pollen is impurified with dust and dirt particles, which may cause false positive detections. All porate centers were manually labeled. As the porates are not always in the equatorial pose it is sometimes difficult to define an objective ground truth.

We define a detection to be successful if the local maxima of the filter response is at most 10(20) pixels apart from the labeled center (a porate has a length of about 40 pixels). All local maxima of the filter responses are collected as detection hypotheses. The filter strength at the putative detection sites are assigned to each hypothesis.

We compared our approach with two different methods. In a first approach we extract SIFT-features at DoG-interest points (following [19]). For compactification of the features we used a PCA. Based on the SIFT(PCASIFT)-features we perform a GHT-like probabilistic voting procedure as done in the ISM model [20]. To achieve rotation invariance we steered the features at the gradient’s main orientation and cast votes relative to the orientation as it is done in [28, 29]. For training we used agglomerative

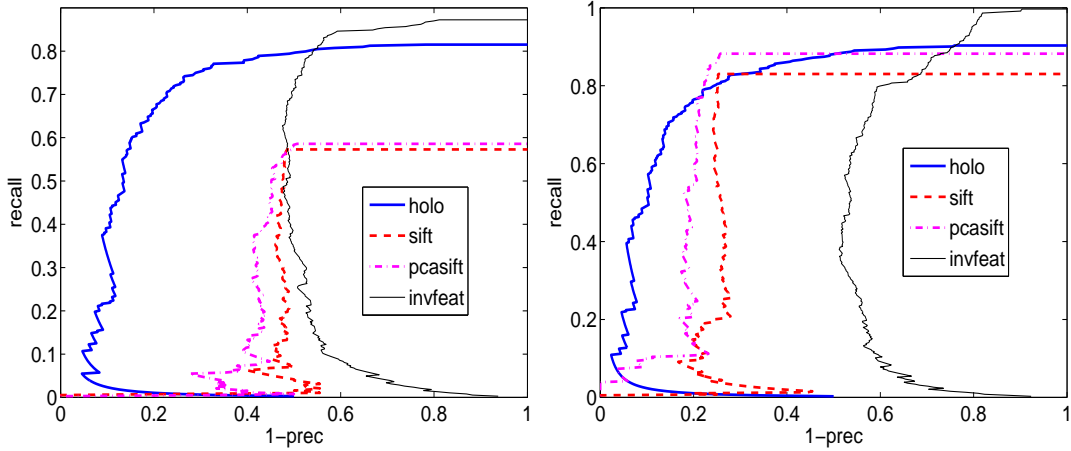


Figure 7: 1-Precision/Recall graphs for the porate test set. Left: with high detection accuracy of 10 pixels. Right: with low detection accuracy of 20 pixels.

clustering to obtain local appearance clusters. The training set is the same as for the holomorphic filter. As the porate dataset does not require for a scale invariant treatment the Hough voting map is only two dimensional, just the location of the object. Local maxima of the smoothed voting map serve as detection hypotheses, the absolute values of the maxima as a confidence value. We also tried to use a three dimensional voting map of object position times orientation, but we found that it mostly performed worse, because we got a lot of spurious local maxima and the final localization of the objects were very poor. Secondly, we used an approach (INVFEAT) which extracts a set of rotation invariant features for each pixel and classifies them whether they are an object center or not. A similar approach was used in 3D for detection of cell nuclei [30]. As features we use multiple complex derivatives, that is $f_{k,j} = \frac{d^{k+j} x_1}{dz^k d\bar{z}^j}$, up to an order of 8 resulting in $8 \cdot 9/2 = 36$ features per pixel. Rotation invariance is obtained by just taking the absolute value of the feature images $|f_{k,j}|$ as it was done in [16]. To keep the running times comparable to our approach we used a linear classifier for classification and the same training procedure as for our approach.

In Figure 7 we show the results in form of 1-Precision/Recall graphs. We make two runs with 10 and 20 pixels detection accuracy. In comparison to our approach (HOLO) the localizations of SIFT and PCASIFT detections are much more imprecise. This might be explained by the imprecise localization of the keypoints which are the basis for the subsequent voting. Our approach does not have such problems because all pixels are taken under consideration. For a low detection precision PCASIFT can slightly outperform HOLO in a certain threshold area. Only INVFEAT is able to

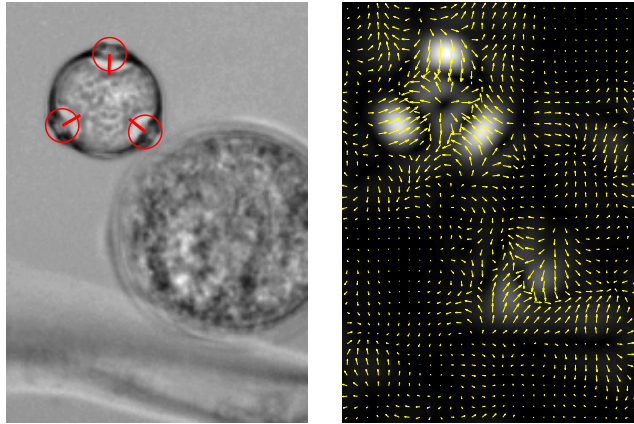


Figure 8: Results for the Orientation Filter. On the left the original image and the detection and its orientation are marked with a red circle. On the right the filter response is shown. The gray values indicate the magnitude of the response, the arrows the phase of the complex number.

finally detect nearly all porates by the cost of very high false positive rates.

We have already shown in the contour experiment that it is possible to build a filter, whose output is some kind of 'orientation' image. Again, we want to use this approach to detect the location of the porate as well as its orientation in the image plane. As the porate shows no rotational symmetry the output has to behave like an ordinary 'orientation' image, i.e. according to equation (9). The index combination has chosen according to $i_0 + 1 = i_1 + i_2 - i_3$. The target image for the training of the filter parameters is also very similar. The only difference is that instead of placing a 1 at each object center we place a $e^{i\varphi}$ pointing to the center of the current pollen. In Figure 8 we show the filter response for a test image. We selected the local maxima of the magnitude of the filter response as detection hypotheses. The porate are detected nicely and the orientation is right. But there is also a drawback of this approach. By taking the local maxima of the magnitude much more false positive detections can occur. In our first approach the sign (or complex phase) of the filter response was also carrying information about the presence of the object. Responses below zero were promptly neglected. Now, the complex phase is abused for the orientation estimation. We do not know whether the local maxima of the magnitude is a 'local minima' of the original response with the opposite orientation. But this drawback can easily be overcome by combining the 'detection'-filter with the 'orientation'-filter. And even the computational complexity does not double because intermediate results can be used

twice.

6.3 Detecting Fungal Spores

Asthma is one of the major respiratory diseases. There may be multiple factors, but it seems that sensitivity to *Alternaria* spores plays an important role with the onset asthma in certain areas [31]. *Alternaria* is one of the most common fungi worldwide, found literally everywhere, both indoors and outdoors. This makes the counting, detection and forecasting of *Alternaria* an important task. In this experiment we want to propose a fast method for detecting *Alternaria* spores in microscopical images by the use of the presented holomorphic filter.

We have demonstrated in Figure 6 that the filter gets severe problems in the presence of dirt. If we want to use the filter for object detection this is a real problem. Suppose that the objects we search for are located in images that are full of additional particles and dirt, we would get a lot of false positive detections. As mentioned above this problem is inherent with such a direct approach, because the filter response directly scales with the input gray values. We want to use a preprocessing step to attenuate this effect. Instead of working directly with the gray value image we compute the gradient image before and normalize softly with the magnitude as follows,

$$(\mathbf{d}_x)(z) = \frac{\frac{\partial x}{d\bar{z}}}{1 + \gamma \left| \frac{\partial x}{d\bar{z}} \right|},$$

with $\gamma > 0$. For large γ only the direction of the gradient remains. Note, that \mathbf{d}_x is $SE(2)$ -equivariant, i.e. $g\mathbf{d}_x = \mathbf{d}_{g\mathbf{x}}$, where the group action $g\mathbf{x}$ on the right hand side obeys eq. (3) and on the left hand side $g\mathbf{d}_x$ to eq. (9). By using the orientation-like \mathbf{d}_x image as the input, we have again to modify the filter algorithm to satisfy the equivariance constraint. This time the index combination has to fulfill $i_0 - 1 = i_1 + i_2 - i_3$.



Figure 9: Positive Training Examples.

The training procedure is similar as before, merely also typical background parts which are around the spores are used for the training, this additionally regularizes the solution and makes the approach more discriminative. In Figure 9 the positive training

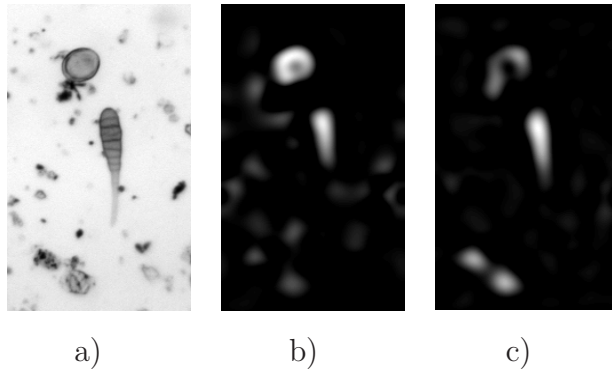


Figure 10: Comparison for different γ . In a) original image, in b) with $\gamma = 0$, in c) with $\gamma = 0.5$.

examples are shown. The training image for background is similar to the clutter in Figure 14. We used again a third-order filter with $m = 5$. The input and output gaussian we adapted according to the explanation in the last Section.

First we show some results concerning the choice of γ . In Figure 10 the filter responses for $\gamma = 0$ and $\gamma = 0.5$ are given. In both cases the filter finds the spore, but for $\gamma = 0$ we also get a equally strong response for the pollen grain in the upper left in the image. This is not the case if we choose $\gamma = 0.5$, the false responses are damped but also new are appearing that are not caused by strong gradients.

detect prec.	HOLO	SIFT	PCASIFT	INVFEAT
10 px	37%	44%	37%	60%
20 px	37%	29%	27%	60%

Table 2: Equal error rates for the spore dataset.

6.3.1 Results

For testing we labeled 50 spores manually in overall 20 images. By varying the threshold over the filter response we obtain the 1-Precision/Recall graphs in Figure 11 and 13. Figure 11 shows 1-P/R-graphs for different choices of γ . The preprocessing discussed above improves the results for a moderate choice of γ . For larger γ the performance drops. In Table 2 the obtained equal error rates are reported. In comparison to (PCA)SIFT we can observe the same behavior as for the porate data set. The (PCA)SIFT detection are much more imprecise, but the voting values are slightly

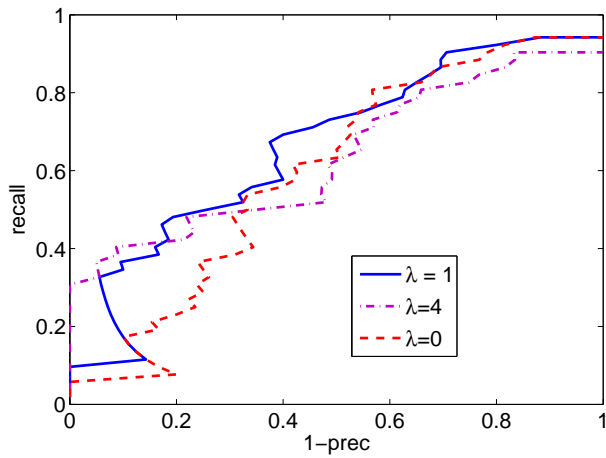


Figure 11: A Precision/Recall plot for different values of the normalization parameter γ .



Figure 12: Spores that were detected by our approach, but missed by (PCA)SIFT.

more reliable. Our HOLO approach has the advantage that the recall rates are better for low thresholds, that is most of the spores are finally detected. The missing detections by (PCA)SIFT are mostly due to missing keypoints caused by low contrast and fuzziness. This is an inherent problem with such keypoint based methods. In Figure 12 three examples of spores are shown that are detected by our approach but missed by the (PCA)SIFT approach.

Another advantage of our approach is that it is fast and does not depend on the complexity of the scene because it works in a pixel-wise manner. For an image of size 1400×1000 the filter needs only about 4 seconds (without initial and final blur). Suppose that such an image is full of non interesting airborne particles like in Figure 14 (a small 300×200 -sized clipping of the original image). The original SIFT implementation by Lowe needs about 11 seconds on our machine to detect and extract about 5000 keypoints.

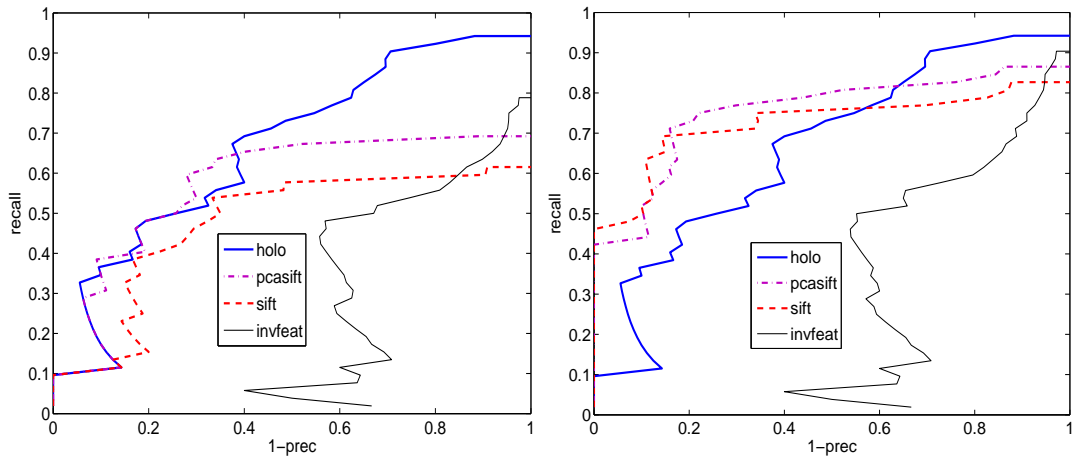


Figure 13: 1-Precision/Recall graphs for the sporen test set. Left: with high detection accuracy of 10 pixels. Right: with low detection accuracy of 20 pixels.

7 Conclusion

In this work we presented a rotation equivariant image filter. The filter can be interpreted as joint Volterra filter for rotations and translations. The approach is also deeply related to steerable filters. The output of the holomorphic filter is polynomial in terms of the individual filter responses of a special kind of steerable filter. Thereby the monoms are chosen such that the rotation equivariance is fulfilled. The filter design, i.e. the choice of polynomial coefficients, is accomplished by a training procedure based on a simple regression scheme. We also reported a fast, approximative computation of the proposed filter, which makes use of the correspondence of holomorphic functions with complex gaussian derivatives.

In a first experiment we used the filter for denoising of contours and compared the results to steerable filters. We found that the results are competitive and show superior behavior at sharp corners. An advantage of our adaptive approach is that the filter can easily be trained to fulfill the user’s demands. It does not rely on a predefined template as a steerable filter.

In a second experiment the filter is used for rotation invariant detection in microscopical images. Compared to a GHT-based approach relying on SIFT features our approach is competitive for the microscopical data. The holomorphic filter provides a much more precise localization of the object, because it does not rely on an intermediate representation by uncertain localized keypoints. Another drawback of the keypoint based approach is that in fuzzy regions of low contrast no keypoints are detected and

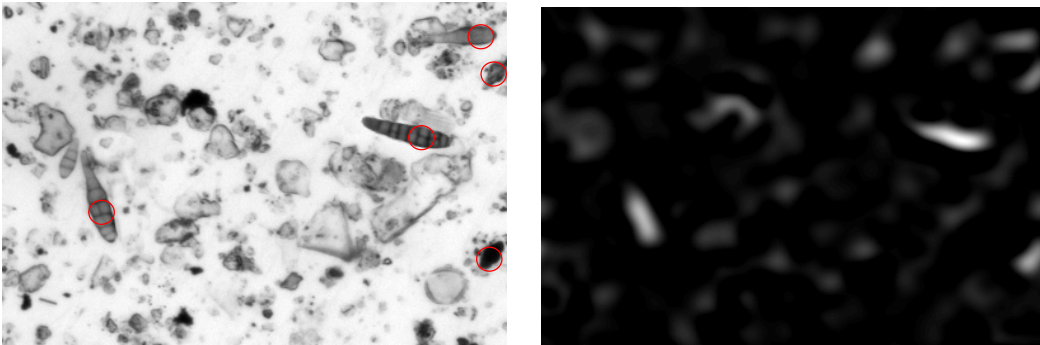


Figure 14: Detection results. On the top the input image is given. Detections with the EER-threshold are marked with red circles. The bottom image shows the filter response.

hence a detection becomes impossible. The holomorphic filter does not have such problems because all pixel are taken under consideration. Of course, for more complex vision problems the non-parametric GHT-based approaches will outperform the holomorphic filter, because the model complexity of the filter is very limited. But therefore, due to the small number of parameters, the filter is able to show better generalization ability for certain data.

In conclusion, the presented framework offers a handy tool for several low level vision problems that require for a rotation invariant treatment. There are many possible directions for future work. A generalization of the framework to multiple scales is straight forward. Different training procedures are also worth to consider, e.g. boosting. A composition of several filters in the manner of a convolutional neural network may also be interesting, whereas the training would be iterative and much more costly. In some way the holomorphic filter can be seen as a heavy weight interest point detector. In the context of keypoint based object detection it may be used instead of a DoG interest point detector, for example. In this context it may be possible to adapt it individually to a specific task and robustness demands. More generally, the holomorphic filter can be seen as the link between the classical interest point detectors that are only able to detect very simple structures and the more complex and expensive object recognition systems that are able to learn very complex models like the ISM.

References

- [1] D. Mumford, J. Fogarty, and F. Kirwan, *Geometric Invariant Theory*. Springer, 1994.
- [2] H. Schulz-Mirbach, “Anwendung von Invarianzprinzipien zur Merkmalgewinnung in der Mustererkennung,” Ph.D. dissertation, TU Hamburg Harburg, Feb. 1995, reihe 10, Nr. 372, VDI-Verlag.
- [3] H. Burkhardt and S. Siggelkow, *Invariant features in pattern recognition - fundamentals and applications. In Nonlinear Model-Based Image/Video Processing and Analysis*. John Wiley and Sons, 2001.
- [4] M. Reisert and H. Burkhardt, “Invariant features for 3d-data based on group integration using directional information and spherical harmonic expansion,” in *Proceedings of the ICPR06*, 2006.
- [5] —, “Using irreducible group representations for invariant 3d shape description,” in *Proceedings of the 28th DAGM Symposium, Berlin*, 2006.
- [6] S. Thurnhofer and S. Mitra, “A general framework for quadratic volterra filters for edge enhancement,” *IEEE Trans. Image Processing*, pp. 950–963, 1996.
- [7] V. J. Mathews and G. Sicuranza, *Polynomial Signal Processing*. J.Wiley, New York, 2000.
- [8] G. Sicuranza, “Quadratic filters for signal processing,” in *IEEE Proceedings*, vol. 80, no. 8, 1992, pp. 665–677.
- [9] J. August, “Volterra filtering of noisy images of curves,” in *Proceedings of the ECCV*, 2002.
- [10] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 9, 1991.
- [11] P. Perona, “Deformable kernels for early vision,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 5, pp. 488 – 499, 1995.
- [12] E. P. Simoncelli and H. Farid, “Steerable wedge filters for local orientation analysis,” *IEEE Trans. Image Processing*, vol. 5, no. 9, 1996.
- [13] D. Ballard and L. Wixson, “Object recognition using steerable filters at multiple scales,” in *Proceedings of IEEE Workshop on Qualitative Vision*, 1993, pp. 2–10.

- [14] X. Shi, A. R. Castro, R. Manduchi, and R. Montgomery, “Rotational invariant operators based on steerable filter banks,” *IEEE Signal Processing Lett.*, vol. 13, no. 11, pp. 684–687, 2006.
- [15] M. Jacob and M. Unser, “Design of steerable filters for feature detection using canny-like criteria,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 82, pp. 1007 – 1019, 2004.
- [16] F. Schaffalitzky and A. Zisserman, “Multi-view matching for unordered image sets, or ‘how do i organize my holiday snaps’,” in *Lecture Notes In Computer Science; Proceedings of the 7th European Conference on Computer Vision-Part I*, vol. 2350, 2002, pp. 414 – 431.
- [17] J. Flusser, “Moment invariants in image analysis,” in *Proceedings of the International Conference on Computer Science. ICCS’06*, 2006, pp. 196–201.
- [18] D. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13-2, 1981.
- [19] D. Lowe, “Distinct image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [20] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model,” in *Proceedings of the ECCV’04 Workshop on Statistical Learning in Computer Vision, Prague*, 2004.
- [21] R. Lenz, *Group theoretical methods in Image Processing*. Springer Verlag, Lecture Notes, 1990.
- [22] L. Nachbin, *The Haar Integral*. D. van Nostrand Company, Inc., Princenton, New Jersey, Toronto, New York, London, 1965.
- [23] R. Silverman, *Introductory complex analysis*. Dover Publications, 1972.
- [24] R. K. Pearson, “Scale-invariant nonlinear digital filters,” *IEEE Trans. Signal Processing*, vol. 50, no. 8, pp. 1986–1993, 2002.
- [25] V. Mathews, “Adaptive polynomial filters,” *IEEE Trans. Signal Processing*, vol. 8, no. 3, pp. 10 – 26, 1991.
- [26] F. B. Hildebrand, *Finite-Difference Equations and Simulations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

- [27] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ, Prentice Hall, 1990.
- [28] K. Mikolajczyk, B. Leibe, and B. Schiele, “Multiple object class detection with a generative model,” in *Proceedings of the CVPR*, vol. 1, 2006, pp. 26 – 36.
- [29] A. Teynor and H. Burkhardt, “Localization of visual object class instances,” in *accepted for publication on the MVA, Japan, 2007*.
- [30] O. Ronneberger, J. Fehr, and H. Burkhardt, “Voxel-wise gray scale invariants for simultaneous segmentation and classification,” in *In Proceedings of the 27th DAGM Symposium, Vienna, Austria, 2005*.
- [31] R. Bush and J. Prochnau, “*Alternaria*-induced asthma,” *J Allergy Clin Immunol*, vol. 113, no. 2, pp. 227–234, 2004.