Real-Time Dense Stereo for Intelligent Vehicles

Wannes van der Mark and Dariu M. Gavrila

***Abstract*— Stereo vision is an attractive passive sensing technique for obtaining three-dimensional (3-D) measurements. Recent hardware advances have given rise to a new class of real-time dense disparity estimation algorithms. This paper examines their suitability for intelligent vehicle (IV) applications. In order to gain a better understanding of the performance and computational cost trade-off, the authors created a framework of real-time implementations. This consists of different methodical components based on Single Instruction Multiple Data (SIMD) techniques.**

Furthermore, the resulting algorithmic variations are compared with other publicly available algorithms. The authors argue that existing, publicly available stereo data sets are not very suitable for the IV domain. Therefore, the authors' evaluation of stereo algorithms is based on novel realistically looking simulated data as well as real data from complex urban traffic scenes. In order to facilitate future benchmarks, all data used in this paper is made publicly available.

The results from this study reveal that there is a considerable influence of scene conditions on the performance of all tested algorithms. Approaches that aim for (global) search optimization are more affected by this than other approaches. The best overall performance is achieved by the proposed multiple window algorithm which uses local matching and a left-right check for robust error rejection.

Timing results show that the simplest of the proposed SIMD variants are more than twice as fast than the most complex one. Nevertheless, the latter still achieve real-time processing speeds while their average accuracy is at least equal to that of publicly available non-SIMD algorithms.

***Index Terms*— Dense disparity, real time, single instruction multiple data (SIMD), stereo vision.**

I. Introduction

An appealing application of intelligent transport systems (ITS) is the automatization of the transport of people and goods in inner city environments. In order to preserve safety in such complex environments, current operational systems, such as people movers, need areas or lanes that are separated from other traffic. Reliable, robust and real-time obstacle detection methodologies are needed to enable the safe operation of these types of intelligent vehicles (IV) among other traffic participants such as cars and pedestrians.

Stereo vision has the advantage that it is able to obtain an accurate and detailed 3D representation of the environment around a vehicle, by passive sensing and at a relatively low sensor cost. The work by Labayrade et al. [18] is an example of a real-time stereo system that is able to detect vehicles up to 80 m away. This and other previous applications (e.g. [6]) for IV have mostly used sparse, feature-based approaches to stereo vision. Here only a subset of image pixels (e.g. vertical edge pixels) are matched, in order to meet real time processing requirements.

However, by only using sparse depth data, it is more difficult to perform a subsequent object segmentation step. For example, the vertical edges of a single object are often separated. If edges of different objects are near to each other it is difficult to determine which of them belong to the same objects. This complicates the application of other processing steps such as classification and tracking because these require some form of image segmentation. For this reason, it is attractive to use dense stereo vision, that tries to estimate disparity for all image points.

A large research community centres around dense stereo vision because it is attractive for a number of applications such as robot navigation, surveillance systems, 3D modelling, augmented reality and video conferences. Many systems for dense stereo vision or disparity estimation have been presented, as discussed in two large surveys of the field, one by Scharstein and Szeliski [22] and another by Brown et al. [3].

In contrast to previous surveys, this paper does not aim to review the whole field of dense stereo. Our aim is to investigate if certain approaches to dense stereo vision are more suitable for IV applications than others. The criteria of this investigation are founded on practical considerations specifically related to the IV domain.

The first of these considerations is that application of dense stereo in IV is only possible if the disparity map can be calculated in real-time. Single Instruction Multiple Data (SIMD) offers an appealing and straightforward way for speeding up computation by carrying out one operation on multiple values simultaneously. Because the parallelism is only in terms of the data, difficult problems such as process synchronization can be avoided.

Over the past few years, manufacturers have extended general purpose processors with SIMD capabilities in response to demanding multimedia applications (e.g. SSE2 instruction set for Intel processors, as used in this paper). Yet SIMD also forms the basis architecture for special hardware as used in the ITS domain (e.g. DSPs), which faces particular demands with respect to power consumption, cost and compactness. Therefore, in order to evaluate dense stereo vision algorithms from IV perspective, it is important to consider their suitability regarding SIMD parallelism.

In this paper, we identify different methodological components and develop efficient underlying SIMD implementations. The latter are combined in a single framework enabling

Manuscript received April 8, 2005; revised September 9, 2005 and October 13, 2005. This work was supported in part by the TNO project "Automatised Safety for Traffic and Transport" (AV3) and by the 5th Framework EU Project SAVE-U (IST-2001-34040).

W. van der Mark is with the Electro-Optics Group at TNO Defence, Security and Safety, Oude Waalsdorperweg 63, P.O. Box 96864, 2509 JG The Hague, The Netherlands (e-mail: wannes.vandermark@tno.nl).

D. M. Gavrila is with the Intelligent Systems Group at the Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands (e-mail: gavrila@science.uva.nl). He is also with the Machine Perception Department of DaimlerChrysler Research, Ulm, Germany.

comparisons and analysis of the various approaches on an equal basis.

A second consideration is that the output of stereo algorithms on itself is not interesting for IV applications. Only subsequent steps, such as obstacle detection or segmentation, can provide useful information about the vehicle surroundings. Other work on dense stereo vision has often used error measures where only the quality of the disparity values was evaluated. We will present error measures and evaluation techniques which are more related to typical applications of stereo vision in the IV domain.

The outline of this paper is as follows. Section II first discusses a number of concepts for dense disparity computation from the literature. In Section III, we present the corresponding real-time SIMD implementations. Section IV compares the resulting algorithms with additional, publicly available approaches [1], [22] on both simulated and real data depicting complex urban traffic scenes. Section V contains the conclusion.

II. Approaches to stereo vision

The goal of stereo disparity estimation is finding the correct correspondences between image points from the left and right camera. For each point, the positions of possible matches in the other image is constrained to a single epipolar line, if the stereo camera geometry is known. Most approaches to disparity assume that the epipolar lines run parallel to the image lines, so that corresponding points lie on the same image lines. This situation can be achieved for stereo cameras by using a rectification technique [7]. If images are rectified, the disparity d between a corresponding left point l and a right point r can be expressed as:

$$d = l - r \tag{1}$$

The disparity space contains all possible matches for the same left and the right stereo image line. The possible matches for a point of the left line are a column in this space, the possible matches for a point on the right line form a row. Often, a minimum d_{min} and maximum d_{max} disparity are used to bound this space. Fig. 1 shows a drawing of the disparity space with $d_{min} = 0$.

Usually, one can distinguish two stages in a disparity estimator. In the first stage, cost values are calculated for comparing the different points in the disparity search space. These cost values are used in the second stage for searching the correct points (matches) in the disparity space. Some algorithms use additional pre- and post-processing steps. In order to simplify the matching step, pre-processing is applied to reduce the illumination differences between the stereo images. A typical post-processing step is the detection of occlusions, that are image regions only visible in one of the stereo images.

In the following sections we describe methods from literature that can be used for each step.

A. Pre-processing

A complicating factor for stereo matching is that the intensities of corresponding pixels from the stereo images can be different. This can be caused by unequal left and right camera sensor characteristics such as brightness and contrast. It is also due to differences in lighting conditions at each of the camera positions. If raw input images are used, it is necessary to use an illumination invariant similarity measure. Because invariant measures are computationally more expensive, preprocessing of input images is often applied to reduce the illumination differences beforehand. One approach subtracts median filtered versions from the original input images [24]. A more popular approach is the convolution of the input images with a Laplacian of Gaussian (LoG) kernel [4]. This reduces illumination influences because the response of the Laplacian is zero in areas with constant intensity while it is either positive or negative near edges with high intensity gradient.

Preprocessing can also be used to extract extra information in order to aid the subsequent disparity search. Hong and Chen [11] use colour based segmentation to find similarly coloured patches. They match patches instead of individual pixels because the assumption is that no large disparity discontinuities occur within the homogeneous coloured patches.

B. Similarity measures

The simplest similarity measures are based on the difference in pixel intensity, such as absolute difference (AD) or squared difference (SD). Algorithms that only use these single intensity measures in order to compare points are known as "pixel-to-pixel" algorithms. Unfortunately, discrete images have a quite limited number of different gray-level intensity values. It is possible to use colour instead [20]. However, colour is difficult to use during nighttime conditions due to monochromatic street lightening.

Pixel-to-pixel measures are not very distinctive when intensities of different pixels are the same or corrupted by noise. Birchfield and Tomasi [1] also pointed out the sensitivity to image sampling. For example, the pixel intensities on corresponding stereo edges can be different due to aliasing. They have therefore designed a measure that is less sensitive to image sampling.

A more commonly used approach for improving distinction, is using a larger support region for aggregating the cost values. Typically, the sum of the pixel differences in a window around a pixel of interest is used, such as the sum of absolute differences (SAD) or the sum of squared differences (SSD).

The use of larger windows will lead to more robustness against noise. However, larger windows with fixed size and centre point will lead to less accuracy in disparity estimates. This is due to the fact that areas on slated surfaces will warp projectively between the stereo images. A square window on such a surface will therefore only correspond correctly to a projectively warped version in the other image.

Occlusions near object edges can also cause problems. When a large window is centred around a background point near a object edge it will almost certainly encapsulate a portion of the foreground object. In the case of an occlusion, a large amount of background pixels in the window will not be visible in the other image. The resulting similarity measure will wrongfully be biased towards disparities that belong to the foreground pixels.

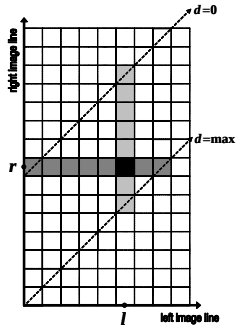


Fig. 1. Disparity search space. The light gray area shows the left-to-right search area for a point match, while the dark area shows the right-to-left search area.

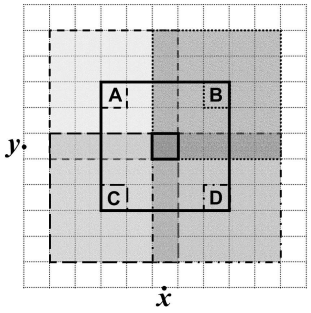


Fig. 2. Multiple cost windows. Apart for the centre window indicated by the solid lines, four additional windows centred at A,B,C and D are used.

In the literature, several ways can be found to improve window based matching. An adaptive size cost window was proposed by the Kanade and Okutomi [12]. Given an initial guess of disparity, they use a statistical technique in order to estimate the optimal size and box shape of each matching window. However, estimating the optimal window size at all points is computationally expensive.

Other ways of improving the cost computation include multi-scale techniques and the use of multiple windows. In some multi-scale approaches several matching windows of different sizes are used. The larger windows provide the robustness, while the smaller windows provide precision. On the downside, matching errors at the coarse scale might propagate to the finer scales.

A somewhat similar approach is not changing the size of the windows, but having different options for the location of the centre point. Both Bobick et al. [2] and Fusiello et al. [8] use nine different window centre points in their approaches. For real-time applications, Hirschmüller et al. [10] suggest the use of five-window configuration. Fig. 2 shows the regular matching window centred in the middle. On the edge points (A,B,C,D) of this window four additional windows have been indicated. Of these four windows the two with the lowest SAD values are searched. They are added to the value of the centre window. This step acts as a sort of deformable window, which means that the resulting window does not have to be centred on the point itself.

C. Disparity search

When similarity or cost values in the disparity search space have been calculated, the correct matches can be searched. A straightforward way of doing this is searching the disparity interval per point for a optimum value. This approach is known as the Winner-Takes-All (WTA) approach.

A drawback of the simple WTA approach is that it does not consider the presence of occlusions. This will result in wrong disparity estimates for those regions. Because WTA only searches for optimum cost values it is very sensitive to the results from the cost computation. The cost values computed for a textureless region often do not indicate a optimum match due to their similarity. Repetitive texture, such as bars, can lead to several ambiguous optimums. In these areas, the WTA is prone to errors.

Assumptions about the scene geometry can be exploited to improve disparity estimates. Three often applied assumptions are the "smoothness constraint", "uniqueness constraint" and "ordering constraint". The smoothness constraint is based on the observation that changes in depth on surfaces are much smaller than those at the edges of objects.

The uniqueness constraint states that a 3D point only has exactly one projection in each of the stereo cameras. Therefore, only one correspondence has to be found for every stereo point pair. The constraint is only violated by image points on transparent material or occluded image parts. The ordering constraint states that the sequence in which points are ordered on a left image line, is the same for the right image line. If occlusions are present, points can be missing from one of the line sequences, but the ordering will remain. There are scene situations possible where the ordering constraint is violated [26]. However, it is assumed that they are not very common.

A cost function (CF) is one approach to enforcing smoothness. It can be used to locally add penalties to the values in the disparity search interval of each next point on the scanline. Mismatches can be suppressed by penalizing large jumps in disparity between the scanline points. The difficulty of this approach is how to choose the penalty function. If penalties are too high, disparity jumps near edges are missed and if they are too low the smoothness in not enforced.

Given a correct stereo match in the disparity space, both the uniqueness and ordering constraint limit the number of possible matches for the following pixel on the reference image. Because of the limitations, finding the correct disparities is now akin to finding a valid path that takes the shortest route through the cost values. Special rules for how to transverse the search space can be added in order to handle occlusions and jumps in disparity. Dynamic-programming (DP) techniques are often used for this approach [2].

Another search optimization technique called Scanline Optimization (SO) was proposed by Scharstein and Szeliski [22]. Instead of enforcing special rules for occlusion and disparity jumps, they use a global variant of a smoothness cost function to add penalties to the matching costs. Then a disparity is searched for each scanline point that minimizes the total cost.

In both the SO and DP the optimization and search techniques are constrained to one single scanline. Just like CF, they are very dependent on the choice of rules that govern the path propagation or the cost functions that are used for searching the optimum solution. This can cause them to miss or wrongfully suppress large jumps in disparity, for example near object edges. These types of errors cause "streaks" of erroneous disparity estimates along scanlines. Interline consistency is difficult to maintain with these techniques. Recently, some work appeared on optimizing in more than one direction. The two pass approach of Kim et al. [14], uses the results of the first optimization pass along the scanlines to optimize the estimates across the scanlines in the second pass.

An alternative approach is to view disparity search as

a directed graph labelling problem. The graph nodes can represent left and right pixel pairs with disparity as label. They can also represent whole pixel patches extracted with a pre-processing step. Each transition between nodes has a cost specified by a CF that partially depends on the assigned disparity label. In several approaches [17], [11], [5] graph cuts are used to the find disparity label assignments that minimize the transition costs. The graph optimization approach carries a high computational cost. However, the algorithms based on it are among those that currently reside in the top positions of the performance ranking on the Middlebury stereo vision web site [28].

D. Post-processing

This section is devoted to some of the additional steps performed next to the disparity estimation. These include error detection, subpixel interpolation and occlusion removal.

It was mentioned earlier that image regions with little or repetitive texture are problematic for stereo algorithms. Some approaches try to detect possible errors in the disparity estimates. Fusiello et al. [8] use the variance of the cost values in the disparity range of a point as a measure of confidence. Other approaches [10] select the optimum C_1 and second best value C_2 from the disparity search range. The confidence in the selected optimum is expressed as:

$$C = \frac{C_2 - C_1}{C_1} \tag{2}$$

The idea is based on the fact that the cost values of a textureless region will be very similar. Repetitive texture will lead to several optimums in the disparity interval. The measure of Eq. 2 will be low in both cases. However, Mühlmann et al. [20] argue that there are occurrences where the optimum correspondence is actually between two pixels. This also leads to two very similar optimum values in the disparity search space. To prevent discarding good estimates on these points, they suggest using the third best optimum instead of the second best.

Besides using integer values for disparity correspondence, floating-point values can be also used for subpixel accuracy. Many algorithms use an additional post-processing step for improving the disparity estimate to subpixel accuracy. If the SSD similarity measure is used, the cost values near an optimum can be approximated by a second degree polynomial. Given the cost values of the optimum and its two nearest neighbours, the subpixel estimate can be computed by:

$$d_{subpixel} = d + \frac{C_{d-1} - C_{d+1}}{2(C_{d-1} - 2C_d + C_{d+1})} \tag{3}$$

Although this formula is intended for the SSD measure, many SAD based approaches [10], [20], [25] do also apply it for subpixel interpolation.

As indicated earlier, an extra step is necessary to detect and remove erroneous estimates on occlusions by WTA. The assumptions about scene geometry can also be exploited for this purpose.

One technique often used is the left-right consistency check. This check exploits the uniqueness constraint. It is assumed that the left to right and the right to left disparities are known. The algorithm checks that the disparity from left to right correspondence is the same value as when it is done conversely, searching the match from the right image to left image. Different value means inconsistency that could be caused by an occlusion and therefore, these matches are removed.

A drawback of this approach is that the minimum disparity has to be searched twice for every pixel. Stefano et al. [25] show that the uniqueness constraint can also be exploited for occlusion and error removal. During the search for left to right disparities, the cost value of a best match is stored for every newly matched right pixel. If a right pixel is encountered that has been matched before, the new and the old cost value of the match are compared. The old match is removed when the new cost value is smaller. This allows the algorithms to 'recover' from previously made bad matches.

III. Dense real time stereo framework

In order to investigate the suitability of real-time dense algorithms for the IV domain on an equal basis, we have implemented our own framework of real-time dense stereo algorithms. These algorithms use the latest SIMD SSE2 instruction set available on the Intel Pentium 4 or AMD Athlon 64 processors. The SSE2 instruction set uses 128-bit registers. For integer operations these can contain packed 8, 16, 32 or 64-bit data buffers.

Implementation based on SIMD is not straightforward because it places restrictions on what kind of operations can be used. An algorithm can only be sped up if the necessary operations can be performed in parallel. Conditional constructs must be avoided because they often lead to stalls of the code path prediction unit on the processor [9]. Furthermore, processing can only achieve its maximum speed if the values reside in the (Level 1) cache of the processor. Because of the much lower speed of main memory, loads and stores to and from it should be kept to a minimum by doing all the operations on the data in one go.

These considerations and the real-time requirements for ITS limit our choice of approaches from the previous section. For example, the current top ranking algorithms on the Middlebury stereo vision web site [28], based on the graph cuts technique, are not suitable due to their computational complexity and memory requirements. Our implementation therefore consists of several components, discussed in the previous section, that are more suitable for real-time operation. They can be combined to form different stereo algorithms.

The following sections explain how the components in our framework have been optimized.

A. SAD value computation

The SAD similarity measure can be computed efficiently by exploiting the fact that neighbouring windows overlap. For neighbouring windows with the same disparity, the overlapping pixels will contain equal absolute difference (AD) values. Therefore, a new SAD can be computed out of an old one by

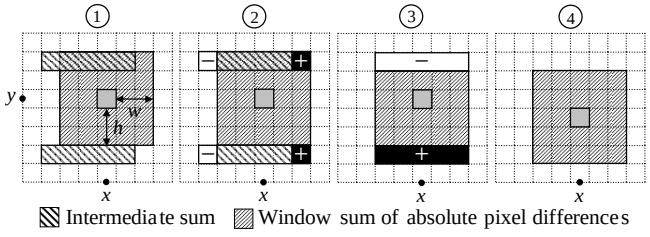


Fig. 3. Computation of a new SAD value out of the value of an old one using intermediate row sums.

subtracting the values, which are only parts of the old window, and adding the values, which are only parts of the new window.

Fig. 3 shows how a new SAD value of a window of size width $2w+1$ and height $2h+1$ centred at $(x, y+1)$ can be computed from the previous sum of a window at (x, y). This process utilizes the intermediate sums of AD values in rows of the same width as the matching window. Once such a sum is initialized at the beginning of a line, successive sums can be calculated recursively by first subtracting the most left AD value and adding the new AD value at the right-hand side. Calculated AD values for the additions are stored temporally in order to re-use them for the subtractions.

Two intermediate sums are applied in order to calculate the new SAD value of a window at $(x, y+1)$ from the old value of the window at (x, y). The first intermediate sum is used for the AD values that belong to the upper row of the old window. The second sum holds the AD values of the lower row of the window at the new position. By subtracting the upper row and adding the lower row, the new SAD value can be calculated. By repeating this process, a cascade function is created that only requires two pixel AD's, four subtractions and four additions in order to compute a new window sum.

The previously described steps can be executed for multiple disparity values simultaneously with SIMD type processing. For the computation of the AD values, SSE2 register $xmm0$ is filled with 16 copies of the left pixel value. The register $xmm1$ holds 16 pixels values from the right disparity interval. Only two saturated subtractions ($xmm0-xmm1$ and $xmm1-xmm0$) and the logical *or* of the results is needed to calculate the 16 AD's. The update of the intermediate sums and the SAD values themselves are carried out with regular SSE2 subtractions and additions.

B. Multiple windows

As explained earlier, the main deficiency of window based algorithms is their poor performance near object edges and on textureless regions. Multiple window approaches can improve the results.

Our implementation of the five window method by Hirschmüller et al. [10] uses a vertical and horizontal step. In the vertical compare step, newly computed SAD values are compared with the values of $2h$ processed image lines earlier. The results are temporally stored as the minimum and maximum values for the current 'centre' image line of h image lines earlier. In the horizontal step, the vertical values on either side ($x-w$ and $x+w$) of a point are compared. The maximum of the left hand side is compared with the minimum of the right hand side and vice versa. The two resulting minimum values are also the two smallest values of the four windows. On average, only three compares are needed to find the smallest values with this method.

C. Left to right minimum search

An important step in a stereo algorithm is searching for the disparities with the lowest SAD values. Fig. 1 shows the search space for a disparity interval of $0 \leq d \leq max$. For a point l on the left image line, the possible matches with points on the right image line are indicated with light grey in the search space. Finding the correct match by searching the match with the lowest SAD value is computationally expensive because the whole disparity interval has to be searched.

In [24], a method was shown for finding a minimum using multimedia extensions (MMX) instruction set that we have modified to work with SSE2. In this approach two types of SSE2 registers are used, one holding the range of disparity values and the other holding a copy of the associated SAD values. All values of two SAD registers can be compared in pairs using a single 'compare smaller than' instruction. Both the smaller SAD values and accompanying disparity values can be selected afterwards, using the resulting binary mask. This algorithm can be used to process a large disparity range recursively, after which only eight values remain to be searched in order to find a single minimum.

D. Right to left minimum search

Disparity search can also be performed from right to left. Dark gray is used in Fig. 1 to indicate possible matches on the left line for a right point r.

Right to left search does not require recomputation of the SAD values, the earlier computed left to right values can be re-used. However, because of the arrangement of the values in memory, a different algorithm is needed in order to apply SIMD techniques. By evaluating the SAD value of multiple right points, instead of one point, this problem can be solved. Similar to the approach chosen for left to right search, we start out with an array holding the SAD values and another holding the disparity values. However, now the SAD values are compared with the SAD values in the next array. In order to align the next SAD values, the values are shifted down one position. By repeating these steps, all the SAD values in the disparity interval of a right pixel are compared sequentially. Eventually, the lowest SAD value and its disparity are the ones discarded from the arrays during the shift step.

E. Search optimizing technique

Almost all real-time SIMD based algorithms known from literature use the simple WTA technique for disparity search. We have also implemented a search optimizing method based on dynamic programming. It is based on a technique, proposed by Kraft and Jonker [15], that uses two stages; a cost value propagation stage and a collection stage for retrieving the best disparity estimates.

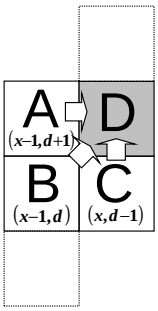


Fig. 4. Possible predecessor points in cost propagation step. Predecessor point B has the same disparity as point D, while the points A and C are respectively an occlusion or part of a discontinuity.

In the cost propagation stage, which runs from left to right through the disparity space, each point receives a cost value from a preceding point. The number of preceding points is limited to three possibilities, that are indicated for a single point in Fig. 4. Each preceding point has a different weight cost added to its accumulated cost value. Constant weights W_A and W_C are used for the points A and C which are occlusion and discontinuity points, respectively. The actual SAD value of point B is used as its weight because its disparity is equal to that of the current point. The predecessor with the lowest total cost value is selected as predecessor of the current point. Each point also stores the location of its predecessor. These references link up to form a path trough the disparity search space. At the end of the propagation stage, the best path is simply selected by searching for the lowest accumulated cost value. The best disparities are found in the collection stage by backtracking this path.

F. Subpixel interpolation

The algorithms of our framework estimate disparity with subpixel accuracy. Until now, we have described integer based operations with SSE2. The (older) SSE instruction set also contains SIMD operations for operations on 128 bit registers with packed 32 bit floats. We use SSE instructions for computing Eq. 3, which enables subpixel estimation for four points simultaneously.

G. Occlusion removal

We have implemented two types of algorithm components for removing pixels in occlusions and erroneous matches. The first type is the left-right check [10], [20], [24]. The second type is the "recover" approach [25]. Detected occluded pixels are set to a predefined error code. This can be a value that is higher than the disparity maximum or simply zero.

Because of the conditional dependencies used in both approaches no straightforward application of SIMD is possible. Fortunately, each check only has to be executed one time for every pixel. The required overhead is insignificant when compared to the other steps.

H. Algorithms

Using the described components we have created seven different stereo algorithms:

1) SAD_L: SAD WTA only left to right search.
2) SAD_{Rec}: SAD WTA with recover approach.
3) SAD_{LR}: SAD WTA with left-right check.
4) $SAD_{MW5\ L}$: same as 1, but with multiple windows.
5) $SAD_{MW5\ Rec}$: same as 2, but with multiple windows.
6) $SAD_{MW5\ LR}$: same as 3, but with multiple windows.
7) SAD_{DP}: SAD with dynamic programming disparity search method.

IV. Experiments

The presented implementations have been tested and compared, together with four other publicly available algorithms. These are the SSD, DP and SO implementations, created by Scharstein and Szeliski (S&S) for their survey [22], that are available on the Internet [28]. Furthermore, an implementation of Birchfield and Tomasi's (B&T) [1] algorithm in the OpenCV library [27] is used.

The SSD algorithm is a WTA type algorithm like the majority of our implementations, however it uses the sum of squared differences for matching cost computation. Both DP and B&T use dynamic programming for searching the correct disparities. In contrast to the other algorithms, B&T does not use matching windows. Its measure is based on interpolating values between real pixels to achieve sampling invariance. The SO algorithm uses scanline optimization for improving the disparity estimate.

A. Stereo image test sequence

For evaluation purposes, several standard stereo image pairs with ground truth are available. A well know example are the stereo pairs of Tsukuba University. Unfortunately, the disparity range of these pairs is quite small (16 pixels) and the ground truth disparity is only given with 1 pixel accuracy.

Another well known test set was introduced by Scharstein and Szeliski in their survey [22]. It does provide wider baseline stereo images with subpixel accurate ground truth disparity. However, in order to keep the acquisition of ground truth disparity simple, the images only contain planar surfaces.

Because our goal is to investigate the suitability of dense stereo vision algorithms for application in the IV domain, test images are needed which show realistic traffic scenes. Unfortunately, none of the commonly used test sets resemble this type of data.

Instead, we use both real and synthetic data of traffic scenes to evaluate the algorithms. The real image sequences were recorded with a vehicle mounted stereo camera. These depict typical traffic scenes with obstacles such as pedestrians and other cars. Unfortunately, the real images do not come with a ground truth disparity. It is possible to obtain a ground truth disparity image with techniques such as active lighting [23]. However, this is not a practical approach for outdoor scenes and moving stereo camera rigs. Our analysis of the results with real stereo images is therefore limited to qualitative comparisons.

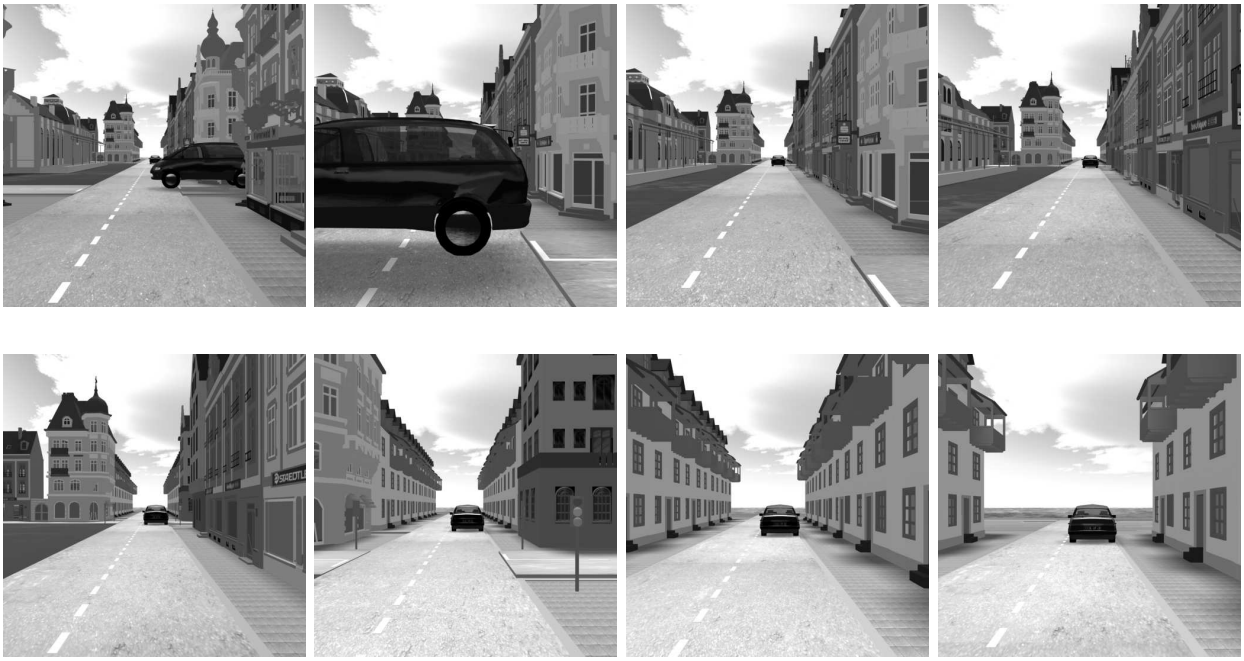
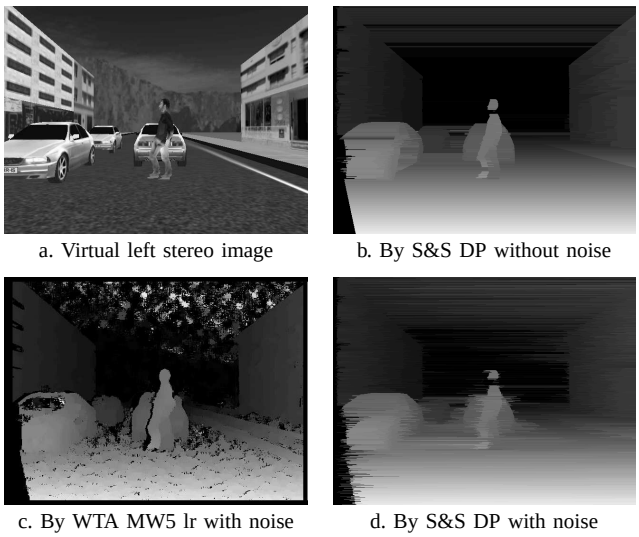


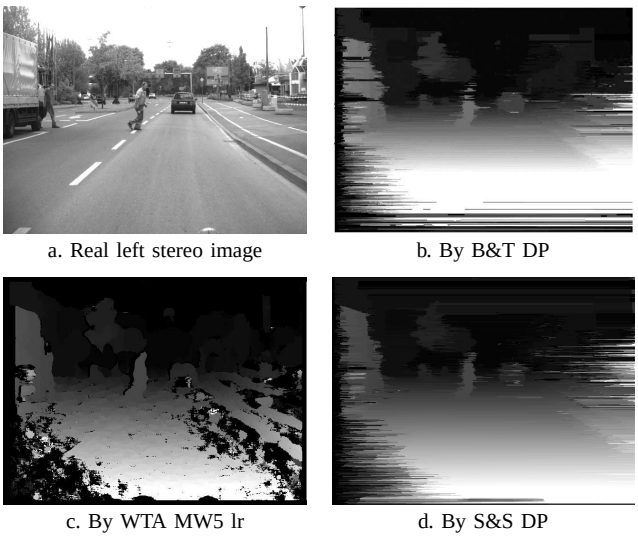
Fig. 5. Virtual left stereo images from the sequence used for our experiments (frame nr. 20, 60, 100, 140, 160, 220, 260 and 300).



a. Virtual left stereo image b. By S&S DP without noise

c. By WTA MW5 lr with noise d. By S&S DP with noise

Fig. 6. Results with simulated stereo images.



a. Real left stereo image b. By B&T DP

c. By WTA MW5 lr d. By S&S DP

Fig. 7. Results with real stereo images.

A stereo image pair and its ground truth disparity also can be generated synthetically from a 3D computer model. The MARS/PRESCAN software [19], [21] is a framework for simulation of different vehicle mounted sensors; such as radar, laser rangefinder or camera based systems such as stereo vision. With this simulator, a sequence was created of a virtual vehicle equipped with a stereo camera driving through traffic scenes. The used city-like scenery is complex and other moving vehicles are present.

As ground truth, the simulator provides the range images for each of the stereo images in the sequence. For our experiments, these are converted to disparity images.

The MARS/PRESCAN synthetic stereo images and ground truth data used in our experiments is publicly available for download from the Internet [29]. Some of the 326 images of this sequence, which have a resolution of 512 by 512 pixels and disparity range of 48 pixels, are shown in Fig. 5.

B. Adding real image influences to synthetic images

We first looked at the qualitative similarities between the output disparity images generated with the simulated images and real images. The results of the simulated data were very good for all algorithms, see for example the DP result in the noiseless case (fig. 6b). This is caused by the fact that no image noise was added to the simulated images.

However, in the output for real stereo images we could more

clearly see errors. An example of a real stereo image of a vehicle mounted camera is shown in Fig. 7a. The output of dynamic programming approaches such as B&T and DP now shows a lot of "streaking" errors near object boundaries. WTA approaches such as in our framework and SSD, on the other hand, generate mainly errors on areas with insufficient texture. See for example the road surface in Fig. 7c.

In order to approach the conditions of the real images more closely we added a number of stereo camera related perturbations. These perturbations are mainly due to the optics, sensor signal to noise ratio and the calibration of the stereo rig itself.

Light passing through the edges of a camera lens will hit the image sensor under a different angle than the light that passes trough the middle. The light rays at the image edges are scatted over a larger sensor area than those at the middle. This effect is known as 'vignetting' and causes pixels far away from the image centre to be darkened. Fig. 8, shows the pixel weights that are multiplied with the original image pixels to add this effect. The weights where obtained with the $\cos^4$-law, which is the technique for calculating vignetting illumination fall-off [16].

In real cameras two forms of image noise are caused by the image sensor. The first type is called fixed pattern noise and is caused by physical differences between the light sensitive elements on the sensor. However, in almost all modern cameras the influence of this type of noise is negligible because non-uniformity correction is used. The other type is called temporal noise and is due to the sensors signal to the noise ratio. This type of noise was introduced to the synthetic images by adding white Gaussian noise with zero mean and a variance of 1 intensity level.

Because dense disparity estimation relies on steps for removing lens distortion and rectifying the stereo images, the stereo camera calibration itself is also a potential source of perturbations. In order to add this influence to the "perfectly" rectified synthetic images, we performed the undistortion and rectification steps on them with parameters based on typical residual errors of stereo calibration. Fig. 8 shows the magnitudes of the distortion effect between the original and the distorted synthetic images.

Outputs of the algorithms with these corrupted images now involve similar artifacts as observed in the real images, e.g. see $SAD_{MW5\,LR}$ and DP results in Fig. 6c & d and Fig. 7c & d.

C. Error measures

Several approaches to quantitative evaluation of stereo algorithms exists. One of the most simplest error measures is the averaged absolute mean error:

$$E_{abs} = \frac{1}{n}\sum_{i=1}^{n}|g_i - d_i| \tag{4}$$

Where g is the estimate by the evaluated algorithm and d is the ground truth. Larger errors can be accentuated by using the squared error instead of the absolute error:

$$E_{sq} = \frac{1}{n}\sum_{i=1}^{n}(g_i - d_i)^2 \tag{5}$$

The drawback of both error measures is that they do not distinguish well between disparity estimates with a lot of small errors and disparity estimates with only a few large errors.

Another error measure is the bad pixel percentage. It uses a threshold δ to set a maximum allowed absolute error. The absolute differences with the ground truth larger than this value are counted as bad pixels:

$$B = 100\%\frac{1}{n}\sum_{i=1}^{n}(|g_i - d_i| > \delta) \tag{6}$$

In contrast to the previous two error measures, small errors are ignored while other errors are counted regardless of their magnitude.

It is difficult to relate the error measures presented so far to problems encountered when dense stereo vision is used as a sensor on a intelligent vehicle. Issues that are of importance here are the ability to detect objects such as obstacles and determine their range accurately.

Classifying a group of pixels as an obstacle requires that they are distinguishable from other background items such as the road surface, buildings or the sky. We therefore use the ground truth disparity from the MARS/PRESCAN simulator to divide pixels in each stereo image into four classes. These are foreground and background obstacles, road surface and sky. The two cars are the foreground obstacles in our sequence while the buildings are background obstacles. Both pixels on the road and the curb belong to the road surface class. Pixels that have zero ground truth disparity are classified as being sky. Examples of the three pixel classes are shown in Fig. 9.

A range can only be given for pixels that lie on surfaces, such as the foreground, background and road pixels. For these three pixel classes we define the estimation density D of a disparity image as:

$$D = 100\%\frac{m}{n} \quad \text{with} \quad m = \sum_{i=1}^{n} g_i > 0 \tag{7}$$

This measures which percentage of the foreground, background or road surface pixels have been assigned a disparity estimate.

For stereo, range has an inverse relationship with disparity; small disparities correspond to large distances while large disparities correspond to small distances. Thus, an error in disparity for a far away point corresponds to a larger error in range than the same small error in disparity for a nearby point. The error measures of Eq. 4, 5 and 6 do not take this into account. In the Mean Relative Error measure the absolute error is divided by the ground truth disparity for each pixel. Therefore, this measure does actually relate to the expected error in range estimation.

$$E_{rel} = \frac{1}{m}\sum_{i=1}^{m}\left((g_i > 0)\,\frac{|g_i - d_i|}{d_i}\right) \tag{8}$$

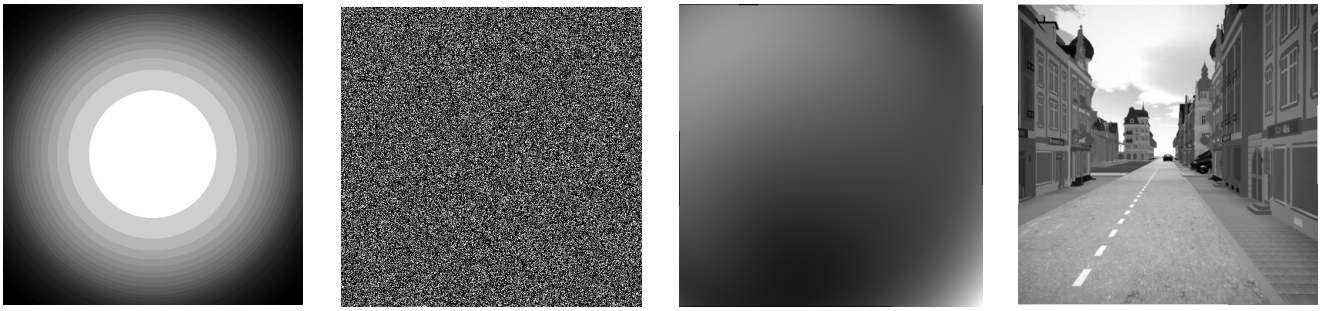


Fig. 8. Influences on the quality of a stereo image pair.

TABLE I
RESULTS OF THE DISPARITY ESTIMATORS FOR DIFFERENT TYPES OF SURFACE PIXELS.

	Foreground			Background			Road			All Surfaces		
	$R\%$	$D\%$	E_{rel}	$R\%$	$D\%$	E_{rel}	$R\%$	$D\%$	E_{rel}	$R\%$	$D\%$	E_{rel}
SAD_L	0	98.8	0.233	0	99.1	0.662	0	99.7	0.142	0	99.2	0.346
SAD_{Rec}	25.4	72.7	0.182	18.3	80.9	0.460	11.5	88.2	0.109	18.4	80.6	0.250
SAD_{LR}	26.8	73.0	0.156	17.7	82.0	0.386	9.0	90.9	0.100	17.8	82.0	0.214
SAD_{MW5L}	0	98.9	0.213	0	99.2	0.535	0	99.8	0.120	0	99.3	0.289
SAD_{MW5Rec}	20.9	75.8	0.172	14.4	84.3	0.397	11.0	88.1	0.098	15.5	82.7	0.222
SAD_{MW5LR}	24.5	75.3	0.149	12.4	87.4	0.340	7.6	92.4	0.088	14.8	85.0	0.192
SAD_{DP}	6.3	82.6	0.188	5.8	90.0	0.361	1.9	94.8	0.076	4.7	89.1	0.208
S&S SSD	0	98.2	0.239	0	97.2	0.574	0	98.8	0.159	0	98.1	0.324
S&S SO	0	93.9	0.153	0	97.5	0.457	0	95.4	0.126	0	95.6	0.246
S&S DP	0	92.8	0.188	0	99.2	0.335	0	95.6	0.087	0	95.9	0.203
B&T DP	0	96.3	0.233	0	99.6	0.333	0	99.2	0.110	0	98.4	0.226

For ITS applications it might be perceived that only the performance measures for the foreground obstacles are important. However, obstacle detection itself involves finding the correct foreground pixels among the other pixels classes. Since the road surface is more easily distinguishable than the (smaller) foreground objects it is actually searched first in some approaches. The performance measures for the other classes are therefore significant because bad estimates here can cause false positives.

D. Algorithms of the framework

The SIMD based algorithms of our framework and others from the public domain were tested with our sequence. The window based algorithms all used the same square window size of 9 by 9 pixels for cost computation. The weights for SAD_{DP} were set so that $W_A = 34000$ and $W_C = 1000$. The tests were conducted on an Intel Pentium 4 3.2 GHz PC with 1.0 GB RAM.

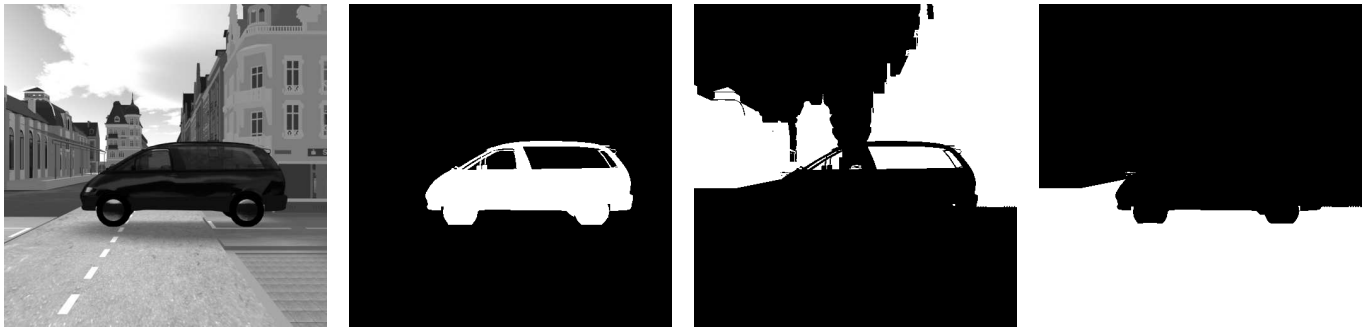
The results for the full resolution frames 1 until 90 are shown in Table I. For each of the surface pixel classes the rejection percentage $R\%$, estimate density $D\%$ and averaged relative mean error E_{rel} are shown. The fourth column shows the overall results averaged over all the surface pixels.

We first studied the overall performance of the different algorithms from our framework. The algorithms that do not reject pixels show the highest error rates, see Table I. These are mainly caused by erroneous estimations in difficult areas such as occlusions and textureless regions. The algorithms with post-processing have lower error rates because they manage to reject many of these pixels. Comparing the percentages of pixels rejected by the recovery and the left-right approach, it is clear that the left-right approach rejects more pixels. Our DP approach rejects the least amount of pixels.

Considering the timing results for both half and full resolution frames in Table II, it is clear that the improvement by the post-processing steps comes at the price of a higher computational cost. The left-right consistency check is more expensive than the recovery approach. If the processing times of SAD_{Rec} and SAD_{LR} algorithms are compared to the time needed for SAD_L, the recovery approach shows a 10% increase, while the left-right check shows a 20% increase. The DP approach has the highest computational cost of our optimized algorithms. This is due to the fact that references have to be stored for back propagation phase, which increases the number of expensive memory operations.

It should be noted that the achieved run times only give an outlook on future performance because they have been achieved on general purpose computer hardware. For ITS it is much more likely that more dedicated and low-power embedded SIMD hardware will be used.

As follows from Table I, the multiple window approach does improve results. It also decreases the number of rejected pixels. Of all our algorithms the $SAD_{MW5\ LR}$ algorithm has the lowest mean relative error for all surfaces pixels. From the timing results however, it is clear that the multiple window approach is expensive compared to the single window approach. The multiple window approach increases computation by about 50%.



Original Image | Foreground pixels | Background pixels | Road surface pixels

Fig. 9. The different pixel classes used for evaluation.

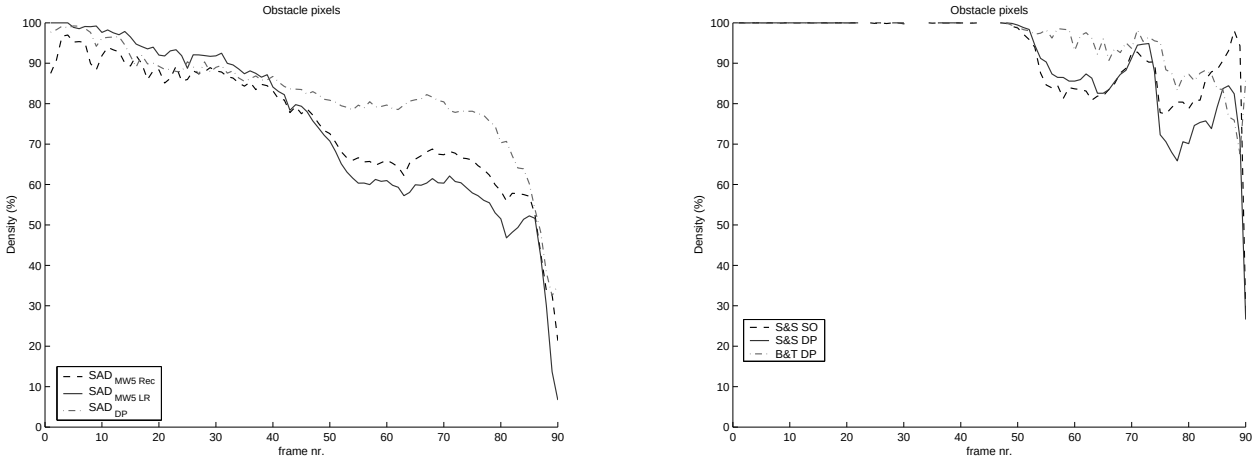


Fig. 10. The estimation densities of six algorithms on foreground pixels.

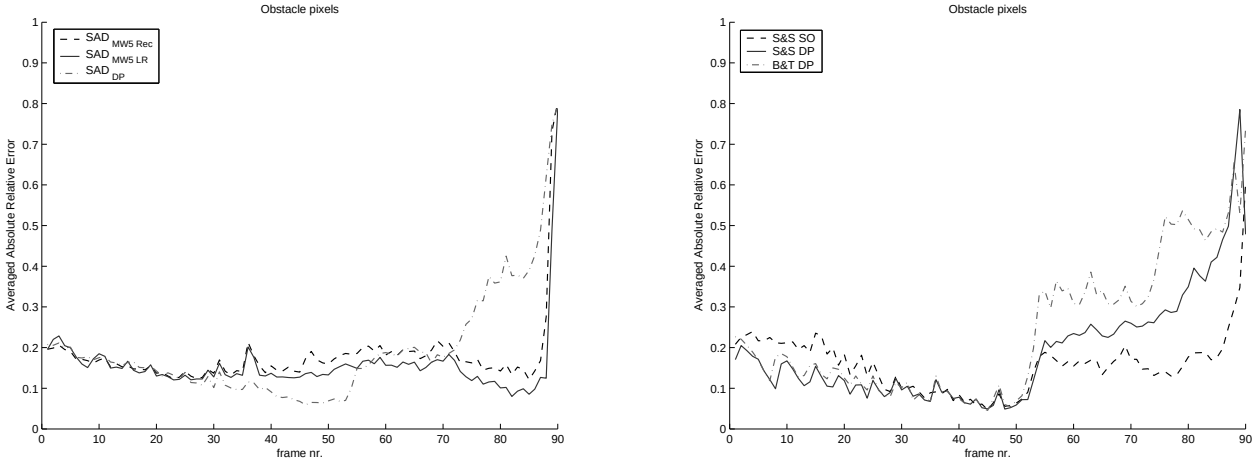


Fig. 11. The mean relative error of six algorithms on foreground pixels.

E. Performance for different surface pixel types

The previous section provided an insight how well our various algorithms compare to each other and what computational cost they incur. In this section we investigate the performance figures for separate pixel classes. We have also included the results of other, publicly available algorithms in this analysis.

The relative mean error for all surface pixels in Table I shows that the $SAD_{MW5\ LR}$ has the lowest overall error rate, followed by S&S DP as second best. However, this is not the case for all individual surface pixel classes.

For the background and road pixel classes it can be seen that

TABLE II
MEAN RUNTIME FOR THE TESTED ALGORITHMS. THE ALGORITHMS ARE ORDERED BY RUNTIME.

	Image dimensions: (width, height, disparity)			
	(512,512,48)		(256,256,24)	
Algorithm	Runtime	Fps	Runtime	Fps
SAD_L	0.0879 s	11.3827	0.0189 s	52.9393
SAD_{Rec}	0.0976 s	10.2432	0.0202 s	49.5065
SAD_{LR}	0.1096 s	9.1240	0.0234 s	42.7709
$SAD_{MW5\ L}$	0.1665 s	6.0061	0.0322 s	31.0802
$SAD_{MW5\ Rec}$	0.1698 s	5.8905	0.0342 s	29.2219
$SAD_{MW5\ LR}$	0.1775 s	5.6350	0.0358 s	27.9397
SAD_{DP}	0.2456 s	4.0721	0.0466 s	21.4573
B&T DP	0.4594 s	2.1768	0.0742 s	13.4711
S&S SSD	5.3012 s	0.1886	0.7123 s	1.4039
S&S DP	6.5724 s	0.1522	0.8645 s	1.1567
S&S SO	10.5968 s	0.0944	1.0484 s	0.9538

approaches based on dynamic programming such as SAD_{DP} and S&S DP perform better than many of the WTA based algorithms. This is due to the fact that the road and background classes do not contain many disparity discontinuities, their disparity profiles adhere to the smoothness constraint. Because disparity discontinuities do occur at the edges of the foreground objects, the DP based approaches have more problems with these type of pixels. The WTA based algorithms that use the left-right check to remove errors are more successful on foreground pixels.

Regarding the processing time of various algorithms, Table II reveals the benefit of the pursued SIMD SSE2 implementation. Our optimized algorithms are much faster than the SIMD non-optimized algorithms S&S DP, S&S SO, S&S SSD and B&T DP. The B&T DP is much faster than the S&S DP algorithm, because although both use dynamic programming, it is based on a pixel-based sampling invariant similarity measure, rather than the window-based measure used by S&S DP.

F. Analysis of performance variations

Until now, we have only considered averaged results for a part of the sequence. However, the performance of the tested algorithms clearly varies from frame to frame. This can seen in plots of the estimation density percentage and relative mean error per stereo image pair.

In Fig. 10 two plots are shown of estimation density percentages ($D\%$) on foreground pixels for frame 0 until 90. The plot on left side shows the results of the $WTA_{MW5\ REC}$, $WTA_{MW5\ LR}$ and WTA_{DP} algorithms from our framework while the plot of the right side shows the results of S&S SO, S&S DP and B&T DP. Fig. 11 shows two plots for the same interval. This time, the relative mean error (E_{rel}) is shown for the six algorithms.

During this part of the sequence, a foreground object (a car) passes a road crossing. The vehicle drives into in the camera field of view from a side street. It is completely visible in frame 25. After frame 48 it starts to leave the cameras field of view. It is completely out of the field of view after frame 90.

If the different density percentage plots are compared it is clear that the algorithms from our frame work show a gradual decrease. This is due to the fact that the surface of the car contains little texture. In the beginning, when the car is far away, features such as edges, provide enough distinct points for the stereo matching. When the car is nearby more pixels in textureless areas are visible. This causes the decrease in estimation density.

Because WTA_{DP}, S&S SO, S&S DP and B&T DP use search optimization techniques they are less effected by textureless regions. The plot of the latter three algorithms does show lower densities percentages after frame 48, which due to 'streaking errors'. Because the disparity jump from background to foreground disparity is not visible anymore after frame 48, the dynamic programming technique wrongfully assigns zero disparities to some of the foreground pixels.

The effects of streaking errors by algorithms with search optimization is more evident in the plots of Fig. 11 where the relative mean error is shown. The algorithms $WTA_{MW5\ REC}$, $WTA_{MW5\ LR}$ that do not use search optimization techniques show a fairly consistent error of about 0.2, while the other algorithms, that do use DP or SO show a sharp increase in error after frame 48.

G. Ground plane estimation experiment

In the previous sections we have studied the quantitative results of the different disparity estimators. We learned that the algorithms which use global search techniques are more effected by the scene complexity. In this section we will show some of the consequences for a typical application of stereo vision in intelligent vehicles: ground plane estimation.

Ground plane estimation is required to distinguish obstacles such as other cars and pedestrians in the disparity image from the road surface. A robust and real-time method for doing this was developed by Labayrade et al. [18].

It is based on the assumption that for scanlines where the road surface is visible, the dominant disparity value is that of road surface pixels. Their method first converts the normal disparity image to a 'V-disparity' image. Each scanline of such an image is the histogram of disparity values of a scanline from the original disparity image. The road surface profile can be extracted from the 'V-disparity' image by finding dominant line features. The original method of Labayrade et al. [18] uses the Hough transform to find line features and approximates the road vertical curvature with a piecewise linear curve.

For our experiment, we use a simplified version of the Labayrade et al. [18] approach. To test an estimated disparity image it is first converted to a V-disparity image. Because the road surface of the synthetic images is flat, only a single dominant line feature is searched with the Hough transform. This line is then compared to the line found by the same method in the ground truth disparity image. The difference in angle between the two lines shows how ground plane estimation is affected by the quality of the disparity image.

In Fig. 12 the differences in ground plane angle is shown for all images from test sequence. Each column shows the errors colour coded in degrees for one of the tested algorithms.

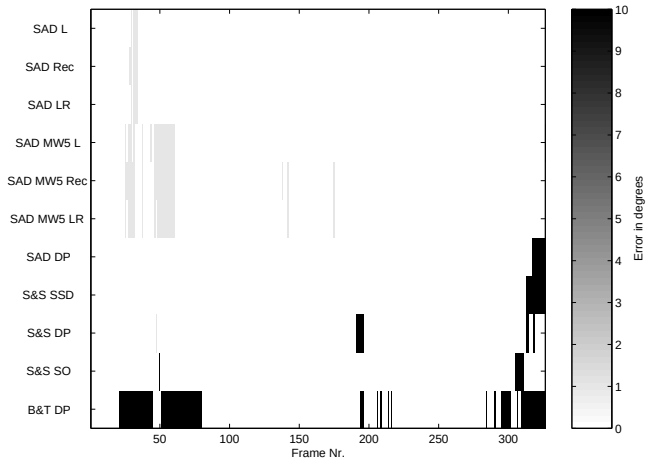


Fig. 12. Error in ground plane angle estimation based on V-disparity.

These results show that error made by algorithms, which use optimization steps, affect the ground plane estimation more than the more simpler approaches. The SAD_{DP}, S&S DP, S&S SO and B&T DP show severe errors in ground plane angle estimation in some the frames. Both the $SAD_{MW5\,L}$ and S&S SSD also show severe errors because they do not use rejection steps. It is also interesting to see that the multiple window approach, when compared to the single window approach, actually increases the error slightly for some parts of the sequence. This is due to the fact that V-disparity assumes that a majority of disparities belong to the road surface. It can therefore become biased in situations where this is not true. The multiple window approach gives more estimates for the textureless surface of the passing car than the single window approach. Surprisingly, this influences the ground plane estimation when the car is nearby and fills a large part of the image.

V. Conclusion

Application of dense stereo vision in intelligent vehicles requires accurate and robust disparity estimation algorithms that can run in real-time on small and power efficient computing hardware. Dense stereo vision algorithms which are based on Single Instruction Multiple Data processing are interesting because this type of instruction level parallelism is currently available on various normal, low power embedded and dedicated computing systems.

We implemented several real-time algorithms to investigate how well different approaches to dense stereo vision can benefit from SIMD optimization and compare them on an equal basis. They are all based on a single framework that provides components for performing SAD cost computation, multiple window selection, disparity search based on WTA or dynamic programming and post-processing for occlusion or error rejection. We came up with fast SIMD optimized versions for most of these components using the SSE2 instruction set.

In order to test the algorithms under the varying conditions, that can be encountered in city-like traffic, we used stereo images from a vehicle sensor simulator. In contrast to the commonly used single pair test images, our sequence contains hundreds of images with varying geometric and image properties. Effects that degrade image quality in real stereo cameras, such as vignetting, sensor noise and imperfect image undistortion and rectification were also added to enhance realism.

The first set of tests show that the post-processing steps can help reduce error percentages at the cost of a sparser disparity map. More advanced algorithms with multiple window technique or dynamic programming improve the disparity estimates on difficult image areas, albeit at higher computational cost. Nevertheless, the slowest of our algorithms still achieves real-time processing speeds.

We also studied the performance of our algorithms together with a set of four other publicly available stereo algorithms for different types of foreground disparity pixel classes. The results confirm that non-greedy matching algorithms (i.e. scan line optimization, dynamic programming) perform better on surfaces with low geometrical complexity such as the road surface. However, the more simpler WTA techniques combined with error detection techniques can outperform these algorithms on more challenging surfaces such as nearby obstacles. This is caused by the fact that search optimization algorithms can miss important disparity jumps at the edges of nearby objects and therefore assume the wrong disparities in later search or optimization stages.

The consequences of these type of errors are demonstrated with an road surface inclination estimation experiment. The results show that ground plane estimation based on disparity estimates of the algorithms which do use optimization techniques are less reliable.

Our research has shown that simple WTA techniques for dense stereo, combined with robust error rejection schemes such as the left-right check, are more suitable for intelligent vehicle applications. Because they do not use search optimization techniques their processing speed is higher. Our investigation has also revealed that the basic search optimization techniques such as dynamic programming can cause errors which interfere with subsequent steps needed for intelligent vehicle related sensing tasks.

The results suggest that future improvement of dense stereo algorithms can be achieved by applying search optimization techniques only to the parts of the stereo images that can benefit from them, as discussed above. Preprocessing steps to detect textureless and edgeless regions could be exploited to detect which pixels are suitable.

Acknowledgements

The authors would line to thank M. G. van Elk from the Imaging Systems (IST) Signal Processing Department at TNO Science and Industry for generating the sequence of synthetic stereo images and ground truth range data used in our experiments. The authors also thank Harris Sunyoto for his assistance in performing the early-stage experiments.

REFERENCES

[1] S. Birchfield, C. Tomasi,"Depth Discontinuities by Pixel-to-Pixel Stereo", *International Conference on Computer Vision*, pp. 1073-1080, Bombay, India, 1998.

[2] A.F. Bobick, S.S.Intille, "Large occlusion stereo", *International Journal of Computer Vision*, Vol. 33, Issue 3, pp. 181-200, 1999.

[3] M.Z. Brown, D. Burschka, G.D. Hager, "Advances in Computational Stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8), August 2003.

[4] R. Bunschoten, B. Kröse, "Range Estimation from a Pair of Omnidirectional Images", *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2000.

[5] H.Y. Deng, Q. Yang, X. Lin, X. Tang, "A Symmetric Patch-Based Correspondence Model for Occlusion Handling", *To appear in Proceedings IEEE International Conference on Computer Vision*, Beijing, China, October 15-21, 2005.

[6] U. Franke, S. Heinrich, "Fast Obstacle Detection for Urban Traffic Situations", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 3, September 2002.

[7] A. Fusiello, E. Trucco, A. Verri, "Recification with unconstraint stereo gemometry", *Proceedings of the Eighth British Machine Vision Conference*, 1997.

[8] A. Fusiello, V. Roberto, and E. Trucco."Symmetric stereo with multiple windowing". *International Journal of Pattern Recognition and Artificial Intelligence*, 14(8) pp. 1053-1066, December 2000.

[9] R. Gerber, *The Software Optimization Cookbook*, Intel Press, Intel Cooperation, Hillsboro, OR; 2002.

[10] H. Hirschmüller, P.R Innocent, J.M. Garibaldi, "Real-Time Correlation-Based Stereo Vision with Reduced Border Errors", *International Journal of Computer Vision*, Vol. 47(1/2/3), pp. 229-246, 2002.

[11] L. Hong, G. Chen, "Segment-based stereo matching using graph cuts", *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Vol. I, pp. 74-81, Washington, DC, USA, 27 June - 2 July, 2004.

[12] T. Kanade, M. Okutomi, "A stereo matching algorithm with an adative window: theory and experiment", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Issue 9, pp 920-932, September 1994.

[13] T. Kanade, H. Kano, S. Kimura, "Development of a VideoRate Stereo Machine". *International Robotics and System Conferences*, pp. 95-100, August 1995.

[14] J.C. Kim, K.M. Lee, B.T. Choi, S.U. Lee, "A Dense Stereo Matching Using Two-Pass Dynamic Programming with Generalized Ground Control Points", *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. II, pp. 1075-1082, San Diego, CA, USA, June 20-25, 2005.

[15] G. Kraft, P.P. Jonker, "Real-Time Stereo with Dense Output by a SIMD-Computed Dynamic Programming Algorithm", *International Conference on Parallel and Distributed Processing Techniques and Applications*, Vol. III, pp. 1031-1036 , Las Vegas, Nevada, USA, June 24-27, 2002.

[16] C. Kolb, D. Mitchell, P. Hanrahan, "A Realistic Camera Model for Computer Graphics", Computer Graphics (Proceedings of SIGGRAPH '95), ACM SIGGRAPH, pp. 317-324, 1995.

[17] V. Kolmogorov, R. Zabih, "Computing Visual Correspondence with Occlusions using Graph Cuts", *Proceedings IEEE International Conference on Computer Vision (ICCV)*, Vancouver, Canada, July 9-12, 2001.

[18] R. D. Labayrade, J.P. Tarel,"Real Time Obstacle Detection on Non Flat Road Geometry through 'V-Disparity' Representation", Proceedings of IEEE Intelligent Vehicle Symposium, Versailles, France, 18-20 June 2002.

[19] K. Labibes, Z. Papp, A.C.H. Thean, P.P.M. Lemmen, M. Dorrepaal, F.J.W. Leneman, "An integrated design and validation environment for intelligent vehicle safety systems (IVSS)". *10th World Congress and exhibition on ITS*, Proceedings on CD-ROM, Madrid, Spain, 16-20 Nov 2003.

[20] K. Mühlmann, D. Maier, J. Hesser, R. Männer, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation", *International Journal of Computer Vision*, Vol. 47, Nr. 1-3, pp. 79-88, April - June 2002.

[21] Z. Papp, K. Labibes, A.H.C. Thean, M.G. van Elk, "Multi-Agent Based HIL Simulator with High Fidelity Virtual Sensors", *IEEE Intelligent Vehicles Symposium*, pp. 213-218, Columbus (OH), June 9-11, 2003.

[22] D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", *International Journal of Computer Vision*, Vol. 47 (April-June), pp. 7-42, 2002.

[23] D. Scharstein, R. Szeliski, "High-accuracy stereo depth maps using structured light". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 195-202, Madison, WI, June 2003.

[24] L. Di Stefano, S. Mattoccia, "Fast Stereo Matching for the VIDET System using a General Purpose Processor with Multimedia Extensions", *Fifth IEEE International Workshop on Computer Architectures for Machine Perception*, pp. 356-362, Padova, Italy, September 11 - 13, 2000.

[25] L. Di Stefano, M. Marchionni, S. Mattoccia, G. Neri, "A Fast Area-Based Stereo Matching Algorithm", *15th IAPR/CIPRS International Conference on Vision Interface*, Calgary, Canada, May 27-29, 2002.

[26] M. Ziegler, *Region-based analysis and coding of stereoscopic video*, PhD Thesis, Technische Universiteit Delft, The Netherlands, 1997.

[27] The Open Source Computer Vision Library, available online: `http://www.intel.com/technology/computing/opencv`

[28] Middlebury College Stereo Vision Research Page, available online: `http://www.middlebury.edu/stereo`

[29] Stereo Image Data for Algorithm Evaluation, available online: `http://stereodatasets.wvandermark.com/`

Wannes van der Mark was born in Leiderdorp, the Netherlands, on 22 June, 1975. He obtained the M.Sc. Degree in Artificial Intelligence from the University of Amsterdam in 2000. He currently works as a PhD student at both TNO Defence, Security and Safety in The Hague and the University of Amsterdam. His current research interest is in stereovision for autonomous vehicle guidance in unstructured terrain.

Dariu M. Gavrila obtained the M.Sc. Degree in Computer Science from the Free University in Amsterdam in 1990. He received the Ph.D. Degree in Computer Science from the University of Maryland at College Park in 1996. He was a Visiting Researcher at the MIT Media Laboratory in 1996. Since 1997 he is a Research Scientist at DaimlerChrysler Research in Ulm, Germany. In 2003, he was appointed Professor at the University of Amsterdam, chairing the area of Intelligent Perception Systems (part-time).

Mr. Gavrila's long-term research interests involve vision systems for detecting human presence and activity with applications in intelligent vehicles and surveillance, in which he has numerous publications. His personal website is "`www.gavrila.net`".