

# Nonlinear functionals in the construction of multiscale affine invariants

Esa Rahtu<sup>1</sup>, Mikko Salo<sup>2</sup>, and Janne Heikkilä<sup>1</sup>

<sup>1</sup> Machine Vision Group

Department of Electrical and Information Engineering  
P.O. Box 4500, 90014 University of Oulu, Finland  
{erahtu,jth}@ee.oulu.fi

<sup>2</sup> Department of Mathematics and Statistics / RNI  
P.O. Box 68, 00014 University of Helsinki, Finland  
mikko.salo@helsinki.fi

**Abstract.** In this paper we introduce affine invariants based on a multiscale framework combined with nonlinear comparison operations. The resulting descriptors are histograms, which are computed from a set of comparison results using binary coding. The new constructions are analogous to other multiscale affine invariants, but the use of highly nonlinear operations yields clear advantages in discriminability. This is also demonstrated by the experiments, where comparable recognition rates are achieved with only a fraction of the computational load. The new methods are straightforward to implement and fast to evaluate from given image patches.

## 1 Introduction

In computer vision, invariant features have provided an elegant way of identifying objects under geometric transformations. The appropriate transformations depend heavily on the application, but in many cases the affine transformation provides a reasonable model. Several affine invariant features have been considered. The first method, the affine invariant moments [1, 2] was introduced already in 1962. Since then affine invariant spectral signatures [3], crossweighted moments [4], and the trace transform [5] have been presented. Unfortunately these methods often suffer from sensitivity to nonaffine distortions, implementational difficulties, and the lack of discriminating features.

One recent approach to affine invariants is the multiscale framework. The idea in this framework is to extend the given image to a set of affine covariant versions, each carrying slightly different information, and then to extract some known invariant characteristics from each of them separately. The construction of the affine covariant set is the key part of the approach, and it is done by combining several scaled representations of the original image. The advantage is the possibility for variations, which is also demonstrated by the amount of methods created using this framework: multiscale autoconvolution [6], spatial

multiscale affine invariants [7], generalized affine invariant moments, multiscale autoconvolution histograms [8], and ridgelet-based affine invariants.

Hitherto, the invariants introduced using the multiscale framework apply simple pointwise products and convolution for creating the affine covariant sets from scaled representations. While these linear operations offer robustness to noise and other distortions, they can also easily compromise the discriminability of the features. In this paper we introduce new multiscale affine invariants, which apply nonlinear comparison operations to the combinations of the scaled representations. We also introduce a way for combining several comparisons together using a binary code construction. These nonlinear operations and the binary code construction have not been used in previous multiscale invariants. The experiments performed demonstrate that the use of comparison operations has a clear impact on the performance, and the new methods achieve comparable or better results with only a fraction of the computational load of the earlier methods.

## 2 Multiscale approach

We begin with a description of the multiscale approach in constructing affine invariants. First we recall the definition of an affine transformation.

**Definition 1.** *A spatial affine transformation  $\mathcal{A}$  of coordinates  $x \in \mathbf{R}^2$  is given by  $\mathcal{A}(x) = Tx + t$ , where  $t \in \mathbf{R}^2$  and  $T$  is a  $2 \times 2$  nonsingular matrix with real entries. Further, let  $f: \mathbf{R}^2 \rightarrow \mathbf{R}$ ,  $f \geq 0$ , be an image function. The affine transformed version  $f_{\mathcal{A}}$  of  $f$  is given by*

$$f_{\mathcal{A}}(x) = f \circ \mathcal{A}^{-1}(x) = f(T^{-1}x - T^{-1}t).$$

The construction of multiscale affine invariants can be done in three steps.

1. The image  $f$  is represented in  $n$  different scales  $f(\alpha_1 x), \dots, f(\alpha_n x)$ .
2. The scaled images are combined to a new image  $Gf(x)$ . The combination is required to be affine covariant, which means that for any affine transformation  $\mathcal{A}$  one has

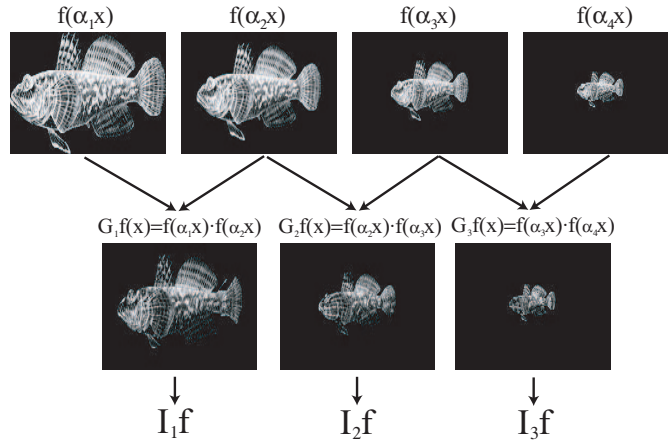
$$G(f \circ \mathcal{A}^{-1})(x) = (Gf)(\mathcal{A}^{-1}(x)).$$

3. An affine invariant operation is applied to  $f$  and  $Gf$  to obtain the full invariant  $If$ .

The procedure is illustrated in Figure 1. The advantage of the method is that by varying the scales  $\alpha_i$ , the combinations  $G$ , and the affine invariant operations, it is possible to create a great variety of different features for many purposes.

The first step, scaling of images, is straightforward. The third step can also be quite simple. Possible choices for the affine invariant operation include the normalized integration

$$If = \frac{1}{\|f\|_{L^1}} \int_{\mathbf{R}^2} Gf(x) dx, \quad (1)$$



**Fig. 1.** Example of multiscale scheme, where two scaled representations are combined together with pointwise product. The final invariants are computed from the resulting images  $G_i f$ .

or more generally, any affine invariant moment [2] can be used. Also, one may apply a histogramming operation to  $Gf$ .

The second step is often the most complicated one in this approach. One needs to take the scaled images  $f(\alpha_i x)$  and combine them to a new image  $Gf(x)$ , so that  $Gf$  and  $G(f \circ \mathcal{A}^{-1})$  are related by the affine transformation  $\mathcal{A}$ . The reason for using scaled images is that scaling commutes with matrix products, i.e.  $\alpha T = T\alpha$ . The translation causes some problems. It may either be normalized away by computing the image centroid, or one may choose  $G$  more carefully so that translation invariance is obtained without finding the centroid.

We will discuss a few examples for  $G$ . First we consider the spatial multiscale invariants (SMA) [7], given here in a slightly modified form. We choose  $Gf$  to be a product of the original  $f$  and two scaled representations of it,  $f(\alpha x)$ , and  $f(\beta x)$ . In this formulation, the translation component must be normalized and it is done by computing the image centroid  $\mu(f)$ . The operator  $G$  is given by

$$Gf(x) = f(x)f(\alpha x + (1 - \alpha)\mu(f))f(\beta x + (1 - \beta)\mu(f)), \quad (2)$$

where  $\alpha, \beta \in \mathbf{R}$ . We then construct the invariant feature  $I$  by using the normalized integration (1). The result is the SMA transform, given by

$$Sf(\alpha, \beta) = \frac{1}{\|f\|_{L^1}} \int_{\mathbf{R}^2} f(x)f(\alpha x + (1 - \alpha)\mu(f))f(\beta x + (1 - \beta)\mu(f)) dx. \quad (3)$$

Due to its simplicity,  $Sf(\alpha, \beta)$  is very fast to evaluate, and the possibility of varying the scales results in an infinite number of different descriptors.

Another example is the multiscale autoconvolution (MSA) [6], where one uses a combination of convolutions and products to form  $Gf$ . One advantage of

this descriptor is the fact that the translation component does not have to be considered separately. Define

$$Gf(x) = \frac{1}{\|f\|_{L^1}^2} f(x)(f_\alpha * f_\beta * f_\gamma)(x), \quad (4)$$

where  $\alpha, \beta, \gamma \in \mathbf{R}$ ,  $\alpha + \beta + \gamma = 1$ ,  $f_a(x) = a^{-2}f(x/a)$ , and  $*$  denotes convolution. The actual invariant features are again constructed by normalized integration, which gives the MSA transform

$$Mf(\alpha, \beta) = \frac{1}{\|f\|_{L^1}^3} \int_{\mathbf{R}^2} f(x)(f_\alpha * f_\beta * f_\gamma)(x) dx. \quad (5)$$

This formulation is not computationally appealing, but fortunately  $Mf$  can be computed using the Fourier transform  $\hat{f}$  as

$$Mf(\alpha, \beta) = \frac{1}{\hat{f}(0)^3} \int_{\mathbf{R}^2} \hat{f}(-\xi)\hat{f}(\alpha\xi)\hat{f}(\beta\xi)\hat{f}(\gamma\xi) d\xi.$$

These two examples illustrate the application of the multiscale framework in constructing affine invariants. Further examples could include generalized affine invariant moments, multiscale autoconvolution histograms, and ridgelet-based affine invariants. The basic idea of applying the multiscale framework is similar in all the examples, although the preprocessing and the computation of the actual invariant characteristics may differ.

### 3 The new approach

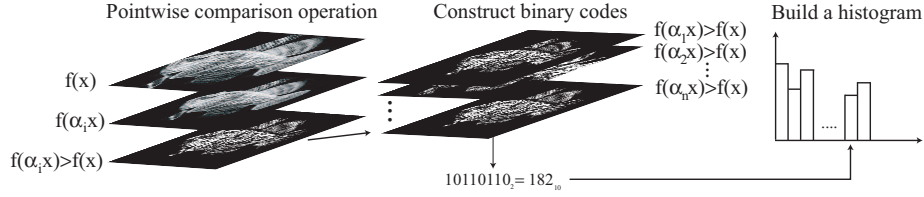
All the multiscale affine invariants presented above perform the combination  $Gf$  of scaled images by using convolutions or pointwise products. These operations, which have a linear character, seem to behave robustly under noise, but in many applications they can compromise the discriminability of the methods. For this reason, better performance might be achieved by using other, nonlinear functionals in the combination of the scaled images.

We propose here to replace the products in the earlier constructions by pointwise comparison operations. This approach is motivated by the excellent performance of the local binary patterns (LBP) [9], where similar comparison operations were used to construct highly discriminative texture descriptors. Another motivation is the fact that comparison operations perform well under illumination distortions, which are very common in real applications. We demonstrate the new approach in two cases, based on similar formulations as in SMA and MSA.

#### 3.1 Invariant based on comparison of scaled images

The first new invariant is analogous to SMA. However, instead of two scales we take only one, and we replace the product by the comparison operation

$$Gf(x) = G_\alpha f(x) = X(f(x), f(\alpha x + (1 - \alpha)\mu(f))), \quad (6)$$



**Fig. 2.** Illustration of the process for generating the invariant histograms.

where  $\alpha \in \mathbf{R}$  and  $\mu(f)$  is the image centroid, and

$$X(a, b) = \begin{cases} 1 & \text{if } a > b, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

It is easy to show that  $G(f \circ \mathcal{A}^{-1})(x) = Gf(\mathcal{A}^{-1}(x))$ , which was required in the multiscale framework. Now one could simply compute the normalized integral from the binary image  $Gf$  to form the affine invariant. However, this would destroy the spatial information in  $Gf$ , and one would need to use many different  $\alpha$  values to achieve desired discriminability. Instead, we propose to compute  $G_{\alpha}f(x)$  with only a few different  $\alpha$  values, say  $\alpha_1, \alpha_2, \dots, \alpha_n$ , and then to use a binary code construction as in LBP to combine all the information to a new function

$$Bf(x) = B_{\alpha_1, \dots, \alpha_n}f(x) = G_{\alpha_1}f(x) + G_{\alpha_2}f(x) \cdot 2 + \dots + G_{\alpha_n}f(x) \cdot 2^{n-1}. \quad (8)$$

The function  $Bf$  has integer values from 0 to  $2^n - 1$ , and it completely encodes the information in the functions  $G_{\alpha_1}f, \dots, G_{\alpha_n}f$ . It follows immediately that  $B(f \circ \mathcal{A}^{-1})(x) = Bf(\mathcal{A}^{-1}(x))$ , so also  $Bf$  fulfills the requirement of affine relationship in the multiscale framework. Thus, to compute the final affine invariant, one could just evaluate the normalized integral of  $Bf$ . This approach, however, does not make sense, since the different combinations would have an uneven impact to the resulting invariant value. Instead we construct a histogram  $HSf(k)$ , with  $2^n - 2$  bins, from nonzero values of  $Bf$ , and we normalize the histogram so that the sum over all bins is equal to one. Compared to the direct integration of  $G_{\alpha}f$ , this makes it possible to preserve the relative spatial arrangements in the functions  $G_{\alpha_1}f, \dots, G_{\alpha_n}f$ . The construction of the invariant is illustrated in Figure 2.

### 3.2 Invariant based on comparison of image and its scaled autoconvolution

The previous construction has the disadvantage that one needs to eliminate the translation by computing the image centroid. Also, it shares the same incompleteness issues as SMA [7]. For these reasons, we base the next construction on a formulation which is similar to MSA. Consider the convolution of two scaled

representations of  $f$  as

$$Cf(x) = C_\alpha f(x) = \frac{1}{\|f\|_{L^1}} (f_\alpha * f_{1-\alpha})(x).$$

Here  $\alpha \in \mathbf{R}$  and  $f_a(x) = a^{-2}f(x/a)$ . It is easy to see that  $C(f \circ \mathcal{A}^{-1})(x) = Cf(\mathcal{A}^{-1}(x))$ , and  $Cf$  is an affine covariant operator.

We take the functions  $C_\alpha f$  as a basis for the new invariant, and use comparison operations for combining them. If  $X$  is the comparison operator in (7), we define

$$G_\alpha f(x) = X(f(x), C_\alpha f(x)). \quad (9)$$

It immediately follows that  $G_\alpha(f \circ \mathcal{A}^{-1})(x) = G_\alpha f(\mathcal{A}^{-1}(x))$ , and the requirement in the multiscale approach is satisfied. With each  $\alpha$  value we get a binary image  $G_\alpha f$ , and we use these binary images in the expression (8) to get  $Bf$ . One may then form a normalized histogram of  $Bf$  as in the preceding invariant. The resulting histogram is denoted by  $HMf(k)$ .

## 4 Implementational issues

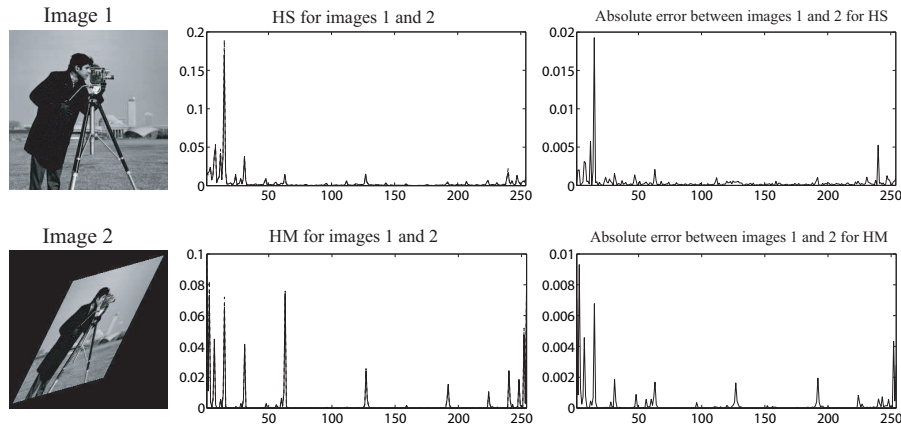
Evaluating  $HS$  can be done similarly as SMA [7], with a few straightforward modifications. Basically the only differences are that instead of a product we use a comparison operation, which is then followed by the binary coding, and the histogram operation instead of the sum. It is also possible to apply a similar interpolation scheme as in SMA, where the interpolation grid is designed so that all the required samples are computed at once. This gave a significant speed advantage in SMA. Due to the very similar implementations,  $HS$  is almost as efficient to evaluate as SMA. The only difference is the histogramming, which is slightly slower than summing. It is easy to show that the asymptotical complexity for an  $N \times N$  image is  $O(N^2)$  for both  $HS$  and SMA.

The computation of  $HM$  is similar to MSA [6]. However, the product in MSA can be handled in Fourier domain, which is not possible for the comparison operation. The convolution involved in (9) can still be evaluated using the Fourier transform, and we also have the advantage that if  $0 < \alpha < 1$  the Fourier transform does not have to be zero padded for accurate evaluation. In addition to this, we need to select a way to perform the scaling to produce  $f(x/\alpha)$  and  $f(x/(1-\alpha))$ . We could do this also in Fourier domain, but as in MSA, better results are achieved by scaling in the spatial domain before taking the Fourier transform. The final asymptotical complexity of the method is the same as in discrete Fourier transform, i.e.  $O(N^2 \log N)$ .

The Matlab programs which were used to evaluate both  $HS$  and  $HM$  are available at the website: [http://www.ee.oulu.fi/research/imag/cmp\\_inv/](http://www.ee.oulu.fi/research/imag/cmp_inv/).

## 5 Experiments

In this section we assess the two new invariants  $HS$  and  $HM$  in classification experiments and compare their performance to MSA, MSA histograms

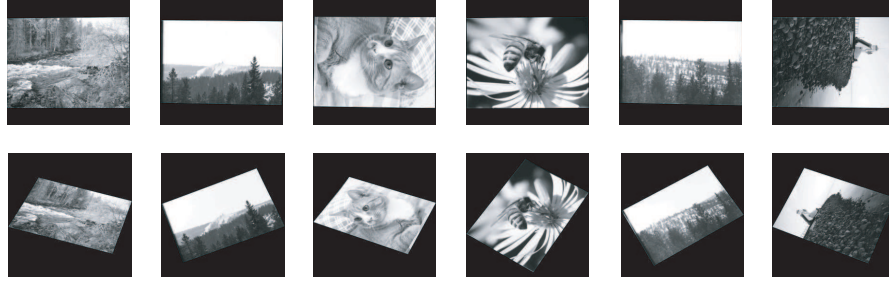


**Fig. 3.** Examples of the proposed affine invariant histograms. The first row has histograms  $HS$  for images 1 and 2, and the absolute difference between these. The second row has the same data for the histograms  $HM$ .

[8], SMA, and affine invariant moments (AMI) [2]. We start by giving some examples of the invariant histograms  $HS$  and  $HM$  in Figure 3. The scaling parameters used with  $HS$  were  $\{0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4\}$  and with  $HM$   $\{0.02, 0.055, 0.09, 0.11, 0.16, 0.205, 0.35, 0.45\}$ . The same parameters are also used in the classification experiments later. Notice the slight differences between histograms for images 1 and 2, which are due to nonaffine discretization errors.

In the classification experiments, we selected the parameters for comparison methods so that they have approximately the same computational load as the corresponding new methods. In the case of MSA we used 5 invariants with  $(\alpha, \beta) = \{(-0.1, 0.1), (-0.1, 0.3), (-0.2, 0.2), (-0.2, 0.4), (-0.3, 0.4)\}$ , and in the case of SMA 10 invariants with  $(\alpha, \beta) = \{(-1, -1), (-1, -0.25), (-1, 0.75), (-0.75, -0.5), (-0.75, 0.5), (-0.5, -0.5), (-0.5, 0.5), (-0.25, -0.25), (-0.25, 0.75), (0.25, 0.5)\}$ . In addition to these we also computed the MSA and SMA with a larger amount of features, namely 19 for MSA and 36 for SMA. For other comparison methods we selected the MSA histogram with  $(\alpha, \beta) = (-0.1, 0.3)$  and AMI with 60 independent invariants. We will refer to these methods as MSA5, SMA10, MSA19, SMA36, MSAhist, and AMI, respectively. Since the methods except for  $HS$  are not illumination invariant, we normalized the images so that they have mean 128 and standard deviation 30. The classification was performed using a simple nearest neighbor classifier, where the distance measure was the histogram intersection for the histogram approaches and the Euclidean distance for the others. The calculation of Euclidean distance was preceded by PCA decorrelation and dimension reductions in order to enhance the classification performance.

In the first experiment we classified  $256 \times 256$  gray-scale images of postcards, obtained from photographs taken from different viewing angles. The training



**Fig. 4.** Some examples of training images (first row) and angle distorted images (second row).

set included one image from each of the 50 different postcards and the test set included 175 images of the same postcards taken from various angles in natural light. The camera we used was Canon EOS 10D with EF 17-40mm F/4L USM lens. The resulting classification problem is not easy, since in addition to the affine deformations, the test images are subjected to many other distortions due to real photographing conditions. Some examples of the training and test images are shown in Figure 4. The achieved classification results are given in Table 1, along with the approximate computation times per one  $256 \times 256$  image. The results indicate that *HS* has the clearly the best performance if we take into account the computation time and the classification accuracy. The difference is especially large compared to *SMA*. Also *HM* performs well, outperforming *MSA5* and *MSA* histograms in classification. With 19 features for *MSA* we were

**Table 1.** Classification error percents under real view angle distortions.

	HS	HM	MSA5	MSA19	MSA hist.	SMA10	SMA36	AMI
Classification error	3.4 %	5.7 %	22.9 %	3.4 %	27.4 %	36.6 %	26.3 %	65.7 %
Execution time	1.08 s	1.89 s	2.42 s	11.40 s	0.19 s	2.62 s	5.09 s	0.29 s

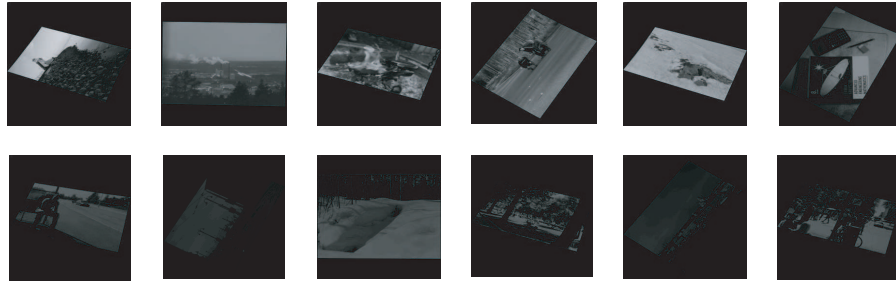
**Table 2.** Classification error percents under illumination distortions.

	HS	HM	MSA19	SMA36
Underexposure by 1.5 apertures	2.4 %	2.4 %	1.6 %	10.4 %
Underexposure by 3 apertures	4.0 %	5.6 %	0.0 %	11.2 %

**Table 3.** Classification error percents under heavy illumination and noise distortions.

	HS	HM	MSA19	SMA36
Underexposure by 2 and ISO 3200	24.0 %	30.0 %	2.0 %	4.0 %





**Fig. 5.** Examples of illumination distorted images. With the images on first row the underexposure is 1.5 apertures and on the second row 3 apertures.

able to achieve better recognition, but with a significantly larger computational load.

We continued by introducing more distortions to the image. First we added illumination distortions by taking a new set of 125 test photos, underexposing them first by 1.5, and then by 3 aperture steps of the camera. The changes in the viewing angle were kept quite small in this experiment. Some examples of the new test sets are shown in Figure 5. The results of this experiment are illustrated in Table 2. We omitted the MSA5, MSA histogram, SMA10, and AMI methods from this experiment due to their high error rates in the first experiment. It can be observed that all the tested methods, except SMA36, are very tolerant to illumination changes. As expected, MSA with 19 invariants works quite robustly, but similar results are achieved with  $HS$  and  $HM$  with only a small fraction of the required computational load. In the case of  $HS$  there was also no need to perform illumination normalization, which simplifies the overall procedure.

As a final illustration we created one more test set of 50 images, by underexposing by 2 aperture steps and increasing the sensitivity (ISO) value of the camera to 3200. The resulting images were severely distorted by noise. The results of this experiment are shown in Table 3. The new methods, which are based on comparison operations, react to the substantial changes in gray-scale values more strongly than MSA and SMA which use product operations. In many cases, reactivity is a desirable feature of the method, but in this experiment robustness leads to better results. The experiment clearly illustrates the trade-off between robustness and discriminability.

## 6 Conclusions

In this paper we introduced a novel way of creating affine invariants from the multiscale framework by applying comparison operations and binary coding. The application of these nonlinear operations offered a new way to increase the discriminability of the invariants. The simplicity of the new operations made the proposed methods efficient to evaluate. The experiments performed indicated

that using already a few scales in the construction of the invariants can outperform similar approaches with linear functionals. The amount of features in the traditional methods had to be drastically increased to achieve even comparable results. In addition to the two examples provided here, we expect that similar nonlinear constructions can be applied to other multiscale invariants.

## 7 Acknowledgments

This work was supported by the Academy of Finland (project no. 110751). The authors would like to thank Mr. Janne Kenttälä for imparting to us his superior expertise in photographic matters, and Mr. Andrew Kenttälä for his valiant efforts in making this possible.

## References

1. M. Hu. Visual pattern recognition by moment invariants. *IEEE Trans. Information Theory*, 8:179–187, 1962.
2. J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, 1993.
3. J. Ben-Arie and Z. Wang. Pictorial recognition of objects employing affine invariance in the frequency domain. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(6):604–618, 1998.
4. Z. Yang and F. S. Cohen. Cross-weighted moments and affine invariants for image registration and matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(8):804–814, 1999.
5. M. Petrou and A. Kadyrov. Affine invariant features from the trace transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):30–44, 2004.
6. E. Rahtu, M. Salo, and J. Heikkilä. Affine invariant pattern recognition using multiscale autoconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):908–918, 2005.
7. E. Rahtu, M. Salo, and J. Heikkilä. A new efficient method for producing global affine invariants. In *Proc. International Conference on Image Analysis and Processing*, 407–414, Cagliari, Italy, 2005.
8. E. Rahtu, M. Salo, and J. Heikkilä. Multiscale autoconvolution histograms for affine invariant pattern recognition In *Proc. the 16th British Machine Vision Conference*, 3:1059-1068, Edinburgh, UK, 2006.
9. T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.