Extraction of Local Symmetries using 
Tensor Field Filtering

Carl-Fredrik Westin        Hans Knutsson
Department of Electrical Engineering, Computer Vision Laboratory
Linköping University, Sweden
westin1@isy.liu.se

Abstract. Feature extraction from a tensor based local image representation is discussed. This tensor image representation keeps statements of local structure, certainty of statement and energy separate. Further processing for obtaining new features also having these three entities separate is achieved by the use of a new concept, tensor field filtering. Tensor filters for smoothing and for extraction of circular symmetries are presented and discussed in particular. The main reason for deriving a tensor filtering procedure is to find a general operator between levels in a feature hierarchy. A tensor representation of features combined with this operator gives some advantages; e.g. when the model on the next higher level do not have support from data this yields “round” tensors. Thus, events not supporting the model are passed on to the next higher level in the norm of the tensors. It is the shape of the tensor that indicates the validity of the statement. Conventional operators in general do not share this property of having a separation between statements of local structure, certainty of statement, and energy.

1 Introduction

Tensor analysis is a generalization of the notions from vector analysis. The need for such a theory is motivated by the fact that there are many physical quantities of complicated nature that cannot naturally be described or represented by scalars or vectors. The additional degree of freedom tensors can carry, compared to scalars and vectors, make them ideal for representation of information. As an example, local deformation of a solid body due to internal forces is described and represented adequately only by tensors.

A key issue in this paper is that of using a “natural” representation for image features in order to make further processing of the information easier. It may be worth pointing out that the problem of finding representations never becomes critical if only one-level operations are considered [Granlund, 1978]. If for example the final goal is to find all edge points in an image, this can be done using a gradient filter followed by some thresholding. If this is the end station of the processing, the specific representation of the achieved image points is almost

\[1\] This paper is a reprint from ICIP’92, IEEE, 1992;371–375.
irrelevant. However, if the information is aimed at being input to a second level operation, things like having continuous variables and relevant distances between events in the signal space start to become of interest. Neglecting such issues can make further processing hard or impossible. Furthermore, if the output from the second level operation is aimed at being input to a third level operation, a natural question arises: How should, in general, operators extracting information be constructed in order to produce relevant representations for the next higher levels?

2 On the use of Tensors in Image Analysis

Only a few examples of the use of tensors can be found in computer vision literature. Researchers having discussed tensors in relation to image analysis are e.g. [Koenderink and van Doorn, 1979], [Koenderink and van Doorn, 1975], [Cyganski et al., 1983], [Cyganski and Orr, 1985], [Knutsson, 1989], [Westin, 1991]. Tensors were developed as a natural representation for physical quantities and relations; they exhibit specific invariances and transformation properties under coordinate system transformation. These properties inspired the use of tensors for describing the relationship between images that differ by linear transformation.

Cyganski and Orr have defined a tensor moment operator used for object recognition. This method does not require any point-to-point correspondence and can handle affine transformation distortion of the objects. The method can be seen as geometric representation of whole images, i.e. the algorithm is global. However, describing images like this with global moments is not very appropriate when more than one object is present at a time. Although the tensor moment theory is elegant, it is in its present form not suitable for general image understanding problems. Koenderink and van Doorn discussed an internal representation of solid shape with respect to vision. They used a stretching tensor to express deformation of retinal images. Knutsson introduced a tensor representation which is attained by combining signal theoretical methods and geometry. More examples of the use of tensors in can be found in differential geometry literature.

3 Tensor Field Filtering

3.1 Definition of Tensor Field Filtering

In this section a definition of tensor field filtering will be presented. In particular, filtering on the orientation tensor field introduced in [Knutsson, 1989] will be discussed.

Definition 1 Tensor field filtering is defined as:

\[ S_{ijkl\ldots}^{mno\ldots}(x) = \sum_p f_{abc\ldots}^{def\ldots}(p) T_{def\ldots}^{abc\ldots}(x-p) \]  

(1)
where:
\( \mathbf{x} \) is a coordinate vector describing a position in image space.
\( \mathbf{p} \) is a coordinate vector describing a position in tensor filter.
\( F_{abc...ijk...}^{def...mno...}(\mathbf{x}) \) denotes the filter.
\( T_{gbc...}^{def...}(\mathbf{x}) \) denotes components of the input tensor field.
\( S_{ijk...}^{mno...}(\mathbf{x}) \) denotes components of the output tensor field.

This definition does not say anything about how the filter coefficients are obtained. It only provides a scheme how to combine the coefficients present.

### 3.2 Filtering of the Orientation Tensor Field

Bärman performed filtering on orientation tensor field data when estimating curvature in 3D and 4D [Bärman, 1991]. The scheme he used is however different from the approach described here. He divides the elements of the tensors into small sub-matrices which he then associated with vectors. Here, each tensor filter consists of tensor coefficients of operates directly on tensor data. The final result after the filtering is a mapping from one tensor field to another. A filter \( F \) used for the filtering of the orientation field needs four indices if the output field should be of the same order (2).

\[
F_{ij}^{kl}(2)
\]

The sum of the tensor operations over all the filter coefficients in \( F \) is the actual convolution process.

\[
S_{kl}(\mathbf{x}) = \sum_{p} F_{ij}^{kl}(\mathbf{p})T_{ij}(\mathbf{x} - \mathbf{p})
\]

Equation (3) provides a very compact formulation of the tensor convolution. This is an example of where tensor calculus is a useful tool. Note that each coefficient in the filter, \( F(\mathbf{x}) \), is a scheme with \( 2 \times 2 \times 2 \times 2 = 16 \) elements. Tensor calculus provides neat and simple rules for how to manipulate all these numbers. It is a good way of organizing the calculations. However, when the manipulation is ready, it is normally possible to find analogue expressions used in vector algebra.

### 3.3 Averaging a Tensor Field

One of the simplest filtering operations on a tensor field that can be performed is averaging. Each of the components in the tensor field can then be interpreted as a scalar field which is convolved with a scalar smoothing operator, normally a Gaussian kernel. In figure 1, a stylized example of averaging a 2D tensor field is shown. The ellipses are a visualization of the direction of the eigenvectors, scaled with corresponding eigenvalues. Knutsson uses this representation for local structure [Knutsson, 1989]. A thin ellipse, implying only one large eigenvalue, represents a local neighbourhood which is almost 1-dimensional. The tensors in 1, are all aligned in almost the same direction after the smoothing operation. The more inconsistent the orientation field is the “thicker” the averaged tensors
will be. Averaging a field with random orientations would produce a field with only almost “round” tensors where the radius is a measure of local energy, see figure 3.

**Fig. 1.** A stylized example describing what the averaging of a tensor field means. Left: input field. Right: output field.

**Fig. 2.** Averaging example where the field has a magnitude gradient from the left to the right. Left: input field. Right: output field.

Another example of averaging can be found in figure 2. Here a neighbourhood with an energy gradient is averaged. As mentioned above, the tensor components are averaged separately. This requires one scalar convolution over each “component field”. An averaging tensor filter has the following form:

\[
F^{ij}_{kl}(x) = \delta^i_k \delta^j_l f(||x||) \quad (4)
\]

where:

\[
\delta^k_l = \begin{cases} 
1 & \text{if } k = l \\
0 & \text{if } k \neq l 
\end{cases} \quad (5)
\]

and \(f(||x||)\) is a scalar smoothing function.

Most of the components in a filter like this are zero. Consider e.g. the filter for averaging the 2-D orientation tensor field. Only 4 of the 16 components in a
filter coefficient are $\neq 0$. The final convolution can now be expressed by

$$S_{kl}(x) = \sum_{p} \delta^i_k \delta^j_l f(||p||) \, T_{ij}(x - p)$$  \hfill (6)

where $f(||p||)$ is a scalar smoothing function.

### 3.4 Circular Symmetries

Vector operators finding circular symmetries was introduced by Knutsson and Granlund in [Knutsson and Granlund, 1986] and later further theoretically investigated by Bigün in [Bigün, 1988]. Examples of such symmetries are circles, Archimedes' spirals and stars, see figure 4.

The use of a tensor representation ensures that the certainty of statement the filtering produces (relation between the eigenvalues) and the local energy (the norm of the tensor) is kept separate. The “roundness” of the orientation tensors, i.e. the isotropic energy, is passed on to the rotation tensor field as a separate entity. This is done automatically in the filtering process.

The filtering procedure described below is an example of a second level operation where the input to the algorithm is a tensor field describing local orientation producing an output field with the same dimension and order. The interpretation of the tensors at this level should, however, be different; a continuous representation of circular symmetries, see figure 6.

**Detection of Local Circular Symmetries** The filter $F$ used for detecting local circular symmetries is defined as in the following:

$$F^{ij}_{11}(x) = u^i(x)u^j(x)h(||x||)$$ \hfill (7)

$$F^{ij}_{12}(x) = u^i(x)w^j(x)h(||x||)$$ \hfill (8)

$$F^{ij}_{21}(x) = w^i(x)u^j(x)h(||x||)$$ \hfill (9)

$$F^{ij}_{22}(x) = w^i(x)w^j(x)h(||x||)$$ \hfill (10)
where:
The rotating vector fields $\mathbf{u}(\mathbf{x})$ and $\mathbf{w}(\mathbf{x})$ describe the class of symmetries the filters are designed for.

$$h(r) = \begin{cases} r \cos^2(\alpha r) & \text{if } \alpha r \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases}$$ (11)

is the radial weighting function

The tensor filter can be viewed as a matrix of matrices filters.

$$S_{kl}(\mathbf{x}) = \sum_{p} F_{kl}(\mathbf{p}) T_{ij}(\mathbf{x} - \mathbf{p})$$ (12)

Each tensor element in the out-field has its “own” matrix filter. For example, consider the first component of the symmetry tensor $S_{11}$. The part of the filter $\mathbf{F}$ used for this component is $F_{11}^{ij}(\mathbf{x}) = u^i(\mathbf{x})u^j(\mathbf{x})$, which is a tensor filter in itself defined as the outer product of the vector field $\mathbf{u}(\mathbf{x})$ in figure 5.

$$S_{11}(\mathbf{x}) = \sum_{p} F_{11}^{ij}(\mathbf{p}) T_{ij}(\mathbf{x} - \mathbf{p}) = \sum_{p} u^i u^j(\mathbf{p}) T_{ij}(\mathbf{x} - \mathbf{p})$$ (13)
Fig. 6. The largest eigenvector indicates the type of local symmetry estimated. The relation between the eigenvalues expresses the certainty of the statement. The larger $\lambda_1$ is compared to $\lambda_2$ the higher the certainty ($\lambda_1 \geq \lambda_2$).

We will proceed by investigating what this operation means. For simplicity we disregard the looping, $p$, over the filter coefficients and consider only one operation:

$$u^i u^j T_{ij}$$

In vector algebra, this expression can be interpreted as:

$$u^T u = (u^1 u^2 \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix}) = T_{11} u^1 u^1 + 2T_{12} u^1 u^2 + T_{22} u^2 u^2$$

or equivalent

$$\text{trace}[uu^T] = \text{trace} \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix}$$

$$= T_{11} u^1 u^1 + 2T_{12} u^1 u^2 + T_{22} u^2 u^2$$

The third interpretation illuminates similarities with the scalar product of two vectors. We define this as the tensor scalar product between $uu^T$ and $T$:

$$\begin{pmatrix} T_{11} \\ T_{12} \\ T_{12} \\ T_{22} \end{pmatrix} (u^1 u^1 u^2 u^2 u^1 u^2 u^2 u^2) =$$

$$= T_{11} u^1 u^1 + 2T_{12} u^1 u^2 + T_{22} u^2 u^2$$
Fig. 7. Different levels in the tensor feature hierarchy. Note that when the model on the next higher level, do not have support from data this yields “round” tensors. Thus, events not supporting the model are passed on to the next higher level in the norm of the tensors. It is the shape of the tensor indicates the validity of the statement.

Summing this expression over the filter gives the first element in the tensor describing circular symmetry. The other elements are derived similarly.

It turns out that only the following five scalar convolutions are needed for estimating the circular symmetry tensor $S$:

1. $r_1 = \sum_p T_{11}(x - p) + T_{22}(x - p)$
2. $r_2 = \sum_p \left( T_{11}(x - p) - T_{22}(x - p) \right) \cos \left( 2\varphi(p) \right)$
3. $r_3 = \sum_p \left( T_{11}(x - p) - T_{22}(x - p) \right) \sin \left( 2\varphi(p) \right)$
4. $r_4 = \sum_p T_{12}(x - p) \cos \left( 2\varphi(p) \right)$
5. $r_5 = \sum_p T_{12}(x - p) \sin \left( 2\varphi(p) \right)$
The components of $S$ is calculated by:

$$S_{ij} = \begin{pmatrix} r_1 + r_2 + 2r_5 & 2r_4 - r_3 \\ 2r_4 - r_3 & r_1 - r_2 - 2r_5 \end{pmatrix}$$  \quad (18)

The eigenvector corresponding to the largest eigenvalue points out the estimated symmetry, see figure 6. The orientation of this vector can be expressed by:

$$2\varphi = \arg(r_2 + 2r_5, 2r_4 - r_3)$$  \quad (19)

**Experimental Results** The test image used contains all different patterns the operator is designed for, i.e the whole class of circular symmetries.

**References**


Fig. 8. Circularly symmetric neighbourhoods. The patterns vary continuously from stars to circles via left and right rotations. The test image used originally appeared in [Bigün, 1988].
Fig. 9. Certainty of estimated symmetries. It is defined as the difference between the largest and the smallest eigenvalue of the symmetry tensor, $\lambda_1 - \lambda_2$ ($\lambda_1 \geq \lambda_2$).
Fig. 10. The arrows correspond to the orientation of the largest eigenvector of circular symmetry tensor expressed in double angle. The length of the vectors are $\lambda_1 - \lambda_2$. 