TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Motion blur reduction for high frame rate LCD-TVs

Citation for published version (APA):
Heesch, van, F. H., Haan, de, G., & Kaup, A. (2011). Motion blur reduction for high frame rate LCD-TVs. In *Proceedings of the 14th ITG Conference on Electronic Media technology, 23-24 March, 2011, Dortmund, Germany* (pp. 1-5)

Document status and date:
Published: 01/01/2011

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 08. Feb. 2024

Motion Blur Reduction for High Frame Rate LCD-TVs

F. H. van Heesch and G. de Haan
Philips Research Laboratories
High Tech Campus 36, Eindhoven, The Netherlands

***Abstract*—Today's LCD-TVs reduce their hold time to prevent motion blur. This is best implemented using frame rate up-conversion with motion compensated interpolation. The registration process of the TV-signal, by film or video camera, has been identified as a second motion blur source, which becomes dominant for TV displays with a frame rate of 100 Hz or higher. In order to justify any further hold time reduction of LCDs, this second type of motion blur, referred to as camera blur, needs to be addressed. This paper presents a real-time camera blur estimation and reduction method, suitable for TV-applications.**

I. Introduction

At the introduction of LCDs in the TV domain, motion blur was identified as one of the most important picture quality aspects that required improvement to compete with the then dominant CRT displays. From the many motion blur reduction methods that have since been proposed [1], Motion Compensated Frame Rate Conversion (MC-FRC) has been the most successful. The MC-FRC method has the advantage that the hold time of a display system can be reduced, without negative picture quality side-effects such as large area flicker and loss of light efficiency. As a result, 120Hz, 240Hz and even 480Hz LCDs with MC-FRC have been proposed [2]. The improvements in motion blur reduction for such high frame rate displays is limited, however, due to the presence of motion blur that originated during registration [3]. The combination of both motion blur sources, i.e. *display blur* and *camera blur*, has been investigated in [3], verified in [4], and is illustrated in Fig. 1.

Camera blur and display blur can both be described as a temporal averaging filter that, due to motion tracking, is perceived as a spatial averaging along the motion vector [3]. There are, however, two important differences for TV-applications leading to different motion blur reduction algorithms. First, display blur can be known from the display properties, while camera blur needs to be estimated from the TV's input picture. Second, display blur is caused by eye movement relative to the display, while camera blur is caused by camera motion relative to the scene. Because of these differences, camera blur reduction requires a post-processing of the TV-signal, while display blur reduction is a pre-processing method.

In this paper, the implementation of a real-time camera blur reduction method is discussed. Because it attempts to invert the *perceived* motion blur, this filter method is referred to as Motion Compensated Inverse Filtering (MCIF). We will show in Section II, that the filter characteristics of motion blur

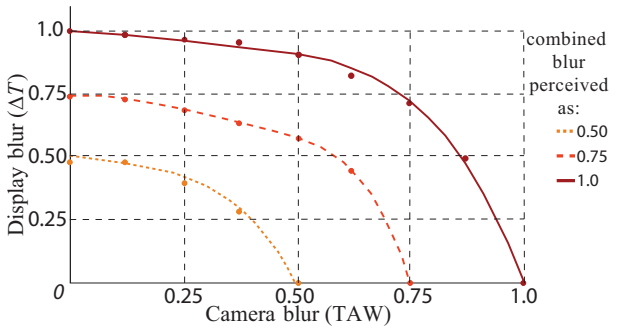


Fig. 1. Motion blur perceived from video that is displayed on an LCD is a combination of display blur (vertical axis) and camera blur (horizontal axis). Both blur sources are relative to the picture delay, ΔT. The measurement points have been obtained from a perception test [4].

are straightforwardly derived from theory, but that a practical implementation for removing this blur is not easily robust. We will describe the signal theory in Section II and from the theory it follows that, for the inverse filter, motion estimation and motion blur estimation are required pre-processing steps. This will be described in Section II-A. In Section II-B, we will discuss two filter implementations and results are shown in Section III. Conclusions are drawn in Section V.

II. Camera blur reduction

The perceived motion blur that is caused by an LCD has been described in [5] and can be analyzed in the frequency domain by describing the perceived picture:

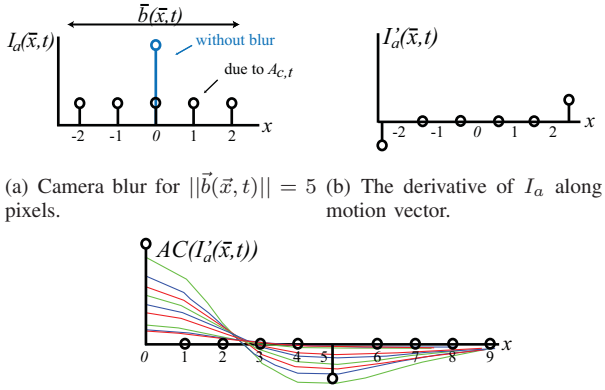
$$I_p^f(f_{\vec{x}}, f_t) = I_a^f(f_{\vec{x}}, f_t) A_d^f(f_{\vec{x}}, \vec{v} \cdot f_{\vec{x}})\, A_p^f(f_{\vec{x}}, f_t), \qquad (1)$$

with I_a^f the reconstructed picture on the display, A_d^f, the spatio-temporal display aperture, A_p^f, the aperture of the Human Visual System (HVS), $\vec{v}$, the motion vector corresponding to tracking and $f_{\vec{x}}$ and f_t, the spatial and temporal frequencies, respectively. The level of perceived motion blur is determined by analyzing the attenuation of the high spatial frequencies of I_a^f for $f_t = 0$.

The reconstructed picture, I_a^f, can be expressed to include camera blur, using a spatio-temporal registration aperture:

$$I_a^f(f_{\vec{x}}, f_t) = I_c^f(f_{\vec{x}}, f_t) A_c^f(f_{\vec{x}}, \vec{v} \cdot f_{\vec{x}}), \qquad (2)$$

with I_c^f the sampled picture during registration and A_c^f the registration aperture. In order to directly compare I_c^f with the



(a) Camera blur for $||\vec{b}(\vec{x},t)|| = 5$ pixels.

(b) The derivative of I_a along the motion vector.

(c) The AC of (b), ideal (black) and measurement examples (lines).

Fig. 2. The position of the minimum of the AC of the image derivative corresponds to the blur length.

reconstructed picture, I_a^f, and to focus on motion blur, we have ignored spatial scaling and the influence of the spatial camera aperture and will describe the temporal camera aperture by the Temporal Aperture Width (TAW), a_t, as reported in [6]:

$$A_{c,t}(t) = \begin{cases} 1, & \text{if } |t| \leq \frac{1}{2a_t} \\ 0, & \text{if } |t| > \frac{1}{2a_t} \end{cases} \multimap\bullet\ A^f_{c,t}(f_t) = a_t \text{sinc}(\pi f_t a_t),$$

with $\text{sinc}(t) = \frac{\sin(t)}{t}$ and $\multimap\bullet$ denoting the Fourier pair.

In order to reduce the attenuation caused by camera blur, an enhancement filter can be constructed that inverts the temporal registration aperture, $|A^f_{c,t}(\vec{v} \cdot f_{\vec{x}})|$, i.e. we can write for the amplitude response of the enhancement:

$$|H^f_c(f_{\vec{x}})| = \frac{1}{\left|A^f_{c,t}(\vec{v} \cdot f_{\vec{x}})\right|} = \frac{1}{|\text{sinc}(\pi(\vec{v} \cdot f_{\vec{x}})a_t)|}. \quad (3)$$

The amplitude response of this filter has infinite gains and is unstable in the presence of noise. Therefore, we approximate the filter response by a filter that boosts high spatial frequencies along the local motion vector. Furthermore, the amplitude response scales with the TAW. In Section II-A, we will explain how the TAW is estimated from the input picture, before discussing the filter design in Section II-B.

A. Camera blur estimation

The estimation of camera blur has been mainly investigated for image processing in the area of astronomy and still picture processing [7], [8], [9]. In [7], a method is proposed that estimates the blur characteristics from the spatial cepstral domain using multiple sub-images to attenuate signal characteristics in favor of the blur characteristics. The cepstral domain method was replaced in [8] with a method that estimates the blur extent, $\vec{b}(\vec{x},t)$, i.e. the length of the blur along the motion vector, using the autocorrelation of the signal's directional intensity derivative. This method was extended in [9] to be able to cope with motion gradients, next to uniform motions. In this section, we will present a TAW estimation algorithm based on the camera blur estimation from [8] that is suitable for arbitrary motions.

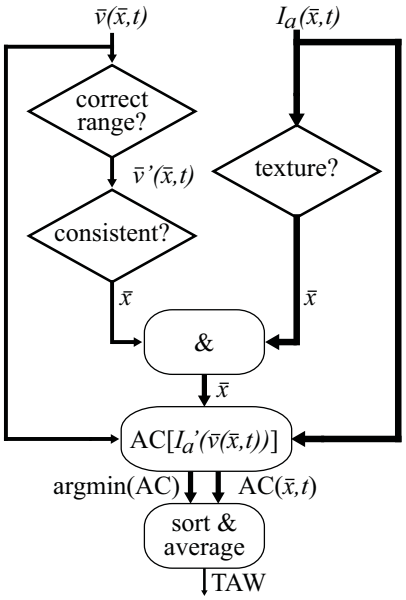


Fig. 3. Camera blur estimation is performed on selected locations. Candidates are discarded based on motion vector size and consistency and based on local image detail (texture). The TAW follows from multiple measurements of the location of the minimum of the AC of the picture derivative along the motion vector.

One of the main limitations of the methods presented in [8] and [9] is a reliable motion estimate. For video, however, a motion estimator can more reliably determine a motion vector at each position in the image. This simplifies the camera blur estimation, as it allows to normalize the estimated blur extent to a blur *time* that directly corresponds to the TAW, according to:

$$a_t = \frac{||\vec{b}(\vec{x},t)|| \cdot \Delta T}{||\vec{v}(\vec{x},t)||}, \quad (4)$$

with ΔT the picture period. Furthermore, the TAW can be expected constant within a picture sequence, allowing for locally selective measurements and the exploitation of temporal consistency to improve robustness by combining multiple measurements.

The camera blur estimation algorithm determines the autocorrelation (AC) along the motion vector at selected locations for the signal derivative. The position of the first minimum in the AC corresponds to the blur length, as explained by Fig. 2.

Combining measurements improves the robustness of the TAW estimate, but further robustness is obtained by discarding measurements that are likely to yield erroneous results, such as measurements at:

- **high speeds:** The video signal is attenuated as a function of motion. Faster motions result in larger blur extents, while noise can be expected constant. As a result, the SNR, and therefore the reliability of the TAW estimate, reduces for increasing speeds.
- **very low speeds:** For low speeds, the influence of the

spatial camera aperture cannot be ignored. The influence of focus blur, scaling and digital compression would typically lead to an over-estimate of the camera blur.

- **flat areas:** A low SNR reduces the reliability of the camera blur estimate. Therefore, flat areas are detected using a local activity metric and removed from the blur extent measurement.
- **motion edges:** The expression of camera blur as stated in Eq. 3 does not hold when locally the motion vector is not constant. In this case, the estimates at edges in the motion vector field are unreliable, even if the motion estimates themselves are accurate. The locations are found by verifying the consistency of the motion along its vector. Locations that fail the consistency check are discarded.

The robustness is further improved by using the resemblance of the autocorrelation function with respect to the ideal shape, shown in Fig. 2(c), as a confidence measure. For example by averaging the results for those measurements meet a certain confidence threshold. This is schematically shown in Fig. 3.

It has been demonstrated that, for HD TV-signals [4], this method is quite resilient to both analog and MPEG noise, and can cope with low contrasts and focal blur. The method fails to provide an accurate estimate only in case of very little motion, or in case of very high noise levels in the picture or motion vector field. In these cases, however, it is unlikely that camera blur reduction will yield a significant improvement in picture quality.

B. Inverse filtering

The camera blur estimate, a_t, and the motion estimate, $\vec{v}(\vec{x},t)$, enable the implementation of the sharpness enhancement filter described by Eq. 3. We refer to this method as inverse filtering, although it is not the goal to obtain the best approximation to the theoretical inverse filter. Instead, we attempt to invert the perceived reduction in sharpness. The infinite gains of the amplitude response complicate a practical implementation of the inverse filter and an approximation is required. An implementation that uses a FIR filter, known as Motion Compensated Inverse Filtering (MCIF), has been proposed in [5] and [10] for display blur reduction.

For display applications, MCIF filtering applies a spatial filter for each pixel in the picture along the motion vector that corresponds to the motion tracking of the HVS. Therefore, a 1D FIR-filter is rotated and scaled to align with the motion vector. In practice, such filtering can be implemented by fetching the luminance of $I_a(\vec{x},t)$ at an interpolated pixel grid along this scaled and rotated vector using e.g. bilinear interpolation. For camera blur, the FIR filter also scales linearly with the TAW as illustrated in Fig. 4.

In this section, we will discuss MCIF for camera blur reduction using two inverse filter design methods: Luminance Transient Improvement (LTI) [11] and Trained Filtering (TF) [12].

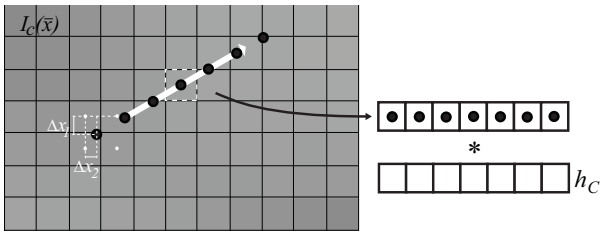
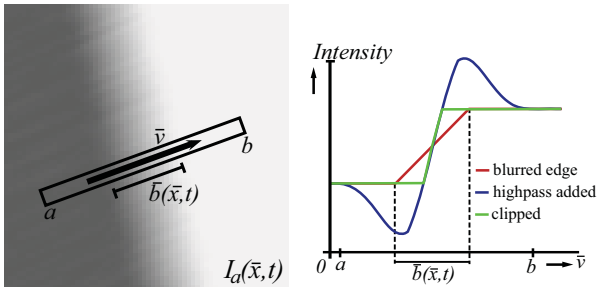


Fig. 4. Filtering along the motion vector requires fetching of intensity values on a non-integer grid along the (scaled) motion vector. The scaled motion vector, indicated by the white arrow, determines the position of the sample positions, indicated by the black dots. The intensity values at these locations are calculated from the nearest intensities. This is illustrated for the bottom left sample position. In case of bilinear interpolation, its intensity is calculated from the surrounding 4 pixel locations (white dots), using the distances Δx_1 and Δx_2 to proportionally weight the intensities.

The LTI method, known from static sharpness enhancement, can be explained as a two-step process. First, a sharpness enhancement filter reduces camera blur at the cost of overshoots at edges. Second, the overshoots at edges are eliminated by clipping the filter output to the local intensity extremes, as illustrated for a 1D signal (along the motion vector) in Fig. 5.

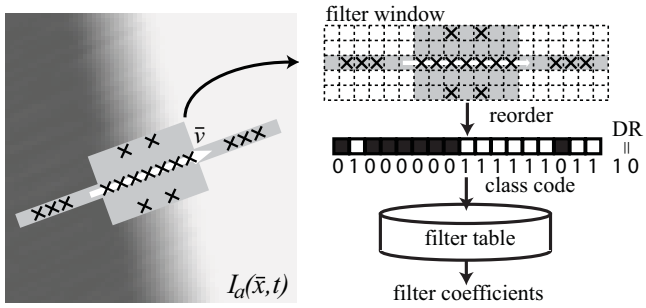
Note that the method does not attempt to approximate an ideal inverse filter, instead it enhances mid-frequencies, limiting the enhancement of noise. This filter method was found to perform well at edges, but yields only a modest sharpness enhancement in textured areas. Also, this method was found to be still sensitive to noise. Improved performance and robustness is obtained by linking a noise estimator to the filter gain and by making the filter coefficients adaptive to the local picture structure.

A second filter design strategy uses a TF method to find Mean Square Error (MSE) optimal inverse filters. With a training set of degraded and "original" video data, containing motion blurred and motion-blur free content, respectively, the filter that obtains the minimum difference between the two data sets can be determined. This requires a classification method to sort data similarity, an energy function that defines



(a) Fetching pixels. (b) Filtering and clipping.

Fig. 5. The LTI method: pixels are fetched at line segment [a-b] at an interpolated grid along the motion vector, $\vec{v}$. The FIR filter is scaled to match the blur length. A highpass filter is added to the blurred edge (red line), yielding overshoots at edges (blue line). Clipping to the local intensity extremes of the blurred edge reduces the overshoots (green line).



(a) (b)

Fig. 6. The TF method: (a) The filter window consists of a line along the motion vector and a box around the center pixel containing 17 sample locations. The intensities within this window are fetched from an interpolated grid. (b) A 19-bit class code is constructed by comparing the sampled intensities with the window average, yielding 17 bits, together with a 2-bit code for the DR. The class code is the index of the filter table.

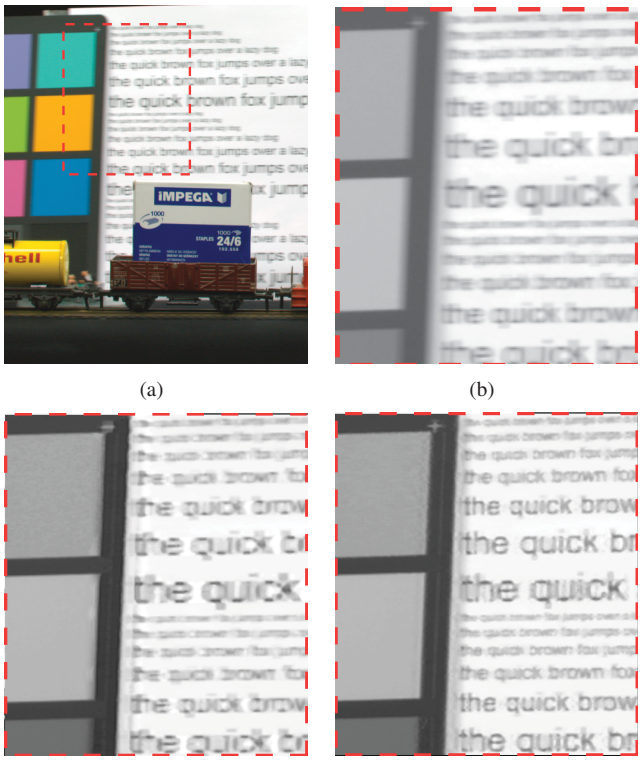
a quantifiable difference metric, and a representative data set of sufficient size. The classification encodes local picture structure using a window aligned along the scaled motion vector. In order to constrain the number of classes to a practical number for training, the number of pixels in the window is limited and the structure is typically encoded with one bit per pixel, e.g. "0" and "1" correspond to a lower or higher luminance value compared to the window average, respectively.

The classification process for camera blur reduction has been thoroughly researched in [13]; an extensive search of all possible symmetric filter structures using up to 19-bits for the classification, as illustrated in Fig. 6(a), using one bit per pixel and an additional two bits to encode the Dynamic Range (DR), yields an MSE-optimal filter structure as illustrated in Fig. 6(b). For this evaluation, a test set of 145 HD images was used. The robustness to noise is obtained with the encoding of the dynamic range, while the content adaptation is encoded by the filter structure. The disadvantage of this method is the large filter table that is required to store all filter coefficients. Furthermore, a re-training is required for every change in (the tuning of) the classification.

Special care is required for the borders of moving objects, i.e. occlusion areas. The inverse filter should take occlusion areas into account and discriminate between covering and uncovering areas, because the motion blur model, described by Eq. 3 does not hold here. Visible artifacts at occlusion areas can be alleviated by filling in the occlusion data from a previous and next picture, or by gradually limiting the enhancement at occlusion areas.

III. Results

The MCIF filter using the LTI and TF methods have been applied to the example HD picture shown in Fig. 7(a) containing a partial panning motion of 30 pixels per frame. The HD picture has been registered with a TAW of $a_t = 0.5$. The filtering methods have been applied to the luminance signal



(a) (b) (c) (d)

Fig. 7. (a) A picture from the *Trainloop* sequence, containing a horizontal motion of 30 pixels per frame, registered with a TAW of $a_t = 0.5$. (b) A close-up of the luminance component, illustrating the camera blur. (c), (d) The same close-up after applying the LTI and TF methods, respectively.

only. The filter outputs are illustrated in Fig. 7(c) and Fig. 7(d).

In addition, the camera blur reduction methods have been applied to a test set and evaluated with an informal subjective assessment on a high frame LCD-TV. From this assessments, we found the LTI method to be most robust to noise, but limited in reconstructing details in textured regions. The trained filter method was found to perform best for recreating details, without creating visible overshoots at edges, although a tuning for several types of picture degradations was found more cumbersome. Further work is required to quantify the quality improvement of these methods and to determine which is most suitable for camera blur reduction in LCD-TV applications.

IV. Discussion

To achieve the best perceived picture quality when implementing camera blur reduction for LCD-TVs, other picture quality improvements that might be present in between a TV's picture reception and rendering must be taken into account. In particular, the combination of MCIF with spatial sharpness enhancement (SSE) [11] and MC-FRC (for display blur reduction) cannot be considered independently.

MC-FRC and MCIF both use motion vector estimates and have to tackle visible artifacts that can appear when moving objects occlude. In addition, the execution order of

MC-FRC and MCIF need to be considered. Applying MCIF after MC-FRC results in higher computational requirements, while applying MCIF before MC-FRC, influences the frame interpolation of MC-FRC.

When combining MCIF with SSE, in general, care has to be taken that the sharpness enhancements of both processing steps do not "overlap". In particular for low speeds, MCIF can influence the spatial sharpnesS, causing an over-sharpening in combination with SSE.

From informal experiments, we found that visible artifacts that result from combining MCIF with SSE or MC-FRC are highly implementation specific and, therefore, need to be addressed on a case-by-case basis.

V. Conclusions

For video and film content, motion blur can only be reduced on high frame rate LCD-TVs by reducing camera blur. The blur characteristics of camera blur are described by the TAW which needs to be estimated from the TV-signal. MCIF is required for camera blur reduction. A camera blur estimation method, two filter design strategies and a system evaluation have been implemented. The TF method was found to perform best in recreating details, while the LTI method was found to be most robust to noise. To implement MCIF for TV applications, the combination with SSE and MC-FRC need to be considered to optimize picture quality and required computational resources.

References

[1] F. H. van Heesch and M. A. Klompenhouwer, "Video processing for optimal motion portrayal," in *Proc. of the IDW*, 2006, pp. 1993–1996.

[2] Hyun-G et al., "45.4: High Frequency LVDS Interface for F-HD 120Hz LCD TV," *SID Symposium Digest of Technical Papers*, vol. 39, no. 1, pp. 681–684, 2008.

[3] M. A. Klompenhouwer, "22.2: Dynamic Resolution: Motion Blur from Display and Camera," *SID Symposium Digest of Technical Papers*, vol. 38, no. 1, pp. 1065–1068, 2007.

[4] W. Yarde, "Estimating motion blur on liquid crystal displays," Master's thesis, Eindhoven University of Technology, 2009.

[5] M. A. Klompenhouwer and L. J. Velthoven, "Motion blur reduction for liquid crystal displays: Motion compensated inverse filtering," in *Proc. of the SPIE*, vol. SPIE-5308, 2004, pp. 690–699.

[6] F. H. van Heesch and M. A. Klompenhouwer, "26.4: The Temporal Aperture of Broadcast Video," *SID Symposium Digest of Technical Papers*, vol. 39, no. 1, pp. 370–373, 2008.

[7] T. M. Cannon, "Blind deconvolution of spatially invariant image blurs with phase," *ASSP*, vol. 24, no. 1, pp. 58–63, February 1975.

[8] Y. Yitzhaky and N. S. Kopeika, "Identification of the Blur Extent from Motion Blurred Images," *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 310–320, 1997.

[9] H. J. Trussell and S. Fogel, "Restoration of spatially variant motion blurs in sequential imagery," *Image Processing Algorithms and Techniques II*, vol. 1452, no. 1, pp. 139–145, 1991.

[10] C. Dolar, "LCD-Modelle und ihre Anwendung in der Videosignalverarbeitung," Ph.D. dissertation, Technische Universität Dortmund, 2010.

[11] J. A. P. Tegenbosch et al., "Improving nonlinear up-scaling by adapting to the local edge orientation," *Visual Communications and Image Processing 2004*, vol. 5308, no. 1, pp. 1181–1190, 2004.

[12] M. Zhao, "Video Enhancement Using Content-adaptive Least Mean Square Filters," Ph.D. dissertation, Eindhoven University of Technology", 2006.

[13] G. Kwintenberg, "Motion-Blur Reduction for Liquid Crystal Displays," Master's thesis, Eindhoven University of Technology, 2009.