A New Similarity Measure for Sub-Pixel Accurate Motion Analysis in Object-Based Coding

Dirk Farin
Dept. Circuitry and Simulation, University Mannheim
68131 Mannheim, Germany

and

Peter H.N. de With
CMG Eindhoven / University of Technology Eindhoven
Eindhoven, Netherlands

Abstract

A new similarity measure for motion-estimation and motion-compensated segmentation applications is presented. The new measure shows low sensitivity to image noise, is more tolerant to small errors in the motion model and it provides a sub-pixel accuracy which is comparable to conventional measures like mean-square error. The proposed measure shows a linear increase of error as a function of the displacement. This beneficial property has been exploited to design a novel motion-estimation algorithm with fast convergence.

Keywords: sub-pixel motion-estimation, similarity-measures, segmentation.

1 Introduction

In video coding systems such as the MPEG standard, motion estimation and compensation are performed for predicting the image to be compressed. A principal role in the design of motion-estimation algorithms is the selection of a measure for comparing parts of two images. The motion estimation process attempts to optimize this *similarity* measure (i.e. minimizing the difference between the images). Many different measures have been defined, for example *Mean Square Error* (MSE), which results in a least-squares approximation, the *Sum of Absolute Differences* (SAD), which is faster to compute with comparable accuracy and *cross-correlation*. However, all previous examples have the disadvantage that they assume the pixels to be compared to lie on a defined grid position. With non-integer motion displacements, this assumption is not satisfied. Traditionally, this problem has been solved by interpolating the values at non-integer positions [1]. Unfortunately, this approach is not robust to image noise. Even worse, small errors in the motion-model lead to large errors at sharp edges in the images. This imposes difficulties with many camera-motion compensated segmentation-algorithms [3]. A common technique in *video-sequence segmentation* for finding objects is to estimate the background movement with a parametric motion model. By comparing background-motion compensated successive images, areas can be identified that do not match with the image contents of the preceding picture. These areas are assumed to be covered by foreground objects, moving in a different direction. The similarity measure is applied to discriminate foreground objects from the background. Thus, large matching errors in the difference of successive motion-compensated pictures at edges and textured regions can lead to background areas which are classified as foreground objects.

The objective of this paper is to present a new measure which is more tolerant to small motion-model errors, yet having the same estimation accuracy as MSE using interpolated pixels. It will be shown that the new measure has also resulted in the design of a fast motion-estimation algorithm.

1.1 Sub-Pixel Accurate Matching

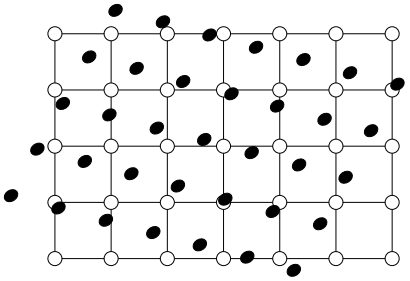
Let $f(x, y)$ be the luminance of a two-dimensional source image at spatial coordinates (x, y) and let $g(x, y)$ be a *template* image (the pattern to be searched), defined over a domain $G \subseteq \mathbb{R} \times \mathbb{R}$. We consider the problem of comparing the deformed template image with the source image f. The template motion is described by an arbitrary transformation $(x', y') = T(x, y)$.

A popular similarity measure for comparing two images is the Mean-Square Error (MSE), which in the case of continuous images is defined as:

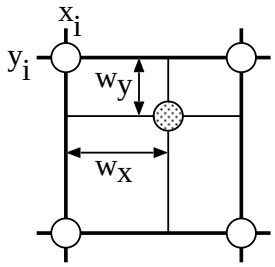
$$d_{c-MSE} = \frac{1}{|G|} \iint_{(x,y)\in G} \Big(f\big(T(x,y)\big) - g(x,y) \Big)^2 \,\mathrm{d}x\,\mathrm{d}y.$$

However, in practical applications, images are stored as an array of space-discrete values, sampled on an orthogonal grid with integer coordinates. This discretization imposes a problem if the transformation T allows integer pixel-positions to be mapped on non-integer positions. Therefore, means are required for comparing a template pixel at an integer position with a source image pixel at a non-integer position (see Figure 1a).

The simplest approach to this problem is to round the non-integer positions to the nearest source image pixel po-



(a) Template pixels (black dots) lying on a different raster than the source image (white dots).



(b) Bilinear interpolation; $w_x, w_y \in [0;\ 1)$.

Figure 1: Definition of non-integer positions of pixels for comparison.

sitions. However, this has the disadvantage that the motion parameters cannot be calculated with high precision. We will refer to the MSE measure using this approach as *n-MSE*:

$$d_{n-MSE} = \frac{1}{|G|} \sum_{(x,y)\in G} \big(f(\lfloor T_x(x,y) + 0.5 \rfloor, \lfloor T_y(x,y) + 0.5 \rfloor) - g(x,y)\big)^2,$$

where T_x, T_y denote the x, y elements of the transformed coordinates, respectively.

Another common technique is to interpolate the source-image pixels around the desired position (x, y) to obtain an estimate of the value $f(x, y)$, even though the function is not defined at that position. The mostly used interpolation formula is simple bilinear interpolation, given by

$$\begin{aligned} f(x,y) = &(1-w_x)(1-w_y) f(x_i, y_i) \\ &+ w_x(1-w_y) f(x_i+1, y_i) \\ &+ (1-w_x) w_y f(x_i, y_i+1) \\ &+ w_x w_y f(x_i+1, y_i+1) \end{aligned}$$

with $x_i = \lfloor x \rfloor$, $y_i = \lfloor y \rfloor$, $w_x = x - x_i$, and $w_y = y - y_i$ (see Figure 1b). In the sequel, we call the combination of a bilinear interpolation and the MSE measure the *i-MSE* measure.

1.2 i-MSE on Noisy Images

In this section, we evaluate the dependence of i-MSE on camera noise in the video sequence. We assume that this camera noise is additive Gaussian noise. Thus, the image f can be written as $f(x, y) = f'(x, y) + n(x, y)$, with f' being the noiseless input image and $n(x, y)$ the output of a random process having the probability distribution $p(n) = 1/(\sigma\sqrt{2\pi}) \exp(-n(x,y)^2/(2\sigma^2))$. To evaluate the influence of interpolation on the i-MSE measure, we consider a constant input image $f'(x, y) = 0$ and a simple horizontal shift transformation $T(x, y) = (x + \Delta x, y)$. We calculate the expectation value for the i-MSE depending on the horizontal shift Δx as

$$\begin{aligned} E\{d_{i-MSE}\} &= E\{((1-\Delta x)\cdot f(x,y) + \Delta x \cdot f(x+1,y) \\ &\qquad - g(x,y))^2\} \\ &= E\{((1-\Delta x)\cdot \underbrace{n(x,y)}_{n_1} + \Delta x \cdot \underbrace{n(x+1,y)}_{n_2})^2\} \\ &= \iint_{n_1,n_2=-\infty}^{\infty} \big((1-\Delta x)n_1 + \Delta x n_2\big)^2 \\ &\qquad \cdot \frac{1}{2\pi\sigma^2} e^{-(n_1^2+n_2^2)/2\sigma^2} \mathrm{d}n_1 \mathrm{d}n_2. \end{aligned}$$

Using the previous model, this expectation value is plotted in Fig. 4a. It is clearly visible that the matching error reaches a local minimum for the non-integer shift $\Delta x = 0.5$. When comparing this theoretical result with real-world data (Fig. 6a), it can be observed that the model matches well with the measured data.

As a consequence, it follows that motion-estimation should be performed without interpolation (i.e. n-MSE should be used) until the estimation converges to a minimum. Sub-pixel accuracy can be obtained by performing a subsequent separate processing step, such as switching to i-MSE.

2 New Measure: Sum of Spatial Distances (SSD)

Many surfaces in real-world images are not completely flat but have a structured surface (carpets, grass, etc.). A simple projective motion-model cannot describe the complicated three-dimensional motion which is caused by camera or object movements. Generally, it is therefore not possible to match the motion-compensated template exactly with the source image, leading to a small difference between the real motion and the estimation. This small error of pixel displacement results in large matching errors at sharp edges, because the luminance difference is a quadratic term in the MSE measure. A comparable effect is caused by geometric lens-distortions and blurring due to a limited camera aperture.

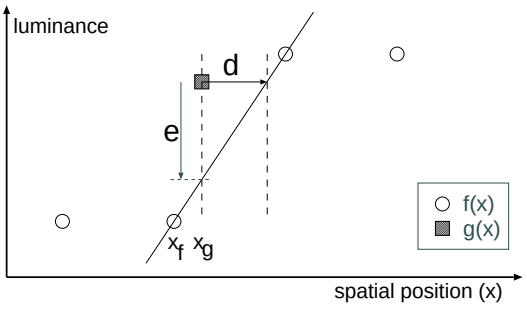
Motivated by these observations, we developed a new measure, which is more tolerant to small displacements of steep edges. Similar to the interpolated sub-pixel search, we also assume a linear interpolation of the source image in the vicinity of the computed template pixel position. However, instead of using a squared difference of the

pixel luminance (MSE), we propose to calculate a measure based on the local deformation of the template which is needed in addition to the motion model to match with the real motion.

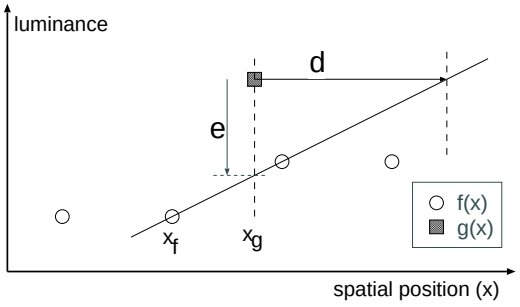
2.1 One-Dimensional Case

For clarity, we first illustrate the new measure in the one-dimensional case (Figure 2). To calculate the error for the template pixel $g(x)$, which is mapped by T on $x_g = T(x)$, we consider the two neighboring pixels in the source image. Since $x_g \in \mathbb{R}$, the two neighboring pixel positions on integer positions are located at $x_f = \lfloor x_g \rfloor$ and $x_f + 1$. With linear interpolation between these two source image pixels, the error d now computes as $(g(x) - f(x_f))/(f(x_f + 1) - f(x_f)) - (x_g - x_f)$. The first term originates from the horizontal distance along the gradient until a pixel of equal brightness is reached, while the second term is the initial distance of the sub-pixel position to the next integer pixel position to the left. In cases where the nominator approaches zero, we limit its value to 1. As opposed to the i-MSE measure, which considers the error e, the SSD error is reduced at edges. This increases the tolerance to luminance variations at edges and areas with fine texture. Summing over all pixels in the template image, we obtain the SSD measure as

$$d_{SSD} = \frac{1}{|G|} \sum_{x \in G} \left| \frac{g(x) - f(x_f)}{f(x_f + 1) - f(x_f)} - (x_g - x_f) \right|.$$



(a) High contrast edge.



(b) Low contrast edge.

Figure 2: Matching-process at an edge. The source-image pixels $f(x)$ are drawn as circles and a pixel of the template $g(x)$ is represented by a square.

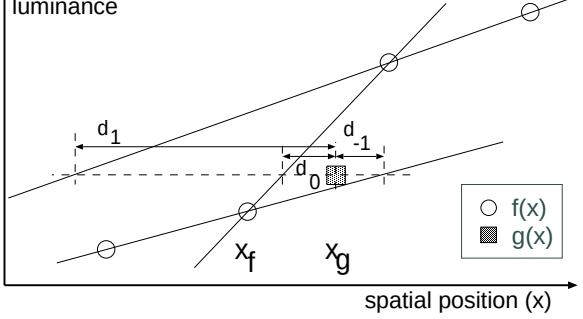


Figure 3: Principle of SSD-3 measure

The SSD measure can be extended in a natural way by considering a larger context of the source image. This increases the robustness in cases where $x_g - x_f$ is about 0 or 1. In order to do this, we have defined the SSD-3 measure, in which case not only one pair of neighboring pixels is examined, but three pairs. The luminance gradients and the corresponding errors d_{-1}, d_0, d_1 are calculated for each of the pixel pairs $(x_f - 1;\ x_f), (x_f;\ x_f + 1), (x_f + 1;\ x_f + 2)$. Consequently, d_i is calculated by means of the pixels $x_f + i$ and $x_f + i + 1$ as (see Figure 3)

$$d_i = \left| \frac{g(x) - f(x_f + i)}{f(x_f + i + 1) - f(x_f + i)} - \big(x_g - (x_f + i)\big) \right|.$$

Note that the special case d_0 is exactly the definition used in the SSD measure mentioned earlier. SSD-3 is now defined by taking the minimum distance,

$$d_{SSD-3} = \frac{1}{|G|} \sum_{x \in G} \min(d_{-1}, d_0, d_1).$$

2.2 Two-Dimensional Case

So far, we have only considered the one-dimensional case of the SSD-measure. For the two-dimensional case, we calculate the 1-D d_{SSD-3} independently for the horizontal and vertical direction (d^H_{SSD-3}, d^V_{SSD-3}, respectively) and define

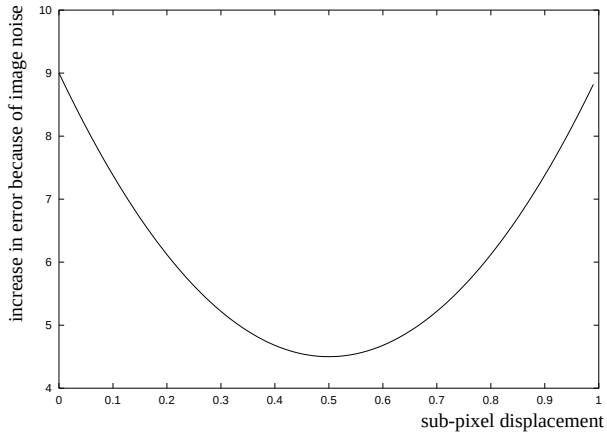
$$d^{2D}_{SSD-3} = \min(d^H_{SSD-3}, d^V_{SSD-3}).$$

2.3 SSD-3 on Noisy Images

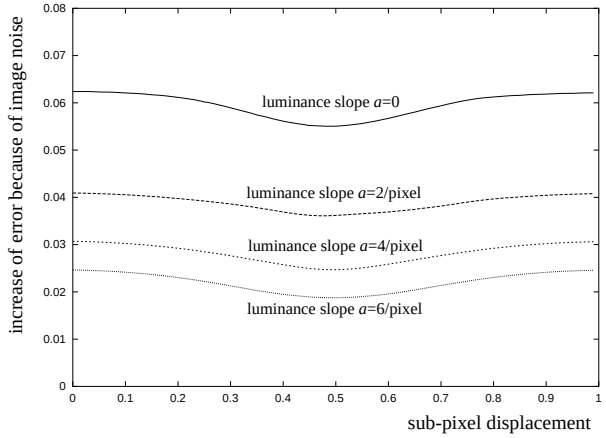
Similar to the approach in Section 1.2, we evaluate the dependence of the SSD-3 measure to image noise. However, in this case, the influence of noise depends on the image luminance gradient. Thus, we assume an increase of a luminance units per pixel and by setting $f'(x_f - 1) = -a$, $f'(x_f) = 0$, $f'(x_f + 1) = a$, and $f'(x_f + 2) = 2a$, we obtain the following expression for the expectation of SSD-3 noise

$$\begin{aligned} E\{d_{SSD-3}\} &= E\{\min(d_{-1}, d_0, d_1)\} \\ &= E\Bigg\{\min\Bigg(\frac{w_x a - (-a + n_3)}{n_1 - (-a + n_3)} - 1 - w_x, \\ &\qquad \frac{w_x a - n_1}{a + n_2 - n_1} - w_x, \\ &\qquad \frac{w_x a - (n_2 + a)}{(2a + n_4) - (a + n_2)} + 1 - w_x\Bigg)\Bigg\}. \end{aligned}$$

For the evaluation, we assume a Gaussian distribution for the random variables $n_1, \ldots, n_4$. Results of the numeric evaluation of this expression for different values of a are shown in Figure 4b. The influence of image noise is still visible, but it is heavily reduced compared to the i-MSE measure. It is also apparent that the influence of noise decreases with increasing luminance gradient a.



(a) sub-pixel noise for i-MSE measure



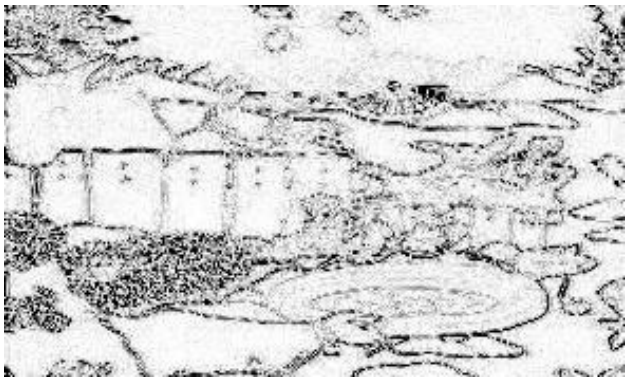
(b) sub-pixel noise for SSD-3 measure at different luminance gradients

Figure 4: Interpolation of pixel value at non-integer position, $w_x \in [0;\ 1)$.

3 Results

We have compared the SSD-3 measure with the i-MSE measure. The well-known interlaced "mobile" sequence was split into its two fields and only the top field was used to eliminate difficulties originating from the interlaced format of the sequence. The image area considered was limited to the upper-left region of the image to obtain a homogeneous motion-model for the whole image. The motion in this area is a shift to the right while the camera is slowly zooming out. As it can be assumed that the picture contents is drawn on a flat plane, the camera motion can be described by a perspective motion-model.

We implemented an 8-parameter perspective-motion estimator and calculated the distance of each pair of corresponding pixels of two consecutive images with both the i-MSE and the SSD-3 measure. Figure 5a portrays that the i-MSE measure is not capable to compensate the error at the edges. The SSD-3 measure, on the other hand, is more tolerant to small displacements of the edges and shows mostly uniform low values. This alleviates the design of video segmentation algorithms that decide if an object is present at a specific position in the image by examining the error map.

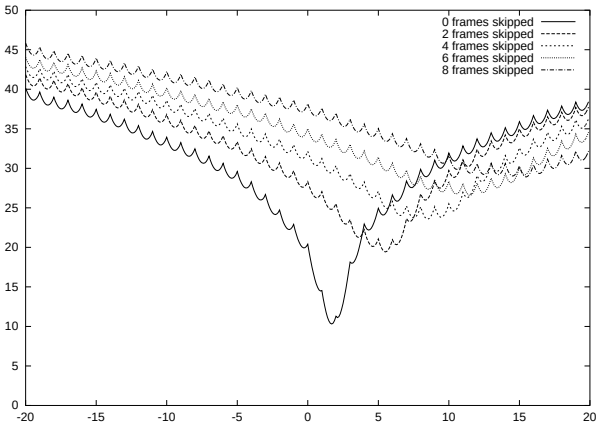


(a) difference image, i-MSE measure

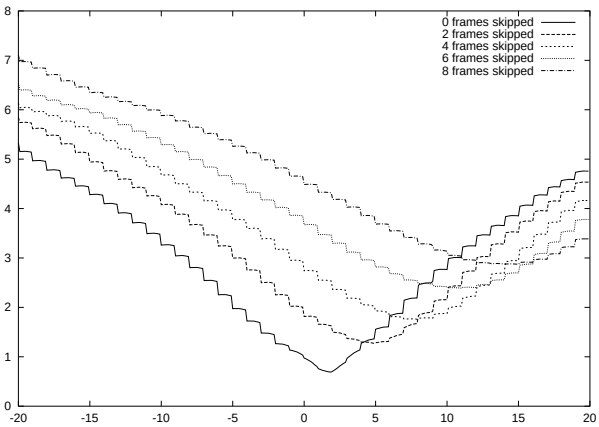
(b) difference image, SSD-3 measure

Figure 5: Error map of two camera-motion compensated images.

In a second experiment, we measured the total error depending on the horizontal displacement between a fixed pair of images. As is visible in Figure 6a, the i-MSE measure exposes a large number of local minima. In fact, a local minimum exists between every integer displacement value. The SSD-3 measure (Figure 6b) does not show this unfavourable effect. For large displacements, the error increases with steps at integer shifts, but for displacements near the global error-minimum, the error function is very smooth. This enables the use of simple gradient-descent algorithms for finding the minimum.



(a) different displacements, i-MSE



(b) different displacements, SSD-3

Figure 6: Error of similarity-measure for a wide search-range. To get larger displacements, several frames of the input have been skipped.

3.1 Fast Motion-Estimation Algorithm

The unit of the SSD-measure is spatial distance. Thus, the measure directly expresses an amount of translation. Experiments with real-world images have shown that the SSD-3-measure increases almost linearly with the displacement from the minimum (Figure 7a). This suggests to use the Newton root-finding algorithm for motion estimation. The outline of such a fast motion-estimation algorithm, which we examined, is as follows (for simplicity described for the one-dimensional case):

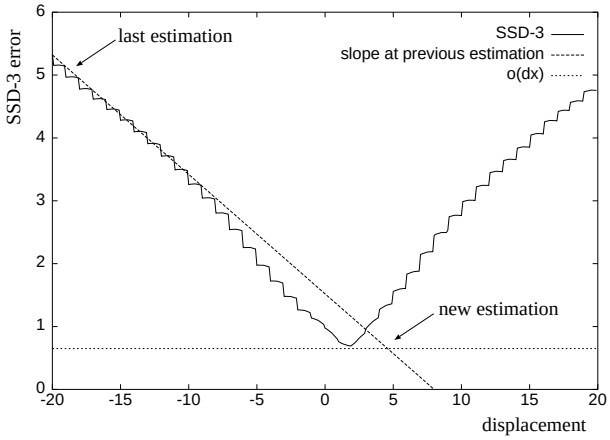
1. Start with an arbitrary, integer displacement Δx_0 ($\Delta x_0 = 0$ for example), $e_{min} = \infty$.
2. Calculate a new estimate for Δx_{i+1} according to

$$\Delta x_{i+1} = \left\lfloor \frac{d_{SSD-3}(\Delta x_i) - o(\Delta x_i)}{d_{SSD-3}(\Delta x_i) - d_{SSD-3}(\Delta x_i + 1)} + \Delta x_i \right\rfloor .$$

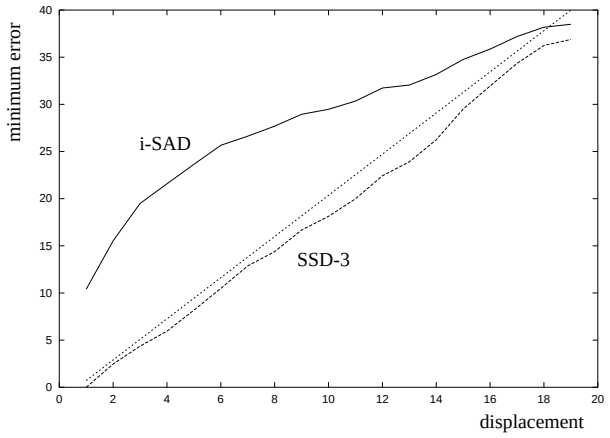
3. If $d_{SSD-3}(\Delta x_{i+1} < e_{min})$, set $e_{min} = d_{SSD-3}(\Delta x_{i+1})$ and repeat step 2.
4. Use a gradient-descent method to find the minimum [2].

The value $o(\Delta x)$ is an estimate for the minimum total error that will be obtained for a displacement of Δx. This value can be calculated on-the-fly after each search. If the total error e_{min} for a displacement Δx is lower than $o(\Delta x)$, then $o(\Delta x)$ is set to e_{min}. Our experiments show a linear increase of $o(\Delta x)$ for increasing values of $|\Delta x|$. This advantageous property can be used to calculate a good initial estimation of $o(\Delta x)$ for those Δx for which $o(\Delta x)$ it is not defined yet.

Simulations have shown that only 1-3 iterations of step 2 are necessary to converge even for large displacements ($\pm$ 20 pixels). The final gradient-descent search converges after about 3 steps.



(a) fast motion-estimation based on SSD3



(b) minimum error at various displacements

Figure 7: Principle of fast motion-estimation algorithm.

4 Conclusions

A new measure for the comparison of images to support motion estimation has been presented. It has been shown that it is more noise robust and tolerant to small motion-model errors as conventional measures such as the MSE and SAD. Evaluation of the error of our new measure shows a linear increase of error with increasing displacement from the real motion. This linearity enables the use of the Newton root-finding technique for the definition of a fast motion-estimaton algorithm.

A topic of further research is the integration of the new measure into an complete segmentation algorithm to evaluate the performance in practice. Moreover, the possibility to generalize the presented motion-estimation algorithm to affine or perspective motion-models has to be evaluated further.

References

[1] Sean Borman, Mark Robertson, and Robert Stevenson. Block-matching sub-pixel motion estimation from noisy, under-sampled frames - an empirical performance evaluation. In *SPIE Visual Communications and Image Processing*, 1999.

[2] Oscal T.-C. Chen. Motion estimation using a one-dimensional gradient descent search. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(4):608–616, June 2000.

[3] Roland Mech and Michael Wollborn. A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera. Technical report, University Hannover.