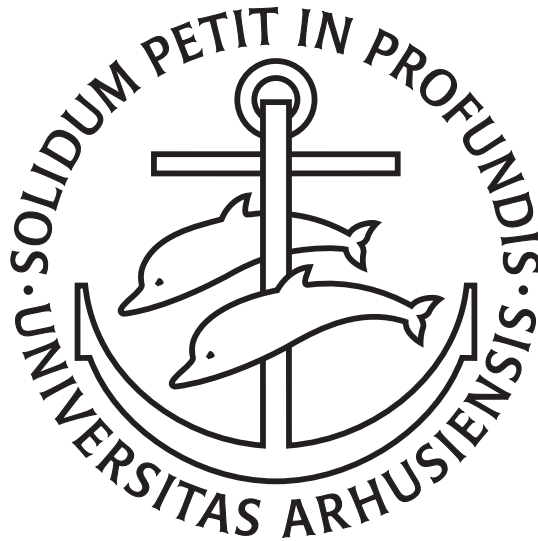


Adaptive picture enhancement

Rasmus Engholm



PhD Thesis

Department of Computer Science
University of Aarhus
Denmark

Adaptive picture enhancement

A Thesis
Presented to the Faculty of Science
of the University of Aarhus

by
Rasmus Engholm
February 12, 2010

Acknowledgements

I wish to thank Eva B. Vedel Jensen for taking on this project and following it to the very end. Your commitment, motivation, positivism and attention to details have been an inspiration for me, especially when the goal seemed unreachable. Furthermore I wish to thank the people at the Thiele Center for always having an open mind and making me feel welcome.

Henrik Karstoft for his project guidance and for continuously providing support, ideas and suggestions to all aspects of the project. For his contributions to the signal processing part of the project and for his willingness to discuss all my questions.

I would also like to thank B&O for initiating this project, for co-funding it and for presenting me with the real problems of video processing. Thanks to the engineers at B&O for many inspirational conversations and eye openers. Thanks to the picture quality group for the deep professional knowledge they never refused to share with me. Furthermore, they should be thanked for their patience with my questions.

Finally thanks to my family: Lene, Mathilde, Johanna and Otto for their love and never ending patience with me.

*Rasmus Engholm,
Viborg, February 2010.*

Summary

Noise reduction in video is an important and challenging subject. One of the important issues is that noise reduction should not destroy the sharpness of the video frames. Furthermore, it should be possible to use the developed procedures in real time - for obvious reasons.

The first part of this thesis presents new methods of noise reduction in video. Video versions of the two dimensional steering kernel are developed. One of the advantages of steering kernels is that edges are not blurred during the noise reduction process. Taking time asymmetry into account improves the performance of the kernel. It is shown that the performance of the developed video kernels is significantly better than that of the traditional two dimensional kernel. Automatic procedures of determining optimal parameters for the steering kernels are also developed.

In the second part of the thesis, the focus is on edge detection and image sharpness. Singular value decomposition (SVD) of an image is used to develop a new sharpness metric, SVD-G. In order to reduce the computational load, a local version of the metric is suggested, the SVD-L metric. It is shown that there is a tight relationship between the value of the developed objective metrics and subjective sharpness judged by test subjects. Finally, new methods of edge detection are developed in the thesis, based on detection of jumps in the intensity levels observed along line scans. The methods are based on theory developed in financial econometrics.

Dansk resumé

Støj reduktion er en vigtig og nødvendig del af video-processering. Eftersom skarphed er en af de vigtigste parametre for, hvordan folk opfatter video kvaliteten, er det vigtigt, at fjernelse af støj ikke sker på bekostning af skarphed. Endvidere er det i sagens natur vigtig at det kan ske i real tid.

Første del af afhandlingen præsenterer nye metoder til fjernelse af støj i video. To udgaver af den to-dimensionale steering kernel udvikles til brug i video. En af fordelene ved steering kernels er at kanter ikke bliver uskarpe ved støjfjernelse. Ved at indføre asymmetri i den temporale retning, forbedrer vi kernernes virkning på støjen. Det vises at de udviklede video kerner er markant bedre end de traditionelle to-dimensionale kerner. Automatiske metoder til at fastslå de optimale parametre til steering kernels bliver også udviklet.

I den anden del af afhandlingen fokuseres der på skarphed og kant detektion. Der anvendes Singular value decomposition (SVD) på et billede til at udvikle en ny skarpheds metrik SVD-G. For at mindske den beregningsmæssige opgave foreslås en lokal udgave, SVD-L metrikken. En tæt sammenhæng påvises mellem de objektive skarpheds mål: SVD-L og SVD-G og indsamlede subjektive skarpheds vurderinger fra testpersoner. Endelig udvikles nye metoder til kant detektion i billeder. Disse metoder baserer sig på bestemmelse af spring i intensiteter som observeres i scan linier. Metoderne bygger på teori udviklet i finansiel økonometri.

Contents

Acknowledgements	iii
Summary	iv
Dansk resumé	v
1 Introduction	1
2 Adaptive kernel filtering used in video processing	1
3 Optimal parameter estimation for anisotropic kernels	5
4 Image sharpness assessment based on SVD	8
5 Edge detection, using realized variance	13
References	17

Accompanying papers

- A** Engholm, R., Karstoft, H. and Jensen, E. B. V. (2009):
Adaptive kernel filtering used in video processing. Earlier version appeared in: Proceedings to Visual Communications and Image Processing 2009
- B** Engholm, R. and Karstoft, H. (2009):
Optimal parameter estimation for anisotropic kernel filtering in image processing. Submitted.
- C** Engholm, R., Karstoft, H. and Bech, S. (2009):
Image sharpness assessment based on singular value decomposition. Manuscript.
- D** Engholm, R., Schmiegel, J., Jensen, E. B. V. and Veraart, A. E. D. (2009):
Edge detection in image processing using realised variance measures. Manuscript.

1 Introduction

Below, a review of the main results of the thesis is given. The emphasis will be on the principle ideas behind the developed methods and their performance compared to existing methods. For a more detailed treatment, the reader is referred to the attached Papers A-D.

2 Adaptive kernel filtering used in video processing

In Paper A, three dimensional adaptive kernels to be used in noise reduction in video are developed. The idea is to construct kernels, based on local gradient information in the spatial as well as the temporal domain, that can handle abrupt scene shifts in a satisfactory manner. The adaptability of the kernels ensures that blurring over edges is minimized. As a consequence, not only improved noise reduction with respect to the rooted mean squared error is obtained but also, and most importantly, images with much sharper edges are produced.

The use of three dimensional filters for noise reduction in video is challenging because an uncritical use of such filters can have the unwanted effect that surrounding frames are too clearly visible. As a consequence, the video may appear blurred or, at scene shift, heavy artifacts may be introduced when two scenes are merged together.

As mentioned above, local gradient information is used in the construction of the kernels. For a given local window the average gradient is calculated and the actual kernel is then a deformation of an isotropic Gaussian kernel, determined by the local gradient. Kernels of this type are called steering kernels [1].

Extending the two dimensional steering kernel to three dimensions introduces some new challenges, since the temporal resolution cannot be compared to the spatial resolution because of different scales. By governing the time direction by a separate parameter, we construct in Paper A a video version of the steering kernel. We restrict the kernel such that two of the three axes in the estimated covariance structure of the anisotropic Gaussian distribution are parallel to the spatial domain (the one from the original two dimensional steering kernel). The third axis is then orthogonal to these and parallel to the temporal directional plane, see Figure 2.1. The three dimensional kernel is constructed in this fashion in order to maintain sharp edges within each frame and not in arbitrary directions in the spatial/temporal space. This three dimensional version of the steering



Figure 2.1: Illustration of the time symmetric three dimensional steering kernel, here using five relatively identical frames. The third frame is the one on which filtering is performed. The superimposed red ellipses represent level curves with same density of the kernel. It is obvious that redistributing some of the kernel weight to surrounding frames may give a better estimate in this case.

kernel is time symmetric since past and future are treated in a similar manner. The kernel illustrated in Figure 2.1 is indeed time symmetric.



Figure 2.2: Illustration of the idea of a time asymmetric three dimensional steering kernel, here using three relatively identical frames and a scene shift. The third frame is the one on which filtering is performed. The superimposed red ellipses represent level curves with same density of the kernel. It is obvious that redistributing some of the kernel weight to frame 1 and 2 will give a better estimate, and that frame 4 and 5 should have a small weight (ideally zero weight).

At scene shifts, the gradients will typically be large and the symmetric three dimensional steering kernel degrades into its two dimensional counterpart. Since either past or future frames are likely to be similar to the present frame, it is natural to modify the symmetric steering kernel such that time asymmetry is taken into account, see Figure 2.2. This provides us with the flexibility to separately exploit the redundancy in the past and future of a frame. The idea is to calculate the gradients in both temporal directions (past and future) and determine the temporal component of the kernel as a combination of the two parts. An example involving a scene shift is shown in Figure 2.2. The resulting kernel is called the time asymmetric three dimensional steering kernel.

We tested the proposed kernels in four different setups illustrating different aspects of the performance. The performance of the new three dimensional kernels was compared with the performance of the traditional two dimensional steering kernel.

For the first three tests, we used video sequences and added $N(0,5)$ normal noise. The three videos were, respectively, a static video sequence (Susie) with no camera movement, as shown in Figure 2.3, a dynamic video sequence with camera motion and a lot of movement from the persons in the scene (Football) and finally a video sequence with multiple scene shifts, as shown in Figure 2.4. The final test was performed on a real broadcasted video with analog noise originating from transmission. In the first three tests, we investigated the performance of the kernels by calculating the Rooted Mean Squared Error (RMSE). In the last test, we used subjective image quality, because no uncorrupted source existed.

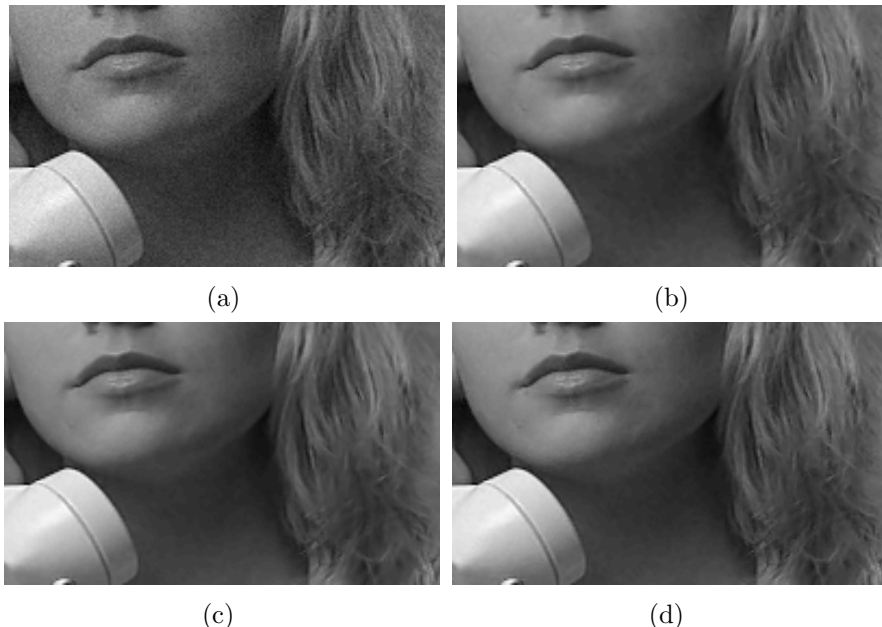


Figure 2.3: (a) Detail of frame number 7 of the 'Susie' video sequence, with $N(0,5)$ noise added. (b)- (d) The resulting frame after filtering with, (b) a steering kernel of size $[5 \times 5]$, (c) a symmetric video steering kernel of size $[5 \times 5 \times 3]$ and (d) an asymmetric video steering kernel of size $[5 \times 5 \times 3]$.

The tests showed that the performance of the proposed kernels was significantly better than the two dimensional version even using only two surrounding frames in the temporal domain. The improvement was most pronounced when dealing with relatively static videos, where the gradients are small resulting in a wide Gaussian kernel in the temporal direction, see Figure 2.3.

The asymmetric version is again better than the symmetric version, with a small improvement within a scene but most pronounced at scene shifts where the sym-

metric kernel, due to the above mentioned large gradients, turns into the two dimensional version, see Figure 2.4. It is however important to have in mind

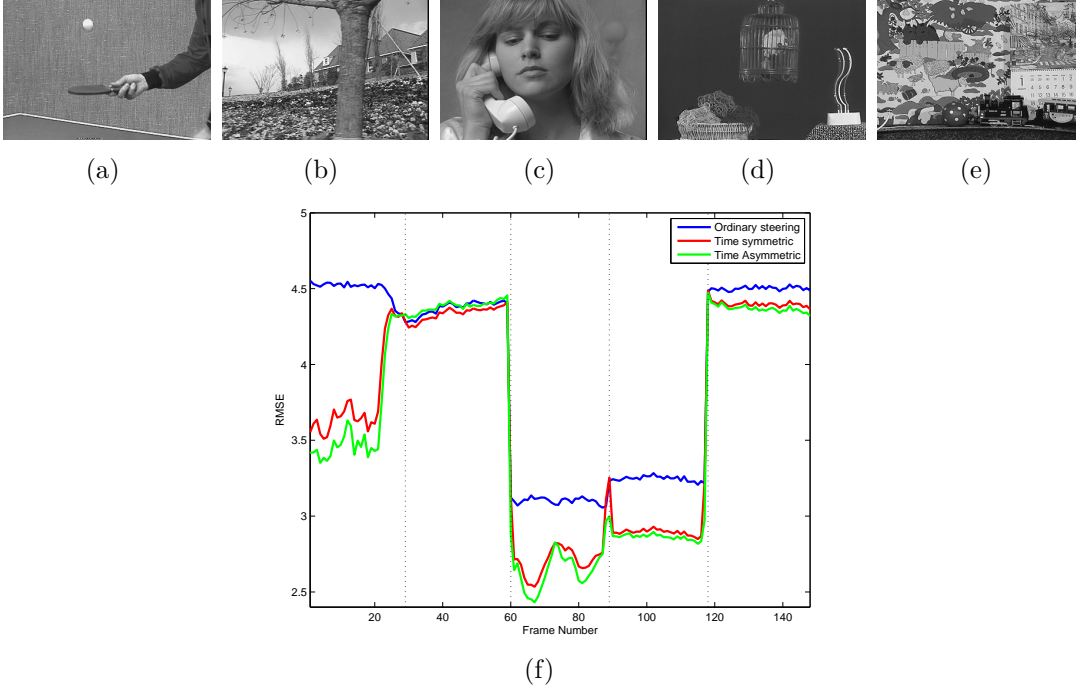


Figure 2.4: (a)-(e) A single frame from each of the five subsequences in the third test. To the scene, $N(0,5)$ noise was added and a filtering with the steering kernel and the two types of video steering kernels was performed. The resulting RMSE of the filtering is presented in (f). The vertical lines marks, where a scene shift occurs. The video kernels outperform the two dimensional kernel. The time asymmetric kernel performs overall best and handles scene shift better than the time symmetric.

that the ultimate goal of noise reduction is not to reduce RMSE, PSNR or other statistical metrics. What we really are interested in is to improve the perceived quality of the video. Future steps should therefore include a subjective perception test in order to verify that filtering with our proposed kernels really contains a quality improvement. Another important subject to deal with in future work is the implementation of the proposed kernels in a color framework. This includes the choice of color space.

3 Optimal parameter estimation for anisotropic kernel filtering in image processing

In Paper B we develop an automatic method for optimal parameter estimation in filtering with primarily anisotropic Gaussian kernels to be used for noise reduction in image and video processing. The proposed parameter estimation is based on a simple model of image degradation and restoration when filtering images. The aim has been to develop a fast and ideally also accurate method.

The challenges of using filtering methods for noise reduction in real time is that, in order to obtain optimal performance, the width of the kernels need to adjust not only to the noise level but also to the presented content. An image containing many sharp edges should be filtered with a sharper Gaussian kernel than an image with few edges to avoid blurring the details, thus decreasing the Peak Signal to Noise Ratio (PSNR).

In Paper B, the case of filtering with an isotropic Gaussian kernel is first considered. The two dimensional noisy signal is modelled as

$$g(n, m) = f(n, m) + \eta(n, m),$$

$n = 1, \dots, N, m = 1, \dots, M$, where f is the noise free signal and the $\eta(n, m)$'s are i.i.d. $N(0, \sigma_\eta^2)$. The filtered signal is denoted by $w_t g$, where

$$w_t g(n, m) = \sum_k \sum_l w_t(n, m) g(n - k, m - l)$$

and w_t is the Gaussian filter kernel

$$w_t(k, l) = \frac{1}{4\pi t} \exp\left(-\frac{k^2 + l^2}{4t}\right).$$

The aim is to find t such that the mean squared error (MSE) between the filtered signal and the true signal is minimized under the model described above. In Paper B, the MSE is decomposed as follows

$$MSE = MSE(noise) + MSE(signal)$$

where $MSE(noise)$ is the remaining noise in the filtered signal and $MSE(signal)$ is the blurring of the true signal, caused by the filtering. In Paper B, their dependency on t is derived,

$$MSE(noise) \approx \frac{\sigma_\eta^2}{8\pi} t^{-1}$$

and

$$MSE(signal) = at^2,$$

where a is a positive constant, depending on the Laplacian of the true signal.

In Figure 3.1, the validity of these approximations is examined in a concrete example. The approximations work well in the interesting region around the optimal t . A similar approach has been used previously to sharpen images by estimating the blurring of the image and then correct the blurred image for this [2].

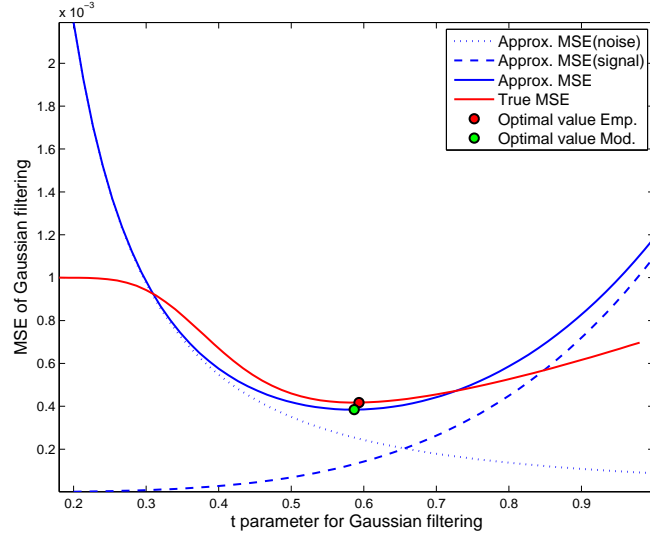


Figure 3.1: The approximate $MSE(noise)$, $MSE(signal)$ and total MSE as a function of the parameter t in the Gaussian filtering. The estimated optimal values of t is shown as a green dot. For comparison, the true MSE is also shown together with the true optimal value of t (red dot). Notice that the approximations work well around the optimal value.

The main challenge of Paper B is to find the optimal parameter for the anisotropic Gaussian kernel, often referred to as the steering kernel. This is an adaptive kernel that is based on local gradient information. The resulting kernel is a deformation of an isotropic Gaussian kernel constructed such that smoothing is not performed across edges but rather along them. Since the kernel adapts to the image content we have to adjust the method mentioned above, because in the anisotropic case a kernel is assigned to each pixel. As a consequence the parameter has to be calculated for each pixel. The final parameter is then chosen to be the median of all parameters to ensure robustness against extreme estimated parameters.

In Paper B we show how $MSE(noise)$ can be estimated analytically in this more involved situation. For the estimation of $MSE(signal)$ we have developed an anisotropic Laplacian operator.

For the filtering with the isotropic kernel the suggested method performs almost as good as manually found parameters at low noise levels. However for very noisy images, the method is less accurate. The performance of the filtering with an anisotropic kernel is more accurate because the anisotropic kernel does not degrade the image as much even at lower noise levels as shown in Figure 3.2. By visual inspection the method is almost indistinguishable from the results obtained by empirically found parameters as shown in Figure 3.3.

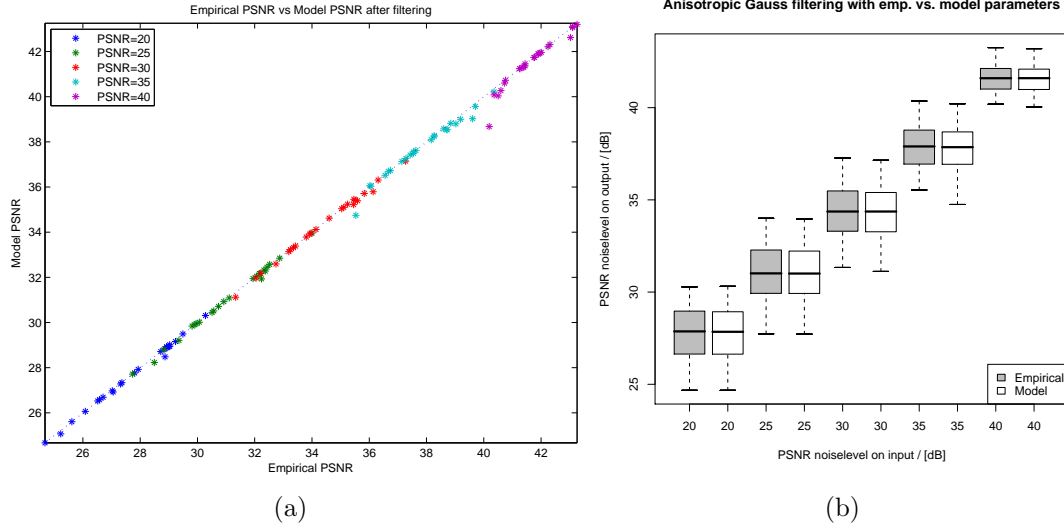


Figure 3.2: (a) The PSNR of 25 images after filtering with the empirically found parameters at 5 different noise levels. They are plotted against the same data but now after filtering with the optimal parameters predicted by the model. (b) Boxplot of the results of filtering with the two sets of parameters. Notice that the performance of the model parameters is very close to the empirically found parameters.

For the anisotropic case we have calculated an optimal parameter for each pixel. This is of course time consuming. Accordingly, we experimented using the median of only a subset of the optimal parameters. Since the parameters are quite clustered, using only a small randomly selected sample of 1 pct. of the pixels gave the same result for the median.

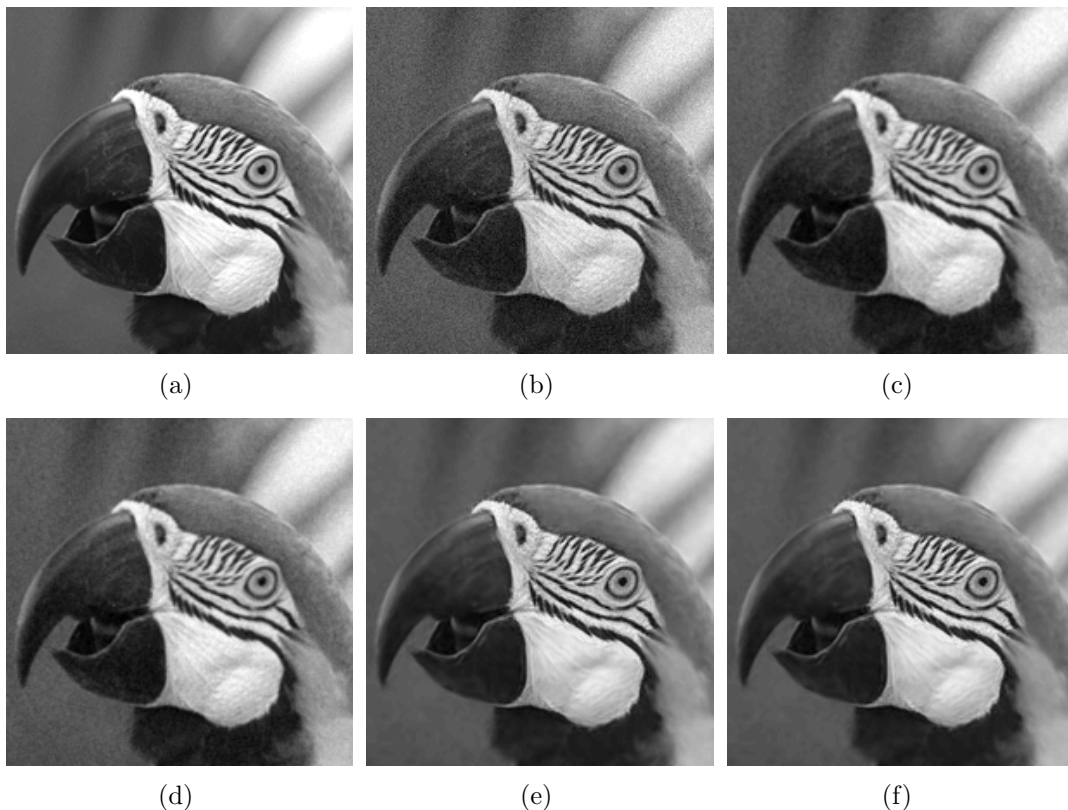


Figure 3.3: (a) Detail from original image, (b) with noise of 30 dB added, (c) isotropic Gaussian filtered with empirically found parameter, (d) isotropic Gaussian filtered with the model predicted parameter, (e) anisotropic Gaussian filtered with empirically found parameter and, (f) anisotropic Gaussian filtered with the model predicted parameter.

4 Image sharpness assessment based on singular value decomposition

Paper C describes the construction of a metric to assess the perceived sharpness/blur of an image. The main idea is to use the distribution of the singular values of the singular value decomposition (SVD) of an image, to predict the perceived sharpness [3]. The metric is a so called no-reference metric, meaning that the assessment is not related to an original uncorrupted image, but only based on the actual image.

By combining the image of the singular vectors and singular values in an interval, we can construct three images, covering the five pct. largest singular values, the singular values from 5-50 pct., and the last 50 pct. An example is shown in Figure

4.1. The decomposition of the image shows that the values related to sharpness is not present in the first singular values. Here the main energy is focused, i.e. the main image content is in a blurred form. The smallest singular values of the image contains close to no image information and can therefore also be discarded.

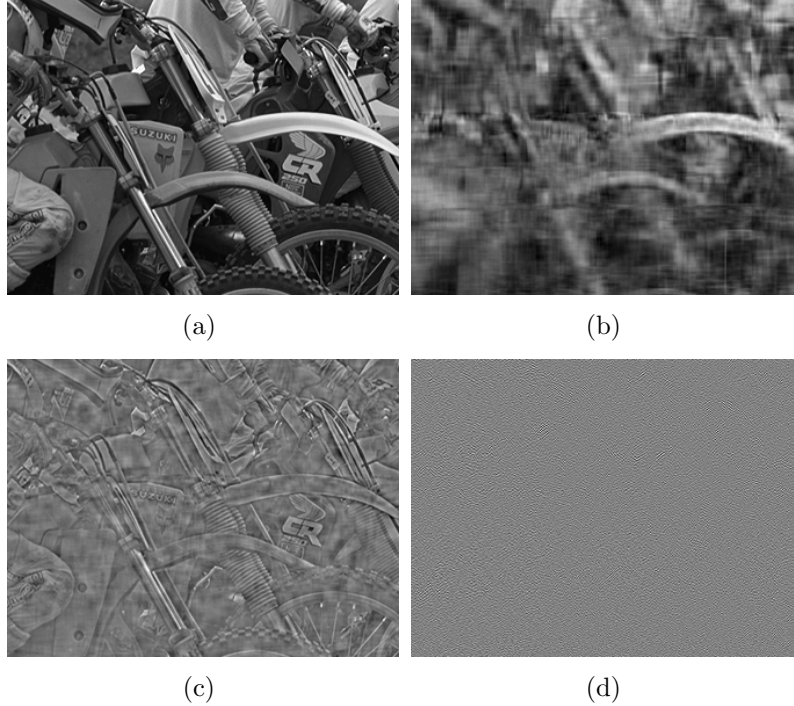


Figure 4.1: (a) Detail from the original 'bike' image, (b) the same image recreated by only using the 5 pct. largest singular values and corresponding vectors, (c) using the singular values between 5-50 pct. in magnitude and corresponding vectors and (d) using the 50 pct. smallest singular values and corresponding vectors. We notice that the sharpness of the image is only present at (c).

Of the remaining singular values, we observe that for sharp images the majority of the singular values are clustered at high values, but as images get more blurred, the majority of singular values clusters at lower values and is therefore primarily noise and imprecision of the computer, see Figure 4.2.

We have in Paper C used this observation to construct a metric that reflects this by taking the average of the logarithm of the singular values. We initially transform the singular values with a logarithmic function to reduce the dominance of the largest singular values. The use of logarithm can also be justified by the often seen logarithmic relationship between physical measurements and the perceived intensity, cf. the Weber-Fechner law [4]. We denote this sharpness metric SVD-G, since it is based on the SVD of the complete image.

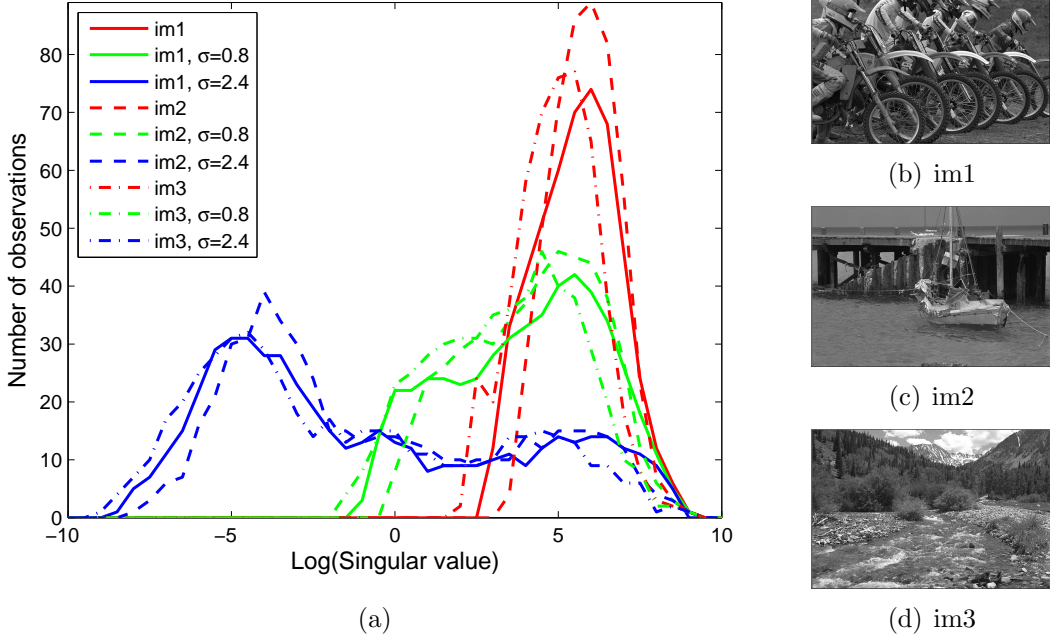


Figure 4.2: (a) The distribution of the singular values for three different images (b)-(d) at different degree of sharpness. We notice a very similar behavior of the distributions as the images get more blurred.

Since the complexity of the SVD algorithm is $\mathcal{O}(\min(nm^2, mn^2))$, where n and m are image width and height, we have in Paper C constructed a faster method, which is based on the same key idea. We couple this with the knowledge that the sharpness of the human visual system only covers 2 degrees of visual angle, also known as the fovea. By initially dividing the image into blocks of approximately this size, we calculate the SVD for each of these blocks. To avoid blocks with no sharpness to clutter the results, we use a fast edge detection algorithm to omit the blocks with no sharpness from the metric. We justify this by assuming that the opinion about the perceived sharpness is formed by looking at blocks containing edges. The final metric is then the mean of the sharpness in all blocks containing edges. This metric is called SVD-L as it uses SVD in local blocks. In Figure 4.3, the performance of both SVD-G and SVD-L is shown for a single image.

To evaluate the performance of the metrics, we have conducted a study among 21 subjects. They were asked to grade on a continuous scale the sharpness of four different images under different conditions. The judgement was performed by adjusting a slider. In order to avoid the common clustering at the markings of the typically used 5 point scale, the scale was with markings only at 'very unsharp' and 'very sharp'. The continuous scale was then mapped to a scale from 0 to 10 where 10 being the sharpest. Each subject was presented to each

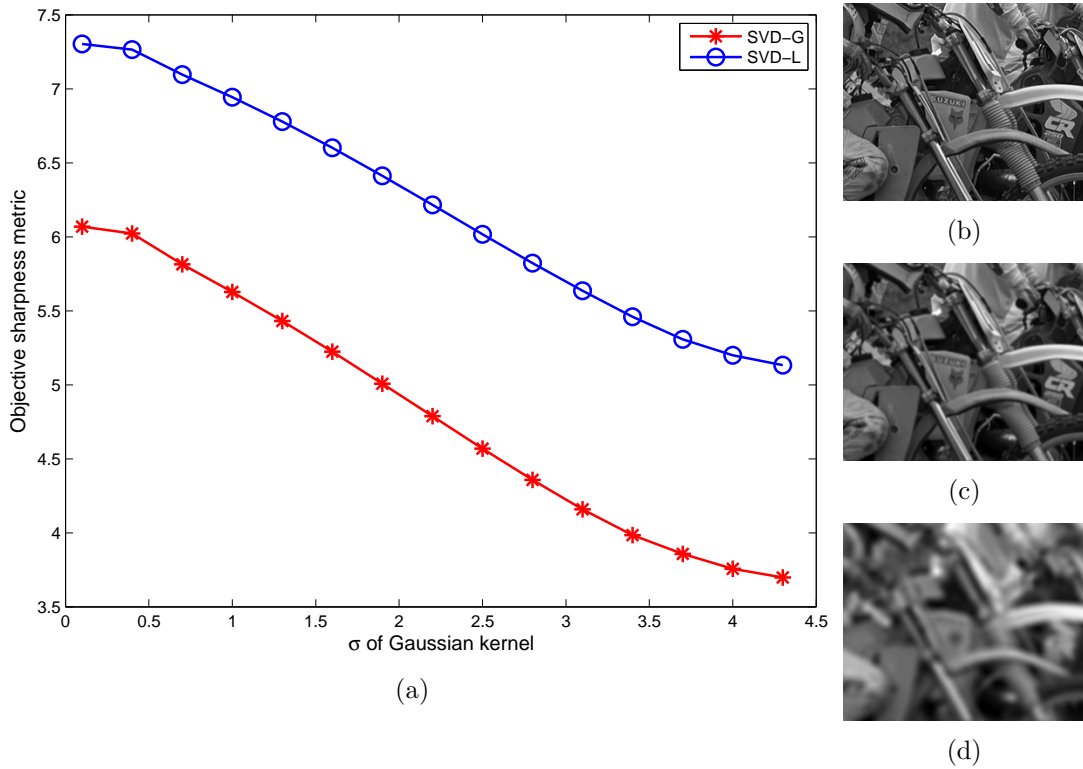


Figure 4.3: (a) The performance of the metrics at the same image at different levels of blur, constructed by filtering with a Gaussian kernel with different σ , (b) detail from image with no blur, (c) detail from image with Gaussian blur of $\sigma = 1$ and (d) detail from image with Gaussian blur of $\sigma = 4$. Notice that both metrics are monotonically decreasing.

image for 10 seconds and then asked to score the sharpness. The images were all from Full High Definition (1920 x 1080) or High Definition (1280 x 720) video sequences presented in native resolution to avoid blur introduced by scaling in the TV. The order of the images was determined by a balanced incomplete latin square design, to handle learning and fatigue bias. The obtained sharpness scores were averaged over all subjects, resulting in an Average Sharpness Rating (ASR) for each image.

The performance of the SVD-G and SVD-L was, together with two other state-of-the-art sharpness metrics, compared to the produced ASR. The competing metrics were the Perceptual Blur (PB) by Marziliano [5] and the Just Noticeable Blur (JNB) metric by Ferzi [6]. The Pearson and Spearman correlation and the Mean Absolute Error and Rooted Mean Squared Error were calculated as indications of the performance. The performance of the two metrics proposed in Paper C are very similar with a slight edge to the faster SVD-L and both outscore the NB and JNB in all four indicators. The Pearson correlation coefficient was 0.903 between SVD-L and ASR and 0.897 between SVD-G and ASR. The performance

of SVD-L is shown in Figure 4.4.

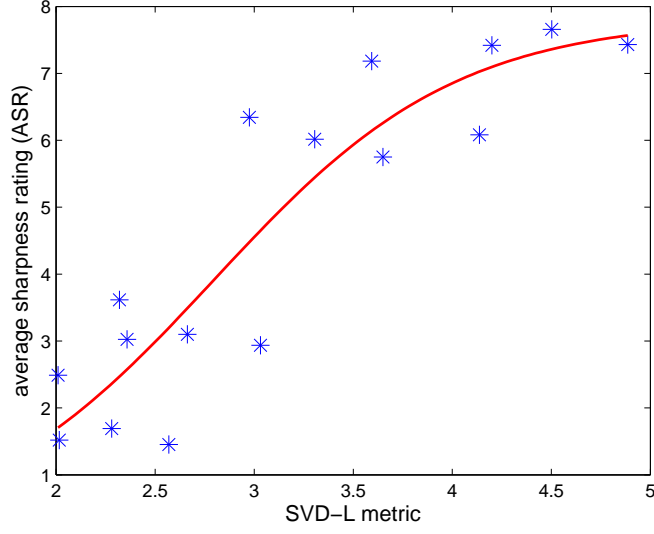


Figure 4.4: Comparison of the subjective ASR rating of sharpness with the objective SVD-L metric of sharpness, developed in Paper C. A total of 21 subjects has taken part in this investigation. The curve is obtained by fitting the data to a logistic function. The correlation is here 0.903 (Pearson) and 0.850 (Spearman).

In Paper C we have constructed a metric for estimating sharpness of images. The impact of sharpness in relation to the human visual system is much more complex when motion is involved. This situation is complicated both on the display technology side and the human visual system side. With regard to display technology, sample hold techniques used in LCD screens play an important role. In the human visual system, fundamental uncertainty about perceived motion is important.

Future work with the metrics developed in Paper C will naturally include modelling video sharpness in a SVD framework. Furthermore the importance of the choice of the thresholds of the singular values should be investigated further.

5 Edge detection in image processing, using realized variance measures

Paper D concerns edge detection in image processing, using realized variance measures. The key idea is to detect edges as jumps in the intensity level observed along line scans. Detection of such jumps is a subtle matter that has been studied in econometrics for years. A well-established asymptotic theory is available for ultra-high frequent time series data from finance. The asymptotics has been developed for the case where the distance between neighbour measurements goes to zero. The challenge of Paper D is to investigate whether these methods coming from econometrics can be modified such that they can be used with success in the detection of edges in digital images. In this transition process from econometrics to image processing, very recently developed finite sample corrections has played an important role, cf. Mancini [7], Jacob [8], Corsi et al. [9] and references therein. This is simply due to the fact that we cannot increase the resolution in the image under study arbitrarily.

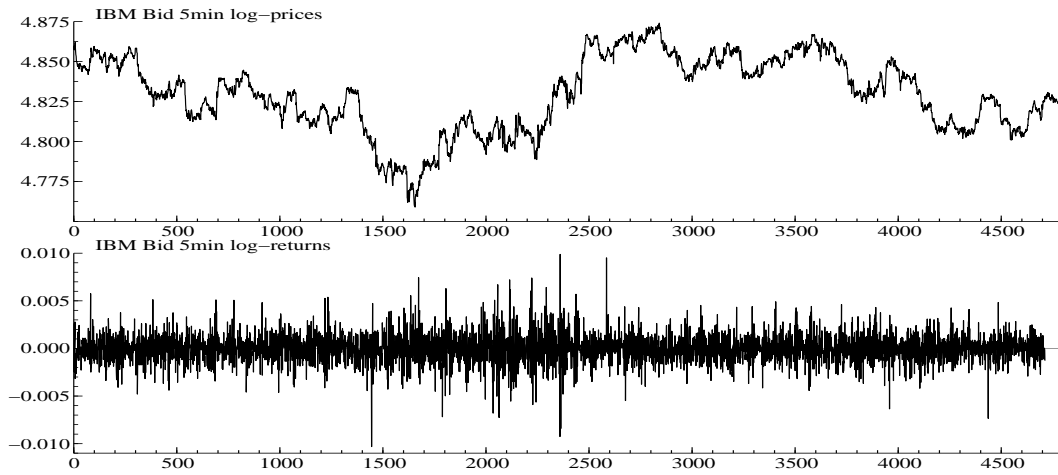
Realized variance measures relate to a discretely observed stochastic process $Y = \{Y_t : t \geq 0\}$. Let the observations be denoted by $Y_{i\delta}$, $i = 0, 1, \dots$. Then, the realized variance process at time t is simply

$$[Y_\delta]_t = \sum_{j=1}^{\lfloor t/\delta \rfloor} y_{j\delta}^2,$$

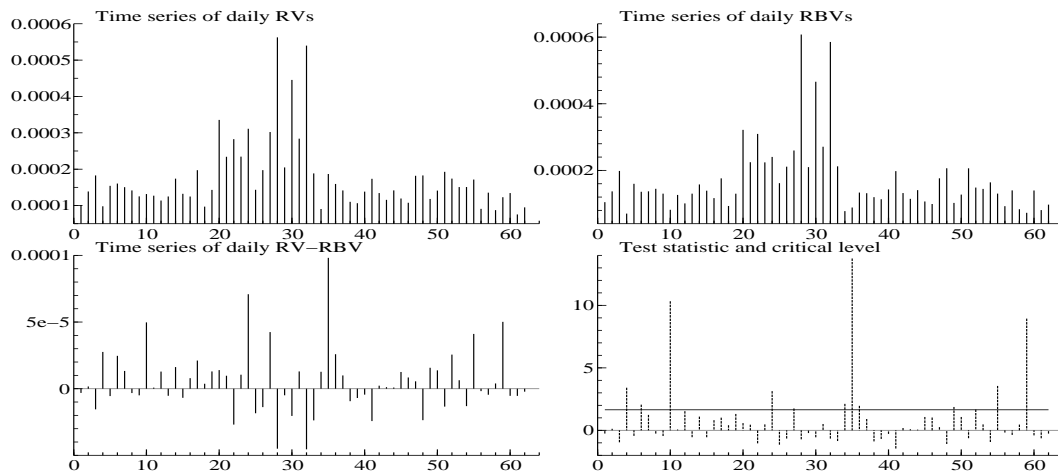
where $y_{i\delta} = Y_{i\delta} - Y_{(i-1)\delta}$ denotes the i th increment of Y and $\lfloor t/\delta \rfloor$ denotes the integer part of t/δ . It can be shown that, for small δ , the realized variance process can be approximated by the sum of a continuous process and a jump process. By various methods (realized bipower variation, truncated realized variance, truncated bipower variation), it is possible to produce a separate estimate of the continuous process and therefore the jump process can be singled out. The methods based on truncated variance are expected to be the most powerful in relation to edge detection since they take into account the finite resolution of the image. In Figure 5.1 below, a traditional application of realized variance measures in econometrics is shown.

In Paper D, the potential of these methods for edge detection in image processing is investigated. The conclusion is that it is indeed possible to detect edges, using the methods from analysis of econometric time series. Three examples are shown in Figure 5.2 below.

A closer look at the 'Bike' image in Figure 5.2 shows that the methods based on



(a) IBM data.



(b) Realized variation measures and jump test.

Figure 5.1: An example of testing for jumps in financial data. Here, (a) plots the logarithmic asset price sampled every five minutes and the corresponding five minutes returns. (b) Shows the results of combining the data into daily blocks of realized variances and realized bipower variation (top row). The estimated jump part is the difference between the realized variance and the realized bipower variation and is plotted together with test of whether the jumps are significantly different from zero. The result is that we have located 12 days with jumps in the price process.

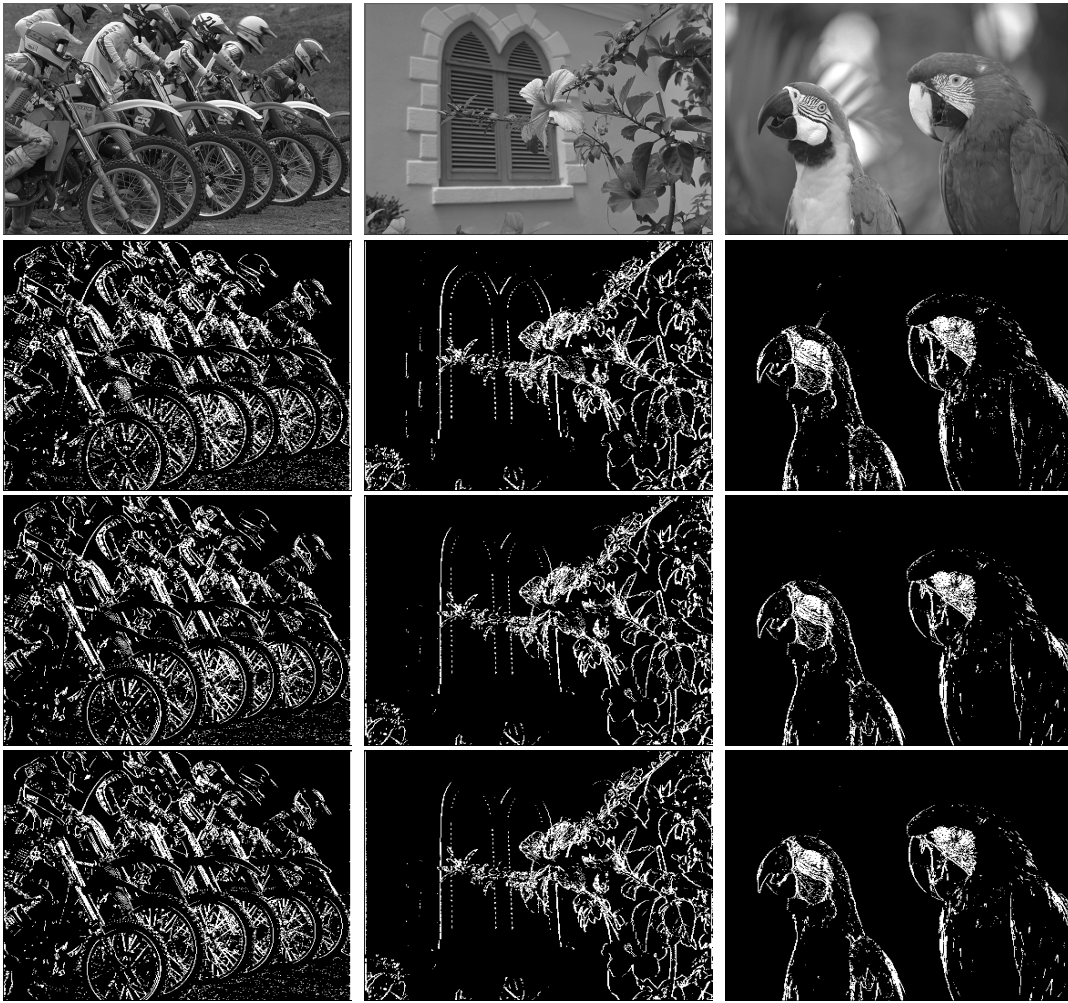


Figure 5.2: (First row): Original grey level images. (Second row): Result of edge detection using bipower variation. (Third row): Edges found, using truncated realized variance. (Fourth row): Edges found, using truncated bipower variation.



Figure 5.3: Detail of edge extraction of the 'Bike' image, using, respectively, (left) bipower variation, (middle) truncated realized variance and (right) truncated bipower variation. Notice that the edges of the rim of the wheel in the last two images appear more connected and less noisy.

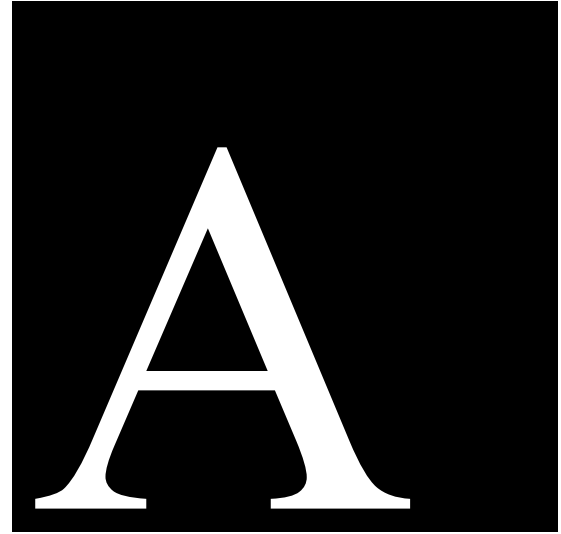
truncated variances are superior. For instance, when observing the rim of the wheel we notice less noise and a more connected structure when using truncated variances, see Figure 5.3.

Another important conclusion of the investigation is that jump detection seems to work also in the case where the jump is estimated from remarkably short series of observations. In Figure 5.2, the length of the series of observations upon which a jump is estimated is as low as 3. This is in marked contrast to the usual situation in applications within econometrics.

In econometrics, the emphasis is often on testing whether a particular jump is significant or not. This issue seems less relevant in image processing since small (although significant) jumps are not likely to be discovered by the viewer.

References

- [1] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, 2007.
- [2] M. Lindenbaum, M. Fischer, and A. Bruckstein. On Gabor’s contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, 1994.
- [3] D. Kalman. A singularly valuable decomposition: The SVD of a matrix. *The College Mathematics Journal*, 27(1):2–23, 1996.
- [4] S. Winkler. *Digital video quality: vision models and metrics*. Wiley, 2005.
- [5] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. Perceptual blur and ringing metrics: Application to JPEG2000. *Signal Processing: Image Communication*, 19(2):163–172, 2004.
- [6] R. Ferzli and L.J. Karam. A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB). *IEEE Transactions on Image Processing*, 18(4):717, 2009.
- [7] C. Mancini. Estimating the integrated volatility in stochastic volatility models with Lévy type jumps. Preprint, 2006.
- [8] J. Jacod. Asymptotic properties of realized power variations and related functionals of semimartingales. *Stochastic Processes and their Applications*, 118(4):517–559, 2008.
- [9] F. Corsi, R. Reno, and D. Pirino. Threshold bipower variation and the impact of jumps on volatility forecasting. Working paper, 2009.



Engholm, R., Karstoft H. and Jensen E. B. V. (2009)
Adaptive kernel filtering in video processing.
Earlier Version appeared in: *Proceedings to Visual
Communication in Image processing 2009.*

Adaptive kernel filtering used in video processing

Rasmus Engholm^{a,b,c}, Henrik Karstoft^b and Eva B. Vedel Jensen^a

^aAarhus University, Department of Mathematical Sciences, Ny Munkegade, DK-8000 Aarhus C

^bEngineering College of Aarhus, Dalgas Avenue 2, DK-8000 Aarhus C

^cBang & Olufsen, Peter Bangs Vej 15, DK-7600 Struer

ABSTRACT

In this paper we present a noise reduction filter for video processing. It is based on the recently proposed two dimensional steering kernel, extended to three dimensions and further augmented to suit the spatial-temporal domain of video processing. Two alternative filters are proposed - the time symmetric kernel and the time asymmetric kernel. The first reduces the noise on single sequences, but to handle the problems at scene shift the asymmetric kernel is introduced. The performance of both kernels is tested on simulated data and on a real video sequence together with the original steering kernel. The proposed kernels improves significantly the Rooted Mean Squared Error (RMSE) compared to the original steering kernel method on video material.

Keywords: Adaptive image filtering, Noise reduction, Kernel Methods, Video Processing.

Kernel methods are non-parametric methods of noise reduction. In relation to video it is a strength that the methods are non-parametric since broadcasted material have various origins and qualities. Electronic interference and flaws in the transmission and the digital image acquisition add noise to the image/video. Some programs are sent digitally at high bit rates, others are sent as analog signals via cables. Some programs have been stored on magnetic tapes for years and thereby degraded and some are sent directly. It seems impossible to construct a single parametric model that can describe all the different types of material and noise. An additional benefit of the kernel methods are that they have applications within noise reduction as well as within interpolation.

The transition to much larger screens, together with the transition from Catode Ray Tube technology (CRT) to Liquid Crystal Displays (LCD) and similar digital screen technologies, makes noise appear much more disturbing on a television set. It is therefore important that the noise is reduced to get the viewer the best perceived experience possible. However other factors such as sharp edges are also important for the perceived video quality. This is what makes the noise removal difficult because blurring reduces the noise but also removes the sharpness of the edges. On the other hand edge enhancement techniques e.g. peaking enhances the noise. Another similar problem is that texture and noise shares many of the same statistics e.g. a sandy beach or the leafs of a tree can locally appear very random.

We will in section 1 first describe related work on kernel methods with special emphasis on the recently proposed steering kernel method. In section 2 we extend this to the video domain, and augment it to handle some of the difficulties in video. We present experiments on both video with simulated noise and real analog noise in section 3 to compare and show the behavior of various methods considered. The application we have in mind is television but the scope of kernel methods is much broader.

1. RELATED WORK

In recent years methods that utilizes non-static kernels, dynamic kernels, that adapt to the data, have been proposed, see.^{1,2} These kernels add flexibility to the performance as well as introduce new challenges.

Instead of just using the Euclidean distance to determine the weight of the observation at a given site we may combine the Euclidean distance with other metrics. Examples of this, where the observed value also enters into the computation of the weight are bilateral filter and steering kernels.

Further author information: (Send correspondence to Rasmus Engholm)

Rasmus Engholm: E-mail: engholm@imf.au.dk, Telephone: +45 6070 6077

1.1 Bilateral filter

The bilateral kernel has had a lot of support since the introduction in 1998² due to its edge preserving properties combined with the appealing simplicity and intuitive construction. The bilateral filter is an attempt to build a filter kernel that has the smoothing properties of the ordinary Gauss filter, and at the same time does not smooth the edges.

The intuition is that sites with similar observations which are also located close to each other is from the same structure and therefore can be assigned larger weights than sites with either different observations or location far away from each other. The bilateral kernel is then a product of two kernels. In general any low pass kernel can be used, but usually the Gaussian kernel is used.

This intuitive approach has received a lot of theoretical investigation tying it together with other edge preserving filtering techniques such as anisotropic diffusion, robust estimation, weighted least squares³ and adaptive smoothing.^{4,5} Others have tried to speed up the performance.⁶⁻⁸ The algorithm has also found use within computer graphics where it has been used to perform smoothing of 3D-meshes.⁹ The origin of the bilateral filter is empirical, but it can be derived from the weighted least squares formulation, similar to 0-order local regression.

On artificial examples the filter performs very satisfactory. However on natural images the result can be cartoon-like, because within the areas defined by the edges, all fine details get blurred.

1.2 Steering kernels

Instead of using the difference in intensities to guide the filtering like the bilateral filter, thus indirectly using gradient information, the intuition behind the steering kernel, introduced in¹ is to directly use the gradient information to construct the kernel. This means that the actual kernel will be a deformation of an ordinary smoothing kernel (scaled, rotated, skewed). The steering kernel was introduced with a Gaussian kernel, but any type of kernel could in principle be used. We should note that the original formulation suggest that the kernel is used iteratively. However the optimal number of iterations remains unclear, why we here will only consider it as a one pass filter.

As with the bilateral filter, the objective is to remove noise while preserving the edges. The shortcomings of the bilateral filter described above appear partly because the kernel is designed directly from the observations i.e. a single black pixel in a white neighborhood can be hard to smooth, since the intensity part of the kernel will ensure that all weight will be put on the single observation under consideration.

The steering kernels are defined using the gradient information and performing principal component analysis, to reduce all the gradients in a local window to the dominant orientation in that window. A kernel is then calculated based on this orientation giving weights to the observations in the window such that most smoothing is performed along edges and not across edges.

We let \mathbf{x}_0 denote the pixel in a local window at which we want to estimate the grey value. The sites of the observations are indexed in a two dimensional grid $(x_{1i}, x_{2j}), i = 1, \dots, n, j = 1 \dots, m$, representing our pixels in the local window. To each pair (x_{1i}, x_{2j}) we have a single associated observation, y_{ij} , giving the grey level value in the image.

We consider a local window and define the gradient as

$$(\nabla y)(x_{1i}, x_{2j}) = \left[\frac{\partial y}{\partial x_1}(x_{1i}, x_{2j}), \frac{\partial y}{\partial x_2}(x_{1i}, x_{2j}) \right]$$

Since we are working on discrete values we must use a discrete differentiation operator.

We are interested in finding the dominant orientation of the structure in the local window of the image, so that we do not smooth in that direction. On average we can consider the gradients to be orthogonal to the

dominant orientation. We do a PCA analysis on the gradients in the local window to find the dominant edge orientation. The gradients of a local window is then reordered as an $nm \times 2$ matrix:

$$G = \begin{bmatrix} (\nabla y)(x_{11}, x_{21}) \\ (\nabla y)(x_{11}, x_{22}) \\ \vdots \\ (\nabla y)(x_{11}, x_{2m}) \\ (\nabla y)(x_{12}, x_{21}) \\ \vdots \\ (\nabla y)(x_{1n}, x_{2m}) \end{bmatrix},$$

where each row specifies first the gradient in the horizontal and then in the vertical direction. The steering kernel uses a simple estimate of the covariance structure of the gradients obtained by

$$\begin{aligned} \mathbf{C}_{\mathbf{x}_0} &= \frac{1}{nm} \sum_{i,j} \left[\begin{array}{cc} \left(\frac{\partial y}{\partial x_1} \right)^2 & \frac{\partial y}{\partial x_1} \frac{\partial y}{\partial x_2} \\ \frac{\partial y}{\partial x_1} \frac{\partial y}{\partial x_2} & \left(\frac{\partial y}{\partial x_2} \right)^2 \end{array} \right]_{(x_{1i}, x_{2j})} \\ &= \frac{1}{nm} G^T G. \end{aligned}$$

This symmetric structure, $\mathbf{C}_{\mathbf{x}_0}$, of the gradients in a local window is then used to form the steering kernel. The objective is to spread the kernel parallel to the edges, so the edges remain unblurred. Secondly if the gradients in general are small we would assume that the differences originate from noise and need smoothing, thus requiring a relatively flat kernel. On the other hand if the gradients in general are large we will assume that it originates from structure and need less smoothing which calls for a steep kernel. Considering the eigenvectors

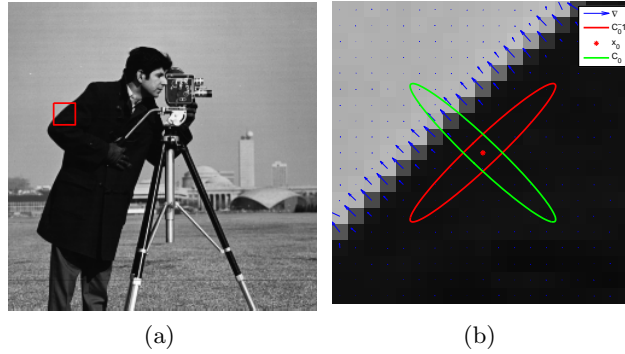


Figure 1. (a) Image with a red square that indicates the neighborhood being examined. (b) Contour ellipses of $\mathbf{C}_{\mathbf{x}_0}$ and $\mathbf{C}_{\mathbf{x}_0}^{-1}$ superimposed onto the current neighborhood together with the estimated gradients.

of $\mathbf{C}_{\mathbf{x}_0}$, the first eigenvector is in the direction of largest variance, i.e. in the direction of on average largest gradient. This structure is exactly the opposite of what we look for. We wish to rotate the structure and make the largest eigenvalue orthogonal to the main gradient direction. This can be accomplished by inverting $\mathbf{C}_{\mathbf{x}_0}$. Since the inverse of a symmetric matrix has the same eigenvectors but reciprocal eigenvalues, we obtain the wanted structure, see Fig. 1.

Using this structure to construct a Gaussian kernel we arrive at the steering kernel

$$K_S(\mathbf{x}_0, \mathbf{x}_w) = c \cdot \exp \left(-\frac{(\mathbf{x}_0 - \mathbf{x}_w)^T (\mathbf{C}_{\mathbf{x}_0}) (\mathbf{x}_0 - \mathbf{x}_w)}{2\sigma_s^2} \right),$$

where $K_S(\mathbf{x}_0, \mathbf{x}_w)$ is the weight associated to pixel \mathbf{x}_w in the local window, $\mathbf{C}_{\mathbf{x}_0}$ is the structure of the gradients for the local window centered around \mathbf{x}_0 , and c is a normalizing constant. σ_s is the parameter governing the basic level of smoothing, e.g. the steepness of the Gaussian kernel before it is deformed by $\mathbf{C}_{\mathbf{x}_0}$. Fig. 2 illustrates the construction of a steering kernel.

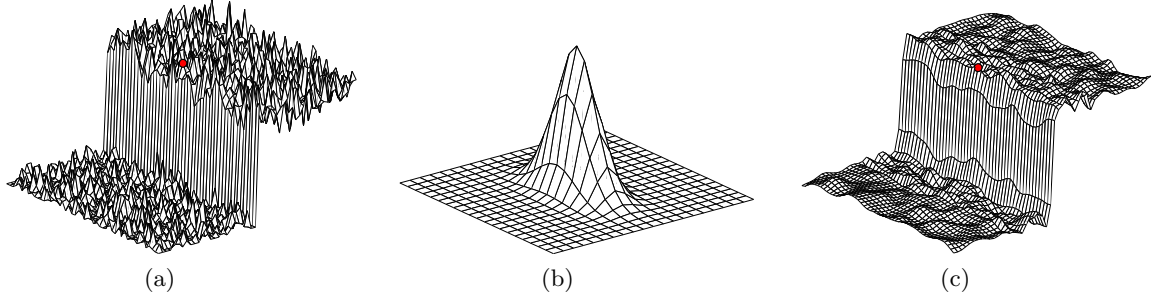


Figure 2. Construction of the steering kernel based on the red dot in (a) in a 21×21 neighborhood. (b) The steering kernel based on the structure of the gradients. (c) The result of filtering with the steering kernel.

2. STEERING KERNELS IN VIDEO

Applying steering kernels on video is an obvious extension. By having more frames we want to use the redundancy of the video to get a better estimate of the pixel value. The redundancy occurs because in video, e.g. 50 frames per second, are available. Of course the type of content determine how useful the surrounding frames are. If a scene shift occurs we cannot use the information to get a better estimate. We propose firstly to extend the idea of the steering kernel such that the kernel is three dimensional and secondly to introduce time asymmetric kernels to handle the problems at scene shifts.

With grey level video we have a three dimensional site input $(x_{1i}, x_{2j}, x_{3k}), i = 1, \dots, n, j = 1, \dots, m, k = 1, \dots, l$, where (x_{1i}, x_{2j}) indicates the spatial location of the site in the frame and x_{3k} the frame number, i.e. the temporal location of the site. The observation at site (x_{1i}, x_{2j}, x_{3k}) is denoted y_{ijk} , representing the grey level value.

2.1 Time symmetric kernel

Extending the steering kernel introduces some difficulties, since the temporal resolution can not be compared to the spatial resolution because of different scales. By governing the time direction by a separate parameter, we construct a video version of the steering kernel. We restrict the kernel such that two of the three axes in the estimated covariance structure is parallel to the spatial domain (the one from the original steering kernel). The third is then orthogonal to these and parallel to the temporal directional plane as shown on Fig. 3. This is chosen because we wish to maintain sharp edges with respect to each frame and not in arbitrary directions in the spatial/temporal space.



Figure 3. Illustration of the time symmetric kernel, here using five relatively identical frames. The third is the one on which filtering is performed. The superimposed red ellipses represent level curves with same density of the kernel. It is obvious that redistributing some of the kernel weight on surrounding frames in this case can give a better estimate.

We construct the matrix similar to the two dimensional case but augment the matrix to a 3×3 matrix. We reuse the $\mathbf{C}_{\mathbf{x}_0}$ unchanged but augment it with the variance of the gradient in the temporal direction.

It should be emphasized that we for the calculation of the temporal variance use all gradients in the temporal direction in the considered block, whereas we only use the gradients in the spatial directions from the central frame. The resulting shape of the video kernel in the central frame is the same shape as in the 2 dimensional case, but with smaller weight since some of the weight is transferred to the surrounding frames. The last entry in the diagonal of our enlarged matrix is, up to a constant,

$$d_{\mathbf{x}_0} = \frac{1}{nml} \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \left(\frac{\partial y}{\partial x_3}(x_{1i}, x_{2j}, x_{3k}) \right)^2.$$

We can now define the covariance structure of the time symmetric kernel as

$$\tilde{\mathbf{C}}_{\mathbf{x}_0} = \begin{bmatrix} \frac{\mathbf{C}_{\mathbf{x}_0}}{\sigma_s^2} & 0 \\ 0 & \frac{d_{\mathbf{x}_0}}{\sigma_t^2} \end{bmatrix},$$

where $\mathbf{C}_{\mathbf{x}_0}$ is exactly as before. Since we want to be able to adjust the overall smoothing in the spatial and temporal directions independently, we introduce a new parameter σ_t governing the temporal degree of smoothing. We can now define the time symmetric steering kernel as

$$K_{TS}(\mathbf{x}_0, \mathbf{x}_w) = c \cdot \exp \left(-\frac{(\mathbf{x}_0 - \mathbf{x}_w)^T (\tilde{\mathbf{C}}_{\mathbf{x}_0}) (\mathbf{x}_0 - \mathbf{x}_w)}{2} \right),$$

where \mathbf{w} index the sites in the local block and c is the normalizing constant.

We notice that the kernel is locked to have one eigenvector parallel to the temporal direction, since there is no correlation between the temporal and the spatial direction. Furthermore we notice that when experiencing small temporal gradients, i.e. the content of the frame is relatively unchanged, the estimate of $d_{\mathbf{x}_0}$ is going to be relatively small. This will lead to a kernel stretching in the temporal direction giving more weight to the surrounding frames. On the other hand if the temporal gradients are large, we will get a large estimate of $d_{\mathbf{x}_0}$, resulting in a kernel shrinking in the temporal direction ultimately becoming the two dimensional steering kernel.

2.2 Time asymmetric kernel

Utilizing the surrounding frames as in the kernel described above improves noise reduction, however the performance not satisfactory at scene shifts and at pan. Obviously at scene shifts the gradient in the temporal direction is relatively large, resulting in a very flat kernel in the temporal direction, almost identical to the two dimensional kernel. We therefore propose an alternative kernel construction, where the temporal direction supports asymmetry. This provides us with more flexibility to exploit the redundancy in only one temporal direction when entering/exiting a scene shift as illustrated in Fig. 4. For estimating the asymmetry kernel we have to estimate the two parts of the kernel separately. We therefore define the asymmetric kernel as

$$K_{TAS}(\mathbf{x}_0, \mathbf{x}_w) = c \cdot \exp \left(-\frac{(\mathbf{x}_0 - \mathbf{x}_w)^T (\tilde{\tilde{\mathbf{C}}}_{\mathbf{x}_0, \mathbf{x}_w}) (\mathbf{x}_0 - \mathbf{x}_w)}{2} \right),$$

where

$$\tilde{\tilde{\mathbf{C}}}_{\mathbf{x}_0, \mathbf{x}_w} = \begin{bmatrix} \frac{\mathbf{C}_{\mathbf{x}_0}}{\sigma_s^2} & 0 \\ 0 & \frac{\tilde{d}_{\mathbf{x}_0, \mathbf{x}_w}}{\sigma_t^2} \end{bmatrix}.$$

Here $\tilde{d}_{\mathbf{x}_0, \mathbf{x}_w}$ is determined as follows. Let $k(x_w)$ be the number of the frame in which x_w lies. Let l_- be the number of frames in the local block around x_0 before the frame in which x_0 lies. Likewise, let l_+ be the number

of frames after the frame in which x_0 lies. Then,

$$\tilde{d}_{x_0, x_w} = \begin{cases} \frac{1}{nml_-} \sum_{i=1}^n \sum_{j=1}^m \sum_{k < k(x_0)} \left(\frac{\partial y}{\partial x_3}(x_{1i}, x_{2j}, x_{3k}) \right)^2 & \text{if } k(x_w) < k(x_0) \\ \frac{1}{nml_+} \sum_{i=1}^n \sum_{j=1}^m \sum_{k > k(x_0)} \left(\frac{\partial y}{\partial x_3}(x_{1i}, x_{2j}, x_{3k}) \right)^2 & \text{if } k(x_w) > k(x_0) \end{cases} \quad (1)$$

By convention, $\tilde{d}_{x_0, x_w} = 0$ if x_w and x_0 belong to the same frame.



Figure 4. Illustration of the idea of the time asymmetric kernel, here using three relatively identical frames and a scene shift. The third is the one on which filtering is performed. The superimposed red ellipses represent level curves with same density of the kernel. It is obvious that redistributing some of the kernel weight to frame 1 and 2 will give a better estimate, and that frame 4 and 5 should have a small or zero weight.

3. EXPERIMENTS

We tested the proposed kernels in four different setups illustrating the performance. We also examined the performance of the regular two dimensional steering kernel for comparison. In the first three setups we used sequences with added normal noise with $N(0,5)$. The last test was done on a real broadcasted video with analog noise originating from transmission. For the simulated data we optimized wrt. the Rooted Mean Squared Error (RMSE). In the last example we used subjective image quality.

3.1 Single static sequence

In the first experiment, we used a standard sequence, *Susie*, representing a very static setting. We added noise with $N(0,5)$. We compared the performance of the steering kernel of size $[5 \times 5]$ with the video steering kernel of size $[5 \times 5 \times 3]$ with respect to RMSE. For parameter estimation we used a frame in the middle of the sequence to determine the steering kernel parameter. These parameters were also used in the video steering kernels. We notice a significant improvement from the two dimensional kernel. The performance of the three dimensional kernels with and without symmetry is similar. There is no clear winner. For this sequence, see Fig. 5, the perceptual quality appears much better for the video steering kernels than for the original two dimensional steering kernel.

3.2 Single dynamic sequence

The setup of this experiment is identical to the previous. The used sequence, *Football*, represents a very dynamic sequence with both camera and object movement. Adding movement makes the video steering kernels less optimal, see Fig. 6. However, the performance is still constantly superior to the two dimensional kernel. The variance of the video kernels performance is more pronounced than in the previous example shown in Fig.5. The time asymmetric kernel is slightly better than the time symmetric kernel.

3.3 Sequences with scene shifts

We used another video consisting of five very different concatenated standard sequences, see Fig. 7. Those were *Table Tennis*, a relative static scene with a noise like wall paper, *Flower Garden* a panoramic sequence of small flowers, *Susie*, as mentioned before, *Birdcage*, a very static scene with a birdcage spinning slowly around, and finally *Mobcal*, a panoramic video of mobile and calender. We added noise as before to investigate the performance when encountering scene shift. The three dimensional kernels performed better than the two dimensional kernels. The only exception is in the *Flower Garden* sequence where the performance of all three are similar bad. We notice that the largest difference between the two dimensional and the three dimensional kernels is obtained in sequences where the camera is relatively static and the background has a noise like texture, e.g. *Table Tennis*. Comparing the time symmetric kernel and the time asymmetric kernel, we notice that in general the asymmetric kernel performs better. Furthermore at scene shift indicated with a dashed vertical line on Fig.7(f) the performance of the asymmetric kernel is superior to the symmetric kernel (most noticeable around frame number 90).

3.4 Real video sequence

The final test was performed on a real video which had been broadcasted and grabbed and converted to digital video. In Fig. 8 we only show a frame together the effect of the three different methods, since the original sequence is unavailable. Each of the RGB-layers have been filtered separately. The parameter was chosen to make the best visual filtering. On the filtered images Fig. 8(b), 8(c) and 8(d) we notice that the video kernels appear most noise free.

4. CONCLUSION

In this paper we have extended the steering kernel to be used in video with added flexibility. The kernel has been altered progressively to handle the difficulties concerning scene shift. This results in the introduction of a constraint in the temporal direction and the introduction of asymmetric kernels. The proposed kernels improve the performance with respect to minimizing the RMSE compared to the original steering kernel. Further work with optimizing the choice of parameters needs to be performed. Finally it will be interesting to see how the obtained RMSE results correlates with the perceived video quality.

References

- [1] Takeda, H., Farsiu, S., and Milanfar, P., “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing* **16**(2), 349–366 (2007).
- [2] Tomasi, C. and Manduchi, R., “Bilateral filtering for gray and color images,” *Proc. 6th Int. Conf. Computer Vision, New Delhi, India*, 839–846 (1998).
- [3] Elad, M., “On the origin of the bilateral filter and ways to improve it,” *Image Processing, IEEE Transactions on* **11**(10), 1141–1151 (2002).
- [4] Barash, D., “Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation,” *PAMI, IEEE Transactions on* **24**(6), 844–847 (2002).
- [5] Barash, D. and Comaniciu, D., “A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift,” *Image and Vision Computing* **22**(1), 73–81 (2004).
- [6] Pham, T. and van Vliet, L., “Separable Bilateral Filtering for Fast Video Preprocessing,” *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 454–457 (2005).
- [7] Durand, F. and Dorsey, J., “Fast bilateral filtering for the display of high-dynamic-range images,” *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 257–266 (2002).
- [8] Paris, S. and Durand, F., “A fast approximation of the bilateral filter using a signal processing approach,” *Proc. of Eur. Conf. on Comp. Vision* (2006).
- [9] Fleishman, S., Drori, I., and Cohen-Or, D., “Bilateral mesh denoising,” *Int. Conf. on Comp. Graphics and Interactive Techniques*, 950–953 (2003).



(a)



(b)



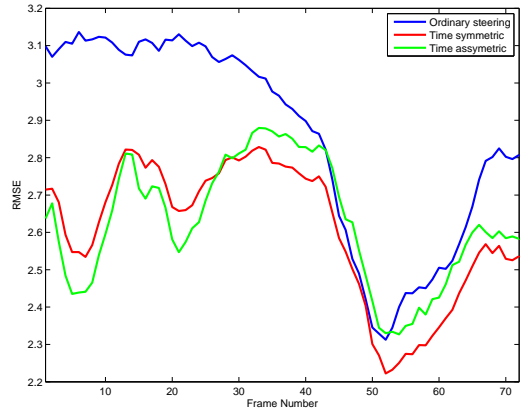
(c)



(d)



(e)



(f)

Figure 5. Detail of frame number 7 of (a) the original frame and (b) the frame with added noise with $N(0, 5)$. The resulting frame after filtering with the (c) steering kernel of size $[5 \times 5]$, (d) symmetric video steering kernel of size $[5 \times 5 \times 3]$ and (e) asymmetric video steering kernel of size $[5 \times 5 \times 3]$ and (f) the frame by frame comparison of the RMSE of the three methods.

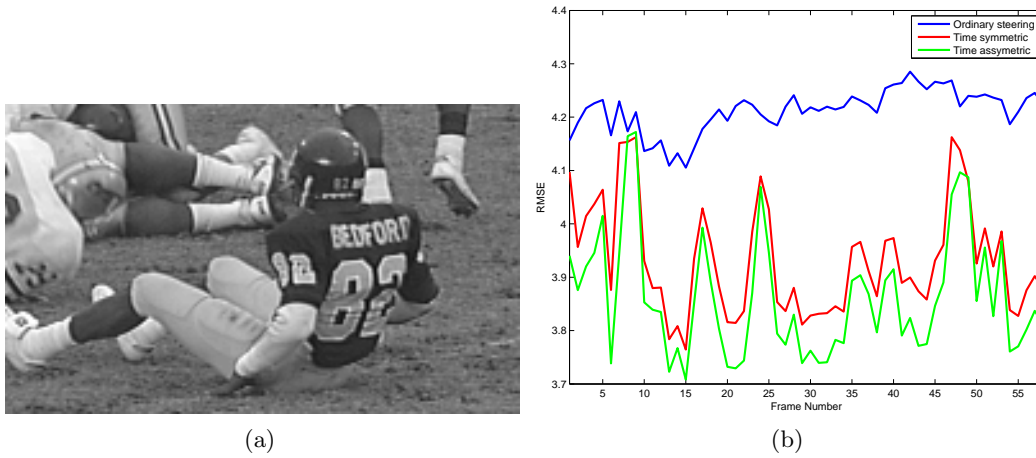


Figure 6. (a) A frame from the football sequence together with (b) the frame by frame comparison of the RMSE of the three methods on the *Football*. The kernel sizes was like the *Susie* example.

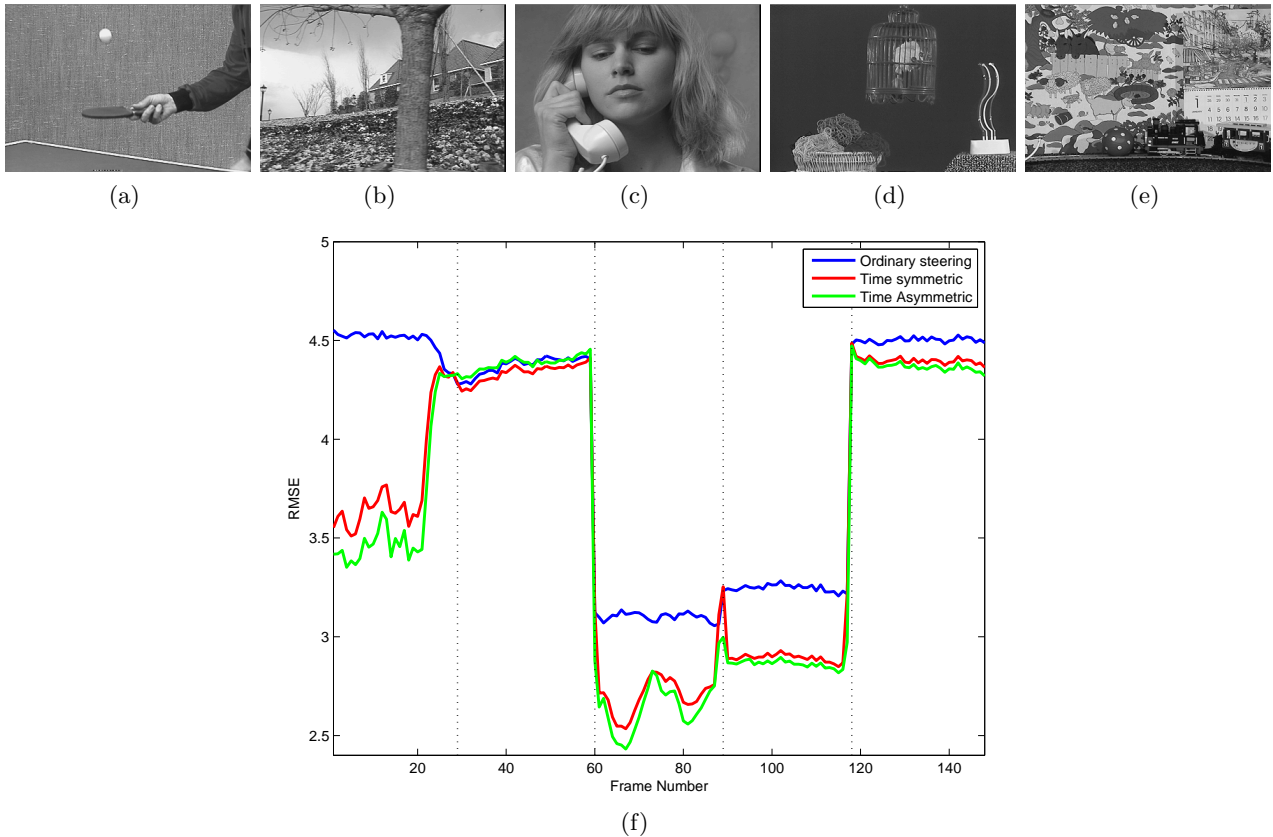


Figure 7. (a)-(e) A single frame from each of the five subsequences in this sequence. To the scene, noise was added with $N(0, 5)$ and a filtering with the steering kernel and the video steering kernels was performed. The resulting RMSE of the filtering is presented in (f). The vertical lines marks, where a scene shift occurs. The video kernels outperforms the two dimensional kernel. The time asymmetric kernel performs overall best and handles scene shift better than the time symmetric.



(a)



(b)



(c)



(d)

Figure 8. (a) A frame from a real video sequence, (b) the result of the filtered frame with the two dimensional steering kernel, (c) the time symmetric kernel and (d) time asymmetric kernel.



Engholm, R. and Karstoft H. (2009).
Optimal parameter estimation for anisotropic kernel
filtering in image processing.
Submitted to *IEEE transaction on circuits and
systems for video technology*.

Optimal parameter estimation for anisotropic kernel filtering in image processing

Rasmus Engholm and Henrik Karstoft

Abstract—In this paper we develop a method for estimating optimal parameters for filtering with isotropic and anisotropic Gaussian kernels to be used in noise reduction in video and image processing. We start by considering the ordinary isotropic Gauss filtering and extend this to edge adaptive kernels. We compare the performance of the proposed methods of estimating optimal parameters with empirically found optimal parameters. The comparison of the performance on noise reduction is based on Peak Signal to Noise Ratio (PSNR) together with the visual result of the filtering. It is concluded that our method of calculating optimal parameters performs as well as the empirically found optimal parameters. Finally a subsampling technique is proposed which increases the speed of the method based on adaptive kernels.

Index Terms—Anisotropic kernel filtering, optimal parameters, noise reduction.

I. INTRODUCTION

Much work within image and video processing focuses on analog noise reduction. Elaborate schemes have been developed that perform very well. Recent advances in adaptive nonlinear filtering techniques aim at combining a good overall noise reduction with respect to mean squared error with edge preserving abilities that are very important for a good perceptual image quality [1]. Anisotropic diffusion [2] and bilateral filtering [3] and more recently, steering kernels [4], [5] are examples of such methods. However these techniques require parameter tuning to perform well. This is often done empirically, but in order for the algorithms to work in a practical situation, e.g. in noise reduction of an incoming video stream, the parameters have to be estimated and adjusted in an automatic way. In reality the omission of this part of the process leads in many situations to the use of a single static parameter, thereby not utilizing the full potential of these algorithms.

In this paper we derive a method to automatically determine a parameter in the proximity of the optimal parameter used in ordinary isotropic filtering, e.g. Gauss filtering, based on a simple model of the signal and the noise. We then carry this result over to be used in the directional filtering algorithm based on steering kernels,

introduced in [4]. In order to increase the performance a simple sampling scheme is presented for the adaptive filtering.

Finally we compare the performance of the model estimated parameters with that of empirically found optimal parameters. We focus on a realistic setting with respect to noise levels, viz. PSNR levels of 20-40 dB, and the real time aspect of filtering on a video stream of frames, thereby disregarding iterative schemes as suggested in [4].

II. A MODEL OF IMAGE RESTORATION AND DEGRADATION

Let $f(n, m)$, $n = 1, \dots, N, m = 1, \dots, M$ be a 2 dimensional signal and η the noise with $\eta \sim N(0, I\sigma_\eta^2)$ and i.i.d. We define:

$$wg(n, m) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} w_t(k, l)g(n-k, m-l) = w_t * g,$$

where

$$g(n, m) = f(n, m) + \eta(n, m)$$

and the Gaussian filter kernel is defined as

$$w_t(k, l) = \frac{1}{4\pi t} \exp^{-\frac{(k^2 + l^2)}{4t}}, \quad (1)$$

that is g is the noisy signal where f is the signal component and η the noise component. We define the filtered signal wg as the discrete convolution of g with a Gaussian kernel w_t , see Figure 1. The parameter t controls the width of the bell shaped kernel.

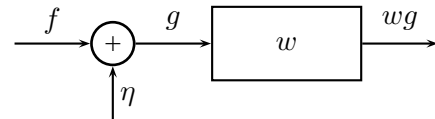


Fig. 1. Illustration of the signal path, with f being the clean signal, η the additive noise resulting in the noisy signal, g . The filtering is done with w , resulting in gw .

On average the estimate of the noisy signal g should be the signal f , since the noise is expected to have zero mean, the variance of the noisy signal is equal to the

variance of the noise. The estimate of the filtered noisy signal is then

$$\begin{aligned}\mathbb{E}[wg(n, m)] &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} w_t(k, l) f(n - k, m - l) \\ &= w_t * f.\end{aligned}$$

We now define the image degradation as the estimate of the squared difference between the filtered noisy signal and the true signal in N observations in a block, i.e. the mean squared error (MSE):

$$\begin{aligned}e &= \mathbb{E}\left[\sum_{n=1}^N (wg(n) - f(n))^2\right] \\ &= \sum_{n=1}^N \mathbb{E}[(wg(n) - wf(n))^2 + (wf(n) - f(n))^2 \\ &\quad + 2(wg(n) - wf(n))(wf(n) - f(n))] \\ &= \sum_{n=1}^N \mathbb{E}[(wg(n) - \mathbb{E}[wg(n)])^2] \\ &\quad + \sum_{n=1}^N (wf(n) - f(n))^2.\end{aligned}\quad (2)$$

We see that the MSE is the sum of two terms, where the first only depends on the variance of the noise, and the other only stems from the signal. The latter is the squared difference between the filtered signal and the signal. We want to minimize the residual e , which is the trade-off between how much noise can we eliminate (the first term), while not destroying the signal (the second term).

III. ISOTROPIC GAUSSIAN FILTERING

The variance of the filtered noisy signal, wg , is then, for the two dimensional Gaussian filtering, equal to:

$$\begin{aligned}\sigma_{wg}^2 &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} (w_t(k, l))^2 \sigma_{\eta}^2 \\ &\simeq \frac{1}{8\pi t} \sigma_{\eta}^2.\end{aligned}\quad (3)$$

The relationship between filtering with a Gaussian kernel as the solution to the heat equation is well known and used for simple deblurring when the amount of blurring is limited [6], [7]. If the diffusion process is determined by

$$\Delta u = \frac{\partial u}{\partial t}, \quad (4)$$

then in a two dimensional setting let $u(x, y) = f(x, y)$ to time $t = 0$. After time t the diffusion of u is equivalent

to folding f with the Gaussian kernel, h ,

$$h(x, y, t) = \frac{1}{4\pi t} \exp^{-(x^2+y^2)/4t}. \quad (5)$$

Using this we turn towards the Gaussian filtering of f , $w_t f$. We assume that for small t , that is for small amount of blurring, we can approximate the blurring with the first two terms of the Taylor series. We then get,

$$w_t f \simeq f + t \left. \frac{\partial w_t f}{\partial t} \right|_{t=0}. \quad (6)$$

Using the heat equation, 4 can be expressed as

$$w_t f \simeq f + t \Delta f. \quad (7)$$

That is, we approximate the blurring of the original signal with the sum of the signal and the Laplacian of the signal scaled by the t parameter.

A. Approximating the optimal t

Returning to the error estimate, we now have an estimation of the noise reduction and an approximation of the signal distortion, $w_t f$. If we insert these results into the expression of e , we find

$$\begin{aligned}e &\simeq \frac{N}{8\pi t} \sigma_{\eta}^2 + t^2 \|\Delta f\|_F^2 \\ &= \frac{N \sigma_{\eta}^2}{8\pi} t^{-1} + \|\Delta f\|_F^2 t^2,\end{aligned}\quad (8)$$

where $\|\Delta f\|_F$ is the squared value of all N values of the Laplacian or the Frobenius norm of the Laplacian. Since we wish to minimize the error, we find the minimum by determining the first derivative,

$$\frac{\partial e}{\partial t} = \frac{-N \sigma_{\eta}^2}{8\pi} t^{-2} + \|\Delta f\|_F^2 2t, \quad (9)$$

and setting this to zero gives us the minimum at

$$t = \left(\frac{1}{\|\Delta f\|_F^2} \frac{N}{16\pi} \sigma_{\eta}^2 \right)^{1/3}. \quad (10)$$

This expression is easy to calculate. However, since the method is applied to images, where the original signal f is unknown, we need to estimate the Laplacian of the signal. Since we require an estimate of the noise, we use that $\|\Delta g\|_F^2 = \|\Delta f\|_F^2 + \|\Delta \eta\|_F^2$. This is only true if f and η are orthogonal, but since η is normal distributed with zero mean this is the case.

We observe that a relatively large Laplacian of f results in a relatively small t , i.e. a smaller degree of blurring. This is not surprising since a large Laplacian implies a lot of edges and blurring the edges results in a large MSE around the edges.

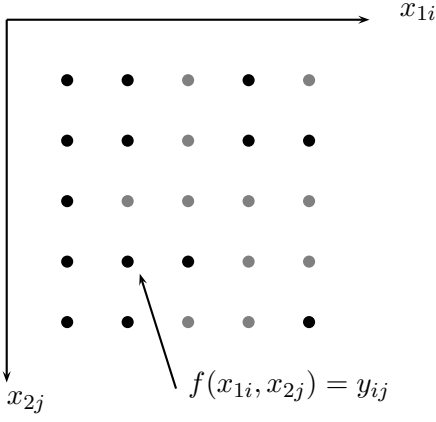


Fig. 2. The naming of the pixels

IV. ANISOTROPIC GAUSSIAN FILTERING

Having found an expression for the optimal regular Gauss filtering with an isotropic kernel, we have not taken the explicit preservation of edges into account. Using a recently proposed edge adaptive filter, the steering kernel, we now wish to find the optimal parameter for this filtering method.

A. Steering kernels

The intuition behind the steering kernel, introduced in [4], is to directly use the gradient information to construct the kernel. This means that the actual kernel will be a deformation of an ordinary Gaussian kernel (scaled, rotated, skewed). We should here note that the original formulation suggests that the kernel is used iteratively. However the optimal number of iterations remains unclear, why we here will only consider it as a one pass filter. The steering kernels are defined using the gradient information in a principal component analysis, to reduce all the gradients in a local window to the dominant orientation in that window. A kernel is then calculated based on this orientation giving weights to the observations in the window such that smoothing is performed mainly along edges rather than across edges.

We let x_0 be the pixel in a local window at which we want to estimate the gray value. The sites of the observations are indexed in a two dimensional grid (x_{1i}, x_{2j}) , $i = 1, \dots, n$, $j = 1, \dots, m$, representing the pixels in the local window. To each pair (x_{1i}, x_{2j}) we have associated a single observation y_{ij} , giving the grey level value at this site.

We consider the local window and define the gradient as

$$(\nabla y)(x_{1i}, x_{2j}) = \left[\frac{\partial y}{\partial x_1}(x_{1i}, x_{2j}), \frac{\partial y}{\partial x_2}(x_{1i}, x_{2j}) \right].$$

We are interested in finding the dominant orientation of the structure in the local window of the image, so that we do not smooth across that direction. On average we can consider the gradients to be orthogonal to the dominant orientation. We perform a PCA analysis on the gradients in the local window to find the dominant edge orientation. The gradients of the local window is then reordered as an $nm \times 2$ matrix:

$$G = \begin{bmatrix} (\nabla y)(x_{11}, x_{21}) \\ (\nabla y)(x_{11}, x_{22}) \\ \vdots \\ (\nabla y)(x_{11}, x_{2m}) \\ (\nabla y)(x_{12}, x_{21}) \\ \vdots \\ (\nabla y)(x_{1n}, x_{2m}) \end{bmatrix},$$

where each row specifies first the gradient in the horizontal and then in the vertical direction. The steering kernel uses a simple estimate of the covariance structure of the gradients obtained by

$$\begin{aligned} C_{x_0} &= \frac{1}{nm} \sum_{i,j} \left[\begin{array}{cc} \left(\frac{\partial y}{\partial x_1} \right)^2 & \frac{\partial y}{\partial x_1} \frac{\partial y}{\partial x_2} \\ \frac{\partial y}{\partial x_1} \frac{\partial y}{\partial x_2} & \left(\frac{\partial y}{\partial x_2} \right)^2 \end{array} \right] \Big|_{(x_{1i}, x_{2j})} \\ &= \frac{1}{nm} G^T G. \end{aligned}$$

This symmetric structure, C_{x_0} , of the gradients in a local window is then used to form the steering kernel. The objective is to spread the kernel parallel to the edges, so the edges remain unblurred. Secondly if the gradients in general are small we would assume that the differences originates from noise and the needed smoothing thus requires a relatively flat kernel. On the other hand if the gradients in general are large we will assume that it originates from structure and less smoothing is needed which calls for a steep kernel. Considering

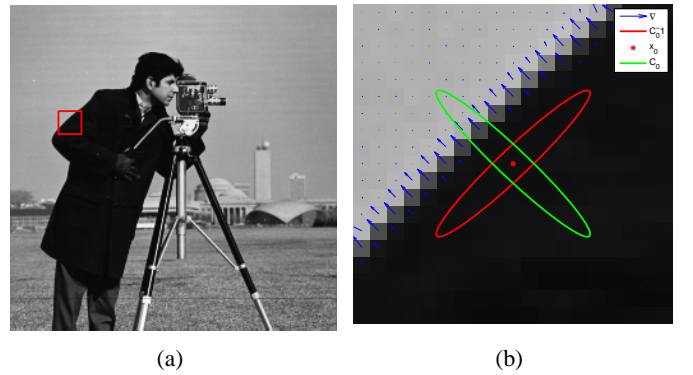


Fig. 3. (a) Image with a red square that indicates the neighborhood being examined. (b) Contour ellipses of C_{x_0} and $C_{x_0}^{-1}$ superimposed onto the current neighborhood together with the estimated gradients.

the eigenvectors of C_{x_0} , the first eigenvector is in the direction of largest variance, i.e. in the direction of the on average largest gradient. This structure is exactly the opposite of what we look for. We wish to rotate the structure such that the eigenvector corresponding to the largest eigenvalue is orthogonal to the main gradient direction. This can be accomplished by inverting C_{x_0} . Since the inverse of a symmetric matrix, has the same eigenvectors but reciprocal eigenvalues, we obtain the wanted structure, see Fig. 3.

Using this structure to construct a Gaussian kernel we arrive at the steering kernel

$$w_s(x_0, x_w) = K \exp \left(-\frac{(x_0 - x_w)^T (C_{x_0}) (x_0 - x_w)}{4t_s} \right) \quad (11)$$

with the normalizing constant K given by,

$$K = \frac{1}{4\pi t_s |C_{x_0}|^{-1/2}}.$$

In 11, $w_s(x_0, x_w)$ is then the weight associated to pixel x_w in the local window, C_{x_0} is the structure of the gradients for the local window centered around x_0 and t_s is the parameter governing the basic level of smoothing i.e. the steepness of the Gaussian kernel before it is deformed by C_{x_0} . We will in the following denote the eigenvalues of C_{x_0} by λ_1 and λ_2 with the first being the largest eigenvalue and corresponding eigenvectors v_1 and v_2 .

The important question of how t_s is chosen in a suitable way to minimize the MSE has to the best of our knowledge not been investigated before.

B. Approximating optimal t_s

Since we are no longer working with a constant kernel size for the entire image, we have to find a strategy for determining the parameter t_s of the steering kernel. The objective is still to minimize the MSE. We use a bottom-up approach determining the optimal t_s locally for all pixels. We then use a suitable statistic to find an estimate of a global parameter.

By using anisotropic kernels we first calculate the variance of the filtered noisy signal in the same way as in the isotropic case. We find

$$\begin{aligned} \sigma_{wg}^2 &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} (w_s(k, l))^2 \sigma_{\eta}^2 \\ &\simeq \frac{1}{8\pi t \sqrt{\lambda_1 \lambda_2}} \sigma_{\eta}^2. \end{aligned} \quad (12)$$

Not surprisingly we get a similar result as in the isotropic case, only scaled with $\sqrt{\lambda_1 \lambda_2}$. For filtering of the signal

f with the steering kernel, we again use the heat equation. However, we have to modify the Laplacian term, such that it is able to model the effect of the steering kernel i.e. less blurring over edges. We will denote this anisotropic Laplacian Δ_C . Assuming that the main axis of our Gaussian kernel is along the axis of the coordinate system, we find that

$$\frac{\partial}{\partial t} w(x, y, t) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) w(x, y, t).$$

Expressing $x(u, v)$ and $y(u, v)$ we have that

$$\begin{pmatrix} x \\ y \end{pmatrix} = C_{x_0} \begin{pmatrix} u \\ v \end{pmatrix},$$

where C_{x_0} is the 2×2 matrix that performs the rotation and the scaling of the kernel. It can be shown that:

$$\frac{\partial^2 w}{\partial (u, v)} = C_{x_0}^T \frac{\partial^2 w}{\partial (x, y)} C_{x_0}.$$

which leads to

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) w = \text{Trace}(C_{X_0}^{(T)-1} \frac{\partial w}{\partial (u, v)} C_{X_0}^{-1}).$$

Since C_{x_0} is symmetric it can be diagonalized as $C_{x_0} = V \Lambda V^{-1}$, where V contains the eigenvectors as columns and Λ the eigenvalues in the diagonal. We find

$$\begin{aligned} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) w &= \text{Trace} \left(V \Lambda^{-1} V^T \frac{\partial w}{\partial (u, v)} V \Lambda^{-1} V^T \right) \\ &= \text{Trace} \left(\Lambda^{-2} V^T \frac{\partial w}{\partial (u, v)} V \right) \\ &= \lambda_1^{-2} v_1^T \frac{\partial w}{\partial (u, v)} v_1 + \lambda_2^{-2} v_2^T \frac{\partial w}{\partial (u, v)} v_2. \end{aligned}$$

This is the anisotropic Laplacian to be used in the estimate of the error

$$\Delta_C f = \left(\lambda_1^{-2} v_1^T \frac{\partial w}{\partial (u, v)} v_1 + \lambda_2^{-2} v_2^T \frac{\partial w}{\partial (u, v)} v_2 \right). \quad (13)$$

Using 12 and 13 we obtain an estimate of the error and we find the minima as in the isotropic case,

$$t = \left(\frac{1}{\|\Delta_C f\|_F^2} \frac{1}{16\pi \sqrt{\lambda_1 \lambda_2}} \sigma_{\eta}^2 \right)^{1/3}.$$

As mentioned above the parameter t_s is calculated for each pixel. We chose the single optimal t_s to be the median. The median was preferred to the mean because a few of the t_s 's for numerical reasons were very large.



Fig. 4. (a) The original image together with (b) the image with added noise of 40 dB, (c) 30 dB and (d) 20 dB, respectively.

C. Improving performance with subsampling

Calculating optimal t_s values for all pixels is a tedious task so reducing the number of calculations is of course always relevant in a real time setting like video playback. To reduce the computational load we tested the change in performance when subsampling the pixels and basing the global optimal t_s on this sample. We compared the full image with respectively 50, 15, 5, 1 % of randomly selected pixels from the image. Figure 5 shows the

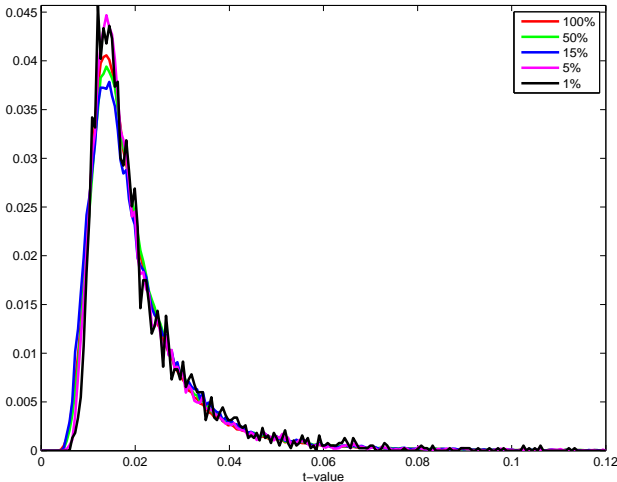


Fig. 5. Distributions of the t_s 's for image number 1 (one of the images on the Kodak image CD shown in Figure 6) with an input noise level of 20 dB, with respectively no subsampling and subsampling of 50, 15, 5 and 1 % of the original image pixels.

distributions of the t_s 's when subsampling. The changes of the distribution are very small until the observations are reduced to 1%. Thus by pixel subsampling we can decrease the calculation time significantly.

V. RESULTS

We have used a setting for realistic noise levels ranging from a PSNR of 20-40 dB on images normalized to values between 0 and 1. The noise is additive Gaussian noise. The filtering is performed in the luminance channel. Figure 4 shows the impact of noise at three different levels. A PSNR of 40 is hardly noticeable while a PSNR of 20 causes severe damage due to noise.

A data set of 24 very different images was used, see Figure 6, covering a broad range of motives of e.g. nature, artificial constructions, humans, etc. All images were 768×512 pixels.

A. Isotropic Gaussian filtering

For each image, we added Gaussian noise and filtered it. We found the optimal parameter manually for each image at each noise level. These parameters were compared with the optimal parameters obtained from the model. An example is shown in Figure 7. To illustrate we have plotted the progression of the error as a function of the parameter together with error predicted by the model. Furthermore we have added the noise and signal error part as a function of parameter.

Figure 8 shows the resulting PSNR of filtering with the empirically found parameters plotted against the result of filtering with the optimal parameters predicted by the model. The performance is almost identical for high PSNR input. However, for the input with PSNR of 20 dB, we notice that the empirically found parameters performs better. This can be explained by looking at Figure 9, where the empirically found parameters are plotted against the model predicted parameters. Here we observe that the model predicted parameters starts



Fig. 6. The images from the Kodak image CD, covering a broad range of objects.

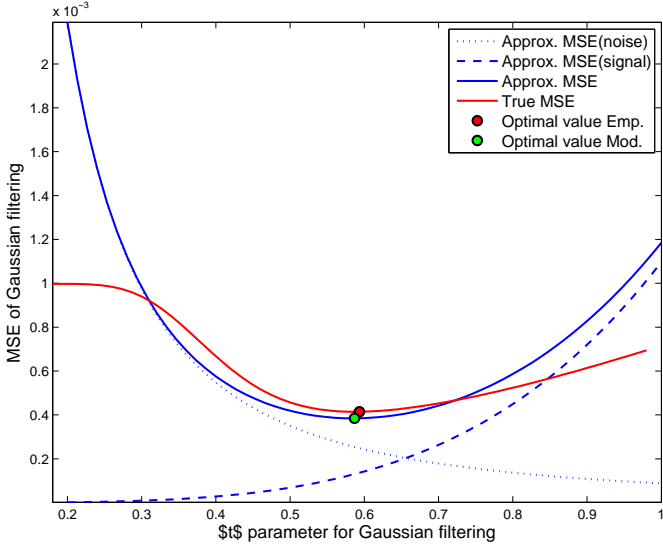


Fig. 7. Resulting MSE of filtering an image with added noise at a PSNR of 25. As the curve drops, the reduction of noise dominates the blurring of the original image, at the optimal value, the rate of the degradation equals that of the noise reduction. After this point the degradation of the image dominates.

to diverge from the empirically found as the PSNR decreases. This is however not surprising, since we assumed the model to be most precise for small degrees of blurring. Finally the average performance of the model predicted parameters over all images is compared with that of the empirically found parameters as shown in table VI. The performance of the model predicted parameters is close to the empirically found.

B. Anisotropic Gaussian filtering

The same procedure as above was followed. Again we compared the empirically results with the theoretical results obtained from the model. Figure 10 shows the

PSNR of the empirically found parameters plotted against the resulting PSNR of filtering with the optimal parameters predicted by the model. Here the results are again very close to the empirically found also at the PSNR level of 20 dB. In Figure 11 the empirically found parameters are plotted against the model predicted parameters. In contrast to the isotropic case above the model predicted parameters does not differ from the empirically found at the low PSNR levels. Finally the average performance of the model predicted parameters over all images is compared with that of the empirically found parameters as shown in table VI. We notice that the model predicted parameters performs almost identical to the empirically found parameters.

Figure 12 shows a detail from image 23 with and without noise with a PSNR of 30 dB, and the result of the optimal filtering with both model predicted and empirically found parameter for both isotropic and anisotropic filtering. No significant difference is seen between the performance of the model predicted parameters and the empirically found. On a side note we notice the superiority of the anisotropic Gauss filtering compared to the ordinary isotropic Gauss filtering.

VI. CONCLUSION AND FUTURE WORKS

In this paper we have proposed a simple model for noise degradation and restoration with filtering. This is used for estimating optimal parameters for kernel filtering. We have extended the to the setting of adaptive kernels. For this purpose we have introduced an anisotropic Laplacian operator to model the degradation of edges.

The results of the methods for ordinary Gauss filtering performs satisfactory to a certain noise level, PSNR levels between 25-40. The model predicted parameters

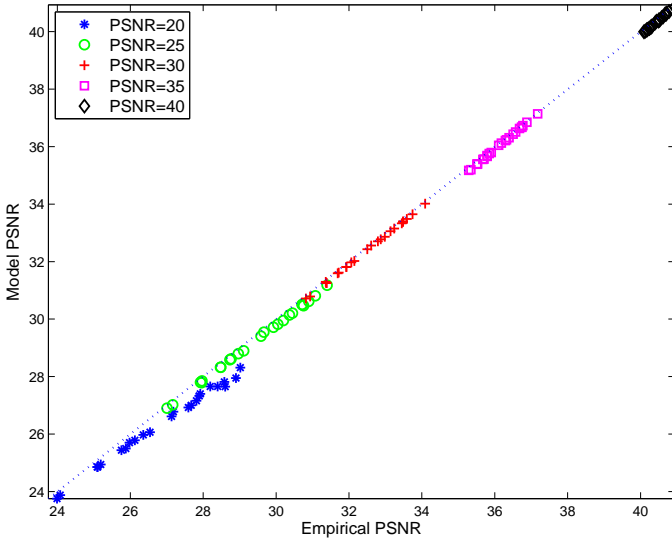


Fig. 8. The PSNR of filtering with the empirical found optimal parameter, plotted against the result of filtering with the optimal parameters predicted by the model.

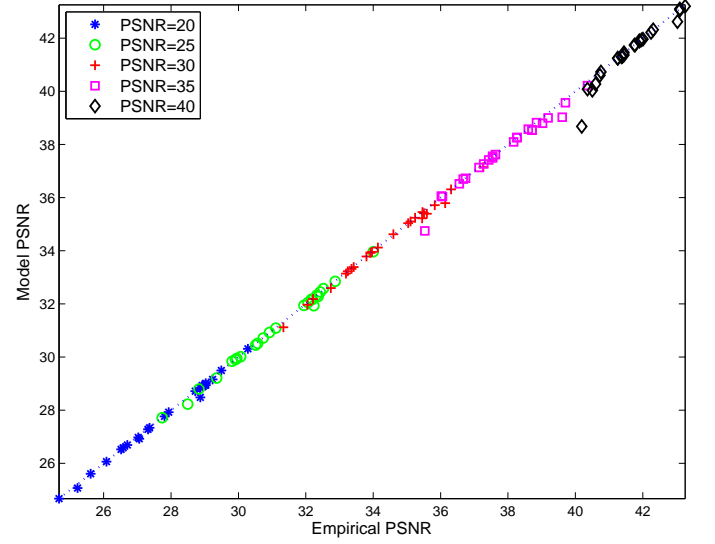


Fig. 10. The PSNR of the empirical found parameters but plotted against the result of filtering with the optimal parameters predicted by the model.

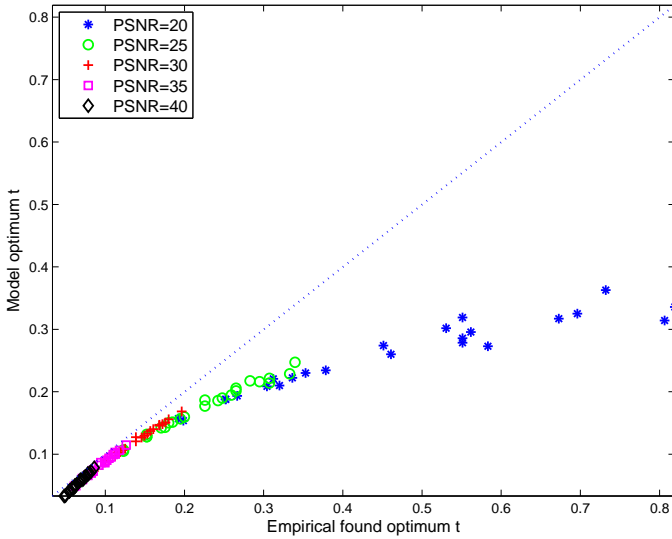


Fig. 9. the empirically found parameter plotted against the model predicted parameter.

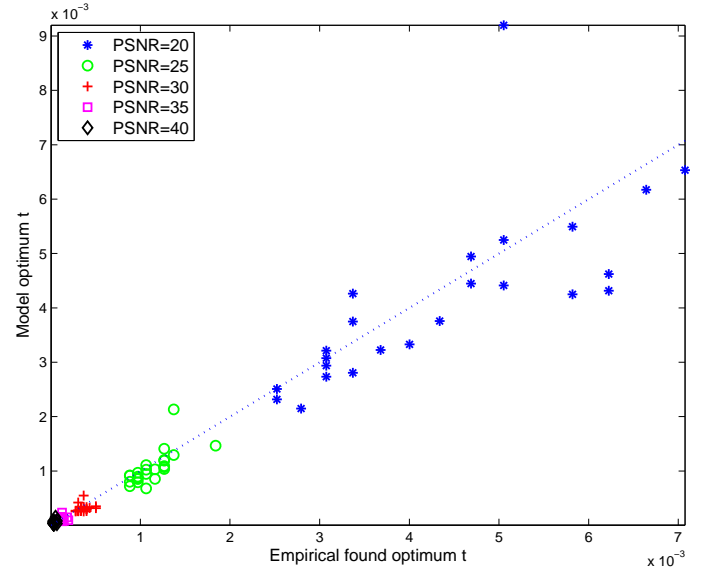


Fig. 11. the empirically found parameter plotted against the model predicted parameter.

show a very high correlation with the empirically found parameters. At very high noise levels (below a PSNR of 25 dB) the predicted values are less correlated.

The results obtained in the anisotropic case are very similar to the ones obtained in the isotropic case. The adaptivity makes the improvement in PSNR levels more pronounced. Furthermore, the visual quality using anisotropic kernel is much more convincing than the value expressed in the PSNR metric. Accordingly, the method is of great relevance to video processing. The performance of both methods is of course very dependent on a reliable estimate of the noise.

The results also show that frames from videos with the same noise level, may require very different parameter settings in order to minimize the error. An important point is therefore that it is not sufficient to only use the noise level as input to these noise removal algorithms, as is often practiced. Features from the actual frame or image should also be included in some way.

REFERENCES

- [1] G.M. Johnson and M.D. Fairchild. Sharpness rules. In *8th Color Imaging Conference*, pages 24–30, 2000.

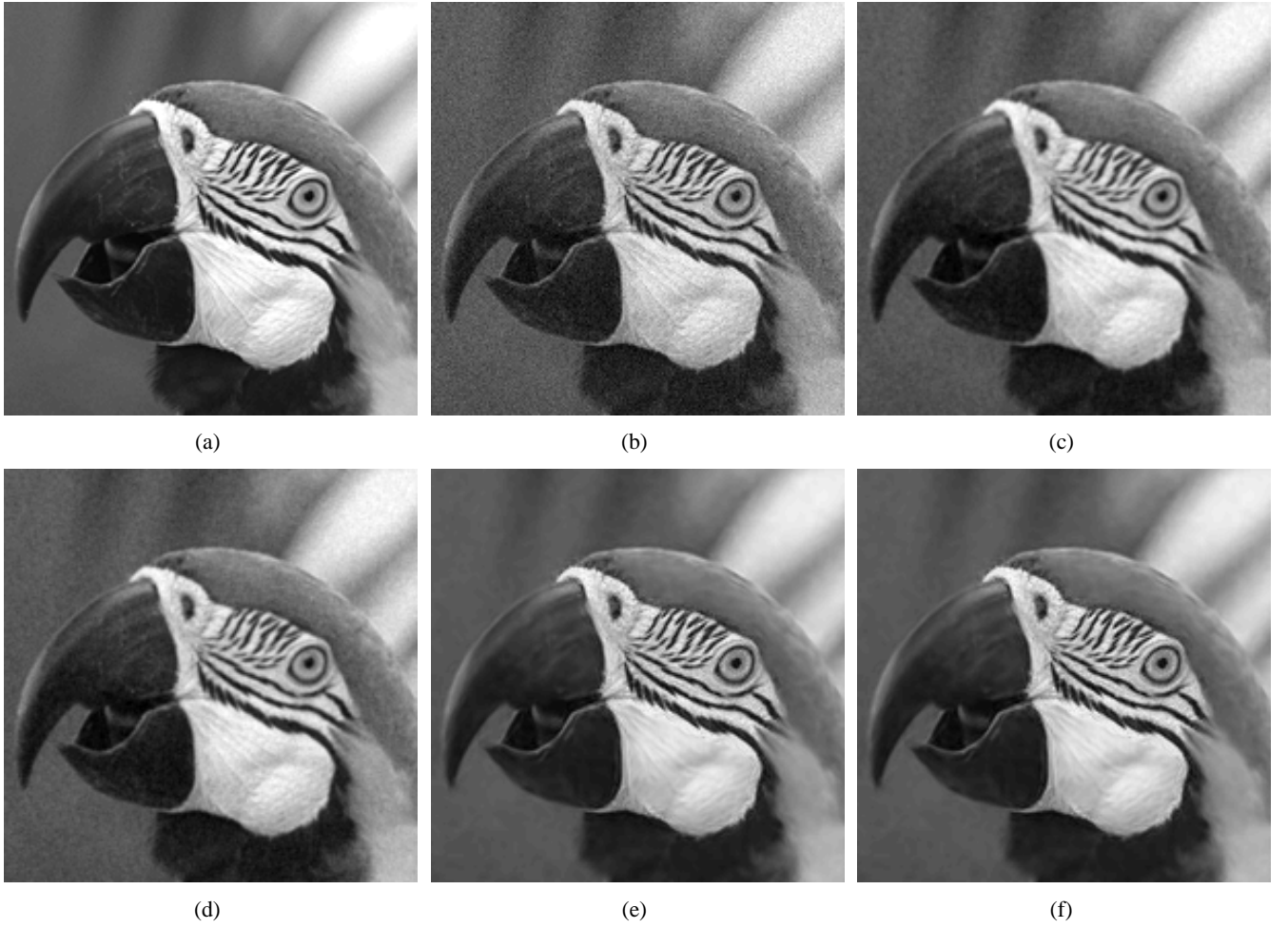


Fig. 12. (a) Detail from original image, (b) and with added noise of 30 dB, (c), Gaussian filtered with empirical found parameter, (d), and with the model predicted parameter. (e) Shows the result of the empirical found parameter with the anisotropic filtering and, (f) with the model predicted parameter.

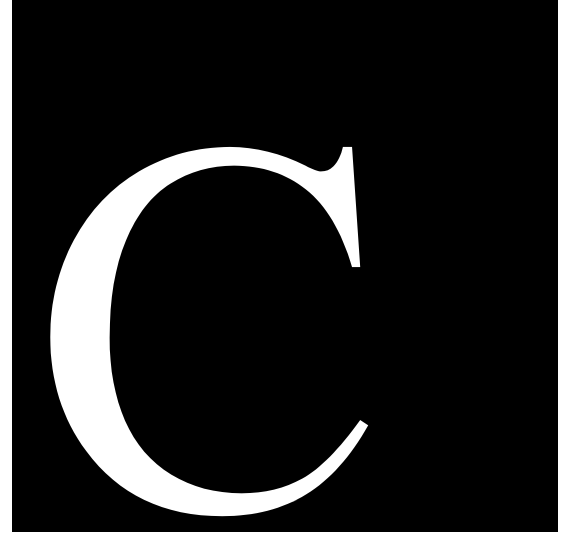
TABLE I
COMPARISON OF PERFORMANCE OF EMPIRICAL AND MODEL
PARAMETERS FOR ISOTROPIC FILTER

Input PSNR	Mean Output PSNR		
	Empirically	Model	Difference
20	26.88	26.37	0.51
25	29.43	29.24	0.19
30	32.47	32.37	0.10
35	36.14	36.05	0.09
40	40.44	40.33	0.11

TABLE II
COMPARISON OF PERFORMANCE OF EMPIRICAL AND MODEL
PARAMETERS FOR ANISOTROPIC FILTER

Input PSNR	Mean Output PSNR		
	Empirically	Model	Difference
20	27.76	27.71	0.05
25	31.05	31.00	0.05
30	34.37	34.30	0.07
35	37.90	37.79	0.11
40	41.65	41.50	0.15

- [2] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
- [3] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proc. 6th Int. Conf. Computer Vision, New Delhi, India*, pages 839–846, 1998.
- [4] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, 2007.
- [5] R. Engholm, H. Karstoft, and E.B.V. Jensen. Adaptive kernel filtering used in video processing (Proceedings Paper). *Visual Communications and Image Processing 2009*, 2009.
- [6] D. Gabor. Information theory in electron microscopy. *Lab Invest*, 14:801–7, 1965.
- [7] M. Lindenbaum, M. Fischer, and A. Bruckstein. On Gabor's contribution to image enhancement. *Pattern Recognition*, 27(1): 1–8, 1994.
- [8] A. Buades, B. Coll, and J. Morel. A Non-Local Algorithm for Image Denoising. In *IEEE Computer society conference on computer vision and pattern recognition*, volume 2, page 60. IEEE Computer Society; 1999, 2005.



Engholm, R., Karstoft H. and Bech, S. (2009).
Image sharpness assessment based on singular value
decomposition.
Manuscript.

Image sharpness assessment based on singular value decomposition

Rasmus Engholm^{a,b,c,*}, Henrik Karstoft^{b,c}, Søren Bech^c

^aDepartment of Mathematical Sciences, Aarhus University, Ny Munkegade 118, DK-8000 Aarhus C, Denmark

^bEngineering College of Aarhus, Dalgas Avenue 2 DK-8000 Aarhus, Denmark

^cBang & Olufsen a/s, Peter Bangs Vej 15, DK-7600 Struer, Denmark

Abstract

A precise estimate of the perceived sharpness is important when evaluating the preprocessing steps needed for improving image and video quality. In this paper we present a sharpness metric based on Singular Value Decomposition (SVD) of an image. A local faster variant is then derived inspired by the human visual system that divides the image into blocks. We test the proposed metric against results from a perceptual study where 21 subjects rated their subjective perception of sharpness on images with various degree of sharpness with and without color.

We find a very high correlation between the proposed two metrics and the subjective ratings. The performance of the proposed metrics is superior to the performance of two other state-of-the-art sharpness metrics.

Keywords: Sharpness, metric, SVD, Perception, Luminance.

1. Introduction

Perceived sharpness plays a very important role when observers form their opinion about the quality of an image or a video sequence [1]. Digital video is a very inhomogeneous quantity where you, within the same program, can experience high resolution quality video and very lossy and low resolution video in e.g. a news program. Therefore it is of utmost importance that a reliable measure of the sharpness is available when processing video and image.

Blurring of video can be introduced at the image formation stage, by being out of focus, but other steps such as compression where the DCT transformation and subsequent quantification of the resulting coefficients as used in MPEG encoding are also a source of blurring. Another source of blurring is upscaling of a frame to fit the screen resolution which also leads to interpolation and loss of sharpness.

To compensate for this, some sharpness enhancement is typically performed in a postprocessing step with various sharpness enhancement methods, before an image frame is presented on the screen. An important example is peaking which is ubiquitous in modern televisions. Sharpness enhancement is however a difficult process to control, because enhancing the sharpness too much not only amplifies the present noise but also introduces new artifacts that degrade the perceived quality. A good estimator of the initial sharpness is therefore important in order to control the amount of sharpness enhancement to be used in order to obtain the best perceived quality.

Different sharpness metrics have in recent years been proposed. These metrics can be divided into two groups: full-reference metrics e.g. [2, 3], and no-reference metrics e.g. [4–6]. Many of the full reference sharpness metrics in the literature is aimed at photography, and are mainly focused on the luminance contribution. For photography the objective often is to adjust the parameters e.g. aperture size, shutter

*Principal corresponding author

Email addresses: rasmus.engholm@gmail.com (Rasmus Engholm), hka@iha.dk (Henrik Karstoft), sbe@bang-olufsen.dk (Søren Bech)

time etc. to capture the sharpest image. This is obviously a full reference setup, where two settings on the camera can be compared and a strategy can be chosen to arrive at the sharpest image. This is very different from the situation we experience when we try to estimate the sharpness of a broadcasted sequence. Hardly ever do we here have an uncorrupted source to refer to with respect to sharpness. This is why the metric of interest for our purpose need to be a no-reference metric.

In this paper we introduce a no-reference sharpness metric based on the singular values of the singular value decomposition of the image. We show that there is a high correlation with the perceived sharpness found in a study among 21 test subjects.

The remaining part of this paper is organized as follows. In section 2 we give an overview of the theory of Singular Value Decomposition and the role it plays as a tool in image analysis. Furthermore we investigate the distribution of the singular values as the blurring increases. Section 3 describes the two proposed metrics based on SVD, while section 4 introduces the experimental setup of the perception study together with the results of the proposed metric and compares it with other state of the art metrics. Section 5 concludes with a discussion of the obtained results together with plans for future research.

2. Theory

Singular Value Decomposition (SVD) plays a vital role in numerous problems in linear algebra, e.g. in the solution of ill posed linear problems, least square problems [7], etc. In image analysis it has also found uses in many different fields e.g. watermarking [8], [9], image compression [10], noise reduction [11] and deblurring of images [12] as well as dimension reduction in face recognition [13] and similar detection type applications.

If A denotes an image with non-negative pixel values, then we can decompose A into three matrices such that $A = USV^T$, where U and V are orthogonal, satisfying $U^T U = I$ and $V^T V = I$ and S is a diagonal matrix with nonnegative elements. The columns of U and V are called respectively left and right singular vectors and are denoted u_i and v_i . The values in the diagonal of S are called the singular values and are denoted σ_i . The values in the diagonal of S are by convention ordered in decreasing order. The decomposition can therefore be formulated as

$$A = \sum_{i=1}^N \sigma_i u_i v_i^T, \quad (1)$$

where N is the smallest of the dimensions of the image.

SVD of an image can be compared to the 2-dimensional Fourier transform (FT), but while FT uses a fixed set of basis vectors built on sinus and cosinus, SVD finds the best possible basis vectors for the particular image, in the sense that the energy is put into as few components as possible. This is very similar to principal component analysis (PCA), the difference is that PCA is based on the covariance matrix and assumes a symmetric matrix. On the other hand, SVD does not impose these restrictions on the structure of the matrix, which makes it suited to be applied directly on images.

2.1. Rank reduction effect on sharpness

Matrix approximation or rank reduction can be used to compress the image [7]. By using only a subset of the k largest singular values, we can make an approximation of the image A by

$$A_{approx} = \sum_{i=1}^k \sigma_i u_i v_i^T. \quad (2)$$

Adding more components gets us closer to our image, ending with a perfect reconstruction when all singular values are included.

After rank reduction, important objects of an image may be reconstructed so they are recognizable to the viewer, even with very low rank. However, using low rank results in loss of some of the details of natural

images, and as a consequence natural images need far higher rank to regain the sharpness. In Figure 1 we show a detail from an image with 512 singular values (b), and the reconstruction using only 5 pct. (c) and 50 pct. (d) of the singular values and associated singular vectors. With 5 pct. the image is hardly recognizable, disregarding the artefacts, and at best must be considered very blurred. With 50 pct. the sharpness appears to be almost completely reconstructed. In Figure 1 we also show the reconstructed image based on the 5-50 pct. largest (e) and the 50 pct. smallest singular values (f). Here we notice that the sharpness and details are clearly present in the mid range whereas the largest 5 pct. and smallest 50 pct. do not contain any visual sharpness.

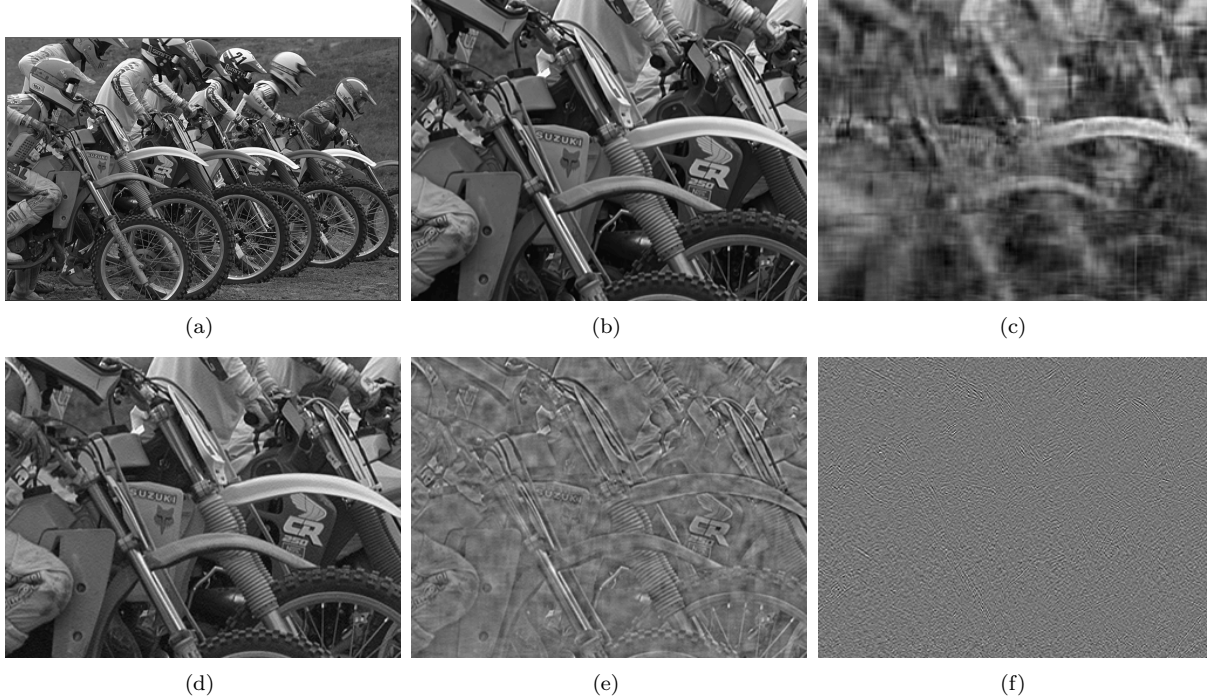


Figure 1: (a) Original 'bike' image with 512 non-zero singular values, (b) detail from 'bike' image, (c) reconstruction using the 5 pct. largest singular values and associated vectors, (d) reconstruction using the 50 pct. largest singular values, (e) reconstruction using the singular values from 5-50 pct. in magnitude, (f) reconstruction of the image using the singular values of the 50 pct. smallest singular values.

Most of the image energy is packed in the largest singular values, while the fine details and noise, which can be hard to distinguish, are associated with the smaller singular values. The image energy is therefore not a very good indicator of sharpness. If we want to use the singular values as a measure of sharpness we have to combine them in a clever way.

2.2. Blur effect on singular values

As mentioned above the image processing often introduces blurring of the image at some stage. During a blurring process for natural image we would expect to see a transition from having relatively much of the energy spread out on many singular values to having most of the energy at the largest singular values, since the largest eigenvalues typically represent low frequent content. We therefore have a decay of the rest of the singular values.

In Figure 2, we plot the singular values of an image together with the singular values of the same image folded with a Gaussian kernel with a variance of 0.8 and 2.4. The magnitude of the singular values are plotted on a logarithmic scale to moderate the dominant appearance of the largest singular value. We notice that the blurring decreases the values and gives a steeper slope.

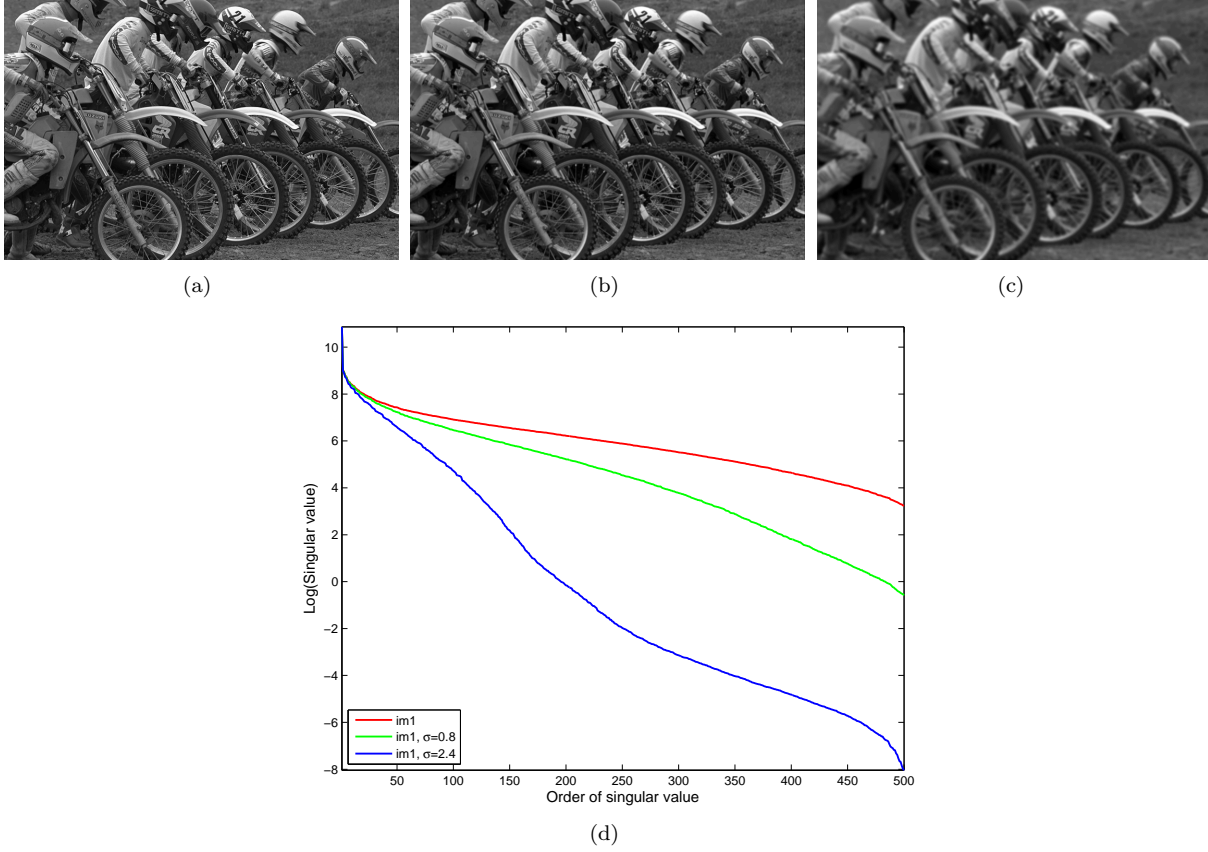


Figure 2: (a) Original unblurred image with 500 non-zero singular values, (b) the same image with moderate blurring ($\sigma = 0.8$), (c) the same image with heavy blur ($\sigma = 2.4$), and (d) the logarithmically transformed singular values plotted in descending order for the three different cases of blur.

Comparing three very different images and performing the same blurring as above, we noticed a similar tendency and observed a very similar pattern for the images, even though the content of the images appeared very different, see Figure 3. Looking at the distribution of the transformed singular values, we see that the images that are treated with the same degree of blur, have a very similar distribution and that the distributions at different degrees of blur have a different shape. The unblurred images have a single steep peak with quite high values. The moderately blurred images are not as steep and have a large proportion of the values at lower values. The heavily blurred image have a relatively flat distribution with a peak in the lowest range of the singular values.

3. Proposed sharpness metric

Using the observations above we propose a metric that aims at capturing the sharpness of the image. We extract the singular values of an image and then perform a logarithmic transformation. The logarithmic transformation is used to remove the dominance of the few largest singular values. The tendency is that sharp images have the majority of the singular values at relatively high level, whereas more blurred images have the majority of the singular values clustered together at a much lower level. A simple measure is then the average value of the log transformed singular values.

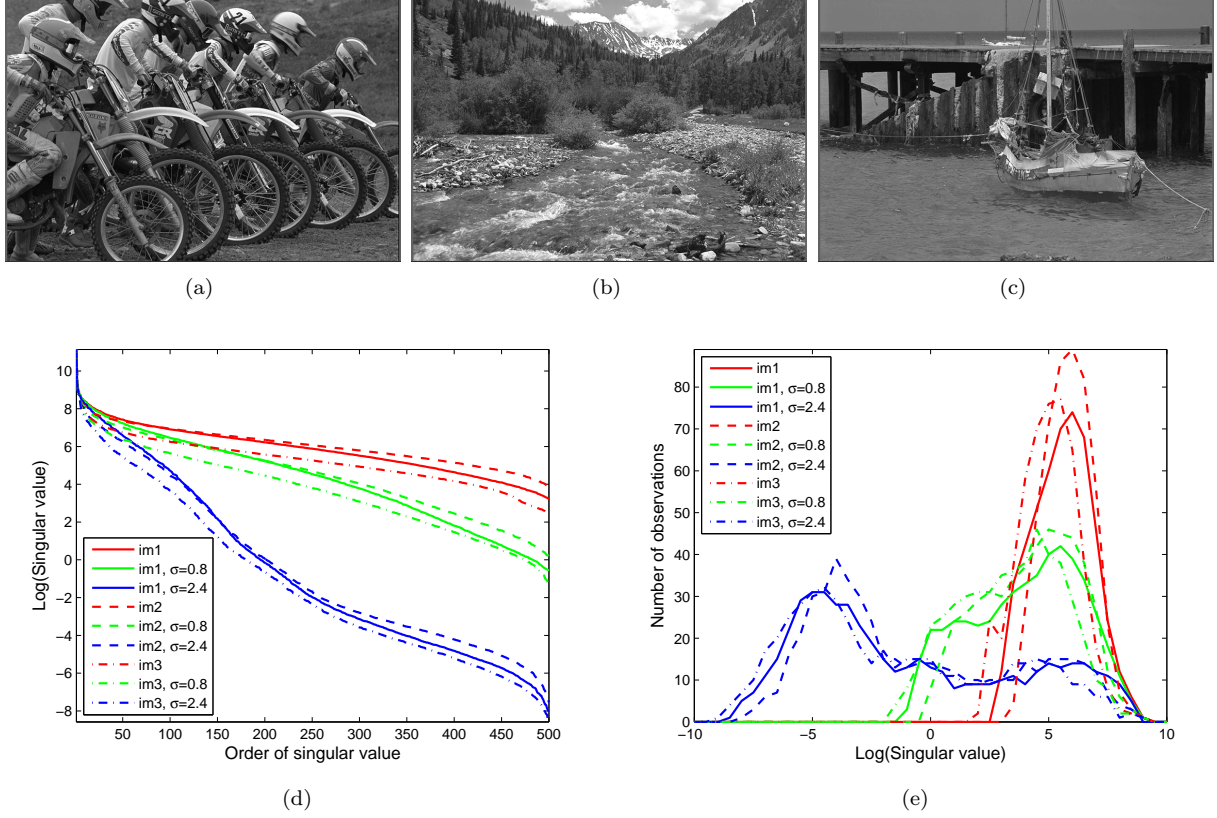


Figure 3: (a) Image with primarily manmade content, (b) image with natural content, (c) image with both natural and manmade content, (d) the logarithmically transformed singular values plotted in descending order for the three different image and three different degrees of blur, (e) a histogram of the transformed singular values from (d).

3.1. Considering a subset of the singular values

Since the sharpness can be reproduced using the 5-50 pct. largest singular values of an image, excluding the lowest 50 pct. and the largest 5 pct. of the singular values should result in a good performance of a sharpness metric. The exclusion of the 5 pct. largest values is motivated by the fact that they have very low correlation with the perceived sharpness. The smallest singular values of the image contain close to no image information and can therefore also be discarded. The metric based on the image is then

$$SVD-G = \frac{1}{U-l+1} \sum_{i=l}^U \log \sigma_i \quad (3)$$

where l denotes the index of the singular value starting at the 5th percentile and U is the index of the median.

3.2. Faster metric

Since the complexity of singular values decomposition grows as $\mathcal{O}(\min(nm^2, mn^2))$ for an $m \times n$ image, making a full SVD of the entire frame is very time consuming. Therefore we propose an alternative metric based on SVD performed on the local patches with edge content. The image is initially divided into regions of the same size as the part of human sharpness vision with the highest spatial resolution, called the foveal region. We assume that the perception of sharpness is formed here. Since the foveal region is only

approximately $\alpha = 2^\circ$, the area covered can be approximated with a square block with side length

$$P = 2 \tan \left(\frac{\alpha}{2} \frac{\pi}{180} \right) dr, \quad (4)$$

where d is the distance to the screen in cm and r the resolution of the screen in pixels/cm. For a 32" screen with Full High Definition resolution 1080×1920 we get a block size of 142×142 pixels when sitting in a distance of 150 cm from the screen. For each block we perform an edge filter e.g. Canny or similar to exclude blocks with few or no edges. This is done because we assume that regions with few or no edges do not influence the perception of sharpness. For each of the remaining blocks we conduct an SVD and calculate the metric as above in the global case. The final metric is then the average over all blocks with edge content

$$SVD-L = \frac{1}{N} \sum_{i=1}^N SVD-G_i \quad (5)$$

where N is the number of blocks with edge content and $SVD-G_i$ is the value of the global metric based on the i^{th} block. We have excluded blocks where less than 1 pct. of the pixels was edge pixels.

Figure 4 shows the performance of the two metrics on the 'bike' image as a function of the blurring parameter of a Gaussian convolution. Both metrics show monotonicity.

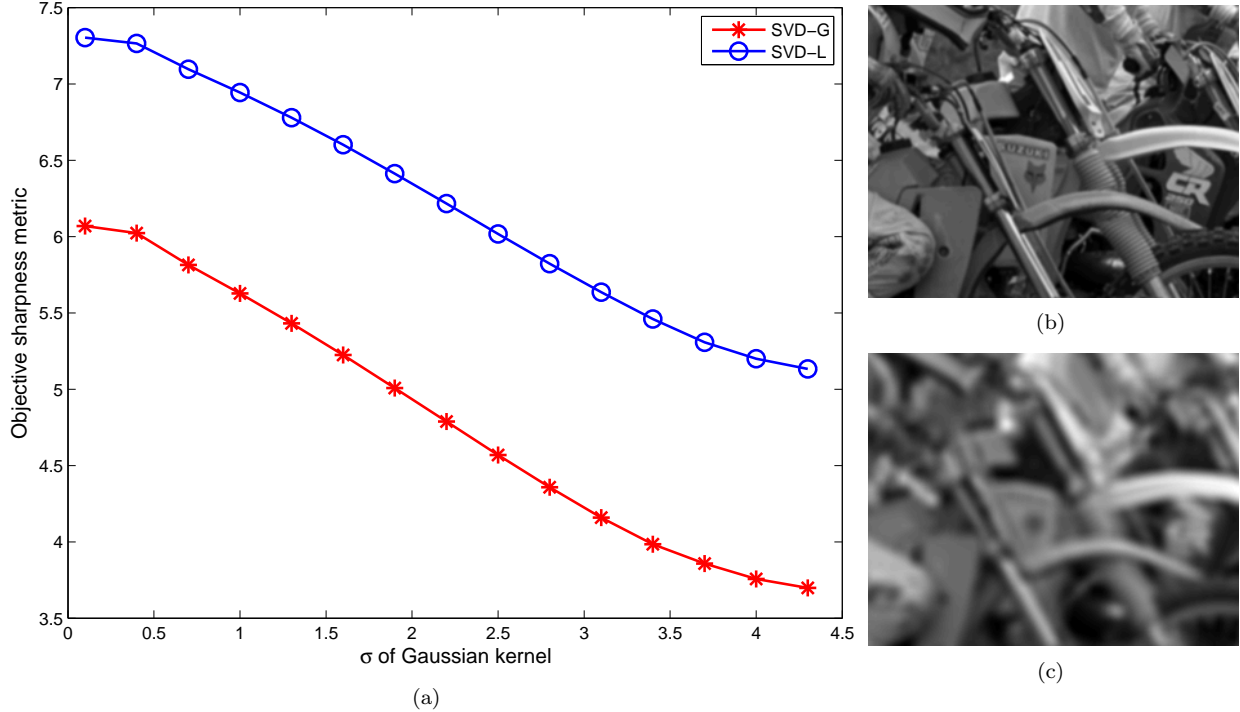


Figure 4: (a) The performance of the two proposed metrics SVD-G and SVD-L as the blurring of the 'bike' image increases. (b) Detail from the bike image filtered with light blur (a Gaussian kernel with $\sigma=1$.) (c) Detail from the bike image filtered with heavy blur (a Gaussian kernel with $\sigma=4$.)

4. Results

To validate our metric we have set up an experiment to measure the subjective perception of sharpness on a human test panel.

4.1. Methodology

Twenty one subjects were asked to rate the sharpness of the four different images shown in Figure 5. The images were distorted with different amount of blur from zero blur to images being blurred with a Gaussian kernel with STD of 0.8, 1.6 and 2.4. The material presented was images of size 1080×1920 and 720×1280 , equivalent to Full High definition and High Definition on a television. The presented images had the same pixel ratio (pixels/cm), to avoid interpolation artefacts when upscaling. All images were presented in both color and grey tones.



Figure 5: The color version of the images used in the test: (a) a running crowd, (b) a golfer, (c) mobile and calendar, and (d) a panoramic flight image over Stockholm.

For the test subjects to be able to discriminate sharpness the viewing distance was 1.5 meters from a 32 inch Beovision7 television (LCD based). The distance was chosen to make the pixel size comparable to that of the recommended ITU 510.11 standard for standard definition (SD) material [14]. The television was set to 'PC mode' to circumvent any image processing (peaking, dynamic contrast etc.). The chosen distance resulted in the fovea covering approximately 142 pixels to be used in the SVD-L metric.

The sharpness scoring of the subjects was done on a continuous scale with only labels of 'very unsharp' and 'very sharp'. This was done to avoid the usual clustering effect around labels. The slider used to report sharpness was set randomly before the presentation of each image. Before the actual experiment a short training session was made, presenting the subject to the range of the parameters in the experiment (color, blur). To avoid bias due to learning and fatigue effects we used an incomplete balanced latin square design in the experiment. For each setting the scoring was averaged over the number of subjects resulting in an Average Sharpness Rating (ASR).

To measure the correlation between the proposed metrics and the ASR, we used the suggestions of the VQEG report [15] for comparing different quality metrics. Here nonlinear regression¹ is suggested to

¹The function nls from R was used to perform the nonlinear fitting

transform the set of metrics to a set of predicted ASR values, which can be compared with the actual ASR values obtained from the subjective tests. The fitting function was:

$$ASR_p = \frac{b_1}{1 + \exp(-b_2(metric - b_3))} \quad (6)$$

where ASR_p is the predicted average sharpness rating based on the result of the proposed metric.

From the transformed results we calculate the Pearson correlation coefficient, to measure the accuracy of the metrics, Spearman correlation to measure the monotonicity of the metrics and rooted mean squared error and mean absolute error to measure the consistency of the metrics.

4.2. Performance of metric

The performance of the two proposed metrics was compared with the performance of two other state of the art sharpness metrics. One of the state of the art metrics, proposed by Marziliano et al. [3], locates edges in the luminance channel and for every edge point measures the distance between the neighboring extrema in each horizontal line. This distance is defined as the edge width, and the value of the metric is the average of all edge widths. The more blurred the image is the wider the edges. This is a commonly used metric due to simplicity and speed. The other state of the art metric used in the comparison is a more complex and novel method by Ferzli et al. [4]. A similar methodology involving edge width is used, but furthermore the notion of just noticeable blur (JNB) is used in a probability summation model. This metric also performs a subdivision of the image into blocks based on the human visual system.

The two proposed metrics are plotted against the average sharpness rating of the color images in Figure 6. The superimposed red line is the result of the non linear regression. The calculated correlations and the errors are presented in Table 1 for the proposed metrics and the state of the art metrics evaluated on the color images. In Table 2 similar results are presented for the corresponding gray level images.

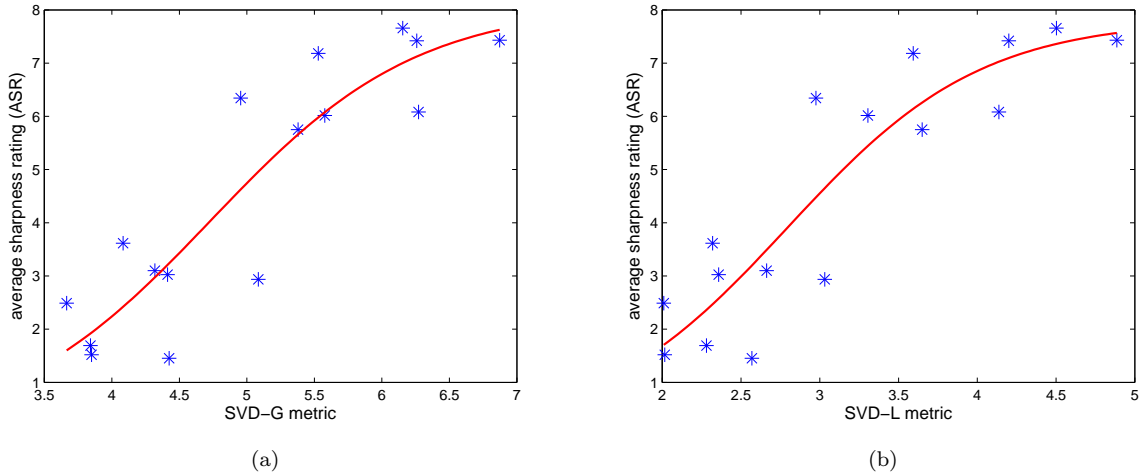


Figure 6: The average sharpness rating (ASR) plotted as a function of (a) SVD-G and (b) SVD-L. The red curve is the non linear regression fitted to the data points.

Table 1 and 2 show that the performance of the proposed metrics are quite similar and that the SVD-L performs slightly better than the SVD-G. Both metrics obtain a very high correlation around 0.9 and both metrics perform better than the Perceptual Blur and the Just Noticeable Blur metric. When comparing the performance of the metrics with and without color, we notice that all metrics perform better on gray level images. This is not surprising, since only the luminance is used to estimate the sharpness.

Table 1: Comparison of the performance of the proposed metrics, SVD-G and SVD-L, and the Perceptual Blur metric (PB) and the Just Noticeable Blur metric (JNB), relative to the Average Sharpness Rating of the test subjects. The subjective rating is here done using the the color images.

Sharpness Metrics				
	Pearson	Spearman	RMSE	MAE
SVD-G	0.897	0.803	0.996	0.758
SVD-L	0.903	0.850	0.969	0.799
PB	0.769	0.791	1.638	1.283
JNB	0.692	0.685	1.639	1.273

Table 2: Comparison of the performance of the proposed metrics, SVD-G and SVD-L, and the Perceptual Blur metric (PB) and the Just Noticeable Blur metric (JNB) relative to the Average Sharpness Rating of the test subjects. The subjective rating is here done on the gray level images.

Sharpness Metrics				
	Pearson	Spearman	RMSE	MAE
SVD-G	0.917	0.835	0.898	0.671
SVD-L	0.923	0.889	0.868	0.705
PB	0.800	0.824	1.601	1.270
JNB	0.713	0.747	1.580	1.260

5. Discussion

We have presented a new sharpness metric based on SVD. We have shown that the distribution of singular values of a grey level image can be used to estimate the perceived sharpness of the image. The metrics have a very high correlation with the ASR and outperforms the other sharpness metrics it was tested against.

The correlation between the proposed metrics and the ASR is marginally higher when using the gray level image instead of the color images. This might be expected since the metrics are only using the luminance channel. It calls for further investigations and the possibility of utilizing the information in the chroma bands to get an even higher correlation with the subjective data.

This is the first step towards a metric to predict the sharpness of video. Another very challenging task is the inclusion of motion vectors in this framework together with the different characteristics of different display types. These are tasks that needs to be explored further in future work.

References

- [1] L. MacDonald, Framework for an image sharpness management system, in: IS&T/SID 7th Color Imaging Conference, Scottsdale, 1999, pp. 75–79.
- [2] C. Wee, R. Paramesran, Measure of image sharpness using eigenvalues, *Information Sciences* 177 (12) (2007) 2533–2552.
- [3] P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi, Perceptual blur and ringing metrics: Application to JPEG2000, *Signal Processing: Image Communication* 19 (2) (2004) 163–172.
- [4] R. Ferzli, L. Karam, A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB), *IEEE Transactions on Image Processing* 18 (4) (2009) 717.
- [5] R. Ferzli, L. Karam, Human visual system based no-reference objective image sharpness metric, in: *IEEE Int. Conf. on Image Processing*, 2006, pp. 2949–2952.
- [6] S. Varadarajan, L. Karam, An improved perception-based no-reference objective image sharpness metric using iterative edge refinement, in: *Proceedings of the 15th IEEE International Conference on Image Processing*, 2008, pp. 401–404.
- [7] D. Kalman, A singularly valuable decomposition: The SVD of a matrix, *The College Mathematics Journal* 27 (1) (1996) 2–23.
- [8] C. Chang, P. Tsai, C. Lin, SVD-based digital image watermarking scheme, *Pattern recognition letters* 26 (10) (2005) 1577–1586.
- [9] A. Mohammad, A. Alhaj, S. Shaltaf, An improved SVD-based watermarking scheme for protecting rightful ownership, *Signal Processing*.
- [10] H. Prasantha, H. Shashidhara, K. Murthy, Image compression using SVD, in: *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)-Volume 03*, IEEE Computer Society, 2007, pp. 143–145.

- [11] Y. Wongsawat, K. Rao, S. Orintara, Multichannel SVD-based image de-noising, in: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Vol. 6, Citeseer, 2005, p. 5990.
- [12] P. Hansen, J. Nagy, D. O'leary, Deblurring images: matrices, spectra, and filtering, Siam, 2006.
- [13] N. Muller, L. Magaia, B. Herbst, Singular value decomposition, eigenfaces, and 3d reconstructions, SIAM REVIEW 46 (3) (2004) 518–545.
- [14] T. I. R. Assembly, RECOMMENDATION ITU-R BT.500-11 (2002) Methodology for the subjective assessment of the quality of television pictures, International Telecommunication Union, Geneva, Switzerland.
- [15] V. Q. E. Group, Final Report From the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II, Tech. rep., VQEG (2003).



Engholm, R., Schmiegel J., Jensen E. B. V. and
Veraart A. E. D. (2009).
Edge detection in image processing using realised
variance measures.
Manuscript.

Edge detection in image processing using realised variance measures

Rasmus Engholm^{*†‡}

Jürgen Schmiegeler^{*}

Eva B. Vedel Jensen^{*}

Almut E. D. Veraart^{*§}

Abstract

Edge detection is one of the key problems in image processing. We relate the problem of detecting edges in images to the problem of detecting jumps/breaks in time series data. Realised measures originating from financial econometrics are used for developing a methodology for edge detection in image processing. Until now realised measures have mainly been used in the analysis of ultra-high frequent time series data from finance. Since line scans do not have the high-frequent nature of finance data various finite sample corrections of realised measures are used in the analysis of the line scans. Finally, we discuss extensions of the methods to two dimensions.

1 Introduction

Edge detection is an essential part of image processing. It plays a vital role in connection with segmentation tasks where detection of edges is often the first step in a chain of processing steps.

Human vision has proven to perform very well as an edge detector, cf. e.g. Marr & Hildreth (1980). In contrast to this, computer-based detection of edges is only a trivial task under optimal conditions. The usual situation is that blurring due to imperfect image acquisition conditions (lenses and focus) and noise introduced in the acquisition and transmission stage makes it very difficult to distinguish between noise and a non-predominant edge.

Edge detection usually consists of a convolution with a small kernel to find edge candidates. This can be done by finding gradients (first order methods) or the Laplacian (second order methods).

^{*}Thiele Centre of Applied Mathematics in Natural Science, University of Aarhus, Denmark

[†]Engineering College of Aarhus, Denmark

[‡]Bang & Olufsen a/s, Struer, Denmark

[§]CREATES, University of Aarhus, Denmark

The most common methods are first order methods, cf. e.g. Sobel & Feldman (1968) and Canny (1986). First order methods are very fast because they only involve a simple convolution with a discrete approximation of a differentiation operator in typically the horizontal and vertical directions. The Canny filter consists furthermore of a smoothing step to reduce noise prior to the convolution and a subsequent step where the edge candidates are connected. All edge detection methods depend on threshold limits, where a pixel is considered an edge candidate if the value associated with the output of the gradient step is equal or above the threshold limit. The introduction of a threshold limit introduces an arbitrary element in the edge detection procedure.

In the present paper we use realised measures from financial econometrics in the development of a methodology of edge detection. In Section 2, we give an overview of the theory of realised measures and exemplify its use by the analysis of a concrete financial time series while Section 3 contains a study of how these measures perform in image processing. Section 4 discusses various possible alternative estimators and plans for future research.

2 Overview on realised variance measures

The problem of detecting breaks or jumps in time series has been studied extensively, in particular in series relating to financial econometrics. The latter area has mainly focused on the question of how to estimate, test and filter out jumps in high frequency financial data. In the following, we will review these methods and discuss how they can be adapted for use in image processing.

2.1 General modelling framework

The general theory can be formulated in continuous time for fairly general stochastic processes, namely for so-called Itô semimartingales, cf. Jacod & Shiryaev (2003) and Jacod (2008). Below, we give an informal exposition of the theory, using the notation from Barndorff-Nielsen & Shephard (2006).

Let $Y = \{Y_t : t \geq 0\}$ be such a stochastic process. The *quadratic variation process* is defined by

$$[Y]_t = \lim_{n \rightarrow \infty} \sum_{j=0}^{n-1} (Y_{\frac{j+1}{n}t} - Y_{\frac{j}{n}t})^2.$$

It can be shown that

$$[Y]_t = [Y]_t^c + [Y]_t^d,$$

where $[Y]_t^c$ denotes the continuous part of the quadratic variation while

$$[Y]_t^d = \sum_{0 \leq u \leq t} \Delta Y_u^2$$

denotes the discontinuous/jump part of the quadratic variation. Here, $\Delta Y_t = Y_t - Y_{t-}$ is the jump of Y at time t .

Usually, we do not observe the stochastic process Y in continuous time, but only at discrete time points. Let us assume that the process Y is observed at discrete times $i\delta$ for $i = 0, 1, \dots, \lfloor T/\delta \rfloor$ for some $T > 0$ and $\delta > 0$. Here, $\lfloor x \rfloor$ denotes the integer part of x . In the following, we will always choose $t \in [0, T]$. Furthermore, we will let $y_{i\delta} = Y_{i\delta} - Y_{(i-1)\delta}$ denote the i th increment of Y .

2.2 The realised variance process

The quadratic variation process can be estimated using the *realised variance process*, defined by

$$[Y_\delta]_t = \sum_{j=1}^{\lfloor t/\delta \rfloor} y_{j\delta}^2.$$

It can be shown, cf. e.g. Jacod (2008), that

$$[Y_\delta]_t \rightarrow [Y]_t = [Y]_t^c + [Y]_t^d, \quad \text{as } \delta \rightarrow 0.$$

Where In order to disentangle the continuous part of the quadratic variation from the jump part, the concept of a realised variance process has been extended in two directions: towards *realised bipower variation* and towards *truncated realised variance*. Both concepts will be briefly presented in the following.

2.3 The realised bipower variation process

The concept of *realised bipower variation*, as defined by Barndorff-Nielsen & Shephard (2004, 2006), can be used for estimating the continuous part of the quadratic variation, see Barndorff-Nielsen et al. (2006), Jacod (2006) and Woerner (2006). The continuous part is the quantity of main interest in financial econometrics whereas jumps are usually referred to as rare events which happen due to macro economic announcements or other more global events.

The realised bipower variation is defined by

$$\{Y_\delta\}_t^{[1,1]} = \mu_1^{-2} \sum_{j=2}^{\lfloor t/\delta \rfloor} |y_{(j-1)\delta}| |y_{j\delta}|,$$

where

$$\mu_1 = E|U| = \sqrt{2}/\sqrt{\pi} \approx 0.79788,$$

and $U \sim N(0, 1)$. It can be shown that

$$\lim_{\delta \rightarrow 0} \{Y_\delta\}_t^{[1,1]} = [Y]_t^c.$$

Hence, the jump part of the quadratic variation process can be estimated by

$$[Y_\delta]_t - \{Y_\delta\}_t^{[1,1]} \quad \text{for small } \delta. \quad (1)$$

The intuitive reason why realised bipower variation is related to the continuous part of the quadratic variation process is the following. If there was one jump in one interval of time of length δ the probability that another jump occurs right in the subsequent time interval is basically zero. So, when one studies realised bipower variation rather than realised variance where $y_{j\delta}^2$ is replaced by $|y_{(j-1)\delta}| |y_{j\delta}|$, then the impact of a jump is much lower. In fact the size of the jump is not squared, but multiplied by a smaller $y_{j\delta}$ which is likely not to be due to a jump. And the same intuition holds when one extends bipower to tripower, quadpower or various other multipower statistics. It is also possible to assess the quality of the estimation by constructing confidence bounds based on the asymptotic theory developed in Jacod (2006).

In contrast to the situation in financial econometrics, we are in this paper interested in estimating jumps and use such estimation in the detection of edges in images. This is done by combining the realised variance with the realised bipower variation as shown in (1).

2.4 Truncated realised variance

Alternatively to the bipower variation, one can use truncation techniques for estimating the continuous part of the quadratic variation process. Such methods have been studied by Mancini (2001, 2006) and Jacod (2008). In particular, we have

$$\sum_{j=1}^{\lfloor t/\delta \rfloor} y_{j\delta}^2 \mathbb{I}_{\{|y_{j\delta}| \leq \gamma \delta^\alpha\}} \rightarrow [Y]_t^c \quad \text{as } \delta \rightarrow 0,$$

where $\gamma > 0$ is a constant and $\alpha \in (0, \frac{1}{2})$.

The intuition behind this truncation technique is straightforward. If an increment $y_{j\delta}$ is larger than a certain threshold, then this increment may be regarded as a jump and hence ignored when computing the realised variance. The threshold has to be chosen according to some regularity conditions, depending on the interval length δ for the asymptotic theory to work. Similarly to the results on realised bipower variation, there is also an asymptotic theory for truncated realised variance which makes this possible, cf. Jacod (2008).

2.5 Truncated bipower variation

Some recent research by Corsi et al. (2009) suggests that combining the ideas of bipower variation and threshold estimators improves the estimation results in finite samples significantly. Truncated bipower variation (TBPV) is defined as follows. Let $\theta > 0$ be the threshold which is usually assumed to depend on the interval length δ , i.e. $\theta = \theta_\delta$ with

$$\lim_{\delta \rightarrow 0} \theta_\delta = 0, \quad \lim_{\delta \rightarrow 0} \frac{\delta \log(1/\delta)}{\theta_\delta} = 0. \quad (2)$$

The truncated bipower variation is defined by

$$\langle Y_\delta \rangle_t^{[1,1]} = \mu_1^{-2} \sum_{j=2}^{\lfloor t/\delta \rfloor} |y_{j\delta}| |y_{(j-1)\delta}| \mathbb{I}_{\{|y_{j\delta}|^2 \leq \theta_\delta\}} \mathbb{I}_{\{|y_{(j-1)\delta}|^2 \leq \theta_\delta\}}.$$

One can then show that

$$\langle Y_\delta \rangle_t^{[1,1]} \rightarrow [Y]_t^c \text{ as } \delta \rightarrow 0.$$

Simulation studies by Corsi et al. (2009) show that truncated bipower variation seems to perform better in finite samples than realised bipower variation or truncated realised variance alone.

Clearly, the question is how to choose the threshold. In the case of high frequency financial data there are methods for choosing a sensible threshold, see Corsi et al. (2009). However, since such methods depend crucially on the existence of ultra high frequent data, which we do not have in image processing, we cannot use those methods. In Section 3 below we simply choose the threshold in an ad hoc manner.

2.6 Testing for jumps

The question of testing for jumps has been studied by Barndorff-Nielsen & Shephard (2004, 2006), Aït-Sahalia & Jacod (2009) and Veraart (2010). The main idea is to use the realised variance for estimating the whole quadratic variation and to use a jump robust statistic such as realised bipower variation or truncated realised variance for estimating the continuous part of the quadratic variation. Clearly, the difference of these two statistics can then be used as a consistent estimator of the jump part of the quadratic variation. And, in fact, one can do even more. In addition to just estimating the jump part of the quadratic variation consistently, one can formally test for jumps since the joint distribution of realised variance and realised bipower variation or truncated realised variance, respectively, are well-known, see Barndorff-Nielsen & Shephard (2006) and Aït-Sahalia & Jacod (2009).

In particular, one can show that, under the assumption that there are no jumps in $[0, t]$ and with $\rho = (\pi^2/4) + \pi - 5 = 0.6090$, the statistic

$$Z_t = \frac{\delta^{-1/2}([Y_\delta]_t - \{Y_\delta\}_t^{[1,1]})}{\sqrt{\rho \mu_1^{-4} \{Y_\delta\}_t^{[1,1,1,1]}}} \quad (3)$$

is asymptotic normally distributed as $\delta \rightarrow 0$ with mean 0 and variance 1. Here,

$$\{Y_\delta\}_t^{[1,1,1,1]} = \delta^{-1} \sum_{j=4}^{[t/\delta]} |y_{(j-3)\delta}| |y_{(j-2)\delta}| |y_{(j-1)\delta}| |y_{j\delta}|.$$

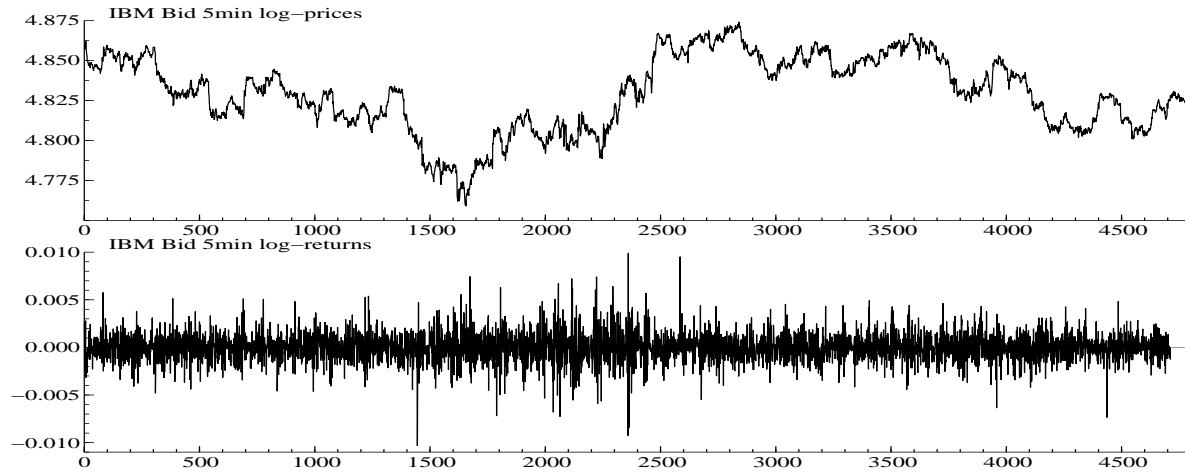
A test for jumps in $[0, t]$ can therefore be performed by computing the critical region associated with the above asymptotic distribution. In particular, if the test statistic Z_t takes higher, positive values than the corresponding 95% percentile of the standard normal distribution, we reject the null hypothesis of Y not having jumps in $[0, t]$.

2.7 Empirical illustration

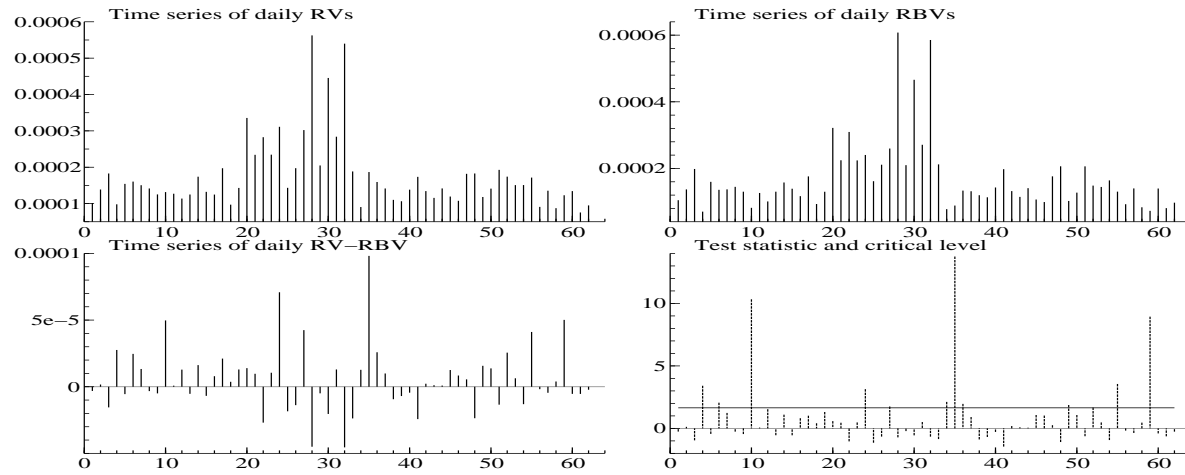
Before we turn to applications of the theory on realised variance measures to image processes, we show how the theory is usually used in financial econometrics. The illustration concerns some high frequency IBM quote data from the WRDS data base which have been cleaned according to the procedure described in Hansen & Lunde (2006a,b).

First of all, we provide a plot of the logarithmic asset prices sampled every five minutes ($\delta = 5$ min) and the corresponding five minute returns $y_{j\delta}$, see Figure 1(a).

In a next step, we compute the daily realised variances and the daily realised bipower variations. In order to do that, we divide the time series of 5 minute data into daily blocks and compute the realised variance and realised bipower variation within each block and repeat the procedure for every day. Hence, we obtain time series of daily realised variances and realised bipower variations. When we look at the corresponding plots in Figure 1(b), we see that there is in particular one period in the middle of the sample where both realised variance and realised bipower variation take fairly high values in consecutive days. This fact is referred to as *volatility clusters* in the financial literature. Next, we plot the difference of realised variance and realised bipower for each day. As previously described, this leads to a time series of the estimated jump part of the quadratic variation. We can clearly spot some days, where the difference between the two statistics seems to be ‘big’. Finally, we check whether these differences are significantly larger than zero. In order to do that, we scale the difference of the realised variance and realised bipower variation by the square root of the estimated asymptotic variance (according to the limit theory derived in Barndorff-Nielsen & Shephard (2006), see (3)) and compare it to the critical value of the standard normal distribution. Over all, we find that there are 12 days (out of 62), where we reject the null hypothesis of not having a jump component in the price process.



(a) IBM data.



(b) Realised variation measures and jump test.

Figure 1: Testing for jumps in financial data.

3 Applying realised measures in image processing

To illustrate how realised variation measures can be used in edge detection in digital images, we initially analyzed the 512×768 image shown in Figure 2.

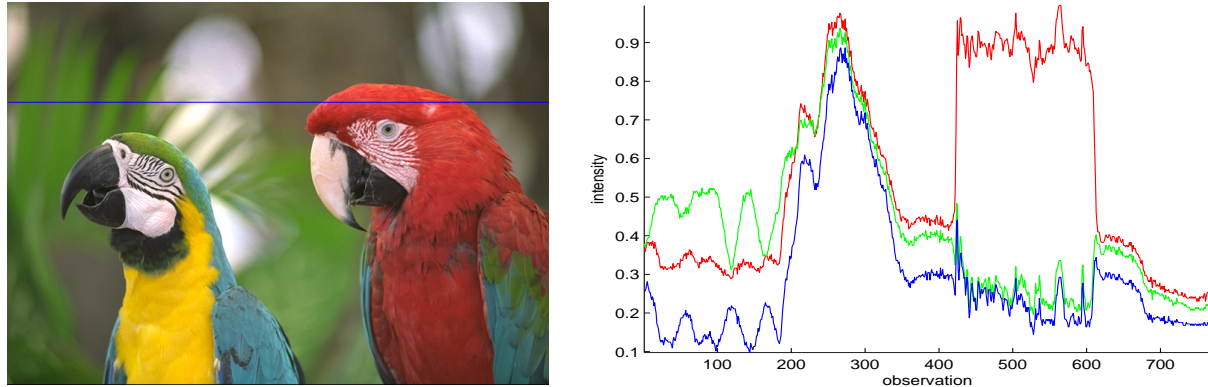


Figure 2: (left) A three component color image with a blue horizontal line superimposed, indicating the location of the scan line considered below. (right) The levels of the red, green and blue component in the scan line, respectively.

In a first examination, the associated grey-level image has been analyzed, cf. Figure 3. In Figure 4, the result of analyzing a particular scan line (number 134, indicated by blue in Figure 3) is shown. In the upper row of Figure 4, the raw data $Y_{j\delta}$ and the first order differences (returns) $y_{j\delta}$ are shown. The remaining part of the figure is based on dividing the scan line under analysis into 50 non-overlapping blocks with 15 observations in each. The middle row of Figure 4 shows the realised variance and the realised bipower variation, while the bottom row shows the difference between the two latter quantities and the normalised difference (the test statistic Z_t).



Figure 3: (left) The associated grey-level image with a blue horizontal line (line 134) superimposed, indicating the location of the scan line analyzed in Figure 4.

The two large differences in block 29 and 40 occur at the locations in the image where we enter and leave the parrot to the right in Figures 2 and 3(left). The results of testing whether the jumps

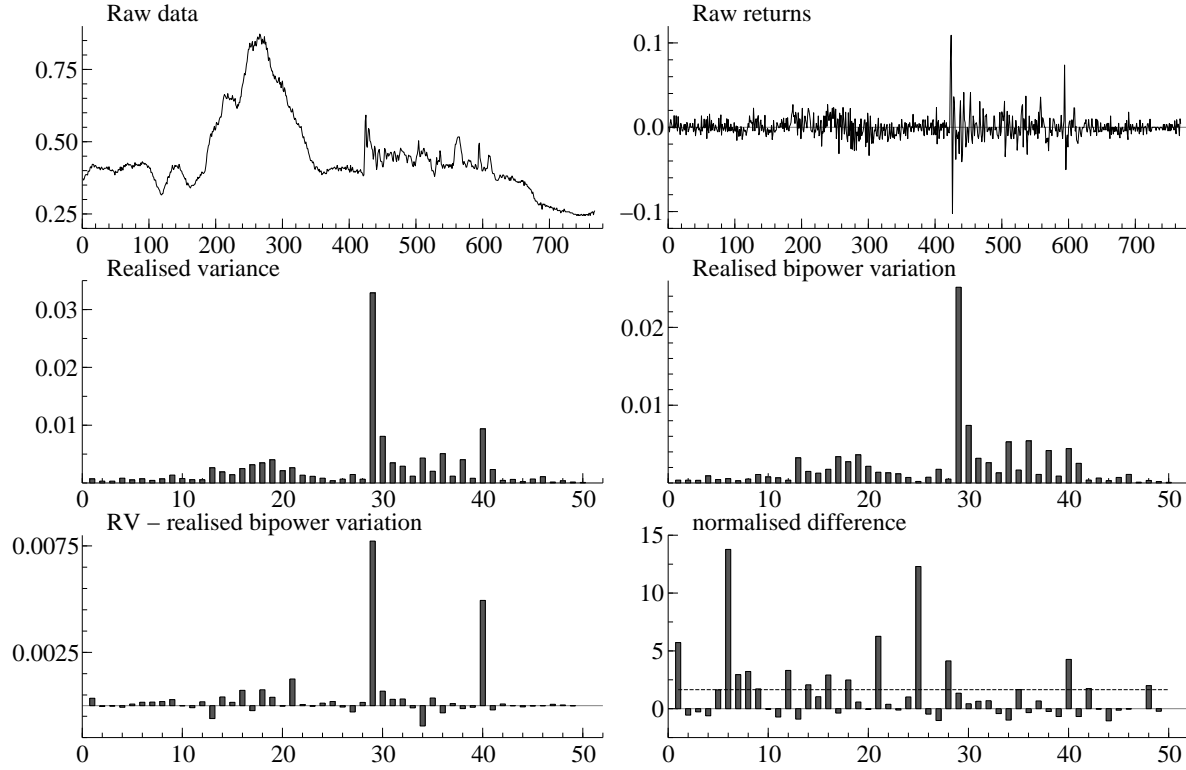


Figure 4: Analysis of the scan line from Figure 3.

are significantly different from zero (lower right corner of Figure 4) are less clear. Additional differences originating from the left part of the scan (the green leaves in the original color image) appear also to be significantly different from zero.

This result calls for further examinations. One important point is here that the test statistic Z_t is not exactly normally distributed but asymptotically normal as $\delta \rightarrow 0$. Simulations in Barndorff-Nielsen & Shephard (2006) indicate that more than 15 observations are needed in order to trust the asymptotics. In the present spatial context, we may compensate for this by also analyzing the columns or, more generally, use the information in a spatial neighborhood of a particular pixel. New asymptotic developments are needed for this purpose.

Quite another point is that in an imaging context it might be more interesting simply to estimate the sizes of the jumps rather than testing whether they are significantly different from zero. A small (although significant) jump is not likely to be discovered by the viewer.

In Figure 5 we illustrate the use of bipower variation on the entire image by processing all horizontal lines in the image in the same way. We illustrate this with block size of 15, 5 and 2. The figure shows those blocks for which the estimated jump (realised variance minus realised bipower variation) is larger than 0.003. It is noteworthy that a block size of 2 works so well.



Figure 5: Estimated jumps larger than a global threshold of 0.003 are shown as white. The jumps are based on horizontal line scans with a block size of 15, 5 and 2 respectively from left to right. The estimation of jumps uses realised bipower variation.

Figure 6 shows a similar investigation of a 512×768 image where again the estimation of the jump component is based on horizontal line scans. The block size is here reduced to 2 pixels to get a higher resolution. The lower right part of Figure 6 shows those blocks for which the estimated jump (realised variance minus realised bipower variation) is larger than 0.003. Note that part of the wing of the plane in the right part of the image is not reconstructed very well. This is not surprising since horizontal scan lines are not well suited to detect horizontal edges.

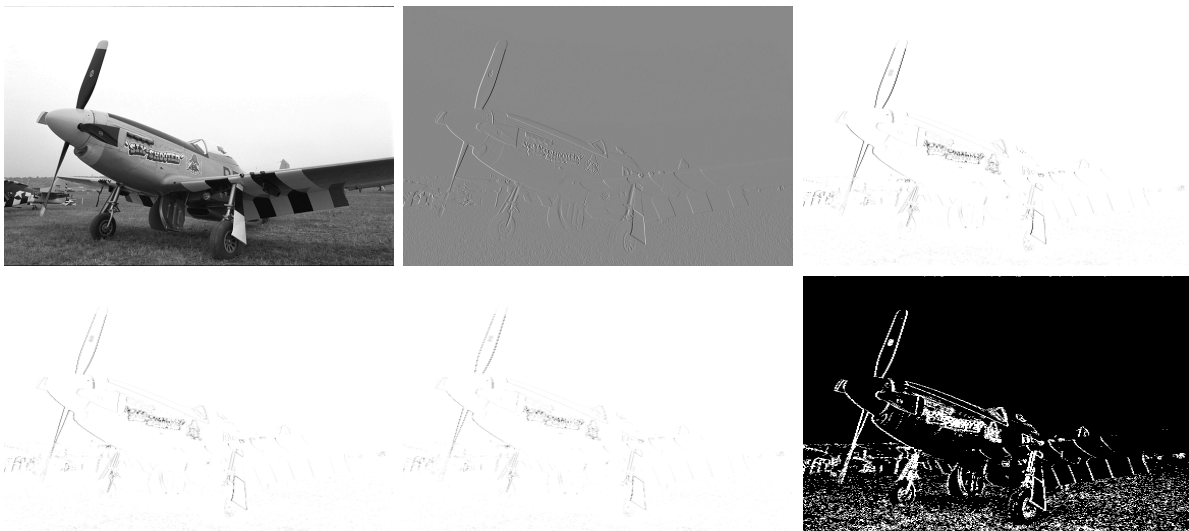


Figure 6: (top left) The original grey level image. (top middle) The raw returns of the image calculated horizontally. (top right) The realized variance calculated horizontally, dark indicates high values. (bottom left) The bipower variation calculated horizontally. The power is 2 and the interval size is 2 (dark indicates high values). (bottom middle) The jump component, realized variance-bipower variation (dark indicates high values). (bottom right) The thresholded jump component (white is edges).

For comparison, Figure 7 shows the analysis of the plane image, now with the use of truncated realised variance ($\gamma\delta^\alpha = 0.05$) instead of realised bipower variation. We notice a rather big difference between the bipower term, Figure 6 (bottom left) and the truncated realised variance term, Figure 7 (left). We notice that the latter appears much more noisy, because the largest values have been excluded in this term, putting more emphasis on the noise.



Figure 7: Reconstruction of edges in the plane image, using truncated realised variance with $\gamma = 0.05$, instead of realised bipower variation. (left) Truncated realised variance. (middle) Jump component (QV- TRV). (right) Thresholded truncated realised variance. Those blocks for which the estimated jump is larger than 0.003 are shown as white.

In Figure 8, the performance of the edge detection methods based on, respectively, realised bipower variation, truncated realised variance and truncated bipower variation are compared for three different images. Although the performance of the methods is similar, some differences between the realised bipower variation on one hand and truncated realised variance and truncated bipower on the other can be observed. This finding is further exemplified in Figure 9, showing a detail from the bike image in the first column of Figure 8. We notice that the use of realised bipower variation results in an edge detection which appears much noisier. In particular at the rim of the wheel, we observe that realised bipower variation does not perform as well as the two methods based on truncation e.g. the rim contains holes, in contrast to unbroken edges when using truncation methods. This is because the truncation methods exclude the largest values in the continuous term, which makes them appear more clear when we estimate the jump term. The two truncation methods are almost identical in performance.

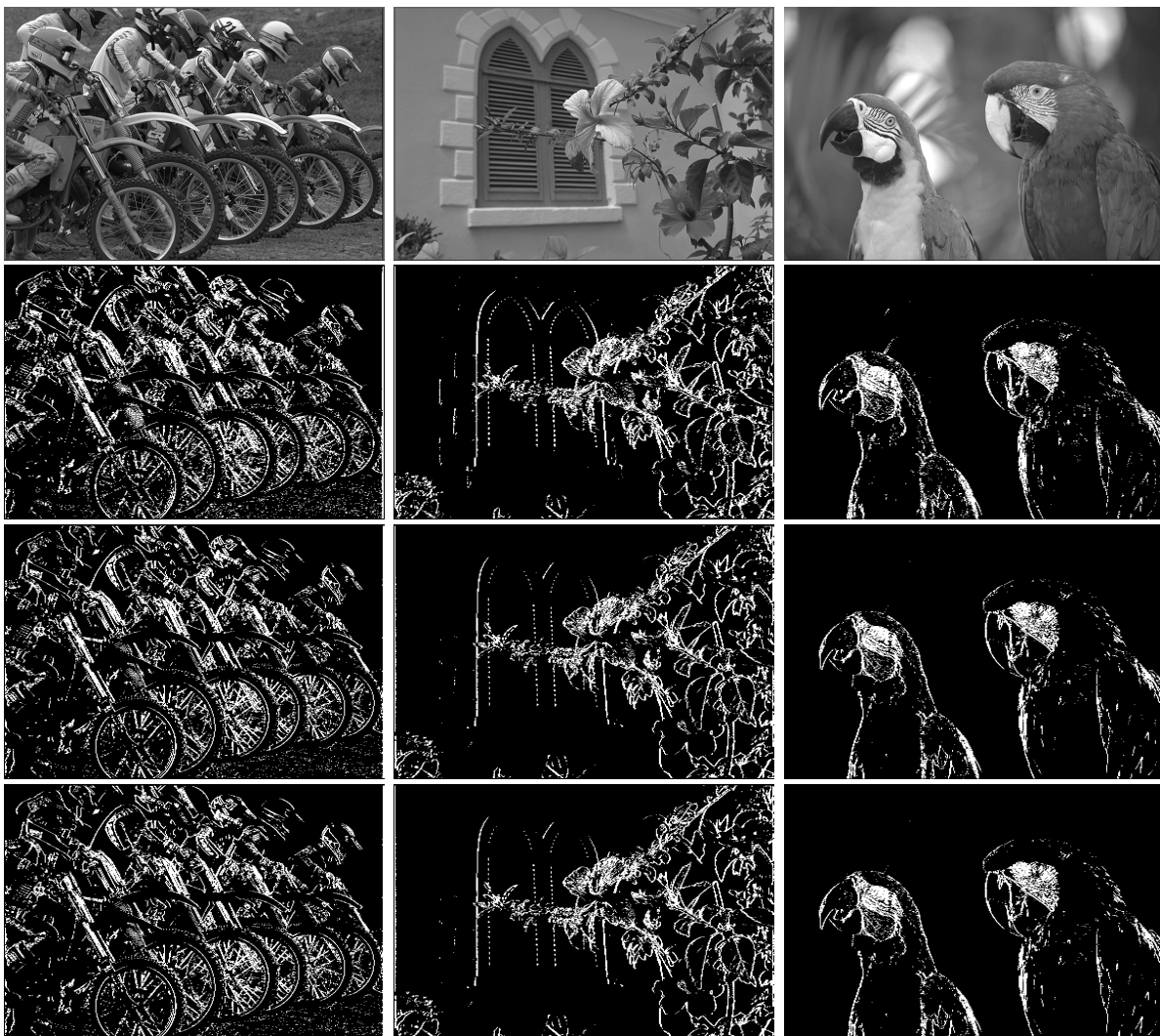


Figure 8: (First row): Original grey level images. (Second row): Result of edge detection using bipower variation. (Third row): Edges found, using truncated realised variance. (Fourth row): Edges found, using truncated bipower variation.



Figure 9: Detail of edge extraction using respectively (left) bipower variation, (middle) truncated realised variance and (right) truncated bipower variation. Notice that the edges of the rim of the wheel in the last two images appear more connected and less noisy.

4 Various improvements and alternative estimators

4.1 Combining horizontal and vertical line scans

As described earlier, using only horizontal line scans is not sufficient for detecting horizontal edges. An obvious solution is to combine horizontal and vertical line scans. As mentioned above the resolution decreases in the observed direction, so combining the two directions is not straight forward. A solution could be to increase the resolution after the edge detection, using up-scaling techniques.

4.2 Local as opposed to global thresholds

Studying the resulting edges as appearing in Figure 6 we notice that the lower part of the image contains a lot of small edges from the grass. This is not in accordance with what we perceive, since the grass should appear as a homogeneous surface. Using a single global threshold to detect edges is obvious not ideal. A threshold function adapting to the variability of the surroundings could be a way to improve the performance.

4.3 Sliding bipower variation

Since the jump component is defined for an interval, we have to decrease the resolution to locate jumps in intervals. In order to maintain the precision an idea is to use sliding intervals, such that the resolution can be regained. However this is only relevant if the frequency of the jumps is larger than the window size, an assumption that makes the application rather limited.

4.4 Spatial realised variance

So far, we have used methods based on one-dimensional data in order to search for edges. Clearly, a natural extension would be to study realised variation measures based on spatial increments. In this case, our process is indexed by two coordinates $Y = \{Y_{s,t} : s, t \geq 0\}$. The discrete observations are $Y_{i\delta, j\delta}$. A spatial increment can be defined as

$$y_{i\delta, j\delta} = Y_{i\delta, j\delta} - Y_{(i-1)\delta, j\delta} - Y_{i\delta, (j-1)\delta} + Y_{(i-1)\delta, (j-1)\delta}.$$

A spatial version of realised variance has been studied by Guyon (1987). Corresponding spatial versions of realised bipower variation and truncated realised variance remain still to be developed.

5 Conclusion

We have shown that realised variance measures can indeed detect edges. In this paper we have presented three different variations of this theme. We believe that the fast computation and the ability of the methods to work with different resolution will be useful in computing an initial edge-map to guide a further process in e.g. vision tasks and avoid using more computational heavy methods on regions that are not of interest. After this initial edge detection a more computational demanding algorithm such as the Canny edge detection can be used in subregions to enhance the edges. Furthermore the foundation on theoretic work within a different field can hopefully be inspirational for others.

References

- Aït-Sahalia, Y. & Jacod, J. (2009), ‘Testing for jumps in a discretely observed process’, *Annals of Statistics* **37**(1), 184–222.
- Barndorff-Nielsen, O. E. & Shephard, N. (2004), ‘Power and bipower variation with stochastic volatility and jumps (with discussion)’, *Journal of Financial Econometrics* **2**, 1–48.
- Barndorff-Nielsen, O. E. & Shephard, N. (2006), ‘Econometrics of testing for jumps in financial economics using bipower variation.’, *Journal of Financial Econometrics* **4**, 1–30.
- Barndorff-Nielsen, O. E., Graversen, S. E., Jacod, J., Podolskij, M. & Shephard, N. (2006), A central limit theorem for realised power and bipower variations of continuous semimartingales, in Y. Kabanov & R. Lipster, eds, ‘From Stochastic Analysis to Mathematical Finance, Festschrift for Albert Shiryaev’, Springer.
- Canny, J. (1986), ‘A computational approach to edge detection’, *IEEE Transactions on pattern analysis and machine intelligence* pp. 679–698.
- Corsi, F., Reno, R. & Pirino, D. (2009), Threshold bipower variation and the impact of jumps on volatility forecasting, Working paper.
- Guyon, X. (1987), ‘Variations de champs gaussiens stationnaires: application à l’identification’, *Probability Theory and Related Fields* **75**, 179–193.
- Hansen, P. R. & Lunde, A. (2006a), ‘Realized variance and market microstructure noise’, *Journal of Business and Economic Statistics* **24**, 127–218.
- Hansen, P. R. & Lunde, A. (2006b), ‘Rejoinder (to comments on realized variance and market microstructure noise)’, *Journal of Business and Economic Statistics*.

- Jacod, J. (2006), Asymptotic properties of realized power variations and related functionals of semimartingales: Multipower variation, Université P. et M. Curie (Paris 6): Unpublished paper.
- Jacod, J. (2008), ‘Asymptotic properties of realized power variations and related functionals of semimartingales’, *Stochastic Processes and their Applications* **118**(4), 517–559.
- Jacod, J. & Shiryaev, A. N. (2003), *Limit Theorems for Stochastic Processes*, second edn, Springer, Berlin.
- Mancini, C. (2001), ‘Disentangling the jumps of the diffusion in a geometric jumping brownian motion’, *Giornale dell Istituto Italiano degli Attuari* **LXIV**, 19–47.
- Mancini, C. (2006), Estimating the integrated volatility in stochastic volatility models with Lévy type jumps, Preprint.
- Marr, D. & Hildreth, E. (1980), ‘Theory of edge detection’, *Proceedings of the Royal Society of London. Series B, Biological Sciences* pp. 187–217.
- Sobel, I. & Feldman, G. (1968), ‘A 3x3 isotropic gradient operator for image processing’, *Presentation for Stanford Artificial Project*.
- Veraart, A. E. D. (2010), ‘Inference for the jump part of quadratic variation of Itô semimartingales’, *Econometric Theory*. Forthcoming.
- Woerner, J. H. C. (2006), Power and multipower variation: inference for high frequency data, in A. N. Shiryaev, M. R. Grossinõ, P. Oliveira & M. Esquivel, eds, ‘Stochastic Finance’, Springer, pp. 343–364.