Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint Nr. 134

A TV Flow Based Local Scale Estimate and its Application to Texture Discrimination

Thomas Brox and Joachim Weickert

Saarbrücken 2005

Fachrichtung 6.1 – Mathematik
Universität des Saarlandes

Preprint No. 134
submitted: April 04, 2005

A TV Flow Based Local Scale Estimate and its Application to Texture Discrimination

Thomas Brox

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Building 27.1,
66041 Saarbrücken, Germany
brox@mia.uni-saarland.de

Joachim Weickert

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Saarland University, Building 27.1,
66041 Saarbrücken, Germany
weickert@mia.uni-saarland.de

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-Mail: preprint@math.uni-sb.de
WWW: http://www.math.uni-sb.de/

Abstract

This paper presents a local region based scale measure, which exploits properties of a certain type of nonlinear diffusion, the so-called total variation (TV) flow. During the signal evolution by means of TV flow, pixels change their value with a speed that is inversely proportional to the size of the region they belong to. From this evolution speed one can derive a local scale estimate based on regions instead of derivative filters. Main motivation for such a scale measure is its application to texture discrimination, in particular the construction of an alternative to Gabor filters. When the scale estimate is combined with the components of the structure tensor, which provides orientation information, it yields a texture feature space of only four dimensions. Like Gabor features, this sparse feature space discriminates textures by means of their orientation and scale, yet the representation of orientation and scale is less redundant. The quality of the feature space containing the new scale measure is evaluated in texture segmentation experiments by comparing results to those achieved with Gabor filters. It turns out that one can gain a total speedup of factor 2 without loosing any quality concerning the discrimination of textures.

Key Words: scale, texture, nonlinear diffusion, segmentation.

Contents

1 Introduction

Texture plays an important role for the analysis of real-world scenes, particularly for the segmentation of images. In the work of Julesz [20, 21], texture has been defined as the repetition of basic image elements, so-called *textons*. From the theoretical point of view, textons are quite attractive; see a recent work in [43].
The arising question in practice is how to extract textons from the image. Assuming that a texture can be represented by a linear combination of basis functions, one can measure at every position how much each basis function correlates with the image, i.e., one convolves the image with the basis functions and represents the texture by means of the correlation coefficients. These coefficients are supposed to contain enough features in order to regenerate or to discriminate textures.
A very popular set of basis functions is described by a so-called *Gabor filter bank* [15]. Each basis function in a Gabor filter bank has one preferred orientation and scale. Thus convolution of the image with each filter yields high responses at those positions in the image where the local structure fits the orientation and scale of the filter. Gabor filters have attained high popularity, in particular because neuroscience has found a similar behavior of neuronal cells in the primary visual cortex of primates. Since then, a lot of research aimed on determining a model that best fits the responses measured in the visual cortex [26, 10, 13]. Moreover, several works on texture segmentation based on Gabor filters are available in the literature, e.g. [5, 11, 34, 29].
Basically a Gabor filter bank extracts nothing else than the magnitude, orientation, and scale of local texture elements. The success of Gabor filters in the context of texture segmentation indicates the importance of these basic features for discriminating textures. However, one may wonder if it is necessary to describe these features by means of the highly redundant coefficient vector provided by a Gabor filter bank. Actually it should be possible to reduce the feature vector to exactly these three components.
This is the motivation of the present paper. Instead of extracting the correlation coefficients of the texture, it is proposed to directly estimate the magnitude, orientation, and scale of the texture elements. For extracting the magnitude and the orientation, one can employ the so-called structure tensor [14], which is a popular tool for orientation estimation. It has been utilized for texture discrimination already in [31] and [4].
Also for extracting the scale of texture elements we could employ methods from the literature [19, 23, 24, 12, 35, 18]. However, all these works have one thing in common: they are all gradient based, i.e. their measure of local scale depends directly on the local gradient or its derivatives. Consequently, the scale cannot be measured in regions without a significant gradient. In order to overcome this problem, the scale is often measured within local windows. Consequently, estimates become inaccurate near texture boundaries.
In this paper we suggest a completely different strategy to measure local scale. In contrast to existing approaches, this method is not based on edges but on regions. This means, our local scale measure does not depend on the behavior of the gradient in scale space, but directly on the size of regions. Since almost all state-of-the-art segmentation techniques are based on regions, a local scale measure that directly employs the size of regions appears to be a more appropriate solution for this kind of application than an edge based measure. A region based technique does not need the definition of any window, and therefore yields the maximum localization accuracy. The work probably most related to this idea is the texture segmentation framework in [16].
With a region based technique, however, we face the problem to define regions and to measure

their size. For this purpose we exploit special properties of a nonlinear diffusion filter called total variation (TV) flow [2]. It is the diffusion counterpart to TV regularization [33, 1] and is known to yield segmentation-like solutions. In [9, 36] it has been shown for the space-discrete one-dimensional setting that TV flow evolves the signal according to a certain set of rules. Among others, the evolution speed of regions is inversely proportional to their size. By means of this rule we can efficiently determine the size, i.e. the scale, of a region by the evolution speed of its pixels.
This article comprises and extends earlier work of the authors presented at a conference [7]. In particular, the extensions contain: (i) a way how to avoid the stopping parameter present in the scale measure presented in [7], (ii) an alternative local scale measure based on a region-merging algorithm, and (iii) an extensive comparison of the derived sparse texture features to Gabor filters.

Paper organization. In the following section, we introduce our new region based scale measure. We review the analytic behavior of TV flow in the one-dimensional setting and show how to derive a local scale estimate from these findings. For comparison, we estimate the local scale in a similar way by means of a region-merging algorithm. In Section 3 we briefly discuss orientation estimation by means of the structure tensor. In Section 4 then both the scale measure and the structure tensor components are combined in order to yield a sparse texture feature space. After a brief review of Gabor filters, Section 5 compares the performance of the new features to that of Gabor features in the scope of texture segmentation. The paper is concluded by a brief summary in Section 6.

2 A Region Based Local Scale Estimate

2.1 Total Variation Flow

The local scale estimate is based on the nonlinear diffusion technique called *TV flow* [2]. This diffusion method has the tendency to yield segmentation-like results, so it implicitly provides the regions needed for measuring the local scale. Starting with an initial image $I(x, y)$, the denoised and simplified version $u(x, y, t)$ of the image evolves under progress of artificial time t according to the partial differential equation

$$\partial_t u = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) \tag{1}$$

with initial condition $u(t = 0) = I$ and with reflecting boundary conditions. In the one-dimensional setting it has been shown that the evolution of u successively leads to larger regions, inside which all pixels have the same value [9, 36]. However, the representation of these regions is not explicit, so it is not straightforward how to determine the region sizes.
For this reason, we exploit another useful property of TV flow besides its tendency to yield segmentation-like results: linear contrast reduction. This allows an efficient computation of the region sizes, without the explicit representation of regions, since the size of a region can be derived by means of the evolution speed of its pixels.

In 1-D, space-discrete TV flow (and TV regularization) have been proven to comply with the following rules [9, 36]:

(i) A region of m neighboring pixels with the same value can be considered as one superpixel with mass m.

(ii) The evolution splits into merging events where pixels melt together to larger pixels.

(iii) Extremum pixels adapt their value to that of their neighbors with speed $\frac{2}{m}$.

(iv) The two boundary pixels adapt their value with half that speed (due to reflecting boundary conditions they can as well be considered to be twice as large).

(v) All other pixels do not change their value.

In higher dimensions, these rules are no longer exactly satisfied. However, one can expect TV flow to evolve the signal in a very similar manner. Thus, although it is only possible to derive an exact scale measure for 1-D signals, the technique provides good scale estimates also in higher dimensions, where TV flow behaves only approximately according to the above listed rules.

2.2 Scale Measure Based on TV Flow

In 1-D, the analytic behavior of TV flow leads to the very useful consequence that by simply sitting upon a pixel and measuring the speed with which it changes its value, it is possible to determine its current local scale. As pixels belonging to small regions move faster than pixels belonging to large regions (iii), the rate of change of a pixel determines the size of the region it currently belongs to. Integrating this rate of change over the evolution time and normalizing it with the evolution time T yields the average speed of the pixel, i.e. its average inverse scale in scale space:

$$\frac{1}{\bar{m}} = \frac{1}{2} \frac{\int_0^T |\partial_t u| \, dt}{T}. \tag{2}$$

This formula may give rise to the question why to measure the *average* speed of the pixels and thereby introducing the integration time T as an unpleasant parameter of the method. Why not just measuring the speed of the pixels at time $t = 0$?
The main reason why the pixel speed at time $t = 0$ is not well suited to estimate the scale of texture elements is the fact that in a real-world image it rarely happens that two pixels have exactly the same value. This means, at $t = 0$ almost all regions are of size 1. Therefore, it is necessary to simplify the image, i.e. to create regions, before it makes sense to measure their size.
The second reason why it is reasonable to compute the average scale over some time is the fact that only extremum regions change their value according to (iii) - (v). This means, while a pixel is not part of an extremum region, it is not possible to derive its scale. By integrating all time instants up to a time T, one increases considerably the chance that all pixels have been part of an extremum region. Respecting for normalization also the time at which a pixel does not move, leads to the following formula:

$$\frac{1}{\bar{m}} = \frac{1}{2} \frac{\int_0^T |\partial_t u| \, dt}{T - \int_0^T \delta_{\partial_t u, 0} \, dt} \tag{3}$$

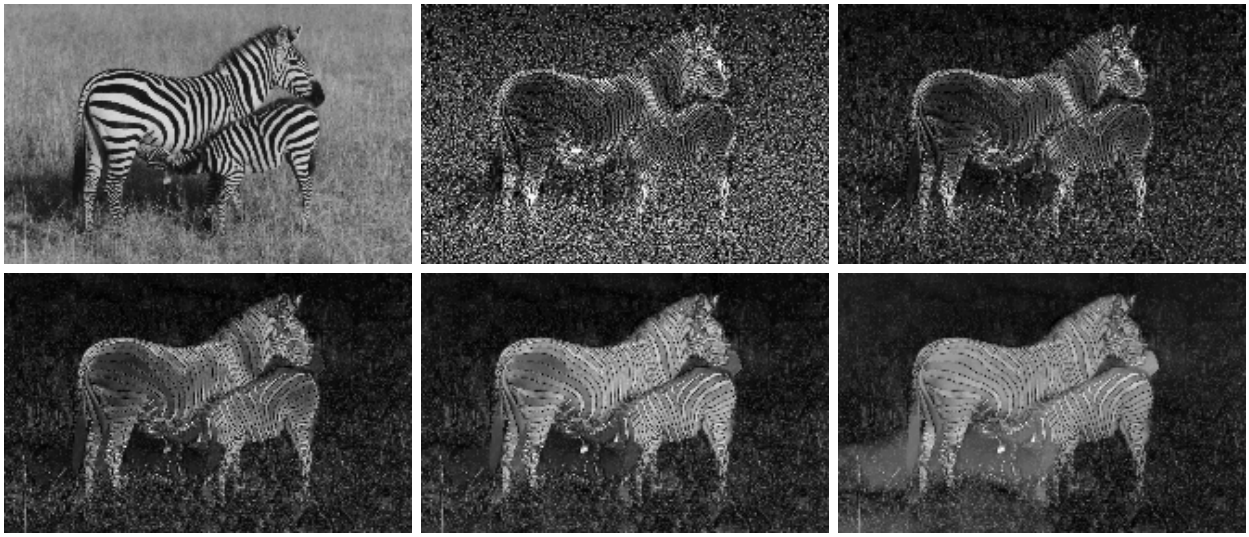


Figure 1: Local scale estimate $\frac{1}{\bar{m}}$ for a zebra image. The measure yields the average inverse scale, so dark regions correspond to large scales, bright regions to fine scales. Values have been normalized for better visibility. From Left to Right, Top to Bottom: (a) Input image. (b) $T = 5$. (c) $T = 20$. (d) $T = 50$. (e) $T = 100$. (f) $T = 500$.

where $\delta_{a,b} = 1$ if $a = b$, and 0 otherwise. In the rare case that a pixel has never been part of an extremum region, the result is set to 0. Alternatively, one could interpolate between the values of neighboring regions. Besides this shortcoming the measure in (3) yields exactly the regions' sizes.

As already mentioned before, the rules for TV flow, from which the scale measure in (3) is derived, do not strictly carry over to the higher-dimensional cases. However, we assume that the scale measure applied to two-dimensional data still yields very good approximations, in fact approximations that compare favorably to alternative scale estimation techniques. This assumption is empirically justified by the experiments shown in this paper; see e.g. Fig. 1.

Thus in 2-D, (3) is only slightly changed by introducing the pre-factor $\frac{1}{4}$ due to the 4 neighbors of a pixel in 2-D:

$$\frac{1}{\bar{m}} = \frac{1}{4} \frac{\int_0^T |\partial_t u| \, dt}{T - \int_0^T \delta_{\partial_t u, 0} \, dt}. \tag{4}$$

Fig. 1 depicts the local scale measure for different stopping times T. At first glance it appears a bit counterproductive to have a scale parameter in a scale measure. However, the scale involved in the choice of T and the local scale measured in the image are of different nature. While T *globally* defines the range of expected scales of texture elements, the estimated scale of different texture elements varies *locally*. While T is determined by the image size and by the predefinition which image structures are still considered as texture, the purpose of the local scale measure is to distinguish regions of different scale.

For texture discrimination one is in general interested in small structures. Thus it is usually advantageous to choose T not too large. For instance, one can see that in Fig. 1e,f the pixels of the stripes of the zebras have been part of a large zebra region for such a long time, that one can no longer distinguish their different scales. In Fig. 1f, the most dominant structure besides the stripes is already the shadow of the zebra, though its scale is very large and should actually not be considered as texture anymore. On the other hand, Fig. 1b shows what happens if T is chosen very small. In such a case mainly the image noise contributes to the scale

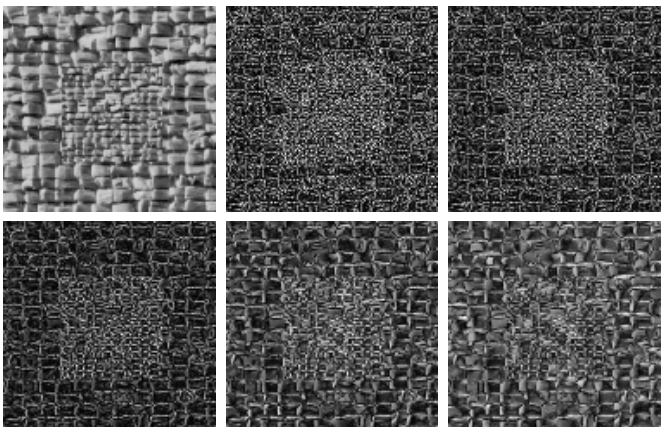


Figure 2: Average inverse scale for a synthetic texture image. From Left to Right, Top to Bottom: (a) Input image. (b) $T = 1$. (c) $T = 5$. (d) $T = 20$. (e) $T = 50$. (f) $T = 100$.

estimate. Setting T to values that correspond to the expected maximum of texture scales, however, results in good scale estimates, as shown in Fig. 1c,d. The same value of T yields also nice results for other images, see Fig. 2. For texture discrimination, T can in general be set to a fixed value such as $T = 20$.

Avoiding the stopping parameter. Instead of setting T fixed, one may be interested in removing the parameter completely from the scale measure. Indeed this is possible when exploiting a further property of TV flow that even carries over to the 2-D case: TV flow flattens the image within finite time $T_{\max}$ [3]. This means, with a finite diffusion time it is possible to consider all possible scales. Thus, only the fact that large scales obtain too much weight in the average scale, keeps us from choosing $T_{\max}$ as the stopping time for the scale measure.

The reason for the lower weight of smaller scales is the fact that small regions move faster than large regions and hence have a shorter period of existence in scale space before they are merged to neighboring regions. This bias towards larger scales can be balanced by introducing a weight $w(t)$ in (4):

$$\frac{1}{\tilde{\bar{m}}} = \frac{1}{4} \frac{\int_0^{T_{\max}} w(t) |\partial_t u| \, dt}{\int_0^{T_{\max}} w(t)(1 - \delta_{\partial_t u,0}) \, dt}. \tag{5}$$

The decisive question is how this weight has to be chosen in order to balance the bias towards larger scales. In order to answer this question we go back to the simpler 1-D case and consider two extremum regions with the same contrast c to its neighbors and masses m_1 and $m_2 = a \cdot m_1$ with $a > 1$. Due to the linear contrast reduction, region m_1 disappears after time $t_1 = \frac{c \cdot m_1}{2}$ while region m_2 only disappears after time $t_2 = \frac{c \cdot m_2}{2} = \frac{c \cdot a \cdot m_1}{2}$. In other words, regions that are twice as large survive twice as long. This indicates the weight we have to choose to balance this behavior:

$$w(t) = \frac{1}{m} = \frac{1}{2} |\partial_t u|. \tag{6}$$

As before, we assume that the behavior of regions approximately carries over to the 2-D case. This yields the following scale measure without a parameter:

$$\frac{1}{\tilde{\bar{m}}} = \frac{1}{4} \frac{\int_0^{T_{\max}} |\partial_t u|^2 \, dt}{\int_0^{T_{\max}} |\partial_t u| \, dt}. \tag{7}$$

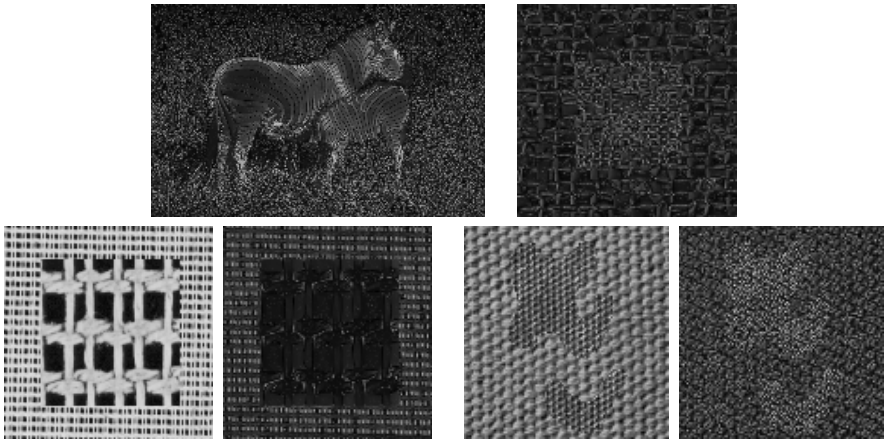


Figure 3: Local scale estimate $\frac{1}{\tilde{\bar{m}}}$ without a stopping parameter. The measure yields the average inverse scale, so dark regions correspond to large scales, bright regions to fine scales. Top Row: Scale estimate for the images from Fig. 1 and Fig. 2. Bottom Row: Input image and scale estimate for two further images.

Fig. 3 depicts the scale estimates with this method that is free of a stopping parameter. Although one gives away the possibility to define a priori a range of simplification levels at which typical texture elements appear as regions, the results are still relatively good. The scale measure without the stopping parameter can therefore be seen as an alternative to the previous setting with $T = 20$ fixed.

Implementation aspects. Numerical implementation of TV flow can cause stability problems when the gradient tends to zero. In order to avoid this, the process is often stabilized artificially by adding a small positive constant ϵ to the gradient (e.g. $\epsilon = 0.001$):

$$\partial_t u = \operatorname{div}\left(\frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \epsilon^2}}\right) \tag{8}$$

The stability condition for the time step size τ of an explicit scheme is $\tau \leq 0.25\epsilon$, so for small ϵ, many iterations are necessary. A much more efficient approach is to use a semi-implicit AOS scheme [42, 37], which is unconditionally stable, so it is possible to choose $\tau = 1$. Let u^k denote u after iteration k. Then the discrete versions of the scale measures for arbitrary τ read:

$$\begin{aligned}
u^0 &= I \\
u^{k+1} &= \frac{1}{2}\Big(\big(E - 2\tau A_x(u^k)\big)^{-1} + \big(E - 2\tau A_y(u^k)\big)^{-1} \Big) u^k \\
\frac{1}{\bar{m}} &= \frac{1}{4\tau} \frac{\sum_{k=0}^{T} |u^{k+1} - u^k|}{T - \sum_{k=0}^{T} \delta_{|u^{k+1}-u^k|,0}} \qquad (9) \\
\frac{1}{\tilde{\bar{m}}} &= \frac{1}{4\tau} \frac{\sum_{k=0}^{T_{\max}} |u^{k+1} - u^k|^2}{\sum_{k=0}^{T_{\max}} |u^{k+1} - u^k|} \qquad (10)
\end{aligned}$$

where $|\cdot|$ is the Euclidean norm and E denotes the unit matrix. A_x and A_y are the diffusion matrices in x and y direction; for details see [42]. The diffusion time $T_{\max}$ can be determined by checking when $\sum_i |u_i^{k+1} - u_i^k|$ becomes 0 or almost 0.

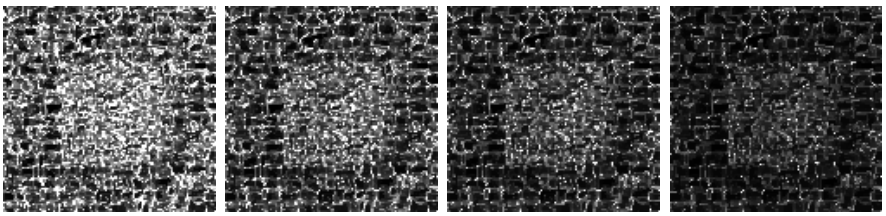


Figure 4: Average inverse scale obtained with the region-merging algorithm for the synthetic texture image in Fig. 2. From Left to Right: (a) $\alpha_{Stop} = 100$. (b) $\alpha_{Stop} = 250$. (c) $\alpha_{Stop} = 500$. (d) $\alpha_{Stop} = 1000$.

2.3 Comparison to an Alternative Scale Measure Based on Region Merging

One may wonder whether TV flow is the only image simplification method that can be used for deriving region based scale estimates. Indeed there is another simplification method that even yields an explicit representation of regions including their size: the region merging algorithm. We derive local scale estimates based on this algorithm to show an alternative region based technique.

In [22], region merging has been motivated as a technique to minimize the cartoon limit of the Mumford-Shah functional [28]

$$E(u, \Gamma) = \int_{\Omega} (I - u)^2 \, \mathbf{dx} + \alpha \int_{\Gamma} ds. \tag{11}$$

This energy functional searches for a piecewise constant solution u with the edge set Γ such that u is close to the image I and the total edge length $\int_{\Gamma} ds$ is minimal. The parameter $\alpha > 0$ determines to which extend region boundaries are penalized, so larger α yield results with less regions.

Global optimization of this functional is very difficult. The region merging algorithm therefore uses a greedy heuristic starting with the trivial segmentation where each pixel is a region, and then successively merges those two neighboring regions that lead to the largest energy decrease. This merging of regions yields successively simplified images with increasingly large regions. The size of these regions is the sought local scale $m(\alpha)$. Similarly to the stopping time of TV flow there appears a scale parameter that defines how much two regions may differ in their value to be still considered as one region.

As in the case of TV flow, the average inverse scale over some range of different values of α yields better results than choosing a single α:

$$\frac{1}{\bar{m}} = \frac{\int_0^{\alpha_{Stop}} \frac{1}{m(\alpha)} \, d\alpha}{\alpha_{Stop}}. \tag{12}$$

In [27] a finite α for merging all regions is shown. Furthermore, it is shown that the upper bound for the number of regions increases proportionally with α^2. From these properties one can derive a scale measure without a parameter in a similar way as in the case of TV flow. However, since the outcome is considerably worse than in the case of TV flow, this possibility is not further discussed in this paper.

Fig. 4 and Fig. 5 show the average inverse scale for two of the test images. One can see that although the very small scale texture can be distinguished from the rest of the image, the

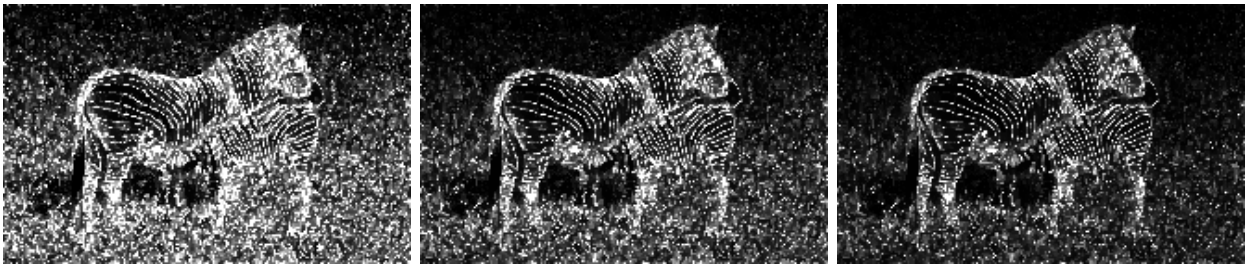


Figure 5: Average inverse scale obtained with the region-merging algorithm for the zebra image in Fig. 1. From Left to Right: (a) $\alpha_{Stop} = 100$. (b) $\alpha_{Stop} = 250$. (c) $\alpha_{Stop} = 500$.

results are quite noisy and the quality of the scale estimates are considerably worse than those obtained with the TV flow based method. This is mainly due to the tendency of the region-merging algorithm to keep single pixels even for quite large α if their value is considerably different from its neighbors. This shows that not only the concept of a region based local scale measure is important, but it is also decisive to choose an appropriate image simplification method.

3 Orientation Estimation with the Structure Tensor

As mentioned in the introduction, the local scale measure is employed to define a sparse alternative to a Gabor texture feature space. Apart from the scale of texture elements, also their orientation and magnitude are important features to discriminate textures. A popular tool for orientation estimation is the structure tensor, also known as second moment matrix [14]. It has already been applied to texture analysis in [31, 4, 25].
The structure tensor integrates the outer product of the image gradient

$$J_0 = (\nabla I \nabla I^\top) = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{13}$$

within a local neighborhood, which is in the classic case a Gaussian window K_ρ with standard deviation ρ:

$$J_\rho = K_\rho * J_0. \tag{14}$$

The result is a symmetric positive semi-definite matrix, from which the dominant orientation in the neighborhood can be extracted by computing the eigenvector to the largest eigenvalue. Additionally to the dominant orientation, the structure tensor contains the magnitude of the structure and the homogeneity of orientation. The magnitude of the structure can be extracted as the trace of the structure tensor, whereas the homogeneity of orientation can be made explicit by the ratio of its two eigenvalues. However, the three features are in general represented just by the three components of the structure tensor without making the separate features explicit.
The non-adaptive Gaussian neighborhood of the classic structure tensor has recently been replaced by adaptive neighborhoods obtained by means of nonlinear diffusion [41, 8]. This concept is called *nonlinear structure tensor* and has successfully been applied to texture segmentation in [32]. An alternative adaptation strategy is to apply robust statistics [40]. The advantage of data-adaptivity is the avoidance of blurring effects near texture boundaries. At such positions, the non-adaptive Gaussian window integrates ambiguous data from two regions reducing the accuracy of the estimation.

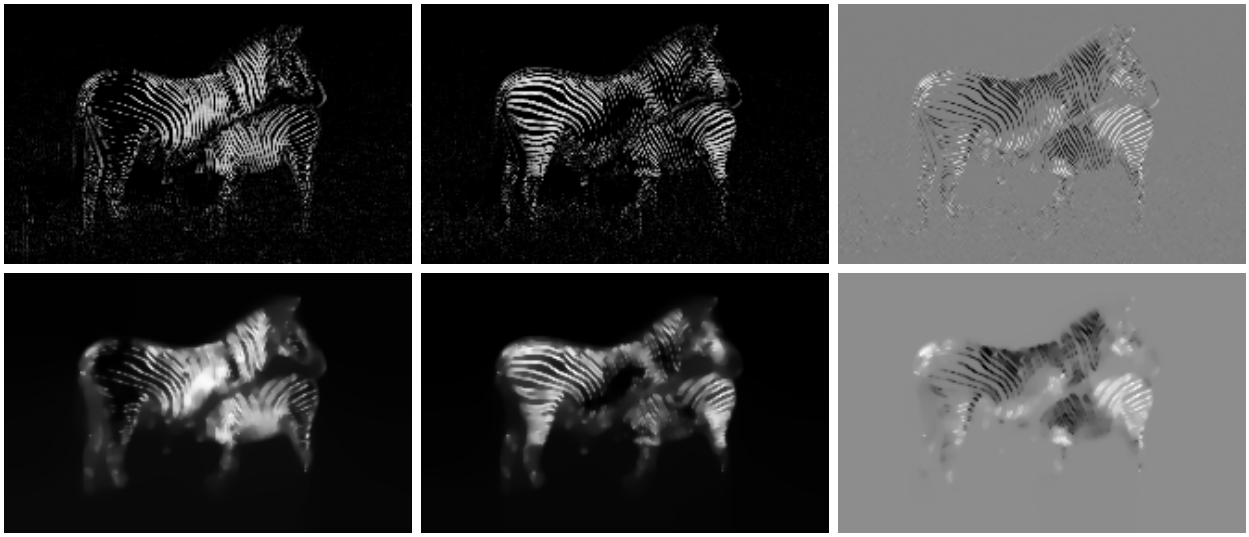


Figure 6: TOP ROW: Components of J_0: $I_x^2, I_y^2, I_x I_y$. BOTTOM ROW: Components of the nonlinear structure tensor J_t with $t = 5000$.

A nonlinear structure tensor J_t is obtained by using J_0 as initial condition in the partial differential equation

$$\partial_t J_{ij} = \operatorname{div}\left(g\left(\sum_{k,l=1}^{2} |\nabla J_{kl}|^2\right)\nabla J_{ij}\right) \qquad i,j = 1,2 \tag{15}$$

which describes matrix-valued nonlinear diffusion with a coupling of all matrix channels [38]. The diffusion time t is a scale parameter that corresponds to the standard deviation of the Gaussian kernel used for the Gaussian neighborhood. The diffusivity g steers the reduction of smoothing in the presence of discontinuities. A very well suited diffusivity function for smoothing the structure tensor leads to TV flow that has already appeared in the last section on scale estimation:

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|}. \tag{16}$$

Fig. 6 depicts the components of J_0 and those of the nonlinear structure tensor J_t with $t = 5000$ for the zebra image introduced in Fig. 1. It can be observed that despite the smoothing, the localization accuracy near texture boundaries is still high.

4 A Sparse Set of Texture Features

We are now in the position to combine the components of the nonlinear structure tensor (J_{11}, J_{22}, J_{12}) at time t and the TV flow based scale measure $\frac{1}{\bar{m}}$. This yields a four-dimensional texture feature vector

$$\mathbf{F} := \left(J_{11}, J_{22}, 2J_{12}, \frac{1}{\bar{m}}\right) \tag{17}$$

which contains the texture magnitude, orientation, scale, and homogeneity of orientation. In comparison to the responses of a Gabor filter bank, the three typical features modelled by Gabor filters are represented sparsely by three feature channels. In addition, $\mathbf{F}$ models also the spatial homogeneity of orientation which arises from the smoothing involved in the structure tensor.

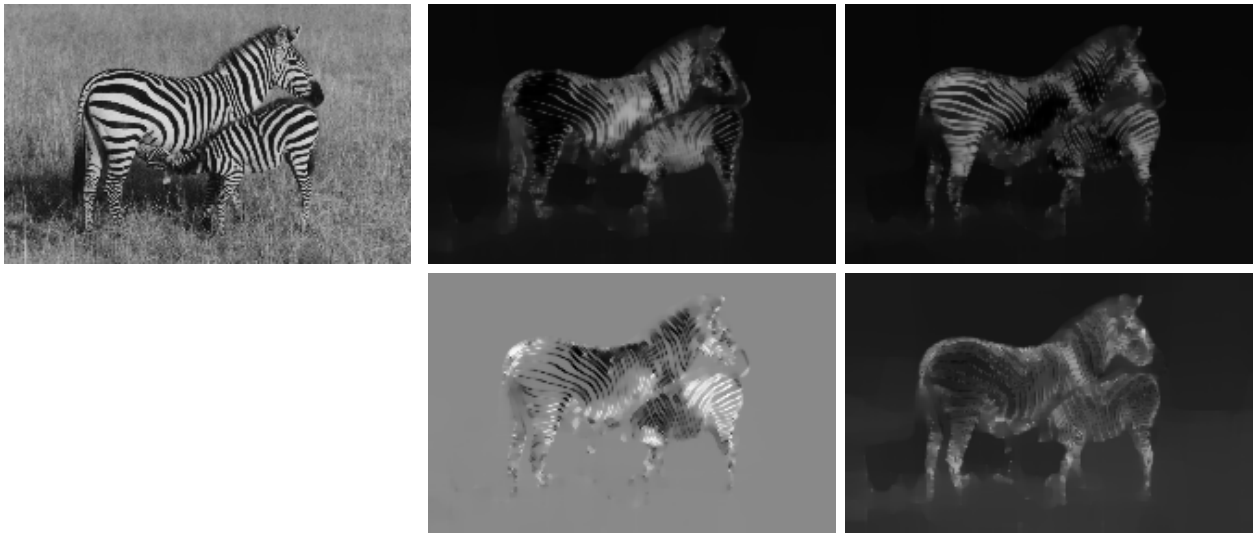


Figure 7: Sparse texture feature space $\mathbf{F}(t)$ with $t = 500$. Top Left: F_0. Top Right: F_1. Bottom Left: F_2. Bottom Right: F_3 (scale).

One may wonder, if it would make sense to apply the coupled nonlinear diffusion process to the structure tensor components *after* the scale measure (and perhaps further features) have been added. Since also the scale measure contains important discontinuities, this could better indicate texture boundaries and avoid smoothing across them. In fact this concept to assemble a feature vector of unsmoothed features and to smooth them with some discontinuity preserving method establishes strong relations to the work in [34] where Beltrami flow is used to smooth the responses of Gabor filters.
In order to ensure a fair coupling in nonlinear diffusion, it is necessary that all feature channels have approximately the same dynamic range. However, the dynamic range of the scale estimates is $[0, 1]$ while the range of the structure tensor components is proportional to the square of the dynamic range of the discrete image; the exact factor depends on the discretization of the derivatives. Thus before the nonlinear diffusion process can be applied, all features have to be normalized to a common dynamic range.
With regard to the possibility to add further features, in particular the gray value or color image itself, we normalize all features to the potential range of the image, which is usually $[0, 255]$. Therefore, J_0 is replaced by

$$\tilde{J}_0 := \frac{J_0}{|\nabla I|}. \tag{18}$$

Multiplying the components of $\tilde{J}_0$ with the factor that is determined by the discretization of the derivatives - for central differences and grid size $h = 1$ this factor is 2 - ensures a potential range of the structure tensor components equivalent to that of the image gray value. The scale measure simply has to be multiplied by the range of the image gray value to achieve the same effect.
This normalized feature vector serves as initial condition $\mathbf{F}(0) = \left(\tilde{J}_{0,11}, \tilde{J}_{0,22}, 2\tilde{J}_{0,12}, \frac{1}{\bar{m}}\right)$ for the partial differential equation [17]

$$\partial_t F_i = \operatorname{div}\left(g\left(\sum_{j=1}^{4} |\nabla F_j|^2\right)\nabla F_i\right) \quad i = 1, ..., 4 \tag{19}$$

which performs coupled TV flow on the feature vector and yields after diffusion time t the

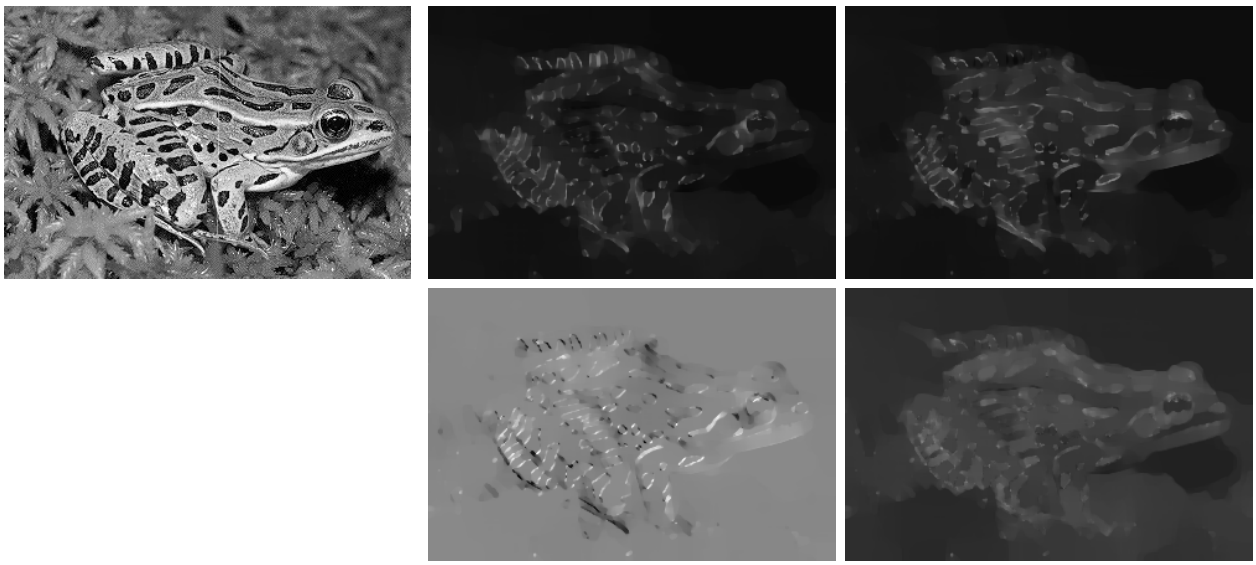


Figure 8: Sparse texture feature space $\mathbf{F}(t)$ with $t = 1000$. Top Left: F_0. Top Right: F_1. Bottom Left: F_2. Bottom Right: F_3 (scale).

smoothed feature vector $\mathbf{F}(t)$, which is depicted in Fig. 7 and Fig. 8 for two different input images.

5 Experiments

5.1 A Brief Review of Gabor Filters

In the following experiments the sparse feature space $\mathbf{F}(t)$ is compared to Gabor filters as the standard approach to texture discrimination. As a further alternative one may consider wavelet representations of an image [39]. The typical implementation of wavelets, however, is neither translation nor rotation invariant. Further on, one is restricted to a small number of discrete scales. Thus one can expect a lower discrimination quality with a number of features that is still significantly larger than with our proposed method. In our view, the question about the quality-efficiency tradeoff of our proposed feature space can be best answered by a comparison to Gabor filters.
In principle, the Gabor function used for Gabor filters is a Gaussian function modulated by an oriented sinusoidal with orientation ϕ and frequency f:

$$G_{\phi,f}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot e^{2\pi f i(x\cos\phi + y\sin\phi)}. \tag{20}$$

The standard deviation σ of the Gaussian is usually set dependent on the frequency f of the wave function. This is reasonable, since both parameters determine the scale of the local structure that should respond to the filter. The relation between σ and f is set to [30]:

$$\sigma = 3\frac{\sqrt{2\ln 2}}{2\pi f}. \tag{21}$$

We choose 4 different orientations $\phi \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ and 3 different scales $f \in \{0.2, 0.35, 0.5\}$ for the Gabor filter bank. Fig. 9 shows the corresponding filter stencils. Sometimes more

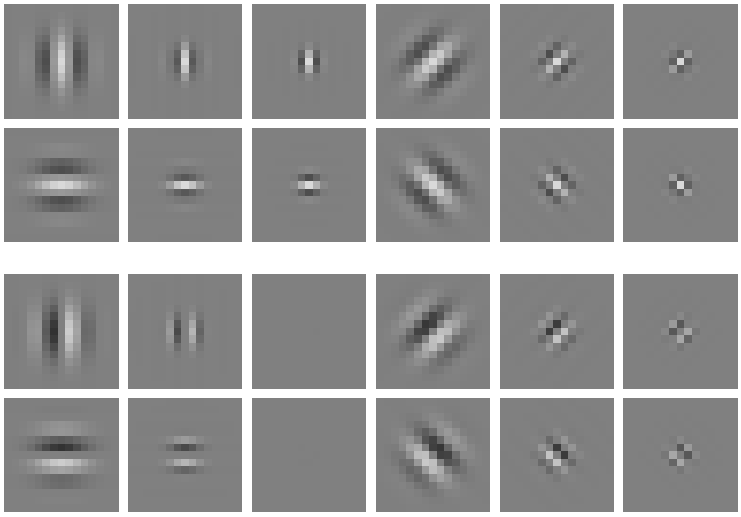


Figure 9: Gabor filter stencils with 4 different orientations $\phi \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ and 3 different frequencies $f \in \{0.2, 0.35, 0.5\}$. TOP: Real part. BOTTOM: Imaginary part.

filters are applied, but this increases not only the sampling accuracy but also the redundancy of the filter bank and the dimension of the feature space.
Instead of using directly the responses of the Gabor filters for texture discrimination, mostly the so-called Gabor energy

$$E_{\phi,f} = \sqrt{(I * \Re(G_{\phi,f}))^2 + (I * \Im(G_{\phi,f}))^2} \tag{22}$$

is used, where $\Re$ and $\Im$ denote the real and imaginary part, respectively. The Gabor energies are depicted in Fig. 10 for the zebra image. One can clearly see the redundancy of these features.

5.2 Discrimination of Separated Textures

One main purpose of a texture feature space is its application to texture segmentation. For segmentation, both good discrimination capabilities *and* a good localization accuracy are important. Before employing the sparse feature space in a segmentation task, however, we test the texture features in a much simpler experiment where only the discrimination capabilities play a role, as the textures are already separated from each other. Fig. 11 shows 8 different textures and the dissimilarities between these textures measured once with the Gabor feature space and once with the sparse representation.
A simple distance measure taking into account the means $\mu_k(T)$ and the standard deviations $\sigma_k(T)$ of each feature channel k of two textures $T \in \{T_1, T_2\}$ serves as dissimilarity measure for each feature channel:

$$\Delta_k = \left(\frac{\mu_k(T_1) - \mu_k(T_2)}{\sigma_k(T_1) + \sigma_k(T_2)} \right)^2. \tag{23}$$

For the total dissimilarity the average of all M texture channels is computed:

$$\Delta = \frac{1}{M} \sum_{k=1}^{M} \Delta_k. \tag{24}$$

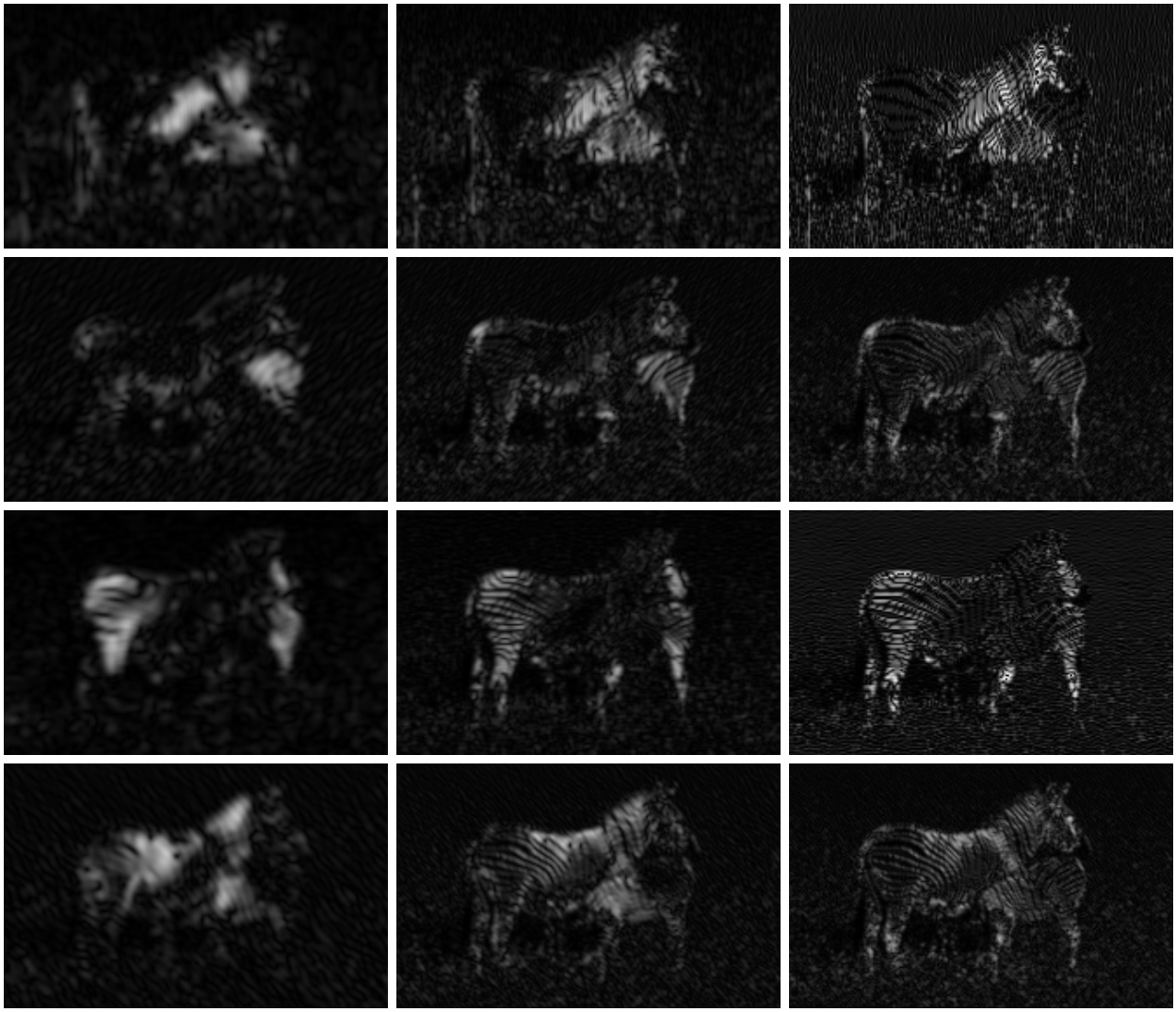
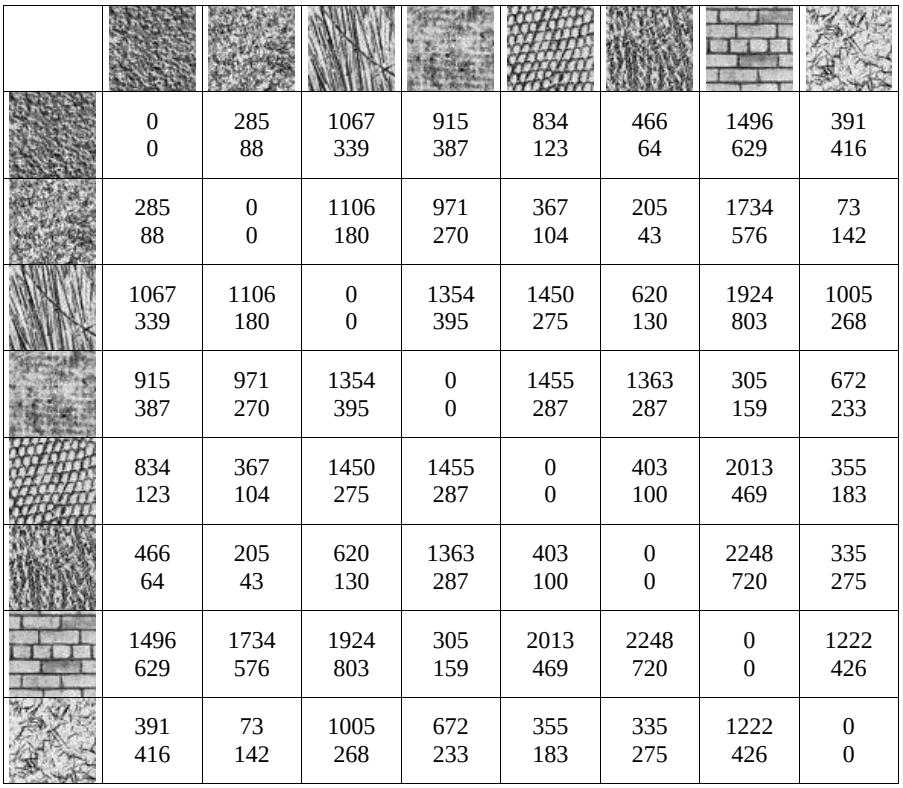


Figure 10: Gabor energy for the filter stencils shown in Fig. 9 and the zebra image in Fig. 7. Same ordering as in Fig. 9 from left to right, top to bottom.

Additionally to the 12, respectively 4, texture channels, also the gray value is added as supplementary feature. This increases the accordance of the results with the visual impression when two textures have different average gray values. The diffusion time t in $\mathbf{F}(t)$ has been 0. There is no smoothing necessary in this experiment, since the mean value already implies an infinite amount of smoothing. The stopping time of the scale measure has been set to $T = 20$ like in all the following experiments. We could have also employed the version without stopping parameter, yet it induces considerably more computational effort as it has to diffuse the image until it is flat. So in practice one may prefer the faster alternative. Note that also the Gabor filter bank captures only textures with a limited scale.
The first value in Fig. 11 indicates the dissimilarity computed with the Gabor features, while the second value shows the distance in the sparse feature space. With both feature spaces the computed values are in accordance with what one would expect from a measure of texture dissimilarity. Mostly the value of the Gabor feature space is around three or four times larger than the value of the sparse feature space, so up to a constant factor both feature spaces yield approximately the same values showing that indeed they discriminate textures by means of the same properties. Only in a few cases there is a significant discrepancy. In such cases, it



	0 0	285 88	1067 339	915 387	834 123	466 64	1496 629	391 416
	285 88	0 0	1106 180	971 270	367 104	205 43	1734 576	73 142
	1067 339	1106 180	0 0	1354 395	1450 275	620 130	1924 803	1005 268
	915 387	971 270	1354 395	0 0	1455 287	1363 287	305 159	672 233
	834 123	367 104	1450 275	1455 287	0 0	403 100	2013 469	355 183
	466 64	205 43	620 130	1363 287	403 100	0 0	2248 720	335 275
	1496 629	1734 576	1924 803	305 159	2013 469	2248 720	0 0	1222 426
	391 416	73 142	1005 268	672 233	355 183	335 275	1222 426	0 0

Figure 11: Dissimilarities measured between some textures. First Value: Dissimilarity in the Gabor feature space. Second Value: Dissimilarity in the sparse feature space.

is difficult to decide which feature space is more in accordance with the expectation. This gives much evidence that, instead of the 12-dimensional Gabor feature space, also the sparse and more efficient 4-dimensional feature space can be used for texture discrimination without compromising the quality of the results.

5.3 Texture Segmentation

This is further supported by the segmentation results shown in Fig. 12 and Fig. 13. In this experiment, the features were feeded into the level set based segmentation framework described in [32, 6]. Like in the previous experiment, the gray value has been added as additional feature. The parameters that appear in the segmentation technique have been optimized separately for the Gabor feature space and the sparse feature space. The same parameters were then used for all test images. To ensure a fair comparison, also the Gabor features have been allowed to be smoothed by coupled TV flow.

Obviously, there are only marginal differences between the results obtained with Gabor features and those based on the sparse feature space. This can be explained by the fact that both approaches discriminate texture by means of the same basic features: magnitude, orientation, and scale. The additional orientation homogeneity present in $\mathbf{F}(t)$ as well as the reduced redundancy may explain why for some images the sparse feature space yields slightly better results. Its main advantage, however, is the improved efficiency. Although the computation of the features itself takes some more time, the segmentation is about three times faster due to the reduced number of features that have to be considered. The total increase in computation speed including feature computation is around factor 2. The exact numbers can be found in Table 1.

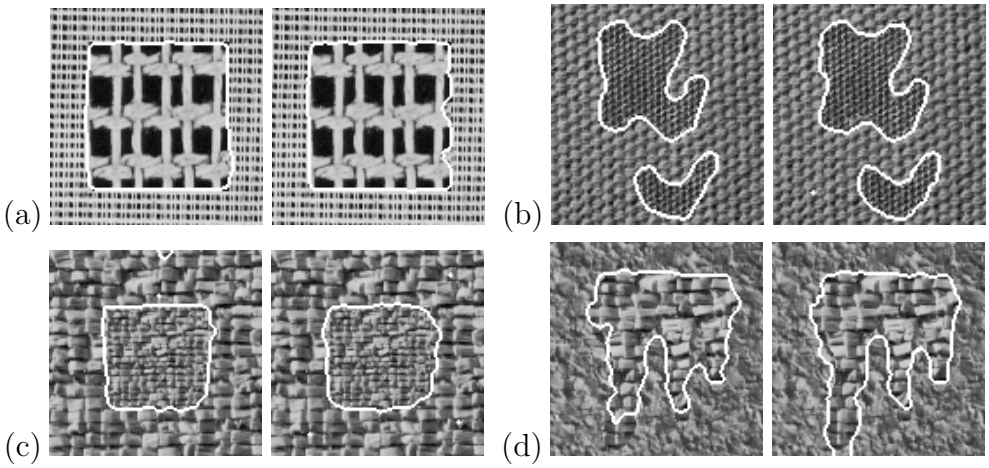


Figure 12: Synthetic texture images. LEFT: Segmentation with smoothed Gabor feature space. RIGHT: Segmentation with sparse feature space **F**.

Image	Size	Gabor features	**F**
Fig. 12a	120×122	3.4	1.7
Fig. 12b	109×113	3.1	1.5
Fig. 12c	123×119	3.5	1.7
Fig. 12d	121×122	3.4	1.7
Fig. 13a	220×140	6.8	3.3
Fig. 13b	329×220	16.9	8.6
Fig. 13c	250×167	9.6	4.8
Fig. 13d	189×244	10.2	5.1

Table 1: Computation times in seconds on an AMD Athlon XP1800+ (C++ implementation).

6 Conclusion

In this article we introduced a new paradigm to estimate the local scale in images. In contrast to conventional gradient based methods, our approach creates regions by means of an image simplification method and estimates the scale of these regions. Since the usage of local windows can be avoided, the estimates yield a very high localization accuracy. Two alternative image simplification methods have been investigated: TV flow and region merging. It has turned out that TV flow is much better suited for estimating the scale of texture elements. We also suggested two alternative ways to choose the simplification level. The possibility to a priori set a maximum simplification level appears most reasonable for texture discrimination, since regions above a certain scale are in general no longer considered as texture. The version that avoids such a parameter may in return be more attractive from the theoretical point of view or for applications besides texture discrimination.

In this paper the principal motivation for a region based scale measure has been the idea to replace the texture feature vector consisting of the responses of a Gabor filter bank by a smaller feature vector too comprising magnitude, orientation, and scale of the texture elements. A direct comparison between these two feature vectors has revealed very similar results. The higher implementation complexity of the sparse feature space is rewarded by a total speedup of about factor 2.

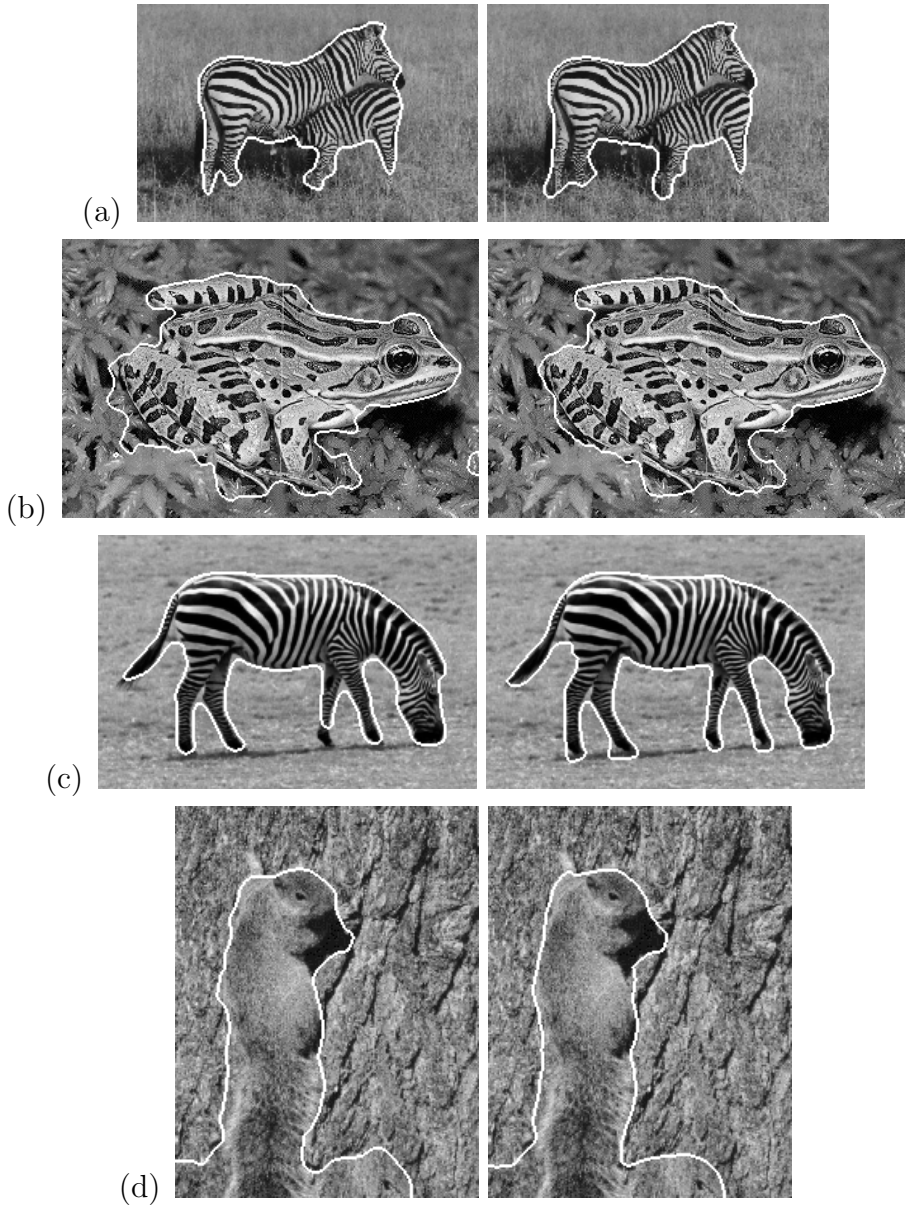


Figure 13: Real-world texture images. LEFT: Segmentation with smoothed Gabor feature space. RIGHT: Segmentation with sparse feature space $\mathbf{F}$.

Acknowledgements

Our research was partly funded by the DFG projects WE 2602/1-1 and WE 2602/1-2. This is gratefully acknowledged.

References

[1] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill–posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[2] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, Mar. 2001.

[3] F. Andreu, V. Caselles, J. I. Diaz, and J. M. Mazón. Qualitative properties of the total variation flow. *Journal of Functional Analysis*, 188(2):516–547, Feb. 2002.

[4] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, Aug. 1991.

[5] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, Jan. 1990.

[6] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 353–360. Springer, Berlin, Aug. 2003.

[7] T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*. Springer, Prague, Czech Republic, May 2004.

[8] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. Technical Report 113, Dept. of Mathematics, Saarland University, Saarbrücken, Germany, Oct. 2004.

[9] T. Brox, M. Welk, G. Steidl, and J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 86–100, Berlin, 2003. Springer.

[10] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[11] D. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, Feb. 1994.

[12] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, July 1998.

[13] D. J. Field. Relation between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4:2379–2394, 1987.

[14] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[15] D. Gabor. Theory of communication. *The Journal of the Institution of Electrical Engineers*, 93:429–457, 1946.

[16] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proc. IEEE International Conference on Computer Vision*, pages 716–723, Nice, France, Oct. 2003.

[17] G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.

[18] G. Gómez, J. L. Marroquín, and L. E. Sucar. Probabilistic estimation of local scale. In *Proc. International Conference on Pattern Recognition*, volume 3, pages 798–801, Barcelona, Spain, Sept. 2000.

[19] H. Jeong and I. Kim. Adaptive determination of filter scales for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(5):579–585, May 1992.

[20] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

[21] B. Julesz. Texton gradients: the texton theory revisited. *Biological Cybernetics*, 54:245–251, 1986.

[22] G. Koepfler, C. Lopez, and J. M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM Journal on Numerical Analysis*, 31(1):282–299, 1994.

[23] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, Boston, 1994.

[24] T. Lindeberg. Principles for automatic scale selection. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook on Computer Vision and Applications*, volume 2, pages 239–274. Academic Press, Boston, USA, 1999.

[25] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[26] S. Marcelja. Mathematical description of the response of simple cortical cells. *Journal of Optical Society of America*, 70:1297–1300, 1980.

[27] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Birkhäuser, Basel, 1994.

[28] D. Mumford and J. Shah. Boundary detection by minimizing functionals, I. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, June 1985. IEEE Computer Society Press.

[29] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

[30] J. Portilla, R. Navarro, O. Nestares, and A. Tabernero. Texture synthesis-by-analysis based on a multiscale early-vision model. *Optical Engineering*, 35:2403–2417, 1996.

[31] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.

[32] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003. IEEE Computer Society Press.

[33] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[34] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi. Geodesic active contours applied to texture feature space. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, volume 2106 of *Lecture Notes in Computer Science*, pages 344–352. Springer, Berlin, 2001.

[35] J. Sporring, C. I. Colios, and P. E. Trahanias. Generalized scale-selection. Technical Report 264, Foundation for Research and Technology - Hellas, Crete, Greece, Dec. 1999.

[36] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, May 2004.

[37] X.-C. Tai. Global extrapolation with a parallel splitting method. *Numerical Algorithms*, 3(4):427–440, 1992.

[38] D. Tschumperlé and R. Deriche. Diffusion tensor regularization with contraints preservation. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 948–953, Kauai, HI, Dec. 2001. IEEE Computer Society Press.

[39] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, Nov. 1995.

[40] R. van den Boomgaard and J. van de Weijer. Robust estimation of orientation for texture analysis. In *Proc. Texture 2002, 2nd International Workshop on Texture Analysis and Synthesis*, Copenhagen, June 2002.

[41] J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251–268. AMS, Providence, 2002.

[42] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, Mar. 1998.

[43] S.-C. Zhu, C. Guo, Y. Wu, and W. Wang. What are textons? In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision - ECCV 2002*, volume 2353 of *Lecture Notes in Computer Science*, pages 793–807, Copenhagen, Denmark, May 2002. Springer.