Kalman-Filter Based Spatio-Temporal Disparity Integration

Sandino Morales, Reinhard Klette

.enpeda.. Group, The University of Auckland, New Zealand

Abstract

Vision-based applications usually have as input a continuous stream of data. Therefore, it is possible to use the information generated in previous frames to improve the analysis of the current one. In the context of video-based driver-assistance systems, objects present in a scene typically perform a smooth motion through the image sequence. By considering a motion model for the ego-vehicle, it is possible to take advantage of previously processed data when analysing the current frame.

This paper presents a Kalman filter-based approach that focuses on the reduction of the uncertainty in depth estimation (via stereo-vision algorithms) by using information from the temporal and spatial domains. For each pixel in the current disparity map, we refine the estimated value using the stereo data from a neighbourhood of pixels in previous and current frames. We aim at an improvement of existing methods that use data from the temporal domain by adding extra information from the spatial domain. To show the effectiveness of the proposed method, we analyse the performance on long synthetic sequences using different stereo matching algorithms, and compare the results obtained by the previous and the suggested approach.

Keywords: Stereo algorithms, Kalman filter, disparity propagation, spatial domain, temporal domain, vision-based driver assistance

1. Introduction

Disparity (or depth) estimation obtained from stereo-vision analysis is commonly used to provide basic 3-dimensional data for complex vision-based applications [33]. Particularly, in vision-based *driver-assistance systems* [8], the calculated disparities have been used to model the environment that surrounds the vehicle where the vision-system is operating in, i.e. the *ego-vehicle*. From road-manifold estimation [28] to object detection and tracking [17], a high accuracy of computed disparity values is required. Accuracy demands actually vary among applications; for example, accurate disparity discontinuities (i.e. at occluding boundaries) are required in driver-assistance systems, but 3-dimensional object modelling focuses on accurate disparities within object regions.

We have identified three general options for the estimation of disparities between the two images in a given stereo pair. The first approach is the design of new, or the

Email address: pmor085@aucklanduni.ac.nz (Sandino Morales)

improvement of existing strategies for stereo matchers. For example the authors proposed in [11] a hierarchical approach for the improvement of the performance of *semi-global* matchers [13].

The second approach is the improvement of functions that define the *data* and *smoothness* term on the *energy function* used to solve the stereo-vision matching problem, see [27]. Using *cost functions* in the data term that do not take as granted photometric consistency between both images in a stereo-pair, has allowed to improve the performance of stereo-vision algorithms when used in real-world applications [23]. In this sort of environments, photometric consistency cannot be assured. For example, *census transform* or *gradient-based* cost functions [11] depend more on the distribution of intensities around a given pixel rather than on a direct comparison of the intensity values themselves.

An improvement of the functions used to define the smoothness term by, for example, moving away from the simple Potts model to truncated linear or quadratic functions, has further allowed the tune-up of stereo-vision algorithms.

The third approach consist in pre-processing the input stereo pair, or post-processing of the resultant disparity map. By preprocessing of the input stereo pairs it is possible to filter-out several types of noise that could potentially affect the matching process. For example, using residuals with respect to smoothing or edge maps as inputs may reduce the influence of *illumination artefacts* [31]. Post-processing methods manipulate the resultant disparity maps, for instance, by using left-right and right-left consistency to detect miscalculated disparities, or by calculating sub-pixel accurate results. Mean or median filtering are further options; see [1].

In this paper we propose a method based on the third approach. We post-process disparity maps using the available *spatial* and *temporal* information in the context of vision-based driver-assistance systems. As the input data for such a system is a time sequence of stereo pairs, it is possible to use the information contained in the *temporal domain* by incorporating disparity data from previous frames into the disparity map generated in the current time instance. The objective is to reduce the influence of miscalculated disparities in particular regions of the disparity maps.

Information contained in the temporal domain has been used before. For example, in [23] some *alpha-blending* of disparity values calculated for the current and previous frames lead to an improved performance for currently measured values at scene points "roughly matching" the scene geometry assumed by this method. This model did not yet consider the motion induced by the ego-vehicle, nor the independent motion of other objects present in the scene.

In [2], the authors merged previous and current information by using an *iconic* (i.e. pixel-wise) Kalman filter-based approach [22] and some ego-motion information, namely the *yaw* and *speed* rate. This approach was designed to improve the disparity measurements in regions of the input images where the visible motion was induced only by the ego-motion, and not by the independent motion of other objects (i.e. by objects that are static with respect to the *ground*, which is the surface manifold where the ego-vehicle is driving on). We refer to this method as the *static approach* from now on.

The static approach was modified in [29] by adding a *disparity rate* term to the Kalman filter. The authors were interested in the improvement of the disparity measurements for objects that move relatively to the ego-vehicle in longitudinal direction. The disparity-rate term introduces into the Kalman filter the change in disparity of a tracked pixel from one frame to the next one. We refer to this method as the *dynamic approach*

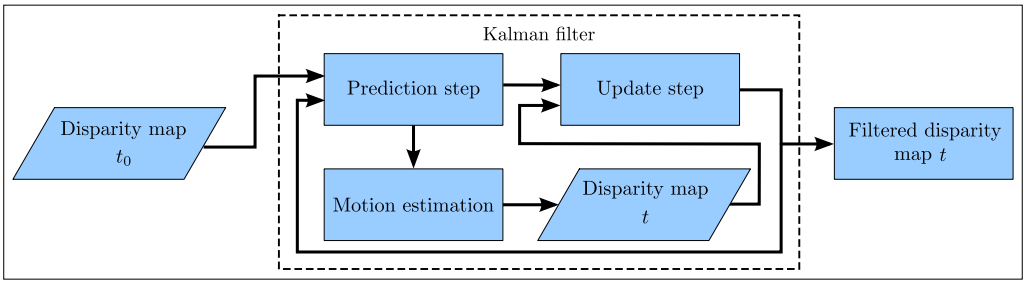

Figure 1: After initialising the filtering process with values from a disparity map at time t_0, the proposed method refines the disparity estimation at time $t > t_0$ of each pixel using data from corresponding neighbourhoods from the current (at time t) and the immediate previous disparity maps.

from now on.

Post-processing of disparity maps can also aim at optimising the computed measurements by considering disparity values within a neighbourhood of the current pixel, i.e. using data from the *spatial domain*. In [23] the authors used once again an alpha-blending approach for modifying the disparity value of a given pixel by using the computed disparities of the *north* and *south* neighbours. Again, no ego- or independent motion was taken into consideration.

We aim to improve the disparity estimation of objects that are static with respect to the ground or moving longitudinally away from the ego-vehicle. The idea is to modify the dynamic and static method by adding to the Kalman-filtering process data from the spatial domain. Spatial information is used from the previous and the current disparity maps, and can be taken from an arbitrarily defined neighbourhood. Actually, those should not be "too large".

After initialising the filtering process with values from a disparity map at a reference time instance t_0, the disparity value of a pixel representing a 3-dimensional point P is improved by considering sets of pixels from the current and previous disparity maps. The *ego-motion data* is used to estimate the relative motion of P to obtain the position where it is projected on the previous and current images. We assume that the ego-vehicle describes a 2-dimensional motion (i.e. horizontal and longitudinal) over the ground assumed to be a plane and exclude any horizontal translation. To describe such motion it is only required to know the speed and yaw rate between consecutive frames. *Roll* and *pitch* are not included into our model, but we acknowledge that both do have a minor influence when modelling the motion of *vision-augmented vehicles*. Figure 1 presents a diagram of the followed approach.

We perform experiments with a computer-generated sequence from [5] with available (stereo and *optic-flow*) ground truth, as introduced in [30]. To generate the input disparity maps, we use three different stereo-vision algorithms. For each of the selected stereo matchers, we use three different configurations based on different cost functions. In Figure 2 we depict sample frames of the sequence used for experimentation. The segmented blue car represents a static object with respect to the ground while the green

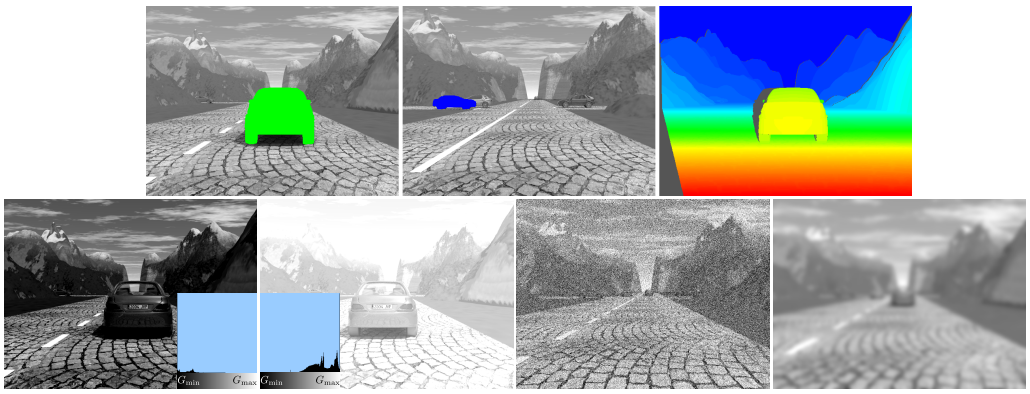

Figure 2: Sample frames from the used data set [30]. *Left*: Segmented object that moves relatively to the ego-vehicle in longitudinal direction. It is used to test the improvements over the dynamic method. *Middle*: Segmented object static with respect to the ground used to test the achieved improvements over the static method. *Right*: Colour coded (red large and blue small) stereo ground truth. We use this colour coding for the sample results presented on Figure 5.

vehicle moves longitudinally away from the ego-vehicle. As expected for perfect input stereo pairs, the generated *raw* disparity maps are very close to the ground truth data, letting almost no room for improvement. Thus, to make our experiments more challenging, we selected three different real-world effects to alter the input stereo pairs: additive Gaussian *white-noise*, *Gaussian blurring*, and a *constant additive intensity change*.

In driver-assistance systems, or any other vision-based outdoor application, the Gaussian noise could be added to the images by a low-quality sensor in the cameras. Blurring could be caused by rainy environments or by a misalignment of the camera lenses. An intensity difference between the images of a given stereo pair could be caused by unexpected shadows, reflections, difference in the gain of the recording cameras and so forth.

We compare the results of the *spatio-temporal* analysis with those obtained when using only information from the temporal domain. We investigate also the effect of the size of the neighbourhood used to incorporate data from the spatial domain.

The approach described in this paper was originally presented in [25]. In this paper we explain further ideas behind the presented method and include also more experimental data. We use three different stereo-vision algorithms to analyse four long stereo-sequences. Each of the matching algorithms was tested with three different cost functions, to define in total nine different stereo-vision matching techniques. The rest of this paper is structured as follows. We start in Section 2 with briefly recalling the structure of Kalman filters. The assumed motion model is discussed in Section 3. In Section 4 we describe the proposed approach. We continue by discussing the results obtained in our experiments along with the specification of the stereo-vision algorithms used in there. We finalise in Section 6 with conclusions.

2. Kalman Filter

We briefly explain the linear Kalman filter [15] as commonly used for a given *discrete dynamic system* [21], with states $\mathbf{x}_t$, measurements $\mathbf{y}_t$, state transition matrix $\mathbf{A}$, process noise $\mathbf{v}$, control matrix $\mathbf{B}$, and measurement noise $\mathbf{w}$.

Given the system

$$\mathbf{x}_t = \mathbf{A} \cdot \mathbf{x}_{t-1} + \mathbf{B} \cdot \mathbf{u}_t + \mathbf{v} \quad (1)$$

$$\mathbf{y}_t = \mathbf{H} \cdot \mathbf{x}_{t-1} + \mathbf{w} \quad (2)$$

The Kalman filter is defined in two steps. In the first step, the information from the previous step is incorporated into the filter by generating a *predicted* state

$$\mathbf{x}_{t|t-1} = \mathbf{A} \cdot \mathbf{x}_{t-1|t-1} \quad (3)$$

$$\mathbf{P}_{t|t-1} = \mathbf{A} \cdot \mathbf{P}_{t-1|t-1} \cdot \mathbf{A}^T + \mathbf{Q} \quad (4)$$

We use the notation $t|t-1$ to denote an intermediate step between $t-1$ and t, while $t-1|t-1$ denotes the state obtained with the Kalman filter at time $t-1$. Matrix $\mathbf{Q}$ represents the process noise variance (obtained from vector $\mathbf{v}$) and $\mathbf{P}_{t|t-1}$, denotes the covariance matrix of the error of $\mathbf{x}_{t|t-1}$ compared to the true value $\mathbf{x}_{t|t}$.

In the second step, the predicted state is *corrected* using data from the current state via the measurement vector $\mathbf{y}_t$ and the predicted matrix $\mathbf{P}_{t|t-1}$:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{K}_t \left(\mathbf{y}_t - \mathbf{H} \cdot \mathbf{x}_{t|t-1}\right) \quad (5)$$

where

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \cdot \mathbf{H}^T \left(\mathbf{H}_t \cdot \mathbf{P}_{t|t-1} \cdot \mathbf{H}^T + \mathbf{R}\right) \quad (6)$$

and $\mathbf{R}$ represents the measurement noise variance obtained from vector $\mathbf{w}$. The matrix $\mathbf{K}$ is the *Kalman gain* as derived in [15]. It follows that

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}^T)\mathbf{P}_{t|t-1} \quad (7)$$

This is an iterative process. The initial state $x_{0|0}$ and the initial covariance matrix $P_{0|0}$ need to be given to start with the process.

3. Motion Model

We assume that the ego-vehicle is driven on a flat road moving according to the so-called *bicycle model* [8]. This type of motion is described by a 2-dimensional transformation defined by a translation in the longitudinal direction or a rotation around the vertical axis.

Given the position of the ego-vehicle at time instance $t-1$ with respect to a known coordinate system, we calculate its spatial position at time instance t with respect to the same reference coordinate system. The parameters necessaries for estimating the new position are the frame rate, the speed and yaw rate. Roll and pitch are not included into our model, but we acknowledge that both do have a minor influence for vision-augmented

vehicles. In the assumed bicycle model, the speed and yaw rate define the *ego-motion data* of the ego-vehicle.

The *tangential speed* v_t, is the speed of the ego-vehicle at time $t-1$. The yaw rate $\dot{\psi}_t$, also known as the *angular velocity*, represents the angular change with respect to the constant time unit $\Delta \in \mathbb{R}^+$ that is defined by the sequence's frame rate.[1]

Assume for now that the reference coordinate system is that of the ego-vehicle at time instance $t-1$. Let O denote the origin of this coordinate system. If $\dot{\psi}_t = 0$, then the position of the ego-vehicle at time t is given by point

$$P_t = (0, 0, \Delta \cdot v_t)^T \tag{8}$$

meaning that the ego-vehicle describes a *uniform rectilineal motion*. If $\dot{\psi}_t \neq 0$, the ego-vehicle describes a *circular trajectory* [9]. To fully describe such a motion, the radius $r \in \mathbb{R}^+$ and centre $C \in \mathbb{R}^3$ of the followed circle $\mathcal{C}$ need to be calculated. The radius is given by

$$r = \frac{v_t}{\dot{\psi}_t} \tag{9}$$

To calculate the coordinates of C, we use the fact that the origin O of the reference coordinate system is an element of $\mathcal{C}$. Thus, as r is already known, $\mathcal{C}$ is defined by the set

$$\mathcal{C} = \left\{ (X, Y, Z)^T \in \mathbb{R}^3 | (X - r)^2 + Z^2 = r^2 \right\} \tag{10}$$

whose centre is at point $C = (r, 0, 0)^T$. Therefore, the position of the ego-vehicle at time t, with respect to the coordinate system at time $t-1$, is given by

$$P_t = \begin{pmatrix} 1 - \cos\left(\dot{\psi}_t \Delta\right) \\ 0 \\ \sin\left(\dot{\psi}_t \Delta\right) \end{pmatrix} \tag{11}$$

Note that we assume that there is no movement in the vertical direction, between $t-1$ and t we drive on a plane. Using this same set of equations me can model the induced motion of static objects with respect to the ground as they describe an analogous motion but in the opposite direction.

Figure 3 presents the described motion model when $\dot{\psi}_t \neq 0$. The ego-vehicle moves across the XZ plane following the circle defined by point C and radius r. Point P_t is defined by Equation (11) and represents the position of the ego-vehicle at time t if its position at time $t-1$ is assumed to be at O.

4. Approach

The basic idea of our approach is to incorporate disparity information contained in the spatial domain [23] into the static [2] and dynamic approach [29]. These two Kalman filter-based methods were designed to handle different kinds of moving objects. The authors of the static approach were interested in improving the disparity values of

[1] We assume that $v_t, \dot{\psi}_t \in \mathbb{R}$. A positive (negative) velocity indicates a forward (backwards) movement. A positive (negative) $\dot{\psi}_t$ represent a right (left) turn.

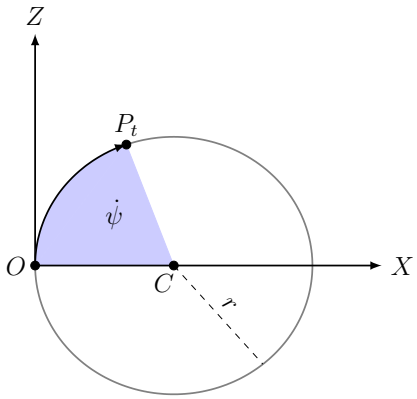

Figure 3: Assumed motion model. The ego vehicle describes a circular motion between time instances $t-1$ and t. At time $t-1$, the ego-vehicle is situated at point O. At time t, it has moved following the circle determined by point C and radius r. Both parameters are calculated from the yaw and speed rate.

static objects with respect to the ground present in the scene. The dynamic approach was designed to improve the calculated disparity values of objects moving longitudinally away from the ego-vehicle. We aim at enhancing both methods by adding data from the disparity spatial domain.

We merge the temporal and spatial information by means of a Kalman filter. For simplicity we use an *iconic* Kalman filter [22], i.e. we define an individual Kalman filter for each pixel under consideration. At each iteration we obtain an *improved* disparity value for each pixel using both spatial and temporal information. In order to do so, it is necessary to consider the motion of the 3-dimensional points that define the pixels in the reference image from the input stereo pair. Each pixel in the reference image that represents a non occluded point will be tracked as the image sequence advances in time and as long as is still visible in the current frame. This is done by considering the ego-motion and the disparity rate [29].

Let P be a 3-dimensional point in the field of view of the recording stereo camera for at least two frames. We denote with p the projection of P on every frame in the sequence where it remains within the field of view of the stereo camera. Besides knowing that its position on the image plane is actually different due to ego-motion or possible independent motion of P, the disparity value assigned to p is also expected to change through the sequence. In other words we do not consider p as a position in the image plane but as the pixel where P is projected in the image plane across the sequence.

Let

$$\mathcal{N}(p) = \{p, p_1, p_2\} \tag{12}$$

be a neighbourhood of p. The generalisation for larger neighbourhoods is straightforward. Consider the dynamic system defined at time t by the state vector x_t and the transition matrix $\mathbf{A}$, given by

$$\mathbf{x}_t = \left(d, d_1, d_2, \dot{d}\right)^T \tag{13}$$

and

$$\mathbf{A} = \begin{pmatrix} \alpha & \beta_1 & \beta_2 & \Delta \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

where d, d_1, d_2 are the computed disparity values of the pixels p, p_1, p_2, respectively, generated at time $t-1$. The value $\dot{d}$ is the disparity rate that introduces into the Kalman filter the change in disparity of a tracked pixel from one frame to the next one. Parameters α, β_1 and β_2 control the interaction of disparity values at pixels p, p_1, and p_2, respectively. Parameter Δ denotes the constant elapsed time between two consecutive frames. We assume that the noise vector $\mathbf{v}$ associated to the system is Gaussian with zero mean and standard deviation σ_d for all disparity values and $\sigma_{\dot{d}}$ for disparity rate.

Measurement data from the current disparity map is not available for the disparity rate. However, it contains the disparity values of all the involved pixels defining $\mathcal{N}(p)$ from the disparity map calculated at time t. Therefore, the dimension of the measurement vector $\mathbf{y}_t$ equals to three, and the matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{15}$$

The noise vector $\mathbf{w}$ associated with the measurements taken from the system is assumed to be the same for all of its coordinates: Gaussian with zero mean and with a standard deviation σ_v.

To start the filtering process, we need to define the initial state and the initial covariance matrix. The initial state is defined using the disparity values of p and its neighbours calculated with a given stereo algorithm at time $t = 0$. The disparity rate is set to be zero. The initial covariance matrix is defined by

$$\mathbf{P}_{0|0} = \begin{pmatrix} \sigma_d^2 & \sigma_{dd_1} & \sigma_{dd_2} & \sigma_{d\dot{d}} \\ \sigma_{dd_1} & \sigma_{d_1^2} & \sigma_{d_1 d_2} & \sigma_{d_1 \dot{d}} \\ \sigma_{dd_2} & \sigma_{d_1 d_2} & \sigma_{d_2^2} & \sigma_{d_2 \dot{d}} \\ \sigma_{d\dot{d}} & \sigma_{d_1 \dot{d}} & \sigma_{d_2 \dot{d}} & \sigma_{\dot{d}^2} \end{pmatrix} \tag{16}$$

where, for example, $\sigma_{d\dot{d}} = \sigma_d \cdot \sigma_{\dot{d}}$. Recall that we assumed that all the calculated disparity values have the same variance. The values used in the experiments (to define $\mathbf{P}_{0|0}$) are specified in Section 5.

Once the filter has been initialised, we can start the iterative process. Assume that we have already calculated $t-1$ steps and that $\mathbf{x}_{t-1|t-1}$ is available. After the prediction step at time t, the first three coordinates of $\mathbf{x}_{t|t-1}$ contain information about a neighbourhood of the disparity map calculated at time $t-1$. In this case it is a 2-neighbourhood.

We incorporate information from the disparity map generated at time t via the measurement vector y_t and just before the update step in the filtering process. In order to do this, we need to calculate the coordinates of the projection of P (i.e. the coordinates of p) in the reference image at time t. As we are are assuming that the ego-vehicle and the objects is the scene are not static, we need to consider the visible movement of the 3-dimensional point P between $t-1$ and t. This is done by calculating the relative motion of P with respect to the ego-vehicle, and this takes two steps.

First, the motion induced by the disparity rate is incorporated into P. This motion is longitudinally away from the ego-vehicle and in the same direction. Thus, only the longitudinal coordinate will be modified, i.e. a uniform rectilineal motion, see Equation (8). Second, the positional change induced by the ego-motion is considered. We use only speed and yaw rate, as described in Equation (11), and assume that they are free of noise.

Once the position of p in frame t is known, we calculate the measurement vector $\mathbf{y}_t$ from the disparity map at time t. This measurement vector can now be used to calculate the Kalman gain and the updated state, so that the next iteration at time t can be performed. From the design of our system [see Equation (2)], disparity information from pixels adjacent to p at time t is included in the measurement vector. To avoid noisy states in both steps of the Kalman process, we follow the validation rules suggested in [29].

We just described how to merge data from the temporal and spatial disparity domain into the dynamic approach. For adding temporal information to the static approach it is necessary to remove the disparity rate term from the state vector as the objects are not "moving away" and modify accordingly the rest of the dynamic system presented in Equations (13) and (14).

5. Experiments

For testing the proposed method, we perform a set of experiments where we expect to detect changes in performance for the static [2] or dynamic [29] approach when incorporating information from the spatial domain.

5.1. Data Set

As experimental data set, we use the Sequence 1 from the Set 2 of the [5] website as introduced in [30]. It is a long (100 frames) 12-bit per pixel computer generated stereo-sequence representing a driving scenario with available stereo ground truth. The *virtual* ego-vehicle drives straight through the entire sequence. Thus, we keep a constant yaw rate equal to zero. We set the speed at 6.99 m/s, calculated from the available stereo-ground truth assuming a frame rate of 25 frames per second.

As stated in Section 4, the dynamic method was designed to improve the calculated disparity values from objects moving away but in the same driving direction as the ego-vehicle. We segmented in the test sequence a vehicle whose movement fulfilled such

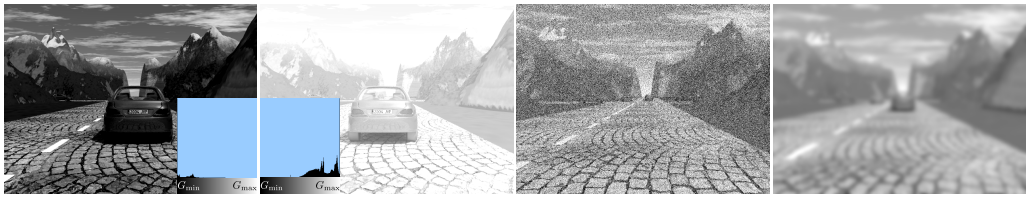

Figure 4: Sample images altered with the chosen real-world effects. From left to right, a stereo pair with maximum brightness difference, an image with added Gaussian noise and a blurred image.

requirements. For the experiments using the static method we segmented a static vehicle (from frame 46 till frame 100) with respect to the ground. In Figure 2 left, the green vehicle is moving away from the ego vehicle, while the static vehicle is highlighted in blue in the middle image.

Computer generated images do not have the multiple biases found in real-world imagery, e.g. multiple light sources, non-Lambertian [12] surfaces, lighting artefacts [31] or camera misalignment in multi-camera systems. To make our experiments more realistic, we selected three different real-world effects (we call it *noise* from now on) to alter the stereo pairs: additive Gaussian white-noise, Gaussian blurring, and a constant additive intensity change.

Let $\mathcal{S}$ denote the set of stereo pairs from the test sequence. All the functionals used to incorporate noise into the elements of $\mathcal{S}$ are functions of the form

$$\mathrm{F}\colon \mathcal{S} \to \mathcal{I}_{\mathcal{S}} \tag{17}$$

where $I_{\mathcal{S}}$ denotes a set of digital images (of the same VGA dimensions) and bit-depth as in the images in $\mathcal{S}$. The particular way how each functional defines a new image is described in the paragraphs below.

Constant Additive Intensity: Differences in intensity values between the two cameras in a stereo-vision system is a common issue in real-world imagery. For driver-assistance systems, this is even more obvious as unexpected shadows, reflections, difference in the gain of the recording cameras and so forth, might affect one camera and not the other.

We simulate the possible difference in brightness by a simple addition of a constant value to the intensity of the pixels in one of the images of a stereo pair. Given $I_t \in \mathcal{S}$, $t \in [1, 100] \in \mathbb{N}$, the *additive intensity* functional $\mathrm{F_i}$ defines the image $\mathrm{F_i}(I_t) \in \mathcal{I}_{\mathcal{S}}$ such that

$$\mathrm{F_i}(I_t)(p) = \begin{cases} I_t(p) + c, & \text{if } I_t(p) + c \in [G_{\min}, G_{\max}] \\ G_{\min}, & \text{if } I_t(p) + c < G_{\min} \\ G_{\max} & \text{if } I_t(p) + c > G_{\max} \end{cases} \tag{18}$$

where $c \in \mathbb{Z}$; $G_{\min}$ and $G_{\max}$ denote the minimum and maximum intensity values of the input images. We use different values of c for each image of the stereo pair depending on t. We chose to have maximum brightness differences at the start and at end of the sequence with no difference at all at the middle of it (i.e. $c = 0$ for both images of the stereo pair at $t = 50$). The values of c used to generate $\mathrm{F_i}(\mathcal{S})$ are defined according to the formulas presented in the first two rows (from top to bottom) of Table 1. The first two images from the left of Figure 4 are the stereo pair at $t = 1$, when $c = -98$ for the image on the left and $c = 98$ for the image on the right. Note the differences between of the histograms of the images.

Gaussian White-Noise: Small amounts of Gaussian noise [14] are present in real-world imagery. It can be added to the images by a low-quality sensor in the cameras (this is more evident in images grabbed in dark environments). However, the level of noise has been decreasing over time as sensor technology improves.

We vary the level of Gaussian noise from low to very high. We add to the pixel values of the images in $\mathcal{S}$ random Gaussian (normal distributed) white noise $N(\mu, \sigma)$ with a mean $\mu = 0$ and a varying standard deviation σ.

Functional	Image	Variable	Frame t	
			$t \in [1, 50]$	$t \in [51, 100]$
$\mathrm{F_i}$	Reference	c	$2t - 100$	
	Match		$100 - 2t$	
$\mathrm{F_g}$	Reference	σ	t	0
	Match		t	$101 - t$
$\mathrm{F_b}$	Reference	k	$2t - 1$	0
	Match		$2t - 1$	$201 - 2t$

Table 1: Parametrization of three different noise functionals used in the experiments. $\mathrm{F_i}, \mathrm{F_g}$ and $\mathrm{F_b}$ denote additive brightness, additive Gaussian and Gaussian blurring functionals, respectively.

The *additive Gaussian* functional $\mathrm{F_g}$, given $I_t \in \mathcal{S}$, defines the image $\mathrm{F_g}(I_t)$ such that

$$\mathrm{F_g}(I_t)(p) = \begin{cases} I_t(p) + N(\mu, \sigma), & \text{if } I_t(p) + N(\mu, \sigma) \in [G_{\min}, G_{\max}] \\ G_{\min}, & \text{if } I_t(p) + N(\mu, \sigma) < G_{\min} \\ G_{\max} & \text{if } I_t(p) + N(\mu, \sigma) > G_{\max} \end{cases} \tag{19}$$

The value of σ is a function of the frame number t. In the first half of the sequence, we add the same level of noise to both images of the stereo pair. The value of σ increases from 1 to 50. For the second half of the sequence we only add noise to the match image. The values of σ used on the experiments are defined according to the formulas presented in the third and fourth row (from top to bottom) of Table 1. The second image from the right of Figure 4 shows the right image of the stereo pair at $t = 51$ using $N(\mu, 50)$.

Gaussian Blur: Blurring can occur in real-world images due to foggy or rainy environments. Also, a minor blurring can be caused by out-of-focus camera(s). We approximate this bias effect with a *Gaussian blurring convolution*, as known from scale space [7].

Given an image $I_t \in \mathcal{S}$, the *Gaussian blur functional* $\mathrm{F_b}(I_t)$ defines the image

$$\mathrm{F_b}(I_t) = I_t * G(k) \tag{20}$$

where $G(k)$ represents a $k \times k$ Gaussian smoothing kernel [3] and $*$ the convolution operator. In the first half of the sequence $\mathrm{F_b}(\mathcal{S})$, both images of the stereo pair are blurred with the same value of k which increases from 1 to 99, while in the second half of the sequence, only the match image is affected with a decreasing amount of noise (from 99 to 1).

The right-most image of Figure 4 depicts the reference image from the stereo pair at $t = 17$. Both images in the stereo pair were modified using $\mathrm{F_b}$ with $k = 33$. All the possible values of k can be calculated using the formulas presented in the last two rows (from top to bottom) in Table 1.

We use the terms *original, intensity altered, blurred*, and *Gauss* to identify $\mathcal{S}$, $\mathrm{F_i}(\mathcal{S})$, $\mathrm{F_g}(\mathcal{S})$, and $\mathrm{F_b}(\mathcal{S})$.

	BPM					GCM				SGM	
	dMax	sMax	λ_d	iter	lev	λ_1	λ_2	thres.	K	c_1	c_2
AD	100	500	0.3			4.2	1.4	1	7		
CEN	75	600	0.6	7	6	3	1	1	5	30	150
EPE	33	200	0.225			2.6	0.86	16	4.33		

Table 2: Parameters for the three versions of stereo algorithms used in this paper. BPM (SGM) uses identical values of iter (c_1) and lev (c_2) for all the three cost functions.

5.2. Stereo-Vision Algorithms

Binocular stereo-vision algorithms compare a pair of images of the same scene and match the *corresponding pixels*. When a couple of corresponding pixels has been identified, it is possible to calculate the coordinates of the "source" 3-dimensional point without having any prior knowledge of its position. (The importance of stereo analysis for driver-assistance systems is discussed, for example, in [19, 26].)

For the experiments to be reported in this paper, we selected three dense stereo-analysis algorithms based on techniques that had shown a good performance in previous studies [23, 24]. We use two (potentially) *global* [27] and one *semi-global* [13] algorithm. We test all of them with three different cost functions. Each cost function, see Section 5.3, considers different local features in the images when matching pixels.

Belief Propagation Matching (BPM): We use a max-product iterative *belief-propagation* algorithm as presented in [6]. This algorithm uses a truncation parameter for both, the data and the smoothness term. The smoothness term is a truncated *everywhere-smooth* quadratic function [27], which allows to obtain a smooth disparity map but without penalising depth discontinuities too much.

To speed up the matching process, a hierarchical approach (i.e. a coarse-to-fine approach) is considered, such that the passing of messages in a 4-adjacency grid is more efficient even with a reduced number of iterations. The original source code in [6] was modified so that we could use different types of cost functions and 12-bit stereo pairs as input images. The truncation parameters for the data (dMax) and the smoothness (sMax) terms, the weighting factor for the data term (λ_d), the number of iterations (iteration), and number of levels (level) of the followed hierarchical approach are shown in Table 2.

Graph Cut Matching (GCM): We use the graph cut-based algorithm presented in [4]. For minimising the energy function, a randomly initialised disparity map is considered as a weighted graph. The optimum disparity map is then calculated using the *α-expansion method* [20].

The implementation of this algorithm uses the binary Potts model as smoothness term to make sure that a global minimum is reached [20]. The values of the three parameters required for defining the Potts model (λ_1, λ_2 and the threshold) and the weighting factor for the cost function (K) are summarised in Table 2.

As with the GCM algorithm, we modified the original algorithm so that a wider range of cost functions could be used, as well as 12-bit stereo pairs could be used as input data.

Semi-Global Matching (SGM): We use the implementation presented in [11] of the semi-global matching algorithm proposed in [13]. A matching strategy followed by BPM or GCM can be characterised as being potentially global (but practically limited by the number of iterations). SGM limits its search space to a predefined set of *paths* to obtain an optimum disparity value with respect to this selected search space. We use the common configuration that considers an eight-path configuration, in contrast to the initially suggested sixteen-path configuration.

For each path in the search space, an energy function is minimised using a recursive dynamic programming approach. The smoothness term penalises small disparity changes of neighbouring pixels with a rather low penalty c_1 to allow slanted surfaces. A second higher penalty c_2 is applied for larger disparity changes. The second penalty is independent of the actual disparity change in order to preserve depth discontinuities [10]. The selected values for c_1 and c_2 are specified in Table 2.

5.3. Cost Functions

Three cost functions are considered for our experiments. Each one of them analyses different characteristics of the stereo input images to calculate the costs of assigning a certain disparity value to a given pixel. Two of them, the *census* and the *gradient-based* ones, have been highlighted as being robust in outdoor environments, see [10, 11]. The third one is the *sum of absolute differences*, which, in contrast to the other two, strongly depends on the photometric consistency between the images of the stereo pairs.

Consider $d \in \mathbb{N}$ as a possible disparity value between the *reference* I_r and *match* I_m image from a given *stereo pair*. Let $p_r = (x_r, y_r)^T \in I_r$ and $p_m = (x_r - d, y_r)^T \in I_m$.

Census Transform (CEN): The census cost function [32] calculates the cost of assigning the disparity d to p_r by analysing the *Hamming* distance between the *signature vectors* of p_r and p_m. Its use supports robustness of a stereo matcher against common types of noise found in real-world images [10].

Given an arbitrary image I and a neighbourhood $\mathcal{N}(p)$ of $p \in I$; the ith coordinate, $i = 1, \ldots, |\mathcal{N}(p)| - 1$, of the signature vector of p is defined as

$$S_g(p)_i = \delta_p\big(p'\big) \tag{21}$$

where $p' \in \mathcal{N}(p) \setminus \{p\}$, and

$$\delta_p\big(p'\big) = \begin{cases} 0, & \text{if } I(p) \neq I(p') \\ 1 & \text{otherwise} \end{cases} \tag{22}$$

The order in which the coordinates of the signature vectors are arranged is irrelevant, but has to be consistent. The comparison of the signature vectors is made coordinate-wise using the Hamming metric. Thus, the cost defined by the census transform of assigning the disparity d to the pixel p_r is given by

$$\mathrm{C_N}(p_r, d) = \sum_{i=1}^{|\mathcal{N}(p_r)|-1} \begin{cases} 0, & \text{if } S_g(p_r)_i = S_g(p_m)_i \\ 1 & \text{otherwise} \end{cases} \tag{23}$$

Following [11], we use a 9×3 neighbourhood as it favours a stronger data contribution along the epipolar line.

Gradient-Based Cost Function (EPE): The selected gradient-based cost function [16] analyses the L_1-distance between the end-points of the *gradient* vectors of p_r and p_m. This cost function is expected to have a good performance when using real-world data [10].

The cost of assigning the disparity d to the pixel $p_r \in I_r$ is given by

$$\mathrm{E_P}(p_r, d) = \Big| \big(\partial_x(p_r), \partial_y(p_r)\big) - \big(\partial_x(p_m), \partial_y(p_m)\big) \Big|_1 \tag{24}$$

where ∂_* denotes a discrete partial derivative calculated using *forward differences*. Symbol $|\cdot|_1$ denotes the L^1-norm.

Sum of Absolute Differences (EPE): The sum-of-absolute-differences is an intensity-based similarity measure. It is known for its poor performance when it comes to real-world stereo sequences [31] as the photometric consistency assumption is commonly violated in those data. The cost of assigning the disparity d to the pixel p_r given a neighbourhood $\mathcal{N}(p_r)$, is given by

$$\mathrm{S_D}(p_r, d) = \frac{1}{|\mathcal{N}(p_r)|} \sum_{p' \in \mathcal{N}(p_r)} \big| I_r(p') - I_m(q') \big| \tag{25}$$

where, if $p' = (x', y')^T$ then $q' = (x' - d, y')^T$ and thus $q' \in \mathcal{N}(p_m)$. For the experiments we use the 8-neighbourhood. Observe that $|\mathcal{N}(p_r)|$ denotes the cardinality of the neighbourhood $\mathcal{N}(p_r)$; however, the $|\cdot|$ symbol in the argument of the addition denotes the absolute value function.

We do not discuss the relationship between the performance of the stereo algorithms and the type of noise used to modify the sequences. See [23] for an evaluation procedure for stereo algorithms using computer generated images altered with real-world effects.

We use the abbreviations BPM-$*$, GCM-$*$ or SGM-$*$, where $*$ denotes CEN, EPE, or SAD, to denote a stereo-*configuration* defined by a stereo-matcher using a specific cost function.

5.4. Evaluation Methodology

We use three different neighbourhoods to define the experimental spatio-temporal configurations. We are interested to find out whether the size of the neighbourhood could affect the results of the filtering. The used spatio-temporal configurations are: *spatial-2* defined using the south and north neighbours, *spatial-4* defined using the 4-neighbourhood, and *spatial-8* defined using the 8-neighbourhood. Note that the larger the neighbourhood, the larger the dimensions of the vectors and matrices used to define the Kalman filter.

We compare the results of the spatio-temporal approaches with those obtained when considering information only from the temporal domain. We refer to the temporal analysis as *Temporal*. In general, the experiments showed that the spatial-n approaches with $n = 2, 4$ or 8 generate better results than Temporal.

On average, it took 0.18 s to process one frame using the Temporal approach. Spatial-2 required an average of 0.39 s to process each frame of the experimental sequence, while the computational time required for spatial-4 and spatial-8 was of 0.50 s and 0.84 s respectively.

To evaluate the post-processed disparity maps, we use the approach proposed in [29]. Given a filtered disparity map, we compute the average of all the values in the object of interest. Let I be an arbitrary image. Let P denote the power-set operator. The average is defined as the function

$$\mathrm{A_v}\colon \mathrm{P}(\Omega) \to \mathbb{R}^+ \tag{26}$$

such that

$$\mathrm{A_v}(\Omega_e) = \frac{1}{|\Omega_e|} \sum_{p \in \Omega_e} I(p) \tag{27}$$

where Ω_e denotes the image domain of I. In our case, Ω_e is the set of pixels defining either a moving object for the dynamic method, or a static object for the static method. We use as reference the average values computed from the available ground truth and the raw disparity maps calculated with any of the algorithms before being filtered. In the plots and tables presented in Section 5.6, we denote the average values from the raw disparity maps as Raw, and as GT for those calculated with the ground truth data.

5.5. Kalman Filter Parametrization

The parameters to initialise the Kalman filter for the dynamic method are as follows (for $n = 2$, and analogously for larger neighbourhoods): the initial state was filled up with data from the disparity map calculated for the first available stereo pair, except for the disparity rate term, which was set to zero.

For the matrix $\mathbf{A}$ [see Equation (14)], let $\alpha = 0.8$, $\beta_1 = \beta_2 = 0.1$. With this values we aim to give a heavier weight to the disparity value of the pixel under analysis. Also, this combination of values has reported to us good results in a previous study [23]. Assuming a frame rate of 25 frames per second, we set $\Delta_t = 0.04$.

The entries of the covariance matrix $\mathbf{P}_{0|0}$ [see Equation (16)] were initialised with the following values: $\sigma^2_{d_*} = 0.3$, assuming a non-perfect disparity map, $\sigma_{d_* d_*} = 0.5$, a relatively large value to represent a strong correlation between the pixel under analysis and its neighbours, and $\sigma_{d_* \dot{d}} = 0.0001$, to show a weak correlation between the disparity values and the disparity rate. d_* represents either d, d_1, or d_2.

Finally, we use $\sigma_{\dot{d}^2} = 1$, a relatively large value to express a high uncertainty in the initial disparity rate.

The parameter initialisation for the static method is analogous. It is only necessary to remove the terms where the disparity rate is involved and modify the corresponding matrices accordingly.

5.6. Results and Discussion

Results for each of the sequence are summarised in tables. Numbers in each table represent the average deviation from the ground truth for the entire sequence. We highlight those numbers (using a light-blue shading) being the lowest average deviation among the three configurations, for each matcher. When two or more filtering approaches reported the same number, we choose to highlight that one which requires less computation time (i.e., which is using a smaller neighbourhood). Using bold font we also highlight the results for Temporal as we aim to emphasise that the use of the spatio-temporal data improved most of the times the results compared with the method which used only data from the temporal domain.

Figure 5: Two sets of raw and spatial-2 filtered results using the colour coding shown in the right image of Figure 2. For each set, the left vehicle corresponds to raw while the right vehicle was encoded using the filtered results. The *left set* was generated with GC-SAD using the original sequence. The *right set* was generated with SGM-SAD using the intensity altered sequence.

For the original sequence, all the stereo-vision configurations, using either the dynamic or static approach, reported better results when we used data from the spatio-temporal domains than when we only used data from the temporal one. For spatial-2 and spatial-4, the disparity values were closer [with respect to the index defined in Equation (27)] to the ground truth values than those obtained with Temporal. The spatial-8 configuration was only better than Temporal for the static approach.

For the modified sequences, and for some of the stereo-vision configurations, the filtering process degraded the already low-quality raw-disparity maps. The more data from the spatial domain were incorporated, the worse the obtained results. However, in general, the spatial-2 and spatial-4 filtering configuration improved the quality of the disparity maps obtained with Temporal. Spatial-8, once again, only improved the results when using the static approach.

In Figure 5 we show two sets of raw (left vehicle) and spatial-2 (right vehicle) filtered results using the colour coding depicted in the right image of Figure 2. For the two sets, spatial-2 (and also spatial-4) managed to discard noisy measurements generating more homogeneous results than those obtained with raw and Temporal. The left set corresponds to the segmented static vehicle (i.e. for the static approach) when using the original sequence and GC-SAD. The evaluation index reported 12.23 and 5.58 for raw and for spatial-2, respectively. The ground truth value for this frame was of 5.3. See the top left chart in Figure 6 for the results over the entire sequence for this particular configuration.

The right set corresponds to the dynamic approach using the intensity-altered sequence and SGM-SAD. The evaluation index in this case reported 47.55 for raw, and 42.95 for spatial-2. The ground-truth value equals to 3.0. Extremely bad initial results made it impossible to generate useful disparity data after the filtering. However, spatial-2 (and also spatial-4) reported better results than raw and Temporal. The results for the entire sequence for this configuration are depicted on the top-right chart of Figure 6. Note that in all of the plots shown in this figure, we chose not to show the graph representing the spatial-4 results, as it is almost identical to that obtained with spatial-2.

Dynamic Approach

For all the sequences, the stereo-configurations showed a consistent behaviour after the filtering process. When compared with the other filtering approaches, Spatial-8 got the worst results. In some cases, it even generated worse results than those obtained

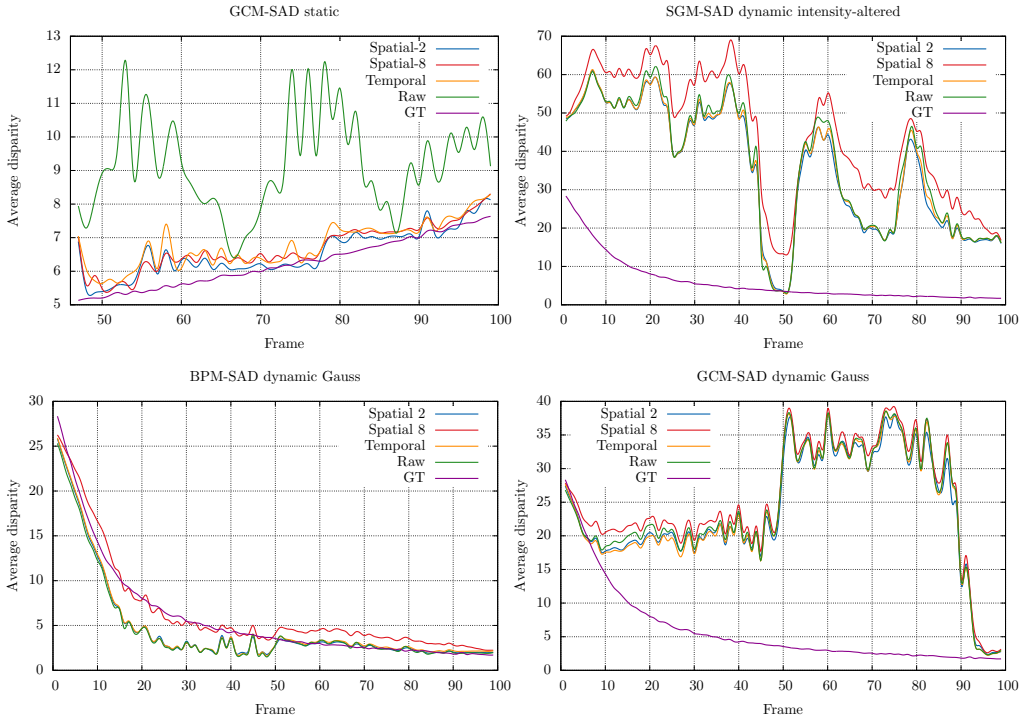

Figure 6: Sample plots of obtained results. *Top left*: GCM-SAD results for the original sequence using the static approach. *Top right*: SGM-SAD results for the intensity-altered sequence using the dynamic approach. *Bottom left*: BPM-SAD results for the Gauss sequence using the dynamic approach. *Bottom right*: GCM-SAD results for the Gauss sequence using the dynamic approach.

with raw. As expected, a large neighbourhood degraded the final filtered disparity value. Spatial-2 and -4 performed almost identically. They showed a better performance than Temporal or raw in most of the cases.

The improvements obtained for the original sequence were quite minor due to the fact that the original disparity maps were quite close to the ground truth values. For two of the stereo-configurations (SGM-CEN and SGM-SAD), the filtering process degraded a bit the original results. However, for all of the stereo-configurations, spatial-2 and -4 generated better results than Temporal. The results for the original sequences are summarised in Table 3. Also, see the top-left chart in Figure 6 for the results of GCM-SAD across the whole sequence. Note how the filtering process damped correctly the noisy raw disparity data.

The results obtained for the intensity-altered sequence were similar to those of the original sequence for stereo-configurations defined using CEN and EPE. These two cost functions have been reported as robust against this common real-world effect [10]. However, as depicted in Table 4, a small degradation could be detected in the raw results.

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	0.15	**0.13**	0.17	**0.25**	0.62	1.46	0.31	0.18	0.17
Spatial-4	0.15	0.13	0.17	0.25	0.62	1.68	0.30	0.20	**0.15**
Spatial-8	3.81	3.85	4.13	3.46	3.48	4.84	3.23	3.02	3.27
Temporal	**0.27**	**0.24**	**0.29**	**0.37**	**0.69**	**1.47**	**0.35**	**0.52**	**0.42**
Raw	0.23	0.27	0.35	0.25	0.78	2.43	0.27	0.44	0.12

Table 3: Average deviation from the ground truth for the three stereo algorithms using the dynamic approach. We highlight (using light-blue shading) the lowest average deviation from the ground truth for every stereo-vision algorithm and, using bold font, we also highlight the results for Temporal for an easier comparison between the spatio-temporal and the temporal approaches.

Spatial-2 and -4 improved the Temporal and raw results for almost all of the stereo-configurations. The configurations defined with SAD generated disparity maps that were near to useless data. For example, the deviation from the ground truth was larger than for GCM-SAD 85 disparity units. The filtering procedures improved the results, but they were still "quite far away" from the ground truth vales.

For the blurred sequence we observed that Temporal was the outperforming technique in 7 out of 9 stereo-configurations. An explanation for this is that the blurring effect caused that the raw disparity maps were also blurred. Therefore, the information contained in the spatial domain would become less reliable. However, spatial-2 and spatial-4 generated better results than all of those obtained with raw.

It is interesting to note that the best overall performing configuration was defined using the simple SAD cost function. BPM-SAD defined a quite robust configuration against this kind of noise. The results for this sequences are summarised in Table 5

For the Gauss sequence, the spatio-temporal approaches performed better than Temporal in 4 out of 9 stereo-configurations, but were always improving the raw results. As with the blurred sequence, BPM-SAD was the best performing configuration. Compare the plots in the lower row in Figure 6, where we depict the BPM-SAD results (left) and GCM-SAD. Even given that spatial-2 and Temporal improve the results by more than 5

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	0.18	0.17	16.39	0.29	2.51	81.26	0.98	5.12	26.48
Spatial-4	0.17	**0.16**	16.40	**0.28**	3.25	81.04	**0.92**	5.41	26.78
Spatial-8	4.18	3.96	25.09	3.42	6.37	109.88	4.10	8.16	42.06
Temporal	**0.36**	**0.27**	**16.03**	**0.41**	**2.79**	**82.36**	**1.18**	**5.13**	**27.12**
Raw	0.31	0.29	16.41	0.22	4.53	85.55	1.56	8.24	33.09

Table 4: Intensity altered sequence results for the dynamic approach. The layout is that of Table 3

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	2.65	1.41	0.66	5.38	18.09	12.77	12.18	12.12	18.46
Spatial-4	2.66	1.43	0.70	5.29	18.11	14.22	12.39	12.39	18.49
Spatial-8	6.18	1.98	2.83	6.79	18.64	20.00	23.23	20.65	19.75
Temporal	**2.68**	**1.26**	**0.53**	**5.06**	**18.07**	**13.15**	**12.94**	**11.79**	**18.42**
Raw	3.07	1.80	1.26	5.97	18.40	15.77	16.55	16.17	18.52

Table 5: Blurred sequence results for the dynamic approach. The layout is that of Table 3

disparity units, this data is still too much biased. The results generated for this sequence are summarised in Table 6.

Static Approach

In general, as with the dynamic approach, the spatial-2 and spatial-4 methods reported better results than Temporal and raw. The main difference (with respect to the dynamic methods) is that for some stereo-configuration in all of the sequences, spatial-8 outperformed Temporal, raw, and the other spatio-temporal methods.

For the original sequence, the raw results were deviated from the ground truth less than 0.25 disparity units for the majority of the stereo-configurations. Thus, the improvement reported from the filtering approaches was marginal and, for some stereo-configurations, it was even worse than that of the raw disparity maps. The exceptions were GCM-SAD and SGM-EPE whose raw results obtained the worst evaluation index. All the spatio-temporal approaches improved by more than 50% compared to the indices reported by raw, as visible in Table 7. See also the top-left chart in Figure 6 for results generated using GCM-SAD. Note that the disparity jumps were correctly damped by the filtering approaches.

Interestingly, for all the filtering techniques, the BPM configurations reported worse results than those reported by the raw disparity maps. There were almost no noisy disparity values to be filtered out. It is interesting to note that Temporal and spatial-8 introduced a large amount of noise. Spatial-2 and -4 reported almost the same index as the raw results.

The results for the intensity altered sequence were similar to those of the original sequence for the configurations defined by the CEN and EPE cost functions. The raw

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	2.54	2.24	**0.98**	3.00	17.63	15.80	**2.21**	34.81	11.19
Spatial-4	2.67	3.25	0.99	**2.90**	19.70	19.03	2.34	35.36	11.21
Spatial-8	2.94	3.03	2.26	3.16	23.97	29.29	2.25	42.50	11.50
Temporal	**2.32**	**2.28**	**1.01**	**3.06**	**16.25**	**15.54**	**2.29**	**34.13**	**11.17**
Raw	2.82	4.62	1.33	3.06	23.45	22.50	2.25	40.44	11.23

Table 6: Gauss sequence results for the dynamic approach. The layout is that of Table 3

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	**0.10**	0.11	0.13	**0.19**	0.34	0.35	0.13	0.11	0.14
Spatial-4	0.10	0.11	0.13	0.19	0.33	0.36	0.16	0.11	0.13
Spatial-8	0.20	0.22	0.20	0.22	0.24	0.45	**0.09**	0.20	0.19
Temporal	**0.22**	**0.24**	**0.23**	**0.25**	**0.37**	**0.55**	**0.18**	**0.32**	**0.28**
Raw	0.09	0.11	0.13	0.22	0.42	2.94	0.19	1.05	0.25

Table 7: Original sequence results for the static approach. The layout is that of Table 3

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	0.10	0.11	23.24	**0.18**	3.65	67.48	0.17	4.51	**28.71**
Spatial-4	0.10	0.11	23.34	0.19	3.67	67.48	0.17	4.52	28.72
Spatial-8	0.10	0.11	23.34	0.18	3.72	67.37	0.19	4.53	28.74
Temporal	**0.10**	**0.11**	**23.36**	**0.19**	**3.70**	**67.47**	**0.20**	**4.49**	**28.67**
Raw	0.11	0.12	23.38	0.21	3.66	67.44	0.18	4.56	28.80

Table 8: Intensity altered sequence results for the static approach. The layout is that of Table 3.

results for the SAD cost functions were more than 20 disparity units deviated from GT, as depicted in Table 8. The filtering process managed to improve "a bit" the noisy measurements with at least one spatio-temporal approach outperforming Temporal.

For the blurred sequence, for all the configurations, at least one of the spatio-temporal approaches reported better results than those obtained using Temporal. Interestingly, for the SGM configurations, spatial-8 was the spatio-temporal approach that obtained the better results. It is also interesting to note the large difference between the results generated with dynamic and static approaches using SGM-EPE. For the static approach, the deviation from the ground truth was 3 times smaller than that obtained with the dynamic approach. Compare Tables 5 and 9.

For the Gauss sequence, the spatio-temporal approaches performed better than Temporal in 7 out of 9 of the stereo-configurations. It is interesting to note that the BPM and the GCM-CEN configurations were "quite robust" against this type of noise. Although the average deviation of raw maps was less than 1 disparity unit, the spatio-temporal approaches managed to improve the evaluation index when using GCM-CEN.

The poor performance of SGM-EPE is also remarkable. The average deviation for this configuration was of about 41 disparity units. Post-processing approaches did not manage to improve the results, which could be an expected result for such a bad kind of initial data. Note that similar results were obtained with this stereo-configuration when using the static approach. The results for this sequence are summarised in Table 10

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	2.89	**2.31**	3.16	**2.58**	3.20	16.55	26.53	5.71	16.22
Spatial-4	2.90	2.31	3.16	2.58	3.20	16.55	26.57	5.72	16.27
Spatial-8	2.90	2.32	3.14	2.60	3.22	16.52	26.52	5.69	16.17
Temporal	**2.91**	**2.32**	**3.16**	**2.60**	**3.23**	**16.57**	**26.54**	**5.69**	**16.22**
Raw	2.90	2.31	3.18	2.60	3.24	16.64	26.64	5.80	16.40

Table 9: Blurred sequence results for the static approach. The layout is that of Table 3

	BPM			GCM			SGM		
	CEN	EPE	SAD	CEN	EPE	SAD	CEN	EPE	SAD
Spatial-2	**0.39**	0.67	0.42	**0.68**	23.16	21.20	5.08	41.43	19.17
Spatial-4	0.39	0.67	0.42	0.68	23.17	21.25	5.09	41.46	19.23
Spatial-8	0.40	0.69	0.44	0.69	23.14	21.29	5.14	41.53	19.39
Temporal	**0.41**	**0.68**	**0.42**	**0.71**	**23.19**	**21.23**	**5.04**	**41.53**	**19.14**
Raw	0.39	0.65	0.42	0.71	23.00	21.09	5.05	41.29	19.10

Table 10: Gauss sequence results for the static approach. The layout is that of Table 3

6. Conclusions

In this paper we discuss a method for post-processing disparity maps generated by stereo-vision algorithms. The idea was to merge information from the spatial and temporal domains using iconic Kalman filters and the available ego-motion data from the ego-vehicle. We aimed at improving previously reported post-processing methods where only data from the temporal domain was used.

The results obtained in the performed experiments, where three different stereo algorithms where tested using four long synthetic sequences, showed that the inclusion of spatial information does have a positive impact on the improvement of the disparity measurements. For the dynamic filtering approaches, if the neighbourhood defining the spatio-temporal technique is kept small, then results are better than for cases where filtering only uses data from the temporal domain. It seems that, as expected, larger neighbourhoods tend to degrade the filtering process. For the static approaches, all the spatio-temporal filtering techniques reported better results than those obtained with Temporal. For a few stereo-configurations, spatial-8 reported the best results.

Not all the filtering methods reported better measurements than those from the raw disparity data. The stereo-configurations generated fairly accurate disparity maps when using synthetic data as input. However, for those stereo-configurations that generated noisy disparity maps, the proposed techniques managed to improve the disparity measurements at a better measurable rate. The used modified (i.e. noisy) sequences indicate also that the proposed approach is potentially useful to improve disparity maps generated for real-world data. However, we also noticed that when filtering is initialised with highly inaccurate disparity maps, the proposed post-processing approach is unlikely to improve disparity estimation.

References

[1] N. Atzpadin, P. Kauff and O. Schreer. Stereo analysis by hybrid recursive matching for real-time immersive video conferencing. *IEEE Trans. Circuits Systems Video Technology*, **14**:321–334, 2004.

[2] H. Badino, U. Franke and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In Proc. *ICCV*, pages 1–12, 2007.

[3] G. Bradsky and A. Kaehler. *Learning OpenCV. Computer Vision with the OpenCV Library* . O'Reilly Media Inc., 2008.

[4] Y. Boykov, O. Veksler and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis Machine Intelligence*, **23**:1222–1239, 2001.

[5] *.enpeda..* (Environment Perception and Driver Assistance) Group, The University of Auckland. EISATS (*.enpeda..* Sequence Analysis Test Site), dataset 2, `http://www.mi.auckland.ac.nz/EISATS`. Retrieved 2012 [online].

[6] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Computer Vision*, **70**:41–54, 2006.

[7] L. Florack, B. Romeny, M. Viergever and J. Koenderink. The Gaussian scale-space paradigm and the multiscale local jet. *Int. J. of Computer Vision*, **18**:61–75, 1996.

[8] U. Franke, C. Rabe, H. Badino and S. Gehrig. $3D$-vision: Fusion of stereo and motion for robust environment perception. In Proc. *DAGM*, pages 216–223, 2005.

[9] D. Halliday, R. Resnick, J. Walker. *Fundamentals of Physics*. John Wiley and Sons, 7th edition, 2004.

[10] S. Hermann, S. Morales and R. Klette. Illumination invariant cost functions in semi-global matching. In Proc. *CVVT*, volume 2, pages 245–254, 2010.

[11] S. Hermann, S. Morales and R. Klette. Half-resolution semi-global stereo match. In Proc. *IEEE IV Symposium*, pages 201–206, 2011.

[12] F. Hill and S. Kelley *Computer Graphics using OpenGL*. Pearson Prentice Hall, 3rd edition, 2007.

[13] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In Proc. *CVPR*, volume 2, pages 807–814, 2005.

[14] K. Kafadar. Gaussian white-noise generation for digital-signal synthesis. *IEEE Transactions on Instrumentation and Measurement*, IM35(4):492–495, 1986.

[15] R. Kalman. A new approach to linear filtering and prediction problems. *J. Basic Engineering*, **82**:35–45, 1960.

[16] A. Klaus, M. Sormann and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In Proc. *ICPR*, volume 3, pages 15–18, 2006.

[17] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel and R. Klette. Moving object segmentation using optical flow and depth information. In Proc. *PSIVT*, pages 611–623, 2009.

[18] R. Klette and A. Rosenfeld. *Digital Geometry: geometric Methods for Digital Picture Analysis*. Morgan Kaufmann Publishers Inc., 2004.

[19] R. Klette, N. Krüger, T. Vaudrey, K. Pauwels, M. Hulle, S. Morales, F. Kandil, R. Haeusler, N. Pugeault, C. Rabe and M Leppe. Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. *IEEE Trans. Vehicular Technology*, **60**:2012–2026, 2011.

[20] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis Machine Intelligence*, **26**:147–159, 2004.

[21] B. Kuo and F. Golnaraghi. *Automatic Control Systems*, 8th edition. John Wiley & Sons, Inc., 2002.

[22] L. Matthies, T. Kanade and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. Computer Vision*, **3**:209–238, 1989.

[23] S. Morales, T. Vaudrey and R. Klette. Robustness evaluation of stereo algorithms on long stereo sequences. In Proc. *IEEE IV Symposium*, pages 347–352, 2009.

[24] S. Morales and R. Klette. Ground truth evaluation of stereo algorithms for real world applications. In Proc. *CVVT*, volume 2, pages 152–162, 2010.

[25] S. Morales and R. Klette. Spatio-temporal stereo disparity integration. In Proc. *CAIP*, pages 540–547, 2011.

[26] Y. Satoh and K. Sakaue. An omnidirectional stereo vision-based smart wheelchair. *EURASIP J. Image Video Processing*, pages 1–11, 2007.

[27] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, **47**:7–42, 2001.

[28] K. Schauwecker, S. Morales, S. Hermann and R. Klette. A comparative study of stereo-matching

algorithms for road-modeling in the presence of windscreen wipers. In Proc. *IEEE IV Symposium*, pages 7–12, 2011.

[29] T. Vaudrey, H. Badino and S. Gehrig. Integrating disparity images by incorporating disparity rate. In Proc. *Robot Vision*, pages 29–42, 2008.

[30] T. Vaudrey, C. Rabe, R. Klette and J. Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In Proc. *IVCNZ*, pages 1–6, 2008.

[31] T. Vaudrey, S. Morales, A. Wedel and R. Klette. Generalized residual images effect on illumination artifact removal for correspondence algorithms. *Pattern Recognition*, **44**:2034–2046, 2011.

[32] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In Proc. *ECCV*, volume 2, pages 151–158, 1994.

[33] X. Zhihui (editor). *Computer Vision*. InTech, online open access, 2008.