# Residual Images Remove Illumination Artifacts for Correspondence Algorithms!

Tobi Vaudrey and Reinhard Klette

The *.enpeda..* Project, The University of Auckland
Auckland, New Zealand

**Abstract.** Real-world image sequences (e.g., recorded for vision-based driver assistance) are typically degraded by various types of noise, changes in lighting, out-of-focus lenses, differing exposures, and so forth. In past studies, illumination effects have been proven to cause the most common problems in correspondence algorithms. We address this problem using the concept of *residuals*, which is the difference between an image and a smoothed version of itself. In this paper, we conduct a study identifying that the residual images contain the important information in an image. We go on to show that they remove illumination artifacts using a mixture of synthetic and real-life images. This effect is highlighted more drastically when the illumination and exposure of the corresponding images is not the same.

## 1 Introduction

This paper applies the structure-texture image decomposition [2, 16] as basic approach for evaluating preprocessing options for image sequences, as recorded in vision-based driver assistance systems (DAS). Currently, technologies are introduced into the DAS market which apply stereo and motion analysis for solving early vision tasks; see [8]. However, the improvement of those modules will be an ongoing challenge for some time to come.

In particular, when evaluating stereo and motion correspondence algorithms on real-world sequences as provided on [7], we realized [20] that illumination artifacts define a major issue, causing serious reductions in accuracy of stereo and motion data.

There might be basically two different approaches for dealing with this problem, either we try to map both images into a uniform illumination model, or we map both into images which carry the illumination-independent information. After some experiments with various unifying mappings we realized that the first approach is basically impossible (or, at least, a very big challenge), considering that impacts of shadow are often just local (e.g., "dancing lights" caused by sunshine through leaves along the road). Thus we moved on to the second approach, and this paper actually shows that this is a very promising direction of research.

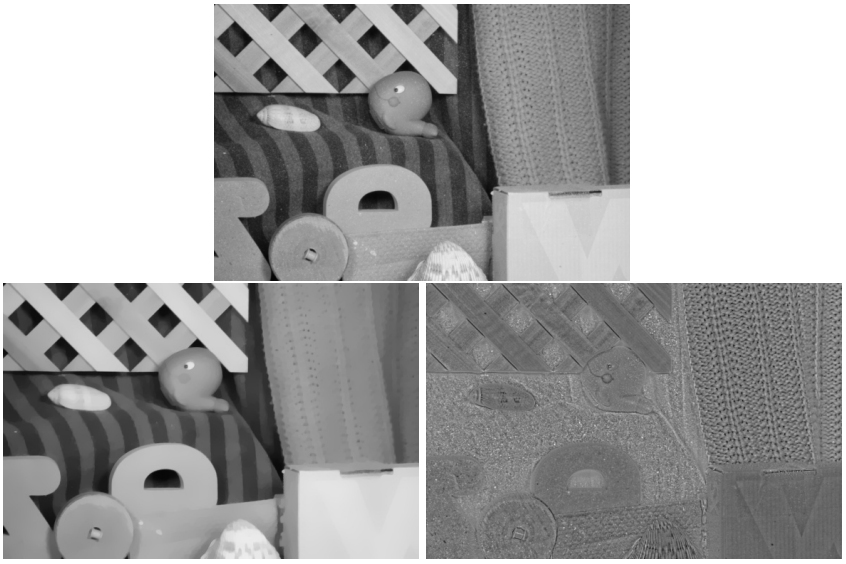

**Fig. 1.** Example decomposition of *RubberWhale* image (top) into its smooth (left) and residual (right) components (example using TV-L$^2$).

For this second approach, we picked up the concept of *residuals* [11], which is the difference between an image and a smoothed version of itself, and generalized it by applying not only the mean operator for smoothing, but also various smoothing operators as known from past and very recent studies in computer vision. (This also includes a small modification of an operator proposed in [12].)

Let $f$ be an image with an additive decomposition $f(x,y) = s(x,y) + r(x,y)$, for all pixel positions $\mathbf{x} = (x,y)$, where $s = S(f)$ denotes the *smooth component* (of an image) and $r = R(f) = f - S(f)$ the *residual*. The residuum is not a (standard) image because it may also contain negative values. See Figure 1 for an example of such a decomposition. We use the straightforward iteration scheme:

$$\begin{aligned} s^{(0)} &= f \\ s^{(n+1)} &= S(s^{(n)}) \\ r^{(n+1)} &= f - s^{(n+1)} \end{aligned}$$

for $n \geq 0$.[1] Co-occurrence matrix [9] based information measures are used to characterize information in $s^{(n)}$ and $r^{(n)}$, for $n \geq 1$.

This paper conducts a study identifying that the residuals $r^{(n)}$ contain the important information in an image. We go on to show that they remove

[1] However, we could also have used $r^{(n+1)} = s^{(n)} - s^{(n+1)}$, or even a different iteration scheme with $r^{(0)} = f$ and $s^{(n+1)} = S(r^{(n)})$, with $r^{(n+1)} = r^{(n)} - s^{(n+1)}$, for $n \geq 0$. This might be a subject for further studies

illumination artifacts using a mixture of synthetic and real-life images. This effect is highlighted more drastically when the illumination and exposure of the corresponding images is not the same.

In this paper we first introduce the chosen smoothing operators in Section 2. This is followed by an overview of the data set we use. We go on to show that the smoothed image $s$ is a good approximation for a low-pass filter, and go on to show that the residual image $r$ does contain the high-frequencies required for correspondence matching (Section 4). Section 5 proposes a methodology to test if the illumination artifacts are, in fact, improved using residuals, then provides results using the proposed methodology. A conclusion and acknowledgments finalise this paper.

## 2 Smoothing Operators

Let $f$ be any frame of a given image sequence, defined on a rectangular open set $\Omega$ and sampled at regular grid points within $\Omega$. Technically, we assume that $f$ is a two-dimensional (2D) function in $L^2(\Omega)$ (i.e., informally speaking, square integrable on $\Omega$), which defines a surface patch above $\Omega$, whose contents (i.e., area) equals $\int_\Omega |\nabla f|$.[2] This integral of the gradient $\nabla f$ of $f$ is also called the *total variation* (TV) of $f$.

[16] assumed an additive decomposition $f = s + r$ into a *smooth component* $s$ and a *residual component* $r$, where $s$ is assumed to be in $L^1(\Omega)$ with bounded TV (in brief: $s \in$ BV), and $r$ is in $L^2(\Omega)$. This allows one to consider the minimization of the following functional:

$$\inf_{(s,r)\in BV\times L^2 \wedge f=s+r} \left( \int_\Omega |\nabla s| + \lambda ||r||^2_{L^2} \right) \tag{1}$$

The TV-L$^2$ approach in [16] was approximating this minimum numerically for identifying the "desired clean image" $s$ and "additive noise" $r$. Actually, further studies (see [2]) then identified $s$ to be the "structure", and $r$ to be the "texture". See Figure 1 for an example of such a TV-L$^2$ decomposition.

The concept may be generalized as follows: any *smoothing operator* $S$ generates a *smoothed image* $s = S(f)$ and a *residuum* $r = f - S(f)$. For example, TV-L$^2$ generates the smoothed image $s = S_{TV}(f)$ by solving Equ. (1). For example, $s$ may also be the result of an ideal low-pass Fourier filter, and $r$ would then be the result of the dual ideal high-pass Fourier filter (due to additivity of the Fourier transform).

Note that the residuum is not a (standard) image because it may also contain negative values; $r$ will be normalized later in this paper into a 2D function with values in [0,1]. (We may also use a *residual operator* $R$ with $r = R(f) = f - S(f)$; but, obviously, $S$ and $f$ already define both the low-frequency term $s$ and the high-frequency term $r$.)

[2] In case of a 1D function $f$, $\int_\Omega |\nabla f| = \int_a^b |f'(x)|\,\mathrm{d}x$ equals the length of the curve $f(x)$, for $x \in \Omega = [a, b]$.

The concept of residual images was already introduced in [11] by using a $3 \times 3$ mean for implementing $S$. We will include this simple smoothing operator $S_{mean}$ into our discussions in this paper. Figure 1 in [11] characterizes the histogram of a residuum $r = f - S_{mean}(f)$ as being a Laplacian distribution of values.

$S_{median}$ is another simple smoothing operator, defined by the $m \times m$ local median operator. Furthermore, the study [1] on comparing edge-preserving smoothing filters points to the (*double-window*) *trimmed mean* operator as introduced in [12]; we use the base principals of this for the trimmed mean filter $S_{TM}$. This smoothing operator uses a $m \times m$ window, but only calculates the mean only for all pixels with values in $[a - \sigma_f, a + \sigma_f]$, where $a$ is the central pixel value and $\sigma_f$ is the standard deviation of $f$. Finally, we also include the bilateral [19] and the trilateral filter [5], defining smoothing operators $S_{BL}$ and $S_{TL}$.

In the bilateral case, offset vectors $\mathbf{a}$ and position-dependent real weights $d_1(\mathbf{a})$ define a local convolution, and the weights $d_1(\mathbf{a})$ are further scaled by a second weight function $d_2$, defined on the differences $f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x})$:

$$s(\mathbf{x}) = \frac{1}{k(\mathbf{x})} \int_{\Omega} f(\mathbf{x}+\mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2\left[f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x})\right] \mathrm{d}\mathbf{a} \tag{2}$$

$$k(\mathbf{x}) = \int_{\Omega} d_1(\mathbf{a}) \cdot d_2\left[f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x})\right] \mathrm{d}\mathbf{a}$$

Function $k(\mathbf{x})$ is used for normalization. In this paper, weights $d_1$ and $d_2$ are defined by Gaussian functions with standard deviations $\sigma_1$ and $\sigma_2$, respectively. The smoothed function $s$ equals $S_{BL}(f)$. The bilateral filter requires a specification of parameters $\sigma_1$, $\sigma_2$, and the size of the used filter kernel in $f$.

The trilateral case only requires the specification of one parameter; it combines two bilateral filters. At first, a bilateral filter is applied on the derivatives of $f$ (i.e., the gradients):

$$g_f(\mathbf{x}) = \frac{1}{k_\nabla(\mathbf{x})} \int_{\Omega} \nabla f(\mathbf{x}+\mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2\left(||\nabla f(\mathbf{x}+\mathbf{a}) - \nabla f(\mathbf{x})||\right) \mathrm{d}\mathbf{a} \tag{3}$$

$$k_\nabla(\mathbf{x}) = \int_{\Omega} d_1(\mathbf{a}) \cdot d_2\left(||\nabla f(\mathbf{x}+\mathbf{a}) - \nabla f(\mathbf{x})||\right) \mathrm{d}\mathbf{a}$$

Simple forward differences

$$\nabla f(x,y) \approx (f(x+1,y) - f(x,y), f(x,y+1) - f(x,y))$$

are used for the digital image. For the subsequent second bilateral filter, [5] suggested the use of the smoothed gradient $g_f(\mathbf{x})$ [instead of $\nabla f(\mathbf{x})$] for estimating an approximating plane

$$p_f(\mathbf{x},\mathbf{a}) = f(\mathbf{x}) + g_f(\mathbf{x}) \cdot \mathbf{a} \tag{4}$$

Let $f_\triangle(\mathbf{x},\mathbf{a}) = f(\mathbf{x}+\mathbf{a}) - p_f(\mathbf{x},\mathbf{a})$. Furthermore, a neighborhood function

$$n(\mathbf{x},\mathbf{a}) = \begin{cases} 1 & \text{if } ||g_f(\mathbf{x}+\mathbf{a}) - g_f(\mathbf{x})|| < A \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

is used for the second weighting. $A$ specifies the adaptive region and is discussed further below. Finally,

$$s(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{k_\triangle(\mathbf{x})} \int_\Omega f_\triangle(\mathbf{x}, \mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2(f_\triangle(\mathbf{x}, \mathbf{a})) \cdot n(\mathbf{x}, \mathbf{a}) \, \mathrm{d}\mathbf{a} \quad (6)$$

$$k_\triangle(\mathbf{x}) = \int_\Omega d_1(\mathbf{a}) \cdot d_2(f_\triangle(\mathbf{x}, \mathbf{a})) \cdot n(\mathbf{x}, \mathbf{a}) \, \mathrm{d}\mathbf{a}$$

The smoothed function $s$ equals $S_{TL}(f)$.

Again, $d_1$ and $d_2$ are assumed to be Gaussian functions, with standard deviations $\sigma_1$ and $\sigma_2$, respectively. The method requires to specify parameter $\sigma_1$ only, which is at first used to be the radius of circular neighborhoods at $\mathbf{x}$ in $f$; let $\overline{g}_f(\mathbf{x})$ be the mean gradient of $f$ in such a neighborhood. Let

$$\sigma_2 = 0.15 \cdot || \max_{\mathbf{x} \in \Omega} \overline{g}_f(\mathbf{x}) - \min_{\mathbf{x} \in \Omega} \overline{g}_f(\mathbf{x}) || \quad (7)$$

(Value 0.15 was recommended in [5]). Finally, also use $A = \sigma_2$.

**Numerical Implementation**

Above we have defined the smoothing filters. All filters have been implemented in OpenCV [15], where possible the native function was used.[3].

For the TV-L$^2$, we use an implementation (with identical parameters) as in [21]. All other filters used are virtually parameterless (except a window size) and we use a window size of $m = 3$ ($\sigma_1 = 3$ for trilateral filter). The only other parameter to set is the bilateral filter color standard deviation $\sigma_1 = 0.1\, I_{\text{range}}$, where $I_{\text{range}}$ is the range of the intensity values.

## 3 Data Sets

For this paper we illustrate our argument with the Middlebury dataset [13]. We use both the optical flow data set and the stereo data sets. We split these into two main datasets.

**Data Set 1**

These are the "good quality" low noise images. They are either synthetically generated, or use good lighting and cameras with good optics. They are also using the same lighting conditions and camera exposures. Specifically, this set includes the 2001 stereo set (provided by [18]): *Barn1, Barn2, Bull, Map, Poster, Sawtooth, Tsukuba,* and *Venus*. It also includes the optical flow set to show how both types of correspondence algorithms have the same issues. The optical flow set (provided by [3]) were used when ground truth was available, specifically: *Dimetrodon, Grove2, Grove3, Hydrangea, RubberWhale, Urban2, Urban3,* and *Venus*. The total dataset is 8 for stereo and 8 for optical flow.

**Data Set 2**

[3] See acknowledgments for numerical implementations.

These are the images that differ in both illumination and exposures. This highlights the major importance to think of removing illumination artifacts. We include both the 2005 and 2006 data sets (provided by [10, 17]). The datasets have 3 different exposures and 3 different illuminations (for both the left and right images). This enables us to test the intensity consistency assumption under extreme conditions. Again, we only use images with ground truth available. For the 2005 set, that includes: *Art, Books, Dolls, Laundry, Moebius,* and *Reindeer*. For the 2006 set: *Aloe, Baby1-3, Bowling1-2, Cloth1-4, Flowerpots, Lampshade1-2, Midd1-2, Monopoly, Plastic, Rocks1-2,* and *Wood1-2*.

We are now no longer interested in "good quality" image pairs. We therefore only use images with differing exposure and illumination. To do this, for each image pair, we keep the left image with illumination = 1 and exposure = 0 (as defined by [13]). But for the right image, we make use of all all the differing illumination $(1, 2, 3)$ and exposure $(0, 1, 2)$ settings (excluding the exact same illumination = 1 and exposure = 0). This is a total of 8 different illumination/exposure combinations, for each image pair. That brings the total number of comparative data to 224 ($28 \times 8$).

## 4 Residual Images Contain the Important Information

This section demonstrates that the important information for correspondence algorithms is contained in the residual image $r$. The residual image is, in fact, an approximation of the high frequencies of the image, and the smoothed image $s$ is an approximation of a low-pass filter. Obviously, by iteratively running a smoothing filter, you will get a more and more smoothed image (i.e., you will be getting lower and lower frequencies). The following subsections demonstrate this well-known fact, but also that (or: how) the residual images still contain the high-frequency image. The texture is in effect the high frequencies of the image.

### 4.1 Co-occurrence Matrix and Metrics

The co-occurrence matrix has been defined for analysing different metrics about the texture of an image [9]:

$$C(i,j) = \sum_{\mathbf{x}\in\Omega} \sum_{\mathbf{a}\in\mathcal{N}\setminus\{(0,0)\}} \begin{cases} 1, & \text{if } h(\mathbf{x}) = i \text{ and } h(\mathbf{x}+\mathbf{a}) = j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\mathcal{N} + \mathbf{x}$ is the neighbourhood of pixel location $\mathbf{x}$, $\mathbf{a} \neq (0, 0)$ is one of the offsets in $\mathcal{N}$, and $0 \leq i, j \leq I_{\max}$ (maximum intensity). $h$ represents any 2D image (e.g., $f$, $r$, or $s$). All images are scaled min $\leftrightarrow$ max for utilizing the full $0 \leftrightarrow I_{\max}$ scale.

In our experiments we chose $\mathcal{N}$ to be the 4-neighbourhood, and we have $I_{\max} = 255$. The loss in information is identified by the following (common)

metrics,

$$\text{Homogeneity:} \quad T_{homo}(h) = \sum_{i\,j} \frac{C(i,j)}{1+|i-j|} \tag{9}$$

$$\text{Uniformity:} \quad T_{uni}(h) = \sum_{i\,j} C(i,j)^2 \tag{10}$$

$$\text{Entropy:} \quad T_{ent}(h) = -\sum_{i\,j} C(i,j)\,\ln C(i,j) \tag{11}$$

($T$ = textureness metric) where an increase in homogeneity represents the image having more homogeneous areas, an increase in uniformity represents more uniform areas, and a decrease in entropy shows that there is less information contained in the image.

### 4.2 Results of Co-occurrence Metrics

If we repeatedly smooth an image (using one of the operators as defined in Section 2) the expected behaviour is to drastically reduce the information (high-frequencies) of the image, as that is what smoothing filters are designed to do. Furthermore, the residual of an image is an approximation of the high frequencies of the image, so should not reduce the information. The following results show this in practice.

The following graphs and explanations are based on iteratively applying a smoothing filter to Data Set 1 defined in Section 3. To get a better representation, we scaled each result by the original image's metric, i.e., $T(s)/\,|T(f)|$, and then average the results for all data (at the specific iteration). The effect of this can be seen in Figure 2, for all the smoothing operators as specified before in this paper.

In this figure it shows that the more iterations performed on an image, the more homogeneous and uniform it becomes. Furthermore, there is a decrease in entropy. All three metrics show that there is a rapid loss of high frequencies initially, and this effect reduces after some time. Some filters come to a steady state (e.g., median and trilateral), some come to a small steady increase (e.g., TV-$L^2$ and trimmed mean), and others behave poorly (e.g., bilateral and mean filter). The main point to note is that all the selected smoothing filters reduce information rapidly.

The residual of an image is an approximation of the high frequencies of the image. Therefore, the information contained in a residual image should be less effected (of course, with any filtering process you are changing the information). The co-occurrence metrics were performed on the residual images (after a number of smoothing iterations), the results are shown in Figure 3. Again the results are for the Data Set 1 so each result is scaled, i.e., $T(r)/\,|T(f)|$ and the graph shows the average of all results (at the specific iteration).

In the homogeneity graph of this figure, it can be seen that the residual images are in fact less homogeneous than the original image (except for median,

which has a slight information loss, and trilateral that increases over time). This could be accounted for by introducing small amounts of (random) noise over the entire image. Note that the mean filter approaches the original graph, this is expected as eventually the mean filter will approximate to a uniform scale

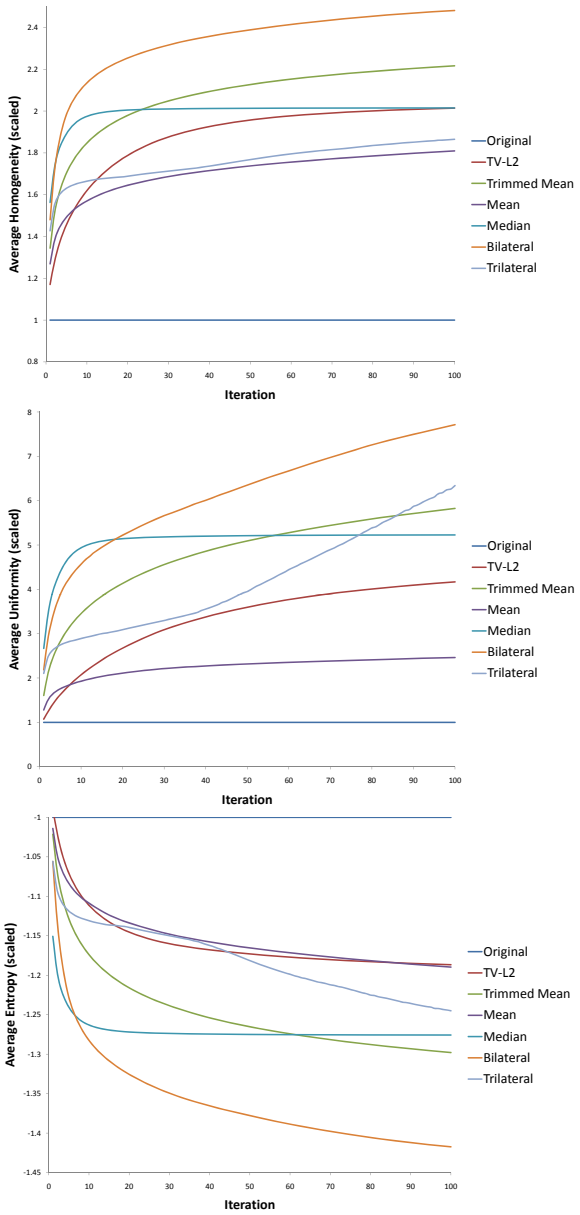


**Fig. 2.** Average homogeneity, uniformity, and entropy (top to bottom) of a smoothed image $s$, averaged over Data Set 1. Shows the reduction of information when repeatedly iterating a filter, for the selected dataset and smoothing operators.

change by the mean of the entire image. Furthermore, the TV-$L^2$ and median filter seems to be more stable than the rest (i.e., not having much range), but the others stabilize very quickly (except the trilateral which increases).

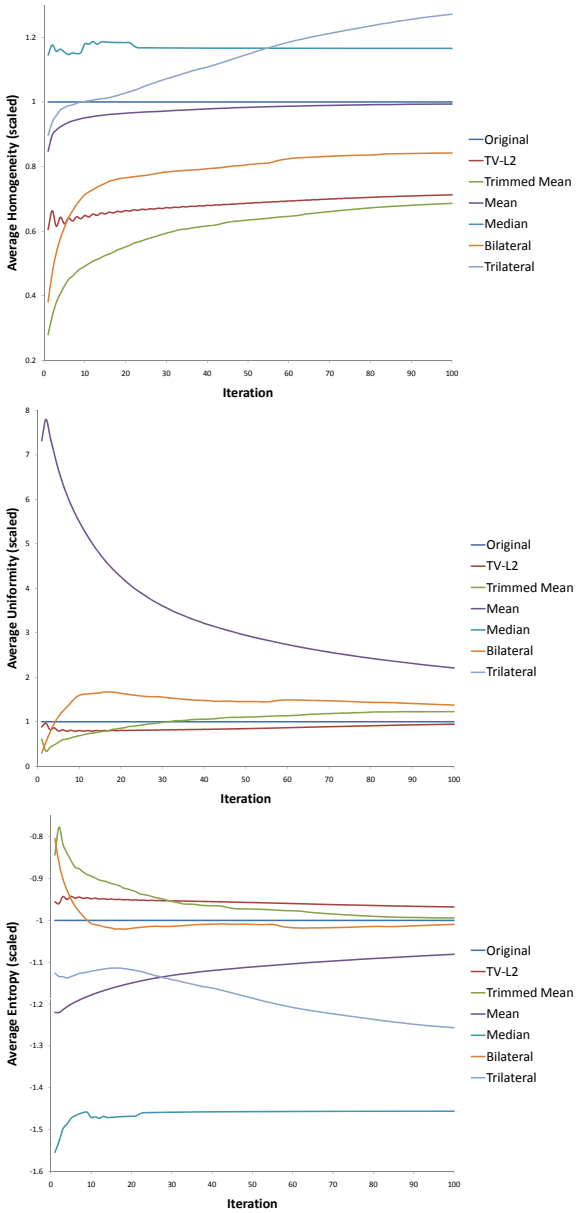


**Fig. 3.** Homogeneity, uniformity, and entropy (top to bottom) of a residual image $r$, averaged over Data Set 1. Shows that the effect of repeatedly iterating a filter (taking the residual) on an image does not necessarily reduce the information. (Note: trilateral and median filter not shown on uniformity graph because of large magnitude).

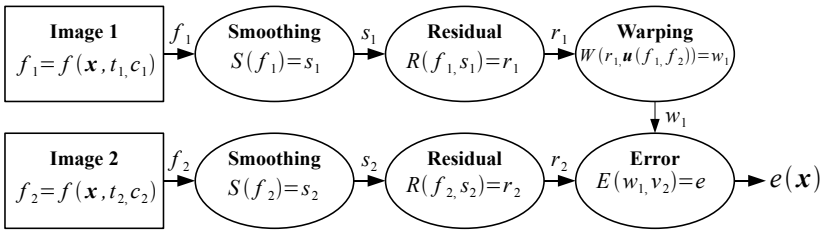


**Fig. 4.** Outline of the methodology used to compare images.

The first thing noticeable about the uniformity graph is that there is no trilateral or median filter. This is because they are much larger in magnitude; the median filter ranges are $30 \leq 50$ and the trilateral filter ranges are $27 \leq 32$. In saying this, the other algorithms (except mean filter) are within similar magnitudes of the original image (if not better), showing that the information is not lost, or only slightly reduced.

The entropy graph shows similar results to the uniformity graph. Most algorithms are within a small band around the original image, except the median filter which is much lower.

## 5 Removing Illumination Artifacts with Residual Images

Correspondence algorithms usually rely on the intensity consistency assumption, i.e., that the appearance of an object (according to illumination) does not change between the corresponding images. A previous study has suggested (by experimental data) that illumination artifacts propose the biggest problem for correspondence algorithms [14]. However, this does not hold true when using real-world images, this is due to, for example, shadows, reflections, differing exposures and sensor noise. We show that the errors from residual images are lower than the errors obtained using the original images. The process for showing this is highlighted in Figure 4. Details about the process are in the following sub-sections. For every set of test data, the filters were iteratively applied.

### 5.1 Image Warping

One way to highlight this (i.e., that the errors from residual images are lower than the errors obtained using the original images) is to warp one image to the perspective of the other (using ground truth) and compare the differences. The forward warping function $W$ is defined by the following:

$$W\Big(h_1(\mathbf{x}, t_1, c_1), \mathbf{u}(\mathbf{x}, h_1, h_2)\Big) = w\big(\mathbf{x} + \mathbf{u}(\mathbf{x}, h_1, h_2)\big) \tag{12}$$

where $h(\mathbf{x}, t, c)$ is the value of an image (e.g., $f$, $r$, or $s$) at 2D pixel position $\mathbf{x} \in \Omega$, at time $t$ (image sequences) from camera $c$ (multiple cameras, e.g., stereo),

and $\mathbf{u}$ is the 2D ground truth warping (remapping) vector from $h_1 = h(\mathbf{x}, t_1, c_1)$ to the perspective of $h_2 = h(\mathbf{x}, t_2, c_2)$. Subscripts 1 and 2 on $t$ and $c$ represent either two different time frames or cameras.

The simplest example is the stereo case, where $t_1 = t_2 = t$, $c_1$ is the left camera, $c_2$ is the right camera, and $\mathbf{u}$ is the ground truth disparity map from left to right (all vertical translations would be zero). Another common example is optical flow, where $c_2 = c_1 = c$, $t_1 = t$, $t_2 = t + 1$, and $\mathbf{u}$ is the ground truth flow field from $t$ to $t + 1$. In practice, this is done using a lookup table using interpolation (e.g., bilinear or cubic).

### 5.2 Image Scaling

For the purposes of this paper, $f$ is discrete in the functional inputs ($\mathbf{x}$, $t$, and $c$), but continuous for the value of $f$ itself. For a typical grey-scale image ($I_{\max} = 2^n - 1$), $n$ is usually 8 or 16. However, we find it easier to represent image data continuously by $-1 \leq f(\mathbf{x}) \leq 1$ with $f(\mathbf{x}) \in \mathbb{Q}^2$, which takes away the ambiguity for the bits per pixel, as any $n$-bits per pixel image can be scaled to this domain.

Therefore, $s$ will also be $-1 \leq s(\mathbf{x}) \leq 1$ with $s(\mathbf{x}) \in \mathbb{Q}^2$. However, the residual images $r$ are in the range $-2 \leq r(\mathbf{x}) \leq 2$ with $r(\mathbf{x}) \in \mathbb{Q}^2$, but in practice the upper and lower magnitude are much less than 1. For better comparison, we scaled the residual images by $\left(\max_{\mathbf{x}\in\Omega} |r(\mathbf{x})|\right)^{-1}$ to bring them into the scale $-1 \leq r(\mathbf{x}) \leq 1$.

### 5.3 Error Images

An error image $e$ is the absolute difference between two images,

$$E(h_1, h_2) = e \quad \text{with} \quad e(\mathbf{x}) = |\, h_1(\mathbf{x}) - h_2(\mathbf{x}) \,| \tag{13}$$

$h_1$ and $h_2$ can be any two images. For this paper, the error image is between $h_2$ and the warped $W(h_1)$; see Figure 5 for an example.

### 5.4 Error Metrics

To assess the quality of an image, there needs to be an error metric. A common metric is the *Root Mean Squared* (RMS) *Error*, defined by

$$RMS(e) = \sqrt{\frac{1}{N} \sum_{\mathbf{x}\in\Omega} \left( e(\mathbf{x})^2 \right)} \tag{14}$$

where $N$ is the number of pixels in the (discrete) non-occluded image domain $\Omega$ (when occlusion maps are available).

The standard RMS error gives an approximate average error for the entire signal. The problem with this metric is that it gives an even weighting to all pixels, no matter the proximity to other errors. In practice, if errors are happening in the same proximity, this is much worse than if the errors are

randomly placed over an image. Most algorithms can handle (by denoising or such approaches) small amounts of error, but if the error is all in the same area, this is seen as signal.

We have defined a more appropriate error to take the spatial properties of the error into account. This *Spatial Root Mean Squared Error* (Spatial-RMS) is defined by

$$RMS_S(e) = \sqrt{\frac{1}{N} \sum_{\mathbf{x} \in \Omega} \Big( G\big(e(\mathbf{x})\big)^2 \Big)} \tag{15}$$

where $G$ is a function that propagates the errors in a local neighbourhood $\mathcal{N}$. For our experiments, we chose a Gaussian error propagation using a standard deviation $\sigma = 1$.

### 5.5 Results on Data Set 1

For this section we again use the Middlebury dataset defined in Section 3. A qualitative example of error images $e$ can be seen in Figure 5. The image is from [3], and has ground truth available (warping from $t$ to $t+1$). In this figure, the image from time $t$ is warped using the ground truth to establish an error map. This highlights that even in relatively good lighting conditions, the differences in intensity between the two images still has a high amount of error (left image). The error image using the TV-L$^2$ residual (right) may appear to have more error but, in fact, it shows that the error is more evenly spread. Sections to notice are areas of shadow (e.g., around the wheel, in the arch and next to the curtain) and also object boundaries (look at the difference in errors of any object boundary). Furthermore the magnitudes of the maximum errors are less; the left image is 1.33 and the TV-L$^2$ residual image is 1.12.

A quantitative evaluation over the entire Data Set 1 has been performed. Again we evaluate the effect of repeated iterations of the smoothing filters, to

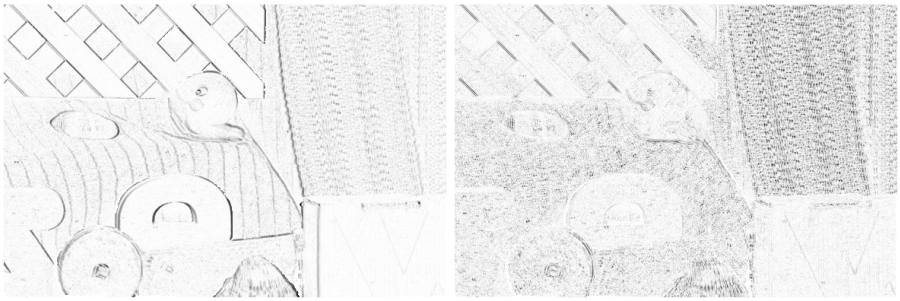

**Fig. 5.** Example error image using *RubberWhale* (see Figure 1 for original image). Left image shows the error between the warped $W(f(\cdot,t,\cdot))$ and $f(\cdot,t+1,\cdot)$, i.e., normal intensity images. Right image shows error between $W\big(r_{TV}(\cdot,t,\cdot)\big)$ and $r_{TV}(\cdot,t+1,\cdot)$, i.e., the residual images using TV-L$^2$ (white $\Leftrightarrow$ black $\equiv 0 \Leftrightarrow \max_{\mathbf{x}\in\Omega}(e)$).

obtain a residual image. The graphs in Figures 6 and 7 show the average RMS and Spatial-RMS for the optical flow dataset and stereo dataset separately. This was to show that although the stereo and optical flow algorithms appear to be quite different (and have differing communities following each), they both suffer from the same correspondence issue and use intensity consistency as their input data.

At a first glance it is obvious that all graphs are similar. There is only a subtle difference in the magnitude of each. The main point to notice is that all residual images get better RMS and Spatial-RMS than the original images after around 20 iterations. Another interesting point to note is that the Spatial-RMS shows similar information to the RMS graph. This may be because the propagation method was not good, or that the even distribution of error (when using residual images) seems to offset the large clusters of errors in the original error images.

From these graphs alone we can not decide which technique is the best, but if we use the graphs from Section 4, we obtain more information about the filters.

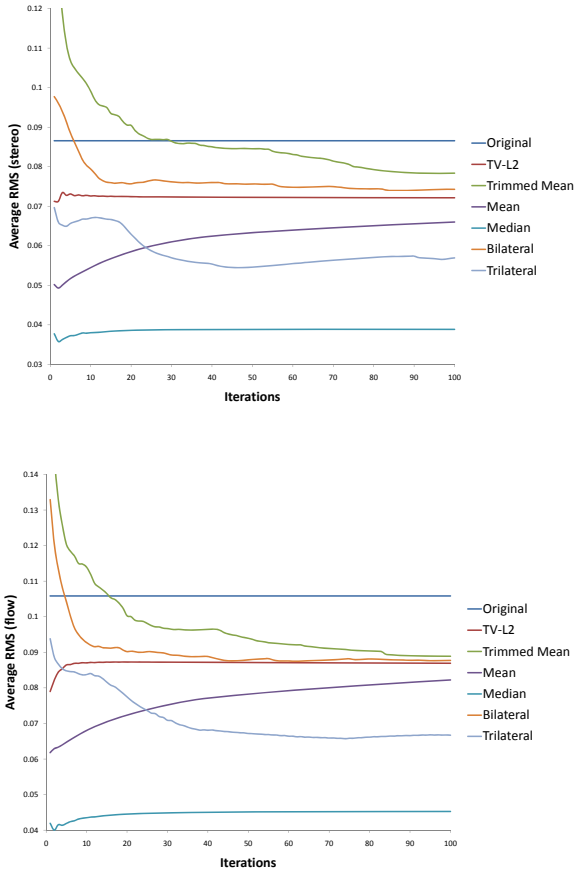


**Fig. 6.** RMS for each iteration. The top and bottom graph represents the average over Data Set 1 stereo and flow data, respectively.

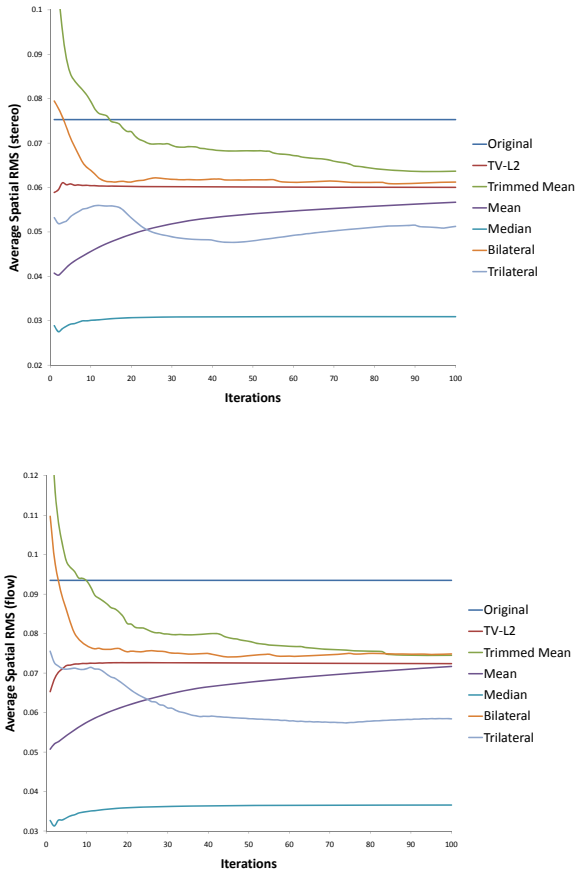


**Fig. 7.** Spatial RMS for each iteration. The top and bottom graph represents the average over Data Set 1 stereo and flow data, respectively.

From the graphs in Figures 6 and 7 it appears that median filtering is the best, however, if we look at Figure 3, the information in the median filter residual is being lost! So this improvement is probably due to the loss of information, rather than the matching error.

So, if we only consider the filters that do not lose information (i.e., TV-$L^2$, bilateral, and trimmed mean) we can see how they rank. TV-$L^2$ shows very good results, on average outperforming both the bilateral and trimmed mean filter. However, after a number of iterations, the difference is not that much. From a computational point of view, less iterations is desirable so this may make the TV-$L^2$ filter much better.

The other filters to consider are the mean and trilateral filter. These two filters retain information at low iterations ($< 10$ for trilateral and $< 3$ for mean). Both these filters provide good results for the RMS metrics, when at low iterations.

### 5.6 Results on Data Set 2

In the previous subsection we have already pointed out that the results for illumination differences is improved using residual images. We now show that these results get even better when illumination is a major issue (not just a minor one). We provide results similar to the previous subsection, but instead use Data Set 2.

The results can be seen in Figure 8. Note: for these results, the trilateral filter was stopped at iteration 10. It is immediately obvious that the original images are far worse than residual images, around 3 times worse on average. The second point to note is that the residual image graphs are almost identical, in magnitude and shape to the results provided in the previous subsection (i.e., Data Set 1). This again highlights that with extremely different exposures and

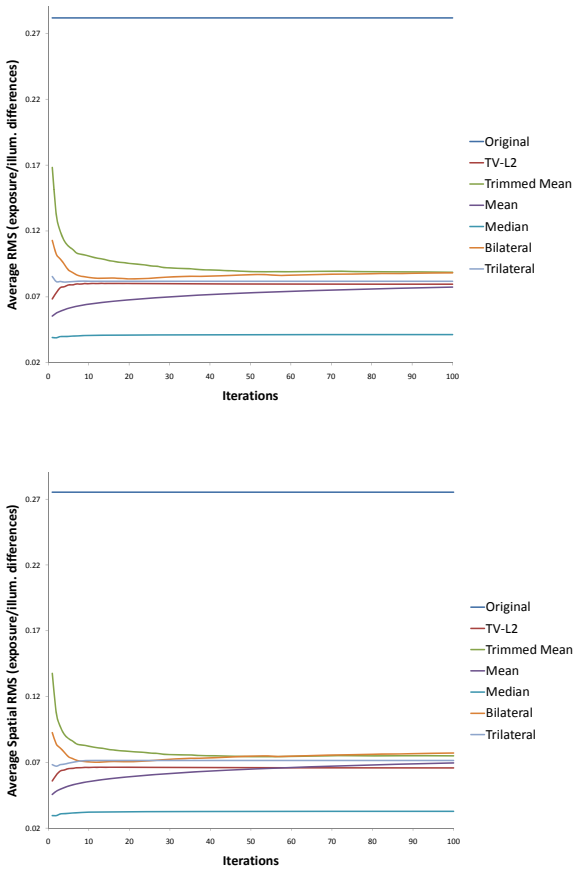


**Fig. 8.** Average RMS (top) and Spatial-RMS (bottom) for each iteration. The result shown is the average over Data Set 2. Notice the huge benefit of using residual images.

illuminations, the residual images provide the best information for matching. Another point to note is that the mean filter still approaches the same RMS as in the previous subsection. This means, that even doing a simple mean balancing between images, you can remove a lot of artifacts, but using the residual images is still much better.

Since most of the filters stabilize around iteration 40 (TV-$L^2$, trimmed mean, bilateral, and median), we have presented statistical results of the RMS after this number of iterations. The trilateral and mean filter were not very stable, so we chose to perform the statistics after 1 iteration. These results are shown in Table 1. You can see from these results that all the statistics for the original images are higher than any of the filters. The mean, trilateral, and median filter seem to be the most robust; showing the lowest range and standard deviation. The TV-$L^2$, mean, and median filters have the best average. The timing information provided in this table is the average time per iteration, on two sizes of images ($450 \times 370$, $752 \times 480$ pixel resolution), this is to highlight the scalability of the filters.

## 6 Conclusions and Future Research

We recalled several smoothing algorithms and the concept of residuals. We then showed that the information is still contained in the residual images (most of the time), and lost in the smoothed images. This leads us onto testing the residual images against illumination differences.

For the relatively "good quality" images (Data Set 1), we showed that you improve your results using a residual image. Furthermore, the errors are spread more evenly over the image, reducing the effect of outliers. We then showed that the results using residual images are extremely effective when using images where the exposure and illumination are different.

From these studies, we conclude that a simple mean filter may produce sufficient (and possibly the best) residual images. The TV-$L^2$ filter is also a

| | | | | | | Time/iteration (ms) | |
|---|---|---|---|---|---|---|---|
| Filter | Average | Min. | Max. | Range | Std. Dev | $470 \times 370$ | $752 \times 480$ |
| Original | 0.282 | 0.072 | 0.637 | 0.565 | 0.136 | - | - |
| TV-$L^2$ | 0.080 | 0.038 | 0.180 | 0.142 | 0.030 | 30 | 60 |
| Trimmed Mean | 0.090 | 0.053 | 0.143 | 0.090 | 0.023 | 30 | 100 |
| Mean | 0.055 | 0.023 | 0.120 | 0.096 | 0.023 | 1 | 1.5 |
| Median | 0.041 | 0.015 | 0.126 | 0.111 | 0.022 | 7 | 15 |
| Bilateral | 0.086 | 0.046 | 0.163 | 0.117 | 0.026 | 160 | 340 |
| Trilateral | 0.085 | 0.056 | 0.125 | 0.069 | 0.017 | 5000 | 11000 |

**Table 1.** Statistics for the RMS evaluation of the filters (average, minimum, maximum, range, and standard deviation), after 40 iterations (TV-$L^2$, trimmed mean, median and bilateral filter) and 1 iteration (mean and trilateral filter). Statistics are for Data Set 2. Average running times per iteration are also included (right) for two image resolutions.

good candidate as it retains information in the residual image, but still improves results. The median filter and trilateral filter appear to be good when looking at RMS, but there is information loss associated with this. The trimmed mean and bilateral filter work well, but not as good as the other filters, so perhaps are better suited to other applications.

Candidates for further studies are the varying weight trimmed mean filter [23], the Kuwahara filter in [4], and the TV-$L^1$ (i.e., TV-norm minimization using the $L^1$ norm) filter of [16]. (See [6] for a comparison of the bilateral with the TV-$L^1$ filter.)

Furthermore, the results from this test need to be compared using a correspondence algorithm. A small study has been conducted using the TV-$L^1$ optical flow (see [21, 22]), but more investigation needs to be done, including the application to a stereo algorithm.

**Acknowledgements**: The authors would like to thank Andreas Wedel (Daimler AG, Germany) for his implementation of TV-$L^2$ smoothing. Also, Prasun Choudhury (Adobe Systems, Inc., USA) and Jack Tumblin (EECS, Northwestern University, USA) for their implementation of the trilateral filter. Furthermore, our thanks go to the hard work put in by the people at Middlebury Vision, to painstakingly generate the dataset with ground truth.

## References

1. Abramson, S. B., and Schowengerdt, R. A.: Evaluation of edge-preserving smoothing filters for digital image mapping. *ISPRS J. Photogrammetry Remote Sensing*, **48**:2–17 (1993)
2. Aujol, J. F., Gilboa, G., Chan, T., and Osher, S.: Structure-texture image decomposition - modeling, algorithms, and parameter selection. *Int. J. Computer Vision*, **67**:111-136 (2006)
3. Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R.: A database and evaluation methodology for optical flow. In Proc. *IEEE Int. Conf. Computer Vision*, pages 1–8 (2007).
4. Chen, S., and Shih, T.Y.: On the evaluation of edge preserving smoothing filter. In Proc. *Geoinformatics* (CD), Nanjing, China, paper C43 (2002)
5. Choudhury, P., and Tumblin, J.: The trilateral filter for high contrast images and meshes. In Proc. *Eurographics Symp. Rendering*, pages 1–11 (2003)
6. Danda, S., and McGraw, T.: A comparison of the bilateral filter and TV-norm minimization for image denoising. In Proc. *ISBI*, pages 676–679 (2007)
7. *.enpeda..* Image Sequence Analysis Test Site (EISATS): http://www.mi.auckland.ac.nz/EISATS
8. Franke, U.: Progress in space-time machine vision. Talk at Freiburg University, http://www2.faw.uni-freiburg.de/kolloquium/ss08/franke.pdf (2008)
9. Haralick, R. M., and Bosley, R.: Texture features for image classification. In Proc. *ERTS Symposium*, NASA SP-351, pages 1219-1228 (1973)
10. Hirschmüller, H., and Scharstein, D.: Evaluation of cost functions for stereo matching. In Proc. *IEEE Conf. Computer Vision Pattern Recognition (CVPR)*, (2007)

11. Kuan, D. T., Sawchuk, A. A., Strand, T. C., and Chavel, P.: Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Trans. Pattern Analysis Machine Intelligence*, **7**:165–177 (1985)
12. Mao, Z., and Strickland, R. N.: Image sequence processing for target estimation in forward-looking infrared imagery. *Opt. Eng.*, **27**:541–549 (1988)
13. Middlebury data set: optical flow data http://vision.middlebury.edu/flow/data/ and stereo data http://vision.middlebury.edu/stereo/data/
14. Morales, S., Woo, Y. W., Klette, R., and Vaudrey, T.: A study on stereo and motion data accuracy for a moving platform. Technical report, MI-tech-32, http://www.mi.auckland.ac.nz/, University of Auckland (2009)
15. Open Source Computer Vision Library, *Intel Reference Manual*, retrieved from http://developer.intel.com (2001)
16. Rudin, L., Osher, S., and Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D*, **60**:259–268 (1992)
17. Scharstein, D., and Pal, C.: Learning conditional random fields for stereo. In Proc. *IEEE Conf. Computer Vision Pattern Recognition (CVPR)*, (2007)
18. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, **47**:7–42 (2002)
19. Tomasi, C., and Manduchi, R.: Bilateral filtering for gray and color images. In Proc. *IEEE Int. Conf. Computer Vision*, pages 839–846 (1998)
20. Vaudrey, T., Rabe, C., Klette, R., and Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In IEEE Conf. Proc. *IVCNZ*, Digital Object Identifier 10.1109/IVCNZ.2008.4762133 (2008)
21. Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D.: An improved algorithm for TV-$L^1$ optical flow. In Post Proc. *Dagstuhl Motion Workshop*, to appear (2009)
22. Zach, C., Pock, T., and Bischof, H.: A duality based approach for realtime TV-$L^1$ optical flow, In Proc. *Pattern Recognition - DAGM*, pages 214–223 (2007)
23. Zhang, D. S., and Kouri, D. J.: Varying weight trimmed mean filter for the restoration of impulse noise corrupted images. In Proc. *Acoustics Speech Signal Proc.*, pages iv/137- iv/140 (2005)