

Holomorphic Filters for Object Detection

Marco Reisert, Olaf Ronneberger and Hans Burkhardt

University of Freiburg, Computer Science Department,
79110 Freiburg i.Br., Germany
reisert@informatik.uni-freiburg.de

Abstract. It is well known that linear filters are not powerful enough for many low-level image processing tasks. But it is also very difficult to design robust non-linear filters that respond exclusively to features of interest and that are at the same time equivariant with respect to translation and rotation. This paper proposes a new class of rotation-equivariant non-linear filters that is based on the principle of group integration. These filters become efficiently computable by an iterative scheme based on repeated differentiation of products and summations of the intermediate results. Our experiments show that the proposed filter detects pollen porates with only half as many errors than alternative approaches, when high localization accuracy is required.

1 Introduction

In image processing the term 'filter' is mostly related to the special class of image transformations that is characterized by the fact that they 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 a mathematical 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.

Complex calculus provides powerful mathematical concepts for the analysis of 2D rotation. For a fast and cheap computation of our filter we propose a special type of kernel function, which has its origin in complex calculus. 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.

The paper is organized as follows: in the following subsection we give references to work that is related to ours. In Section 2 we present the holomorphic filter and show an efficient way of implementation together with a training scheme for the filter parameters. Section 3 presents experiments on microscopical images. The task is to detect so called porates, which are small pores on the surface of pollen grains.

1.1 Related Work

The idea of group integration (GI) to obtain invariants has its origin in classical invariant theory. In pattern recognition it is widely used to obtain invariants 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]. Applications for shape retrieval can be found, e.g., in [9] or [10]. We will use GI to project a nonlinear image transformation onto a rotation- and translation equivariant transformation (see [11]).

Volterra filters are nonlinear transformations that are equivariant with respect to translations. They are widely used in the signal processing community and also find applications in image processing tasks (e.g. [14, 6]).

Steerable filters, introduced in [3], are a common tool in early vision and image analysis. For 2D rotations steerable filters get a very simple form in complex notation and are closely related to complex filters [12].

The generalized Hough transform (GHT) [1] is a major tool for the detection of arbitrary shapes. Many modern approaches [5, 4] 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 filter 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 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 $x(z) = (\mathbf{x})(z)$, where $z = u + \mathbf{i}v$ and (u, v) are the cartesian pixel coordinates. The complex conjugate is denoted by $\bar{z} = u - \mathbf{i}v$. The area measure in the complex plane is denoted by $dz\bar{z}$, which is in ordinary cartesian $dz\bar{z} = du dv$. For further introduction in complex analysis see, e.g., [13]. By calligraphic letters, e.g. \mathcal{A} , we denote image transformations, i.e. mappings from $L_2(\mathbb{C})$ into itself. The special Euclidean group usually acts on the image function by

$$(g\mathbf{x})(z) := x(e^{-\mathbf{i}\phi}(z - t)), \quad (1)$$

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 $\mathbf{T}_g \mathcal{A} = \mathcal{A}$ is identical to the equivariance of the image transformation $\mathcal{A}[g\mathbf{x}] = g\mathcal{A}[\mathbf{x}]$.

The natural extension of a linear image transformation to a non-linear, homogeneous transformation of n th order is given by

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

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 anti-holomorphic in z_k for $k < p$ and holomorphic for $k \geq p$, i.e. we can write h as follows

$$h(z_0, z_1, \dots, z_n) = \sum_{i_0, \dots, i_n} \alpha_{i_0, \dots, i_n} \bar{z}_0^{i_0} \dots \bar{z}_{p-1}^{i_{p-1}} z_p^{i_p} \dots z_n^{i_n},$$

where the sum is $(n+1)$ -fold with indices $i_k \geq 0$ that are bounded by some finite cutoff index m . The $\alpha_{i_0, \dots, i_n} \in \mathbb{C}$ are some expansion coefficients which have to be learned. The natural parameter p will later help us to design the rotation equivariance condition more freely. The choice of the above kernel is driven by the following observation. In [8] 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. Additionally, this type of kernel can be computed very quickly by the use of complex derivatives as we will see below.

To make the image transformation \mathcal{A} equivariant with respect to the Euclidean motion we use the principle of group integration. We integrate the basis image transformation \mathcal{A} over all possible group actions, i.e.

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

It is easy to show that, if \mathcal{H} converges then it is a $SE(2)$ -equivariant transformation. After inserting all the definitions from above we obtain

$$(\mathcal{H}[\mathbf{x}])(z_0) = \sum_{\substack{i_0 + \dots + i_{p-1} = \\ i_p + \dots + i_n}} \alpha_{i_0, \dots, i_n} (-1)^{i_0} \mathbf{g}_0^{(i_0)} * \left(\prod_{k=1}^n (\mathbf{x} * \mathbf{g}_k^{(i_k)}) \right), \quad (2)$$

where $*$ denotes a convolution and $(\mathbf{g}_k^{(i)})(z) = \bar{z}^i e^{-\lambda_k |z|^2}$ for $k < p$ and $(\mathbf{g}_k^{(i)})(z) = z^i e^{-\lambda_k |z|^2}$ for $k \geq p$ (for a detailed derivation see [11]). At first several feature images $\mathbf{x} * \mathbf{g}_k^{(i)}$ 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. According to the constraint $i_0 + \dots + i_{p-1} = i_p + \dots + i_n$, all possible point-wise products of such descriptor images are computed. This constraint achieves the rotation equivariance of the filter. Note, that for $p = 0$ the condition is unsatisfiable. Finally, everything is summed up weighted by the parameters α_{i_0, \dots, i_n} .

2.1 Differential Formulation

The convolutions with the functions $\mathbf{g}_k^{(i)}$ are computationally expensive, even if we use a fast Fourier transform 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 derivative of a Gaussian $\mathbf{g}_k := \mathbf{g}_k^{(0)}$, that is

$$\frac{\partial}{\partial \bar{z}} \left(z^{(i-1)} e^{-\lambda_k |z|^2} \right) = -\lambda_k z^i e^{-\lambda_k |z|^2},$$

where the partial derivative with respect to \bar{z} is defined by $\frac{\partial}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial}{\partial x} + \mathbf{i} \frac{\partial}{\partial y} \right)$. Correspondingly, the z -derivative is defined by $\frac{\partial}{\partial z} = \frac{1}{2} \left(\frac{\partial}{\partial x} - \mathbf{i} \frac{\partial}{\partial y} \right)$. Inserting this relation into equation (2) and using the fact that convolutions and derivations commute on an unbounded domain gives

$$(\mathcal{H}[\mathbf{x}])(z) = \mathbf{g}_0 * \left(\sum_{i_0=0}^m \frac{\partial^{i_0}}{\partial z^{i_0}} \sum_{\substack{i_0 + \dots + i_{p-1} = \\ i_p + \dots + i_n}} \beta_{i_0, \dots, i_n} \prod_{k=1}^n (\mathbf{x}_k^{(i_k)})(z) \right),$$

where we used the abbreviation $\mathbf{x}_k^{(i)} = \frac{\partial^i}{\partial \bar{z}^i} (\mathbf{x} * \mathbf{g}_k)(z)$ for $k < p$ and $\mathbf{x}_k^{(i)} = \frac{\partial^i}{\partial \bar{z}^i} (\mathbf{x} * \mathbf{g}_k)(z)$ for $k \geq p$. The parameters β_{i_0, \dots, i_n} are related to the former by $\beta_{i_0, \dots, i_n} := (-1)^{i_0} \alpha_{i_0, \dots, i_n} \prod_{k=0}^n (-\lambda_k)^{-i_k}$. For a detailed derivation see [11]. We only have to compute $n + 1$ convolutions with Gaussians, the remaining steps are computations of derivatives which can be performed quickly by the use of finite difference schemes. The computation is sketched in Algorithm 1.

After the initial convolutions with the Gaussians, we have to compute m derivatives. By an iterative scheme the number of outer z -derivatives can 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.

To further reduce the number of convolutions we assume that $\lambda_1 = \dots = \lambda_n$. So, we have to compute the derivatives only for one blurred image \mathbf{x}_1 . In Figure 1 a workflow graph of a filter with order $n = 2$ and $p = 1$ and $m = 3$ is shown. One can see that we only need to keep the m 'derivative'-images in memory. The rest can be accomplished in place.

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 $\mathbf{x}_k := \mathbf{g}_k * \mathbf{x}$ for $k = 1, \dots, n$.
- 3: Compute derivatives $\mathbf{x}_k^{(i)}$ for $k = 1, \dots, n$ and $i = 1, \dots, m$.
- 4: **for** $i_0 = m : -1 : 1$ **do**
- 5:

$$\mathbf{y} := \mathbf{y} + \frac{\partial}{\partial z} \sum_{\substack{\Sigma i_0 + \dots + i_{p-1} = \\ i_p + \dots + i_n}} \beta_{i_0, \dots, i_n} \prod_{k=1}^n \mathbf{x}_k^{(i_k)}$$

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

2.2 The Training of the Filter Parameters

As the filter is linear in the parameters $\beta = (\beta_{i_0, \dots, i_n})$ a simple linear regression scheme can be applied to adapt the parameters. For a given input image \mathbf{x} and a desired output image \mathbf{y} we have to minimize $J(\beta) = \|\mathcal{H}_\beta[\mathbf{x}] - \mathbf{y}\|^2$, which can be accomplished by solving the normal equations (for details see [11]).

3 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 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 compute the function $\mathbf{g}_0^{(8)}$ in three different ways. First it is computed 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

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

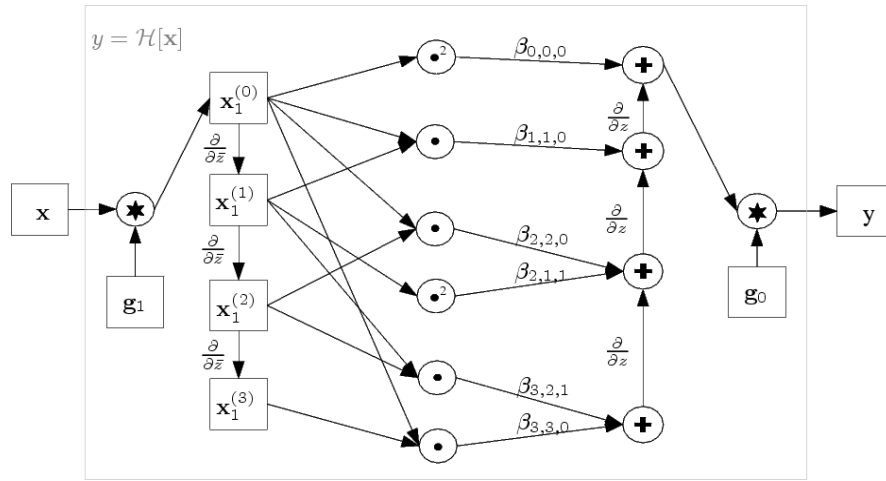


Fig. 1. The workflow of a second order-filter ($n = 2$, $p = 1$). The holomorphic function is expanded up to a degree of $m = 3$. The star \star in the circle is indicating a convolution of the two incoming images. The dot \cdot or the plus $+$ indicate the point wise multiplications or addition of the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

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 high 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.

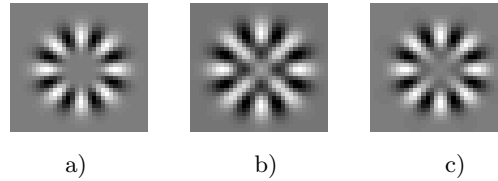


Fig. 2. The real part of the function $g_0^{(8)}$ in a 32×32 grid is shown. In image a) it is computed explicitly, in b) by a crude approximation with finite difference operator Δ_1 of first order. Image c) shows a approximation with second order finite differences by the use of Δ_2 .

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

3.1 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 pollen-forecasts or in forensics. An important feature of certain 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 transmitted light 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 additive change of the gray values the filter must not depend on zero degree $\mathbf{x}_1^{(0)}$ 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. As the local maxima of the filter response will serve as detection hypotheses, a gray scale change will not affect the detection results, because we use a homogeneous filter.

A third order filter ($n = 3$) with degree $m = 5$ is used. We choose $p = 2$, 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. One example is $(i_0, i_1, i_2, i_3) = (4, 1, 3, 2)$. Besides, finding all non-redundant monoms under certain constraints is not a trivial task. The choice of λ_0 and $\lambda_1 = \lambda_2 = \lambda_3$ is motivated by interpreting the filter as some kind of generalized Hough transform [1]. 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 part at position z is described by the derivatives $(\mathbf{x}_1^{(i)})(z)$, which serve as the 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 descriptors of the parts onto a 'voting' function for the object center. 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 have an 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 diameter of 40 pixels (compare to Figure 3). So we used an 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 eight betula (birch) pollen, four of them are shown on the left of Figure 3. Each pollen grain possess 3 porates.

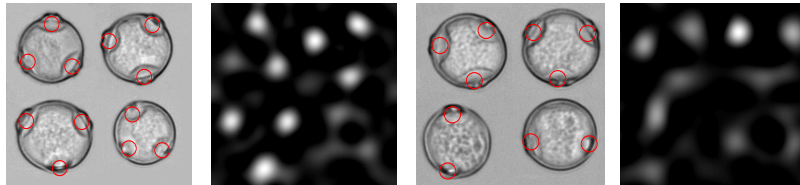


Fig. 3. Example for the porate detection. On the left you see the original input image. On the right the output of the filter. For visualization filter responses below zero are set to zero. The local maxima of the filter output are marked in the left image by red circles.

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.

3.2 Results

The filter response for the training image is shown on the left of Figure 3. 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 and also for an unknown image on the right of Figure 3.

To measure the performance of our system we collected a test set of 150 segmented pollen grains with about 500 porates at all. The pollen in the dataset are sometimes 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 diameter 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 as a confidence value.

We compared our approach with two different methods. In a first approach we extract SIFT-features at DoG-interest points (following [5]). 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 [4]. 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 [7]. For training we used agglomerative clustering to obtain local appearance clusters. The training set is the same as for the holomorphic filter. As the porate dataset does not require 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

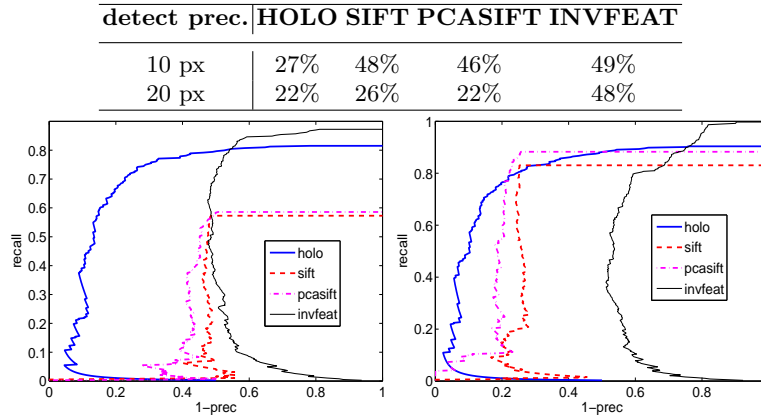


Fig. 4. Equal error rates and 1-Precision/Recall graphs for the porate test set. Left graph: with high detection accuracy of 10 pixels. Right graph: with low detection accuracy of 20 pixels.

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 was 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. 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 [12]. 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 4 we give equal error rates and 1-Precision/Recall graphs for the considered dataset. We made 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 of 20 pixels, PCASIFT can slightly outperform HOLO in a certain threshold area. Only INVFEAT is able to finally detect nearly all porates by the cost of very high false positive rates.

4 Conclusion

We propose a rotation and translation equivariant image transformation. 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 is applied for rotation invariant detection of objects in microscopical images. Compared to a GHT-based approach relying on SIFT features our approach is competitive for 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 hence a detection becomes impossible. The holomorphic filter does not have such problems because all pixels are taken into account. 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 otherwise, due to the small number of parameters, the filter is able to show better generalization ability for certain data.

References

1. D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13-2, 1981.
2. 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.
3. W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(9), 1991.
4. 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.
5. D.G. Lowe. Distinct image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
6. V. J. Mathews and G. Sicuranza. *Polynomial Signal Processing*. J.Wiley, New York, 2000.
7. Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Multiple object class detection with a generative model. In *Proceedings of the CVPR*, volume 1, pages 26 – 36, 2006.
8. P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(5):488 – 499, 1995.
9. 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.
10. M. Reisert and H. Burkhardt. Using irreducible group representations for invariant 3d shape description. In *Proceedings of the 28th DAGM Symposium, Berlin*, 2006.
11. Marco Reisert. Equivariant holomorphic filters - theory and applications. Technical Report 3, Albert-Ludwig University Freiburg imb.informatik.uni-freiburg.de/people/reisert, 2007.
12. Frederik Schaffalitzky and Andrew 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*, volume 2350, pages 414 – 431, 2002.
13. R. Silverman. *Introductory complex analysis*. Dover Publications, 1972.
14. S. Thurnhofer and S. Mitra. A general framework for quadratic volterra filters for edge enhancement. *IEEE Trans. Image Processing*, pages 950–963, 1996.