Affine Invariant Pattern Recognition Using Multi-Scale Autoconvolution

Esa Rahtu, Mikko Salo and Janne Heikkilä

Abstract

This article presents a new affine invariant image transform called Multi-Scale Autoconvolution (MSA). The proposed transform is based on a probabilistic interpretation of the image function. The method is directly applicable to isolated objects and does not require extraction of boundaries or interest points, and the computational load is significantly reduced using the Fast Fourier Transform. The transform values can be used as descriptors for affine invariant pattern classification, and in this article we illustrate their performance in various object classification tasks. As shown by a comparison with other affine invariant techniques, the new method appears to be suitable for problems where image distortions can be approximated with affine transformations.

Index Terms

Affine invariance, affine invariant features, pattern classification, target identification, object recognition, image transforms.

E. Rahtu and J.Heikkilä are with the Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu. P.O. Box 4500, 90014 University of Oulu, Finland. E-mail: {erahtu,jth}@ee.oulu.fi

M. Salo is with the Rolf Nevanlinna Institute, Department of Mathemathics and Statistics, University of Helsinki. P.O. Box 68, 00014 University of Helsinki, Finland. E-mail: msa@rni.helsinki.fi

Affine Invariant Pattern Recognition Using Multi-Scale Autoconvolution

I. Introduction

Every day we confront situations where we have to recognize an object or patterns, like when seeing the face of a friend, finding keys from the table, reading a book, smelling food, listening to a conversation, looking at photos, watching television, etc. These things are often easy for us, but to build a machine with similar abilities is very difficult. Nevertheless, the many results already achieved, combined with ever increasing computational power, insure that nowadays it is difficult to find many scientific or technical disciplines not applying some pattern recognition techniques in one way or another. Fields like medicine, biology, psychology, marketing, computer vision, artificial intelligence, and remote sensing make use of patterns and automated recognition.

One of the most difficult things for a machine to do is to identify an object when seen from different angles. However, this kind of ability is very much desired in many pattern recognition applications. There is even the dream of a general pattern recognizer. While it is difficult to directly achieve such a system, many approximations have been proposed with promising results. One of these is the idea to model the distortions using some geometric transformation, like translation, rotation, scaling etc. and then to find features which are invariant under these transformations. These features can be used as a basis for classification. For example in models where only rotation, scaling and translation are involved, well known global methods are Fourier-Mellin transform [1], Radon transform [2], wavelet and ridgelet based techniques [3], and Haar integrals [4]. However many times these three degrees of freedom are not enough for adequate approximation. Restricting ourselves to essentially linear models, which are usually much easier to handle than strongly non-linear ones, the affine mapping gives the best possible approximation.

It is easy to build affine invariances using segmented information like contours or interest points. Well known methods in this class include boundary based Fourier descriptors [5], [6], [7], discrete interest point based methods [8], [9], and some wavelet based descriptors [10], [11]. Some of these methods are actually even capable of handling the more complex case of projective mappings [9]. The problem is however that these methods stay very dependent on

the success of the segmentation process. Sometimes one also needs additional techniques, like geometric hashing [9] or interest point detectors [12], for matching correspondences.

The group of global affine invariant methods is more sparse despite the fact that the first one, affine invariant moments [13], was already presented in 1962. That presentation, however, contained an error which was later corrected in [14] and [15]. Since then, other methods have been introduced, e.g. Ben-Arie's frequency domain technique [16], [17], cross-weighted moments [18], and trace transform [19]. In this article, we propose a novel approach called Multi-Scale Autoconvolution (MSA) for global affine invariant feature extraction. Our technique is based on applying standard point-based invariants in a novel way, combined with probabilistic ideas. The proposed transform has only moderate computational complexity, and it does not require any image segmentation steps other than background elimination. This idea was originally introduced in [20] and further developed in [21]. It is also possible to use MSA for convexity recognition, and some initial results can be found in [22]. As convexity recognition is not directly related to affine invariant classification we will not consider this here.

In this article, we give a comprehensive treatment of the MSA transform, expanding the earlier results and complementing them with new experiments. The article is organized as follows. The upcoming section introduces the Multi-Scale Autoconvolution with its definition and properties, whereas Section III considers the actual implementation. In Section IV, we examine the computational complexity of the proposed method and compare it with some other similar techniques. Section V illustrates the discriminative power of MSA based features in various object classification tasks, including binary and gray-scale cases also in the presence of noise and different distortions. Sections VI and VII finally summarize the results and give directions for future development. Our goal is to present a novel idea for affine invariant recognition and to discuss its properties and computational aspects, so that it will be easy to implement for the use of applications.

II. Multi-Scale Autoconvolution

In this section, we define an affine invariant image transform called Multi-Scale Autoconvolution (MSA). We start with an introduction to the ideas behind MSA and then go further into the mathematical details and the properties of this transform. But before continuing, let us first define the spatial affine transformation and affine invariance as follows:

Definition 1: Define the affine transformation $\mathcal{A} = \mathcal{A}\{\mathbf{T},\mathbf{t}\}$ by

$$\mathbf{x}' = \mathcal{A}(\mathbf{x}) = \mathbf{T}\mathbf{x} + \mathbf{t},$$

where $\mathbf{t}, \mathbf{x} \in \mathbb{R}^2$ and $\mathbf{T}$ is a 2×2 nonsingular matrix whose elements belong to $\mathbb{R}$. Any such transformation is invertible with inverse $\mathcal{A}^{-1}(\mathbf{x}) = \mathbf{T}^{-1}\mathbf{x} - \mathbf{T}^{-1}\mathbf{t}$.

Definition 2: Suppose $f(\mathbf{x})$: $\mathbb{R}^2 \to \mathbb{R}$, $f \geq 0$, is an image intensity function corresponding to a gray-scale image in $\mathbb{R}^2$. We may apply an affine transformation $\mathcal{A}$ to this image, which gives a new gray-scale image in $\mathbb{R}^2$ with image function f', where

$$f'(\mathbf{x}) = f \circ \mathcal{A}^{-1}(\mathbf{x}) = f(\mathbf{T}^{-1}\mathbf{x} - \mathbf{T}^{-1}\mathbf{t}).$$

We call f' the $\mathcal{A}$ transformed version of f.

Definition 3: A feature I, extracted from the function f, is said to be affine invariant if it produces the same value for f and the $\mathcal{A}$ transformed version of f, for any affine transformation $\mathcal{A}$.

A. The definition of MSA

Let $f(\mathbf{x}) : \mathbb{R}^{\mathbf{2}} \to \mathbb{R}$ with $f \geq 0$ be an image intensity function in $L^1(\mathbb{R}^2) \cap L^2(\mathbb{R}^2)$ and let $p(\mathbf{x}) = \frac{1}{\|f\|_{L^1}} f(\mathbf{x})$ be the normalized version of f, so that $\int_{\mathbb{R}^2} p(\mathbf{x}) d\mathbf{x} = 1$. Then $p(\mathbf{x})$ is a probability density function, and we may take $\mathbf{X}_0$, $\mathbf{X}_1$, and $\mathbf{X}_2$ to be independent random variables with values in $\mathbb{R}^2$ so that $p_{\mathbf{X}_j}(\mathbf{x}_j) = p(\mathbf{x}_j)$ (we write $p_{\mathbf{X}}$ for the probability density function of a random variable **X**). Consider three samples $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$ of these random variables as a basis for the following transformation

$$\mathbf{u} = \alpha(\mathbf{x}_1 - \mathbf{x}_0) + \beta(\mathbf{x}_2 - \mathbf{x}_0) + \mathbf{x}_0 \tag{1}$$

where (α, β) are the coordinates for **u** in the space spanned by the vectors $\mathbf{x}_1 - \mathbf{x}_0$ and $\mathbf{x}_2 - \mathbf{x}_0$ and with the origin at $\mathbf{x}_0$.

Now let $\mathcal{A}\{\mathbf{T},\mathbf{t}\}$ be an affine transformation as in Definition 1. If one takes the $\mathcal{A}$ transformed versions of the sample points $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e. $\mathbf{x}'_0 = \mathbf{T}\mathbf{x}_0 + \mathbf{t}$, $\mathbf{x}'_1 = \mathbf{T}\mathbf{x}_1 + \mathbf{t}$, and $\mathbf{x}'_2 = \mathbf{T}\mathbf{x}_2 + \mathbf{t}$, and defines another transformation by

$$\mathbf{u}' = \alpha(\mathbf{x}'_1 - \mathbf{x}'_0) + \beta(\mathbf{x}'_2 - \mathbf{x}'_0) + \mathbf{x}'_0, \tag{2}$$

it can be rewritten as

$$\begin{aligned}\mathbf{u}' &= \alpha(\mathbf{T}\mathbf{x}_1 - \mathbf{T}\mathbf{x}_0) + \beta(\mathbf{T}\mathbf{x}_2 - \mathbf{T}\mathbf{x}_0) + \mathbf{T}\mathbf{x}_0 + \mathbf{t} \\ &= \mathbf{T}\mathbf{u} + \mathbf{t}. \end{aligned} \tag{3}$$

One can see that also the points $\mathbf{u}$ and $\mathbf{u}'$ are connected by $\mathcal{A}$ if they have the same coordinate values (α, β) in the corresponding spaces. This well known phenomenon is illustrated in Figure 1. As known, affine transformations are bijective, so every $\mathbf{u}$ has exactly one corresponding $\mathbf{u}'$.

Now recall that the points $\mathbf{x}_0$, $\mathbf{x}_1$, and $\mathbf{x}_2$ were samples of the random variables $\mathbf{X}_0$, $\mathbf{X}_1$, and $\mathbf{X}_2$, and define a new random variable

$$\mathbf{U}_{\alpha,\beta} = \mathbf{X}_0 + \alpha(\mathbf{X}_1 - \mathbf{X}_0) + \beta(\mathbf{X}_2 - \mathbf{X}_0) \tag{4}$$

which has $\mathbf{u}$ as the corresponding sample.

Similarly define

$$\mathbf{U}'_{\alpha,\beta} = \mathbf{X}'_0 + \alpha(\mathbf{X}'_1 - \mathbf{X}'_0) + \beta(\mathbf{X}'_2 - \mathbf{X}'_0),$$

where $\mathbf{X}'_0 = \mathbf{T}\mathbf{X}_0 + \mathbf{t}$, $\mathbf{X}'_1 = \mathbf{T}\mathbf{X}_1 + \mathbf{t}$, and $\mathbf{X}'_2 = \mathbf{T}\mathbf{X}_2 + \mathbf{t}$ and hence also $\mathbf{U}'_{\alpha,\beta} = \mathbf{T}\mathbf{U}_{\alpha,\beta} + \mathbf{t}$. It follows (see Lemma 2.1 below) that $p_{\mathbf{X}'_j}(\mathbf{x}'_j) = p'(\mathbf{x}'_j)$, where p' is the normalized version of $f'(\mathbf{x}) = f \circ \mathcal{A}^{-1}(\mathbf{x}) = f(\mathbf{T}^{-1}\mathbf{x} - \mathbf{T}^{-1}\mathbf{t})$.

Substituting $\mathbf{x} = \mathbf{U}'_{\alpha,\beta}$ we get

$$\begin{aligned} f'(\mathbf{U}'_{\alpha,\beta}) &= f(\mathbf{T}^{-1}\mathbf{U}'_{\alpha,\beta} - \mathbf{T}^{-1}\mathbf{t}) \\ &= f(\mathbf{T}^{-1}(\mathbf{T}\mathbf{U}_{\alpha,\beta} + \mathbf{t}) - \mathbf{T}^{-1}\mathbf{t}) = f(\mathbf{U}_{\alpha,\beta}). \end{aligned}$$

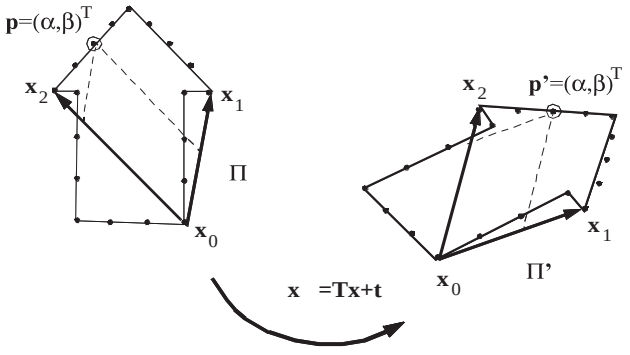


Fig. 1. Illustration of how a point triplet behaves under affine transformation.

Hence $f(\mathbf{U}_{\alpha,\beta})$ and $f'(\mathbf{U}'_{\alpha,\beta})$ are equal as random variables. This gives a method of obtaining affine invariant features of an image function f: the expected value of $f(\mathbf{U}_{\alpha,\beta})$, or any of its moments, or the expected value of $g(f(\mathbf{U}_{\alpha,\beta}))$ for a measurable function g, do not change in an affine transformation of f. Thus, we introduce the following affine invariant features, and single out one of them as the MSA transform of f.

Definition 4 (Multi-Scale Autoconvolution): Let f be a function in $L^1(\mathbb{R}^2) \cap L^2(\mathbb{R}^2)$ with $f \geq 0$, and let $p(\mathbf{x}) = \frac{1}{\|f\|_{L^1}} f(\mathbf{x})$ be the corresponding probability density function. Take $\mathbf{X}_0$, $\mathbf{X}_1$, and $\mathbf{X}_2$ to be independent random variables with values in $\mathbb{R}^2$ so that $p_{\mathbf{X}_j}(\mathbf{x}_j) = p(\mathbf{x}_j)$. For $\alpha, \beta \in \mathbb{R}$ define a new random variable

$$\mathbf{U}_{\alpha,\beta} = \mathbf{X}_0 + \alpha(\mathbf{X}_1 - \mathbf{X}_0) + \beta(\mathbf{X}_2 - \mathbf{X}_0).$$

The kth moment of $f(\mathbf{U}_{\alpha,\beta})$ is defined as

$$F^k(\alpha, \beta) = E[f(\mathbf{U}_{\alpha,\beta})^k],$$

and the MSA transform of f is defined as the first moment,

$$F(\alpha, \beta) = E[f(\mathbf{U}_{\alpha,\beta})].$$

Below we will use the MSA transform for affine invariant classification. The reason for not using the other moments is that higher moments are in some cases more sensitive to noise. We note that already the $F(\alpha, \beta)$ give infinitely many affine invariant features by varying α and β. Also, for binary images where f is the characteristic function of a set in $\mathbb{R}^2$, all the moments are equal and the first moment carries the same information as the rest of the moments.

B. Explicit expressions for MSA

We proceed to derive different expressions for the MSA transform. In doing this, we will need the following basic lemma which gives the probability density functions for sums, affine transformations, and scalar multiples of random variables.

Lemma 2.1: Let $\mathbf{X}, \mathbf{Y}$ be two independent random variables with values in $\mathbb{R}^2$, and suppose that they have probability density functions $p_{\mathbf{X}}(\mathbf{x}) = p(\mathbf{x})$, $p_{\mathbf{Y}}(\mathbf{y}) = q(\mathbf{y})$.

(a) $p_{\mathbf{X}+\mathbf{Y}}(\mathbf{z}) = (p * q)(\mathbf{z}) = \int_{\mathbb{R}^2} p(\mathbf{x}) q(\mathbf{z} - \mathbf{x})\, d\mathbf{x}$.

(b) Let $\mathbf{T}$ be a nonsingular 2×2 real matrix and $\mathbf{t} \in \mathbb{R}^2$. Then $p_{\mathbf{TX}+\mathbf{t}}(\mathbf{z}) = \frac{1}{|\mathbf{T}|} p(\mathbf{T}^{-1}\mathbf{z} - \mathbf{T}^{-1}\mathbf{t})$.

(c) If $a \neq 0$ then $p_{a\mathbf{X}}(\mathbf{z}) = \frac{1}{a^2} p(\frac{\mathbf{z}}{a})$.

For completeness, we include the proof in Appendix I.

Define $\gamma = 1 - \alpha - \beta$. We get (4) in the form

$$\mathbf{U}_{\alpha,\beta} = \alpha \mathbf{X}_1 + \beta \mathbf{X}_2 + \gamma \mathbf{X}_0. \tag{5}$$

Using Lemma 2.1 (a) and (c) several times we obtain

$$p_{\mathbf{U}_{\alpha,\beta}}(\mathbf{u}) = \left(p_\alpha * p_\beta * p_\gamma\right)(\mathbf{u}), \tag{6}$$

where $p_a(\mathbf{x}) = \frac{1}{a^2} p(\frac{\mathbf{x}}{a})$ if $a \neq 0$, and $p_a(\mathbf{x}) = \delta(\mathbf{x})$ if $a = 0$ (Dirac delta).

We are now able to write out the MSA transform of $f(\mathbf{x})$ in terms of the probability density function as

$$\begin{aligned} F(\alpha,\beta) = E[f(\mathbf{U}_{\alpha,\beta})] &= \int_{\mathbb{R}^2} f(\mathbf{u}) p_{\mathbf{U}_{\alpha,\beta}}(\mathbf{u})\, d\mathbf{u} \\ &= \int_{\mathbb{R}^2} f(\mathbf{u}) \left(p_\alpha * p_\beta * p_\gamma\right)(\mathbf{u})\, d\mathbf{u} \\ &= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} f(\mathbf{u}) p_\alpha(\mathbf{x}) p_\beta(\mathbf{y}) p_\gamma(\mathbf{u}-\mathbf{x}-\mathbf{y})\, d\mathbf{x} d\mathbf{y} d\mathbf{u}. \end{aligned} \tag{7}$$

Using $p(\mathbf{x}) = \frac{1}{\|f\|_{L^1}} f(\mathbf{x})$ and assuming that all $\alpha, \beta, \gamma \neq 0$, this may be written in the form

$$\begin{aligned} F(\alpha,\beta) = \frac{1}{\alpha^2\beta^2\gamma^2} \frac{1}{\|f\|_{L^1}^3} \int_{\mathbb{R}^2}\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} f(\mathbf{u}) f(\frac{\mathbf{x}}{\alpha}) f(\frac{\mathbf{y}}{\beta}) \\ f(\frac{\mathbf{u}-\mathbf{x}-\mathbf{y}}{\gamma})\, d\mathbf{x} d\mathbf{y} d\mathbf{u} \end{aligned} \tag{8}$$

If for instance $\alpha, \beta \neq 0$ and $\gamma = 0$, which means that $\beta = 1 - \alpha$, one has $p_\gamma = \delta(\mathbf{x})$ in (7), and (8) becomes

$$F(\alpha, 1-\alpha) = \frac{1}{\alpha^2(1-\alpha)^2 \|f\|_{L^1}^2} \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} f(\mathbf{u}) f(\frac{\mathbf{x}}{\alpha}) f(\frac{\mathbf{u}-\mathbf{x}}{1-\alpha})\, d\mathbf{x}\, d\mathbf{u}.$$

Similar modifications give $F(\alpha,\beta)$ when $\alpha = 0$ or $\beta = 0$. As we can see from (7) the transform is based on multi-scale convolution kernels, and thus it is called Multi-Scale Autoconvolution (MSA).

At this point, we may explain the requirement $f \in L^1 \cap L^2$ in Definition 4. From (7) we see that $F(\alpha,\beta)$ is defined if the function

$$f(\mathbf{u}) \left(p_\alpha * p_\beta * p_\gamma\right)(\mathbf{u}) \tag{9}$$

is integrable. Since $g * h \in L^2$ when $g \in L^1$ and $h \in L^2$, we have $f \in L^2$ and $p_\alpha * p_\beta * p_\gamma \in L^2$ when $f \in L^1 \cap L^2$, so that (9) is integrable by the Cauchy-Schwarz inequality.

It is essential for the implementation of MSA that the transform may be computed in the frequency domain. We define the Fourier transform pair as

$$\hat{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^2} e^{-j2\pi \mathbf{x}\cdot\xi} f(\mathbf{x})\, d\mathbf{x} \quad \text{and} \quad f(\mathbf{x}) = \int_{\mathbb{R}^2} e^{j2\pi \mathbf{x}\cdot\xi} \hat{f}(\boldsymbol{\xi})\, d\boldsymbol{\xi}.$$

Using the Plancherel formula, $\int_{\mathbb{R}^2} f\bar{g} = \int_{\mathbb{R}^2} \hat{f}\bar{\hat{g}}$, in (8), and noting that the Fourier transform takes convolutions into products, we obtain

$$F(\alpha,\beta) = \frac{1}{\hat{f}(\mathbf{0})^3} \int_{\mathbb{R}^2} \hat{f}(-\boldsymbol{\xi})\hat{f}(\alpha\boldsymbol{\xi})\hat{f}(\beta\boldsymbol{\xi})\hat{f}(\gamma\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{10}$$

which holds for all α, β.

C. Properties of MSA

Here we shall present some important properties of the MSA transform. The first one was already considered above.

Property 1: The value of the MSA transform $F(\alpha,\beta)$ and the moments $F^k(\alpha,\beta)$ are invariant against any affine transformation $\mathcal{A}(\mathbf{x}) = \mathbf{T}\mathbf{x} + \mathbf{t}$, where $\mathbf{t} \in \mathbb{R}^2$and $\mathbf{T}$ is a 2×2 nonsingular matrix whose elements belong to $\mathbb{R}$.

The next property considers symmetries of the MSA transform. For this we will use the notation $F(\alpha,\beta) = F(\alpha,\beta,\gamma)$, where it is always understood that $\gamma = 1 - \alpha - \beta$.

Property 2: The transform $F(\alpha,\beta,\gamma)$ has the following symmetries.

1. $F(\alpha,\beta,\gamma) = F(\sigma(\alpha,\beta,\gamma))$ for any permutation σ of three elements,
2. $F(\alpha,\beta,\gamma) = \frac{1}{\alpha^2}F(\frac{1}{\alpha},-\frac{\beta}{\alpha},-\frac{\gamma}{\alpha}) = \frac{1}{\beta^2}F(-\frac{\alpha}{\beta},\frac{1}{\beta},-\frac{\gamma}{\beta}) = \frac{1}{\gamma^2}F(-\frac{\alpha}{\gamma},-\frac{\beta}{\gamma},\frac{1}{\gamma})$.

The first part follows immediately from (10), and the second part is obtained from (10) by changing variables in the integral. Property 2 is particularly important because one can use it to find small regions in the (α,β) plane which determine all the MSA transform values. There are several possibilities, and one is presented in the Figure 2.

Property 3: Multi-Scale Autoconvolution can be generalized to cover a group of transforms that are invariant against other linear transformations of the image coordinates.

Example: Let us consider a transform which is invariant against translation, rotation, and scaling. In such a case, we only need to draw two samples from $p(x,y)$: $\mathbf{x}_0$ which defines the

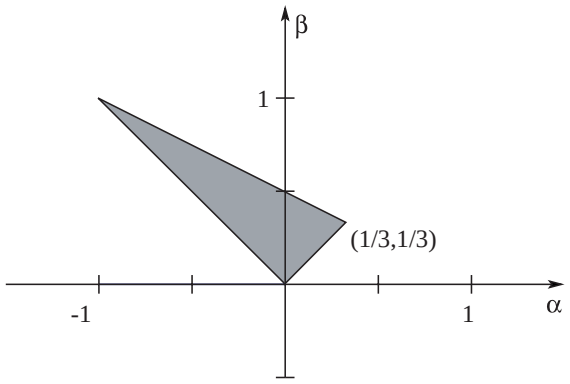


Fig. 2. Illustration of the (α, β) plane region that will give all the MSA transform values.

center of the rotation and scaling, and $\mathbf{x}_1$ which represents the orientation and the extent of the rotated and scaled x axis. Using these points, we can express the transformation as follows:

$$\mathbf{u} = \alpha(\mathbf{x}_1 - \mathbf{x}_0) + \beta\mathbf{S}(\mathbf{x}_1 - \mathbf{x}_0) + \mathbf{x}_0 \tag{11}$$

where $\mathbf{S} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. This transform is invariant against translation, rotation and scaling of the image coordinates. It can be shown that the corresponding probability density function of $\mathbf{u}$ is

$$p_{\mathbf{U}_{\alpha,\beta}}(\mathbf{u}) = (p_{\mathbf{R}_1\mathbf{X}_1} * p_{\mathbf{R}_2\mathbf{X}_0})(\mathbf{u}) \tag{12}$$

where $\mathbf{R}_1 = \alpha\mathbf{I} + \beta\mathbf{S}$, $\mathbf{R}_2 = (1-\alpha)\mathbf{I} - \beta\mathbf{S}$, and $p_{\mathbf{R}_1\mathbf{X}_1}$ and $p_{\mathbf{R}_2\mathbf{X}_0}$ are probability density functions of $\mathbf{R}_1\mathbf{X}_1$ and $\mathbf{R}_2\mathbf{X}_0$ obtained from Lemma 2.1. The transform coefficients are then obtained similarly to Definition 4.

The next two properties are concerned with the MSA transform and convex sets. If $K \subseteq \mathbb{R}^2$ is a compact set define

$$\chi_K(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in K, \\ 0 & \text{otherwise.} \end{cases}$$

We say that K is convex if for any $\mathbf{x}, \mathbf{y} \in K$, the line segment $\{(1-t)\mathbf{x} + t\mathbf{y} \,|\, 0 \le t \le 1\}$ is contained in K. We will also use the notations $\overline{A}$ for the closure and $\mathrm{int}(A)$ for the interior of a set $A \subseteq \mathbb{R}^2$. The proofs of the following two properties are found in [22].

Property 4: Let K be a compact subset of $\mathbb{R}^2$ and let $f = \chi_K$. If K is convex then $F(\alpha, \beta) = 1$ whenever $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1 - \alpha$.

Property 5: Let K be a compact subset of $\mathbb{R}^2$ which satisfies $K = \overline{\text{int}(K)}$. If $f = \chi_K$ and $F(\frac{1}{2}, \frac{1}{2}) = 1$, then K is convex.

In Figure 3, we have plotted the MSA transforms of three images. Notice the symmetries and the behavior with convex objects.

D. MSA in higher dimensions

It is easy to extend the definition of the MSA transform to functions in $\mathbb{R}^n$, where $n \geq 3$. The transform will have natural corresponding properties, including affine invariance. For higher dimensional applications, the most important case is $n = 3$, and there has been interest in 3D shape analysis and classification of 3D gray scale models [23], [24], [25]. However, the main subject of this article is the two-dimensional case, and we will only briefly discuss the theory in $\mathbb{R}^n$.

Definition 5: Let $f \in L^1(\mathbb{R}^n) \cap L^2(\mathbb{R}^n)$ with $f \geq 0$ be an image intensity function in $\mathbb{R}^n$. We let $p(\mathbf{x}) = \frac{1}{\|f\|_{L^1}} f(\mathbf{x})$ be the corresponding probability density function. Let $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_n$

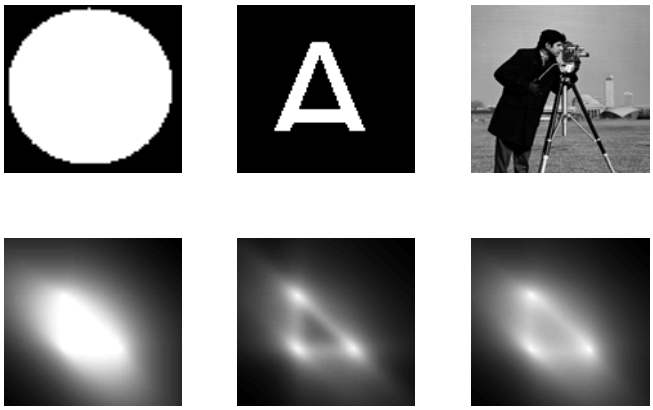


Fig. 3. Three images and their MSA transforms for α, β ranging from -1 to 2. α is on the horizontal axis and β is on the vertical axis.

be independent random variables with probability density function p, and define

$$\begin{aligned}\mathbf{U}_{\alpha_1,\ldots,\alpha_n} &= \mathbf{X}_0 + \alpha_1(\mathbf{X}_1 - \mathbf{X}_0) + \ldots + \alpha_n(\mathbf{X}_n - \mathbf{X}_0) \\ &= \alpha_1\mathbf{X}_1 + \ldots + \alpha_n\mathbf{X}_n + \alpha_{n+1}\mathbf{X}_0\end{aligned}$$

where $\alpha_{n+1} = 1 - \alpha_1 - \ldots - \alpha_n$. We define the kth moment of $f(\mathbf{U}_{\alpha_1,\ldots,\alpha_n})$ as

$$F^k(\alpha_1, \ldots, \alpha_n) = E[f(\mathbf{U}_{\alpha_1,\ldots,\alpha_n})^k],$$

and the MSA transform of f is the first moment

$$F(\alpha_1, \ldots, \alpha_n) = E[f(\mathbf{U}_{\alpha_1,\ldots,\alpha_n})].$$

The MSA transform has similar expressions as in the two-dimensional case. An analogue of Lemma 2.1 gives that

$$p_{\mathbf{U}_{\alpha_1,\ldots,\alpha_n}}(\mathbf{u}) = (p_{\alpha_1} * \ldots * p_{\alpha_{n+1}})(\mathbf{u}),$$

where $p_a(\mathbf{x}) = a^{-n}p(\mathbf{x}/a)$ for $a \neq 0$, and $p_a(\mathbf{x}) = \delta(\mathbf{x})$ for $a = 0$ (Dirac delta). We then have

$$\begin{aligned}F(\alpha_1, \ldots, \alpha_n) &= \int_{\mathbb{R}^n} f(\mathbf{u})(p_{\alpha_1} * \ldots * p_{\alpha_{n+1}})(\mathbf{u})\, d\mathbf{u} \\ &= \frac{1}{(\alpha_1 \cdots \alpha_{n+1})^n \|f\|_{L^1}^{n+1}} \int f(\mathbf{u}) f(\frac{\mathbf{x}_1}{\alpha_1}) \cdots f(\frac{\mathbf{x}_n}{\alpha_n}) \\ &\quad f(\frac{\mathbf{u} - \mathbf{x}_1 - \ldots - \mathbf{x}_n}{\alpha_{n+1}})\, d\mathbf{x}_1 \cdots d\mathbf{x}_n\, d\mathbf{u}\end{aligned}$$

if each $\alpha_j \neq 0$, and with straightforward modifications if some of the α_j are zero. In the last expression there are $n + 1$ integrals over $\mathbb{R}^n$.

We write $\hat{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^n} e^{-j2\pi\mathbf{x}\cdot\xi} f(\mathbf{x})\, d\mathbf{x}$ for the Fourier transform of f. Then the Plancherel formula gives the following expression for the MSA transform:

$$F(\alpha_1, \ldots, \alpha_n) = \frac{1}{\hat{f}(\mathbf{0})^{n+1}} \int_{\mathbb{R}^n} \hat{f}(-\boldsymbol{\xi}) \hat{f}(\alpha_1\boldsymbol{\xi}) \cdots \hat{f}(\alpha_{n+1}\boldsymbol{\xi})\, d\boldsymbol{\xi}.$$

Most properties of the two-dimensional MSA transform have obvious counterparts in n dimensions, and the proofs are similar. Most importantly, $F(\alpha_1, \ldots, \alpha_n)$ and the higher moments $F^k(\alpha_1, \ldots, \alpha_n)$ are invariant under spatial affine transformations $\mathbf{x} \mapsto \mathbf{T}\mathbf{x} + \mathbf{t}$, where $\mathbf{T}$ is an $n \times n$ nonsingular real matrix and $\mathbf{t} \in \mathbb{R}^n$. The symmetry properties are stated using the notation $F(\alpha_1, \ldots, \alpha_n) = F(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$, where always $\alpha_1 + \ldots + \alpha_n + \alpha_{n+1} = 1$:

1. $F(\alpha_1, \ldots, \alpha_{n+1}) = F(\sigma(\alpha_1, \ldots, \alpha_{n+1}))$ whenever σ is a permutation of $n + 1$ elements,

2. $F(\alpha_1,\ldots,\alpha_{n+1}) = \frac{1}{\alpha_j^n} F(-\frac{\alpha_1}{\alpha_j},\ldots,-\frac{\alpha_{j-1}}{\alpha_j},\frac{1}{\alpha_j},-\frac{\alpha_{j+1}}{\alpha_j},\ldots,-\frac{\alpha_{n+1}}{\alpha_j})$ when $1 \le j \le n+1$ and $\alpha_j \neq 0$.

Also Properties 4 and 5 have analogs in higher dimensions.

III. Implementational Issues

When applied to digital images, a discrete form of MSA must be used. A direct evaluation of the triple integral in (8) would be computationally expensive, and it is crucial for the implementation of MSA that the transform may be computed from the single integral in (10) which involves the Fourier transform of f. Fast algorithms for computing Fourier transforms are easily available, e.g. [26]. Discretizing the integral (10) gives

$$F(\alpha,\beta) = \frac{1}{N_1 N_2}\frac{1}{\hat{f}(\mathbf{0})^3}\sum_{i=0}^{N_1N_2-1} \hat{f}(-\mathbf{w}_i)\hat{f}(\alpha\mathbf{w}_i)\hat{f}(\beta\mathbf{w}_i)\hat{f}(\gamma\mathbf{w}_i) \tag{13}$$

where $\mathbf{w}_i$ are $N_1 \times N_2$ points in a rectangular grid in $\mathbb{R}^2$. We may think of $\hat{f}$ as the discrete Fourier transform of a discrete function f, i.e. f is an $N_1 \times N_2$ matrix which represents the digital image.

Formula (13) contains scaled versions of the continuous Fourier transform $\hat{f}$. We know that $\hat{f}(a\mathbf{w}) = \widehat{f_a}(\mathbf{w})$ where $f_a(\mathbf{x}) = \frac{1}{a^2} f(\frac{\mathbf{x}}{a})$, so we may write (13) equivalently as

$$F(\alpha,\beta) = \frac{1}{N_1 N_2}\frac{1}{\hat{f}(\mathbf{0})^3}\sum_{i=0}^{N_1N_2-1} \hat{f}(-\mathbf{w}_i)\hat{f}_\alpha(\mathbf{w}_i)\hat{f}_\beta(\mathbf{w}_i)\hat{f}_\gamma(\mathbf{w}_i). \tag{14}$$

This has the advantage that we have scaled versions of the original image instead of the Fourier transform, and accurate scaling is easier to do for the original image. Therefore we will base our implementation of the MSA transform on (14).

Implementing (14) is quite straightforward, apart from the fact that we need a method to interpolate or decimate the original image f. In our implementation, the interpolation was done by adding zeros between the known samples. The decimation, on the other hand, was done by dividing the image into equal sized regions and then summing the elements in them to form one element in a decimated image. These methods were chosen because in the probability sense they preserve statistical relations between each part of the image. One should also keep in mind that the Fourier transform length must be long enough to avoid the wrap-around error, so if the original image f is an $M_1 \times M_2$ matrix, the transform length N_i must satisfy $N_i \ge (|\alpha|+|\beta|+|\gamma|)M_i - 2$.

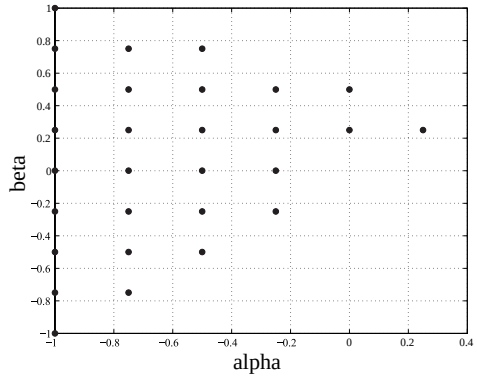


Fig. 4. The 29 (α, β) pairs used to obtain the MSA based features in our experiments.

To ensure this in our implementation, the transform length for both dimensions $i = 1, 2$ was taken to be $(|\alpha| + |\beta| + |\gamma|)M_i$. The actual 29 (α, β) pairs that we selected for obtaining the features in our experiments are shown in Figure 4. The Matlab algorithm that we implemented and used in Section V to compute the MSA transform values for any (α, β) can be retrieved from the URL: http://www.ee.oulu.fi/research/imag/msa/.

IV. Computational Complexity

Let us make some observations about the computational complexity of the Multi-Scale Auto-convolution compared to some other affine invariant feature extraction methods, namely affine

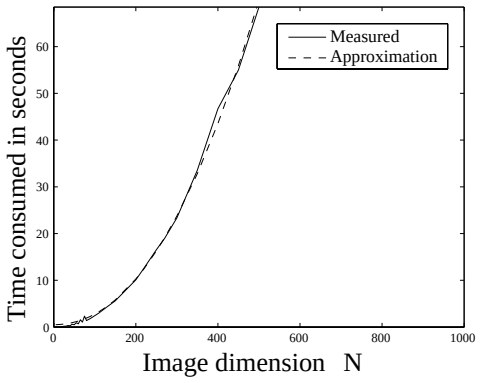
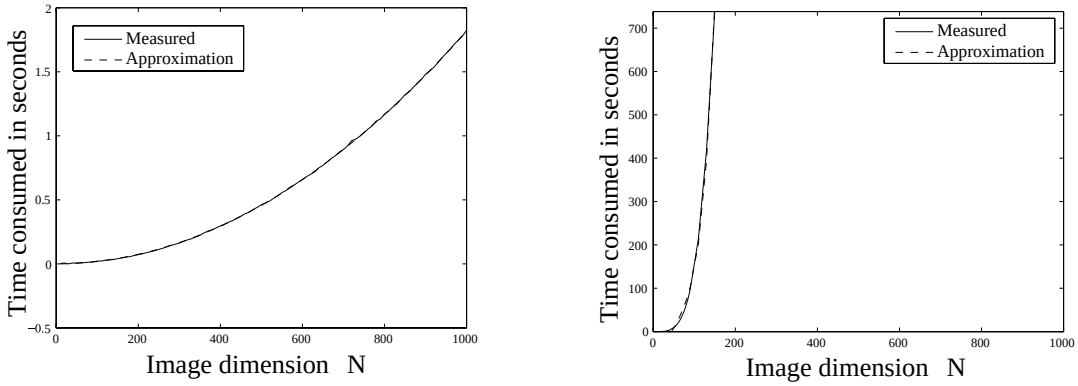


Fig. 5. The measured and approximated computation times for 29 MSA values.



(a) Affine invariant moments

(b) Cross-weighted moments

Fig. 6. The measured and approximated computation times for 4 affine invariant and cross-weighted moment values.

invariant moments [15] and cross-weighted moments [18]. However, one should keep in mind that the actual computation times depend on the specific implementation and computing resources.

We first consider the MSA case, and assume the implementation described in Section III for an $N \times N$ image. We need three scaled versions of the image whose size depends on the (α, β) pair, but, as known from the theory, we only need α and β in the region in Figure 2, so we may assume that $|\alpha| \leq 1$, $|\beta| \leq 1$ and $|\gamma| \leq 2$. If we neglect the operations needed for image decimation, we have to perform 4 discrete Fourier transforms, and each is a $O(N^2 \log_2 N)$ operation with standard Fast Fourier Transform (FFT). After these we must compute their products, requiring $3N^2$ operations, and the sum requires $N^2 - 1$ operations. Also computing $\hat{f}(\mathbf{0})$ uses $N^2 - 1$ operations, but this computation is only needed once. Taking into account the final multiplication with a constant, we have a total of $O(N^2 \log_2 N)$ operations for one (α, β) pair, and we will compute 29 of these. Figure 5 illustrates the measured and approximated MSA computation times, where the approximation curve was computed by fitting the data to an $x^2 \log_2 x$ model.

The affine invariant moment method, using second and third order algebraic invariants, is straightforward to implement according to the theory in [15]. It can be easily seen from this that one needs $O(N^2)$ operations for computing one invariant for an $N \times N$ image, and usually 4 invariants are used for classification. Figure 6(a) shows a graph of measured and approximated computational times, and we can see that at least thus far the second order polynomial model

fits well.

Also in the case of cross-weighted moments [18], the implementation is trivial, only that due to the cross-operation the amount of the operations needed rises to $O(N^4)$ for 4 invariants for an $N \times N$ image. Now Figure 6(b) illustrates the computational time taken by computing the cross-weighted moments, as well as the model approximation.

Comparing these results, we can immediately see that the computational complexity of the cross-weighted moments increases very quickly, leading to the fact that this method is practical only when the image dimensions are much smaller than 100. As expected, the classical affine invariant moments have the smallest computational complexity, but nevertheless the computational load for MSA is still bearable, also for larger images.

V. Experiments

The purpose of this section is to illustrate the performance of the MSA compared to the affine invariant moments and the cross-weighted moments in various object classification problems. The comparison methods are described in [15] and [18], and we have implemented these methods as discussed in the articles. For the cross-weighted moments, we must choose certain values for the exponent s, and according to [18] we chose the values 0.8, 0.95, 0.99, 1.05, 1.1, 1.2, 1.5 and 1.07 to produce 4 absolute invariants. The reason for not computing more than 4 invariants is the high computational complexity of the method. With the affine invariant moments we constructed the four invariants using second and third order moments.

In the experiments, we have used the nearest neighbor classifier, but, before passing the computed features to it, they were normalized by subtracting the mean and dividing by the standard deviation. We used three different test sets: binary images, gray scale images, and gray scale images with 3-D objects viewed from different angles. All the images were normalized to have intensity values in the interval $[0, 1]$, where 0 is black and 1 is white. Random affine transformations were generated by randomly choosing the affine transformation matrix and translation according to following expressions:

$$A = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \quad t = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \tag{15}$$

where $\omega, \phi \in [0, 2\pi)$, $d \in [0.35, 1]$ and $t_1, t_2 \in [-5, 5]$ are uniformly distributed random variables.

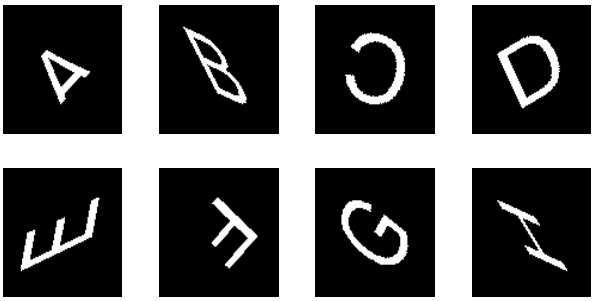


Fig. 7. Randomly affine transformed versions of the first eight letters in the alphabet.

A. Binary image classification

In the first experiment, the classified images were all the capital letters in the alphabet. After constructing the nearest neighbor classifier using the images with no distortions, we took as test images randomly affine transformed versions of the same letters. A few samples of these test images are illustrated in Figure 7. The purpose of this test is to verify that the methods work correctly in the case of real digital images. As a result of 2080 test runs using all three methods, we achieved almost 100% accurate classification results.

In order to make the situation a bit more challenging, we introduced uniformly distributed binary noise into these same letters. As we know, all the methods are, however, quite sensitive to single separated noise pixels in the background, so before recognition we performed a simple connectivity analysis to remove them. The final affine transformed test letters had a rough boundary, and holes in the interior. Samples of these are illustrated in Figure 8. The noise level D here is the probability of a pixel to be disturbed by noise before connectivity analysis.

Figure 9 shows the results of 2080 test runs using the nearest neighbor classifier under different noise levels. As we see from the results, the overall performance of MSA and cross-weighted moments is superior to the affine invariant moment classification at all tested noise levels. The

Fig. 8. Samples of noise disturbed letters after connectivity analysis. The noise level $D = 0.07$.

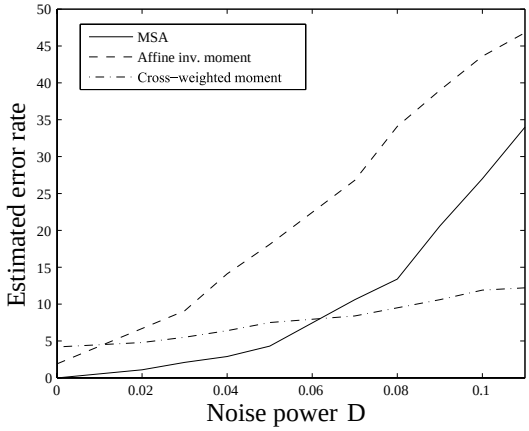


Fig. 9. Error rates for the letter classification at different noise levels.

affine invariant moment method seems to collapse after $D = 0.02$, while the others give bearable error rates up to the noise level $D = 0.06$. If we compare the MSA and cross-weighted moments, it seems that MSA is more reliable with low noise rates, and when a certain limit is reached the error rate starts to increase more rapidly than with the cross-weighted moments.

B. Gray-scale image classification

In the second experiment, we classified gray-scale images of 94 different fish, some illustrated in Figure 10. The resolution of the images was 200x400. First, we trained the classifier using the image with no distortions and then tested the methods by classifying the same images which had now undergone some distortion, not necessarily affine. The experiments were done in order to see how the methods handle different types of deviations from the ideal theory. The cross-weighted moments were left out of these experiments, because one needs fairly large images to get the distortion phenomena clearly visible, and the method is not computationally feasible

Fig. 10. Samples of fish images used in the gray-scale classification problem.

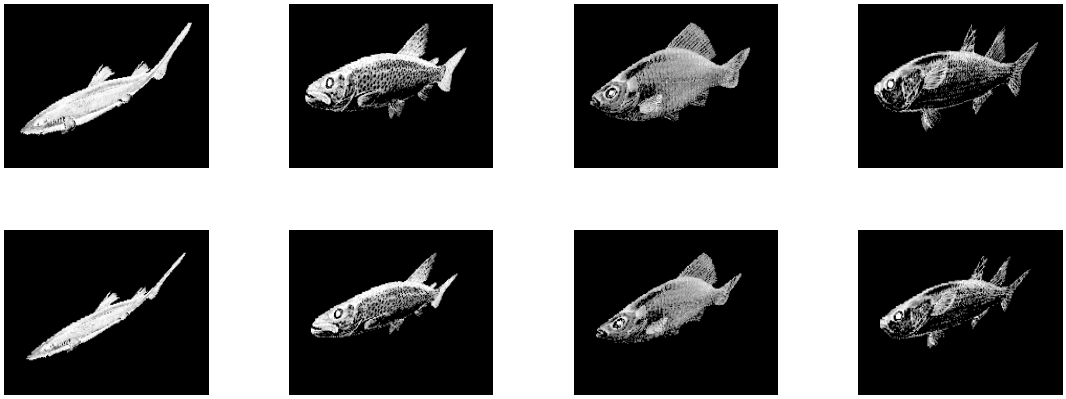


Fig. 11. Samples of projective transformed fish with nonlinearities c from top to bottom $\{20, 40\} \cdot 10^{-4}$.

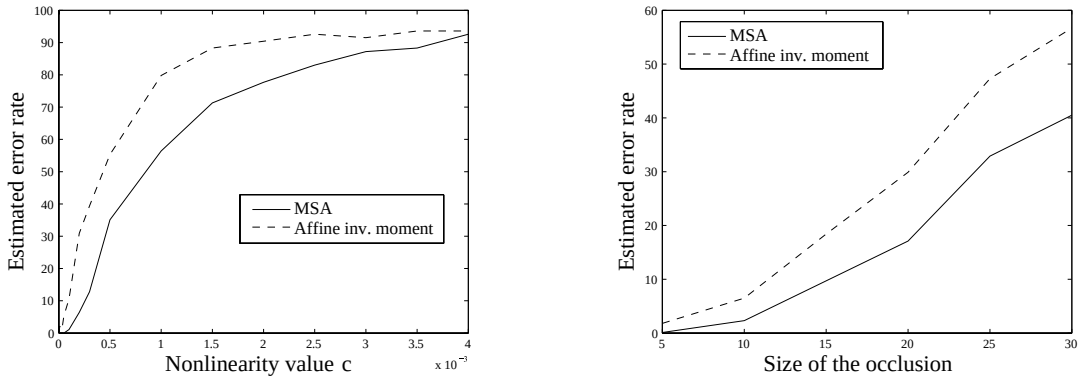
with images larger than 100x100 (see Figure 6).

1) Projective transformation: In the first test case, the perturbation introduced was a computer-simulated projective transformation that is the real model for the pinhole camera and an adequate approximation for many real cameras. Here we recall that a (plane) projective transformation $\mathscr{P} = \mathscr{P}_P : \mathbb{R}^2 \to \mathbb{R}^2$ is a map

$$\mathscr{P}(x) = \Big(\frac{p_{11}x_1 + p_{12}x_2 + p_{13}}{p_{31}x_1 + p_{32}x_2 + p_{33}}, \frac{p_{21}x_1 + p_{22}x_2 + p_{23}}{p_{31}x_1 + p_{32}x_2 + p_{33}}\Big),$$

where $p_{31}x_1 + p_{32}x_2 + p_{33} \neq 0$. In our experiment these parameters were chosen so that $p_{11} = p_{22} = p_{33} = 1$, $p_{12} = p_{13} = p_{21} = p_{23} = 0$, and the value $c = |p_{31}| + |p_{32}|$, which can be seen as a measure for the nonlinearity of the transformation, was varied from $1 \cdot 10^{-6}$ to $4 \cdot 10^{-3}$. We have plotted some samples of the resulting distorted images in Figure 11. The classification errors with different methods are illustrated in Figure 12(a). One can observe that the error rates with both methods start to increase rapidly, but MSA is clearly more tolerant against the nonlinearities caused by the perspective projection.

2) Occlusions combined with affine transformation: In many situations, an object may be disturbed by some other unwanted object which causes partial occlusion. We tested this by introducing different sized and randomly situated occlusions on the fish images. The occlusion was a square shape, with a uniform gray-value 0.5 and the center of the occlusion always on the fish. Before the occlusion, we made a random affine transformation, because the occlusion is many times combined with some view angle distortions. We show some samples of the resulting



(a) Projective distortion (b) Occlusion distortions

Fig. 12. Error rate graphs for gray-scale classification in the case of projective and occlusion distortions.

test images in Figure 13. The actual classification errors as a mean of 1410 test runs, with different methods and different occlusion sizes are then illustrated in Figure 12(b).

3) Illumination changes combined with affine transformation: It is usually very difficult to set up ideally uniform lighting conditions, and we typically get some illumination distortions in the images. We tested how the methods behave in the case of linear illumination distortion combined with the affine transformation. The illumination distortion was done so that starting from the vertical center line, to the left we linearly decreased the gray-scale values and to the

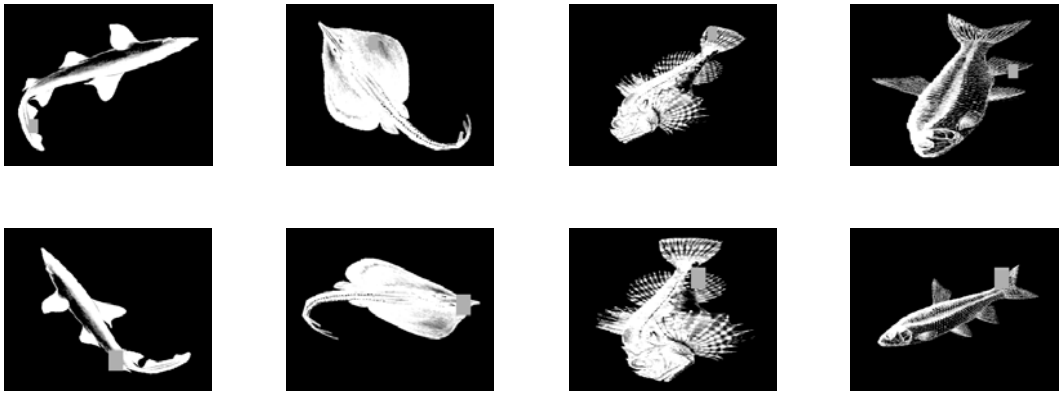
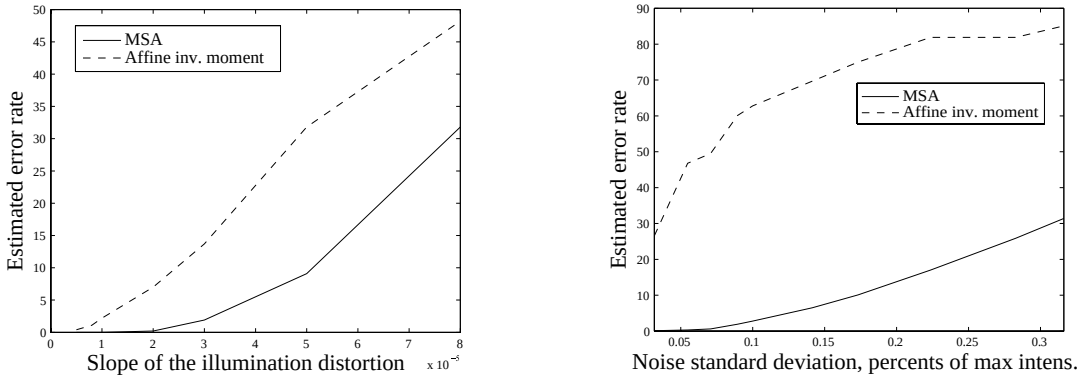


Fig. 13. Samples of occluded and affine transformed objects with occlusion sizes from top to bottom $\{20, 30\}$.



(a) Illumination distortions

(b) Gaussian noise distortions

Fig. 14. Error rate graphs for gray-scale classification in the case of illumination and Gaussian noise distortions.

right we similarly increased them. We performed 1410 test runs with different slopes, and the results are illustrated in Figure 14(a).

4) Gaussian noise combined with affine transformation: Since we do not have perfect cameras, we always get some noise from various sources like CCD arrays. In this experiment, we performed an affine transformation to the image and then added some Gaussian noise. However, the size of the background has a strong effect on both of the methods, since due to noise the large background will acquire considerable probability mass distracting the features. To diminish this effect, we performed a simple threshold operation to remove the background. This type of distortion is difficult since it affects both the gray scale and the shape information. The classification results are shown in the graph in Figure 14(b). With this type of distortion, MSA most clearly outperforms the affine invariant moments, as it seems that they can tolerate almost no noise.

C. Classification in the presence of real view angle disturbances

As a final test, we wanted to see how the methods perform with images having real view angle distortions. In order to test this, we used the Coil-100 image database of Columbia University as a test set. Coil-100 contains 100 different objects each viewed from 72 different angles, i.e. every 5 degrees. For the details of the database arrangements, see [27]. The RGB images from the database had to be first converted to gray scale. Then, the first images of each object,

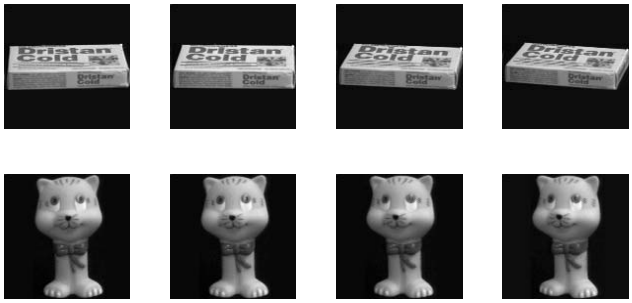


Fig. 15. Two samples of the Coil-100 database objects viewed from angles $0°$, $5°$, $10°$ and $15°$.

those taken at angle 0, were used as the training images to create the classifiers. To also get some idea of the absolute performance that can be achieved, we included 20 randomly affine transformed versions of the original training image in the training set. This improves the results considerably and offers some insight as to how affine transformation can be used as a model for real distortions. With this larger training set we were also able to use a linear discriminant classifier, which gave slightly better results than the nearest neighbor classifier.

The test set was taken to include all the objects in angles $\pm 5°$, $\pm 10°$ and $\pm 15°$, in order to see how the classifier performed in the presence of real perspective transformations. Figure 15 contains some samples of these test objects. This classification task is very challenging, because the objects are not quite planar compared to the distance of the camera. From the test results in

TABLE I

ERROR RATES OF THE COIL-100 CLASSIFICATION PROBLEM AT VIEWING ANGLES $-15 \ldots 15$ USING LINEAR DISCRIMINANT CLASSIFIER.

View angle	MSA	Affine invar. moments	C-w moments
$-15°$	2 %	44 %	46 %
$-10°$	1 %	31 %	27 %
$-5°$	0 %	21 %	15 %
$5°$	0 %	23 %	25 %
$10°$	0 %	41 %	34 %
$15°$	1 %	55 %	49 %

Table I one can see that MSA is superior compared to both comparison methods with all tested angles and classifiers. The error rates $0\% - 2\%$ with the linear discriminant classifier indicate that the proposed method is quite robust against the approximation errors caused by the view angle disturbances.

VI. Discussion

Let us say a few words about the experimental results achieved in Chapter V. In binary image classification, all the tested methods performed well when no noise was present. When binary noise was introduced, we observed that MSA seems to perform reliably up to a certain noise threshold, after which it collapses quickly. One could argue that this does not come as a surprise, but is a consequence of the strong nonlinear structure. By looking at our benchmark method, the affine invariant moments, one can see that both MSA and cross-weighted moments clearly outperform it in the sense of classification accuracy, with any noise level. However, using higher order moments for producing more invariants might increase the performance of affine invariant moments.

The gray scale classification experiments tested the performance of the methods under different types of distortions that are usually encountered in the photographing process. The tested cases were perspective transformation, occlusion, nonuniform illumination and image noise. As it is difficult to simulate these distortions using small images, we had to omit the computationally expensive cross-weighted moments from this experiment. From the achieved results, one can see that the methods may not perform well in the presence of nonaffine distortions, which was to be expected. However, the final test using the Coil-100 database verified that if one is able to eliminate the object background and if the photographing conditions are reasonably good, then these distortions will stay in such limits that a reliable recognition system can be constructed. Altogether one should not pay too much attention to the absolute error levels, but more to the relative performance of the methods. This is because the actual recognition system will in practice use enhancements like more sophisticated classifier structures, or more training images of the actual object. From the error rate graphs, one can observe that, apart from the Gaussian noise test, the methods have graphs of similar shape, but in all the test cases the relative performance of the MSA is significantly better. This result can also be observed from the real world example with the Coil-100 database.

As an overall result, we might conclude that there does not seem to be a single superior feature computing method or classifier which would give the best performance in all circumstances, and this is exactly what we were expecting. Anyway, using the classification error as a measure, we would argue that the best overall performance of all the tested feature computing methods was provided by MSA. Even in the noise disturbed binary classification, the points where the cross-weighted moments outperformed MSA were already at high noise powers. One can, of course, notice that the affine invariant moments are, by any measure, the most efficient to compute, but one should consider the trade off between computational complexity and classification accuracy depending on the application. It should also be noted that the computing time for MSA can be reduced using fewer (α, β) pairs, which might on the other hand lead to some decrease in the classification performance. In fact the optimal choice of (α, β) pairs will be one of the important future research issues.

VII. Conclusions

In this article we presented a novel transform called Multi-Scale Autoconvolution, which is applicable to affine invariant feature extraction. The proposed method was analyzed theoretically and assessed experimentally by comparing it to other similar techniques. We also considered the implementation and computational complexity of the method.

In the theoretical part of the article, we defined the MSA transform and gave different expressions for it. We also proved its affine invariance property along with a number of other useful properties. The experiments performed with similar comparison methods showed that apart from the binary image classification scheme with high noise levels, MSA provided reliable results in the classification error sense.

Theory and experiments made it clear that the computational complexity of the cross-weighted moments rises quickly with the image size, which generally makes the method computationally impractical already for images smaller than 100×100. Although the affine invariant moments provided superior computational speed with $O(N^2)$ complexity, the $O(N^2 \log_2 N)$ complexity of MSA was not much worse, and it will provide a computationally conceivable alternative.

As observed in the experiments, the affine transformation provides an adequate approximation of real view angle distortions, and hence affine invariant feature computing methods remain an important part of statistical pattern recognition. From the comparison with other affine invariant

techniques, it appears that MSA based classifiers provide accurate results for such problems. Bringing MSA to real applications still requires some further investigation of the computational algorithms and feature selection. However, MSA has already proved its discriminative capacity, and we expect to see MSA in 3D object recognition applications in the future.

ACKNOWLEDGMENTS

The authors would like to thank the Academy of Finland for providing funding for this research (project no. 102083). We would also like to acknowledge Prof. Petrou and Dr. Kadyrov for offering the fish image database for our experiments.

APPENDIX I
THE PROOF OF LEMMA 2.1

(a) Let $A \subseteq \mathbb{R}^2$ be measurable. We denote by 1_B the indicator function of a set B in some probability space, so that $1_B = 1$ in B and $1_B = 0$ outside B. Since $\mathbf{X}$ and $\mathbf{Y}$ are independent, we have

$$\begin{aligned} P(\mathbf{X}+\mathbf{Y} \in A) &= E(1_{\{\mathbf{X}+\mathbf{Y}\in A\}}) \\ &= \int_{\mathbb{R}^2} \Big(\int_{\mathbb{R}^2} 1_{\{\mathbf{x}+\mathbf{y}\in A\}} p(\mathbf{x})q(\mathbf{y})\, d\mathbf{x} \Big)\, d\mathbf{y} \\ &= \int_{\mathbb{R}^2} \Big(\int_{-\mathbf{x}+A} p(\mathbf{x})q(\mathbf{y})\, d\mathbf{y} \Big)\, d\mathbf{x} \\ &= \int_{\mathbb{R}^2} \Big(\int_{A} p(\mathbf{x})q(\mathbf{y}-\mathbf{x})\, d\mathbf{y} \Big)\, d\mathbf{x} \\ &= \int_{A} \Big(\int_{\mathbb{R}^2} p(\mathbf{x})q(\mathbf{y}-\mathbf{x})\, d\mathbf{x} \Big)\, d\mathbf{y} \end{aligned}$$

by Fubini's theorem and a change of variables. This shows that the probability density function of $\mathbf{X}+\mathbf{Y}$ is the convolution $p * q$.

(b) If $A \subseteq \mathbb{R}^2$ is measurable we have

$$\begin{aligned} P(\mathbf{T}\mathbf{X}+\mathbf{t} \in A) &= P(\mathbf{X} \in \mathbf{T}^{-1}A - \mathbf{T}^{-1}\mathbf{t}) \\ &= \int_{\mathbf{T}^{-1}A-\mathbf{T}^{-1}\mathbf{t}} p(\mathbf{x})\, d\mathbf{x} = \frac{1}{|\mathbf{T}|} \int_A p(\mathbf{T}^{-1}\mathbf{y} - \mathbf{T}^{-1}\mathbf{t})\, d\mathbf{y} \end{aligned}$$

using the change of variables $\mathbf{x} = \mathbf{T}^{-1}\mathbf{y} - \mathbf{T}^{-1}\mathbf{t}$.

(c) This follows from (b) by choosing $\mathbf{T} = \alpha\mathbf{I}$, $\mathbf{t} = 0$.

REFERENCES

[1] N. Götze, S. Drüe, and G. Hartmann, "Invariant object recognition with discriminant features based on local fast-fourier mellin transform," in *Proc. International Conference of Pattern Recognition*, Barcelona, Spain, 2000, pp. 950–951.

[2] S. R. Deans, *The Radon Transform and Some of Its Applications*. Krieger, 1983, 1983.

[3] H. Xiong, T. Zhang, and Y. S. Moon, "A translation- and scale-invariant adaptive wavelet transform," *IEEE Trans. Image Processing*, vol. 9, pp. 2100– 2108, December 2000.

[4] H. Burkhardt and S. SiggelKow, "Invariant features for discriminating between equivalence classes," in *Nonlinear Model-Based Image/Video Processing and Analysis*, I.Pitas and C. Kotropoulis, Eds. New York: John Wiley & Sons, 2001, pp. 269–307.

[5] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Reading MA: Addison-Wesley, 1993.

[6] M. T. Quang and W. W. Boles, "Wavelet-based affine invariant representation: a tool for recognizing planar objects in 3d space," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 846–857, August 1997.

[7] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant fourier descriptors to recognition of 3-d objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 640–647, July 1990.

[8] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Affine invariant model-based object recognition," *IEEE Trans. Robotics and Automation*, vol. 6, no. 5, pp. 578–589, 1990.

[9] D. Forsyth and J. Ponce, *Computer Vision*. New Jersey: Prentice Hall, 2003.

[10] V. H. S. Ha and J. M. F. Moura, "Affine invariant wavelet transform," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2001, pp. 1937–1940.

[11] M. I. Khalil and M. M. Bayoumi, "A dyadic wavelet affine invariant function for 2d shape recognition." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1152–1164, 2001.

[12] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision*. Copenhagen: Springer, 2002, pp. 128–142.

[13] M.-K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Information Theory*, vol. 8, pp. 179–187, February 1962.

[14] T. H. Reiss, “The revised fundamental theorem of moment invariants,” *EEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 830–834, August 1991.

[15] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, January 1993.

[16] J. Ben-Arie and Z. Wang, “Pictorial recognition of objects employing affine invariance in the frequency domain,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 604–618, June 1998.

[17] J. Ben-Arie and Z. Wang, “Gabor kernels for affine-invariant object recognition,” in *Gabor Analysis and Algorithms*, H. Feichtinger and T. Strohmer, Eds. Birkhäuser, 1997.

[18] Z. Yang and F. Cohen, “Cross-weighted moments and affine invariants for image registration and matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 804–814, August 1999.

[19] M. Petrou and A. Kadyrov, “Affine invariant features from the trace transform,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 30–44, January 2004.

[20] J. Heikkilä, “Multi-scale autoconvolution for affine invariant pattern recognition,” in *Proc. International Conference of Pattern Recognition*, Québec, Canada, 2002, pp. 119–122.

[21] E. Rahtu and J. Heikkilä, “Object classification with multi-scale autoconvolution,” in *Proc. International Conference of Pattern Recognition*, Cambridge, England, 2004.

[22] E. Rahtu, M. Salo, and J. Heikkilä, “Convexity recognition using multi-scale autoconvolution,” in *Proc. International Conference of Pattern Recognition*, Cambridge, England, 2004.

[23] I. Rigoutsos, “Well-behaved, tunable 3d-affine invariants,” in *Proc. Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998, pp. 455 – 460.

[24] O. Ronneberg, H. Burkhardt, and E. Schultz, “General-purpose object recognition in 3d volume data sets using gray-scale invariants – classification of airborne pollen-grains recorded with confocal laser scanning microscope,” in *Proc. International Conference of Pattern Recognition*, Québec, Canada, 2002, pp. 290 – 295.

[25] S. S. M Schael, “Invariant grey-scale features for 3d sensor-data,” in *Proc. International Conference of Pattern Recognition*, Québec, Canada, 2002, pp. 531 – 535.

[26] M. Frigo and S. G. Johnson, “FFTW: An adaptive software architecture for the FFT,” in *Proc. International Conference of Acoustics Speech and Signal Processing*, vol. 3, 1998, pp. 1381–1384.

[27] Columbia University, “Coil-100 image database,” URL: www.cs.columbia.edu, 2003.

Esa Rahtu received his M. Sc. degree in electrical engineering in 2004 from the University of Oulu, Finland. Currently, he is a researcher in the Machine Vision Group at the University of Oulu. His research interests include geometric invariants and object classification.

Mikko Salo received his M. Sc. and Ph. Lic. degrees in mathematics in 2001 and 2003 from the University of Oulu, Finland, and the Doctor of Philosophy degree in applied mathematics in 2004 from the University of Helsinki, Finland. In 2002, he was a visiting researcher at the University of Washington in Seattle. Currently he is a researcher in the Department of Mathematics and Statistics / Rolf Nevanlinna Institute at the University of Helsinki. His research interests include inverse problems, partial differential equations, and harmonic analysis.

Janne Heikkilä received his M. Sc. degree in electrical engineering in 1993, and the Doctor of Technology degree in information engineering in 1998 from the University of Oulu, Finland. From 1998 to 2000, he was a postdoctoral fellow in Machine Vision and Media Processing Unit at the University of Oulu. Since 2000 he has been working as an acting professor of signal processing engineering, and an acting professor of information engineering. From 2002 to 2004, he was a professor of digital video processing at the University of Oulu. His research interests include geometric camera calibration, 3-D vision, geometric invariants, image registration, and motion estimation in video coding.