Stereo Processing by Semi-Global Matching and Mutual Information

Heiko Hirschmüller

***Abstract*— This paper describes the Semi-Global Matching (SGM) stereo method. It uses a pixelwise, Mutual Information based matching cost for compensating radiometric differences of input images. Pixelwise matching is supported by a smoothness constraint that is usually expressed as a global cost function. SGM performs a fast approximation by pathwise optimizations from all directions. The discussion also addresses occlusion detection, sub-pixel refinement and multi-baseline matching. Additionally, post-processing steps for removing outliers, recovering from specific problems of structured environments and the interpolation of gaps are presented. Finally, strategies for processing almost arbitrarily large images and fusion of disparity images using orthographic projection are proposed.**

A comparison on standard stereo images shows that SGM is among the currently top-ranked algorithms and is best, if sub-pixel accuracy is considered. The complexity is linear to the number of pixels and disparity range, which results in a runtime of just 1-2s on typical test images. An in depth evaluation of the Mutual Information based matching cost demonstrates a tolerance against a wide range of radiometric transformations. Finally, examples of reconstructions from huge aerial frame and pushbroom images demonstrate that the presented ideas are working well on practical problems.

***Index Terms*— stereo, mutual information, global optimization, multi-baseline**

I. Introduction

ACCURATE, dense stereo matching is an important requirement for many applications, like 3D reconstruction. Most difficult are often occlusions, object boundaries and fine structures, which can appear blurred. Matching is also challenging due to low or repetitive textures, which are typical for structured environments. Additional practical problems originate from recording and illumination differences. Furthermore, fast calculations are often required, either because of real-time applications or because of large images or many images that have to be processed efficiently.

A comparison of current stereo algorithms is given on the the Middlebury Stereo Pages[1]. It is based on the taxonomy of Scharstein and Szeliski [1]. They distinguish between four steps that most stereo methods perform, i.e. matching cost computation, cost aggregation, disparity computation/optimization and disparity refinement. Matching cost computation is very often based on the absolute, squared or sampling insensitive difference [2] of intensities or colors. Since these costs are sensitive to radiometric differences, costs based on image gradients are also used [3]. Mutual Information has been introduced in computer vision [4] for handling complex radiometric relationships between images. It has been adapted for stereo matching [5], [6] and approximated for faster computation [7].

Cost aggregation connects the matching costs within a certain neighborhood. Often, costs are simply summed over a fixed sized window at constant disparity [3], [5], [8], [9]. Some methods additionally weight each pixel within the window according to color similarity and proximity to the center pixel [10], [11]. Another possibility is to select the neighborhood according to segments of constant intensity or color [7], [12].

Disparity computation is done for local algorithms by selecting the disparity with the lowest matching cost [5], [8], [10], i.e. winner takes all. Global algorithms typically skip the cost aggregation step and define a global energy function that includes a data term and a smoothness term. The former sums pixelwise matching costs, while the latter supports piecewise smooth disparity selection. Some methods use more terms for penalizing occlusions [9], [13], alternatively treating visibility [11], [12], [14], enforcing a left/right or symmetric consistency between images [7], [11], [12], [14] or weight the smoothness term according to segmentation information [14]. The strategies for finding the minimum of the global energy function differ. Dynamic programming (DP) approaches [2], [15] perform the optimization in 1D for each scanline individually, which commonly leads to streaking effects. This is avoided by tree based DP approaches [12], [16]. A two dimensional optimization is reached by Graph Cuts [13] or Belief Propagation [3], [11], [14]. Layered approaches [3], [9], [11] perform image segmentation and model planes in disparity space, which are iteratively optimized.

Disparity refinement is often done for removing peaks [17], checking the consistency [8], [11], [12], interpolating gaps [17] or increasing the accuracy by sub-pixel interpolation [1], [8].

Almost all of the currently top-ranked algorithms [2], [3], [7], [9], [11]–[15] on the Tsukuba, Venus, Teddy and Cones data set [18] optimize a global energy function. The complexity of most top-ranked algorithms is usually high and can depend on the scene complexity [9]. Consequently, most of these methods have runtimes of more than 20 seconds [3], [12] to more than a minute [9]–[11], [13], [14] on the test images.

This paper describes the Semi-Global Matching (SGM) method [19], [20], which calculates the matching cost hierarchically by Mutual Information (Section II-A). Cost aggregation is performed as approximation of a global energy function by pathwise optimizations from all directions through the image (Section II-B). Disparity computation is done by winner takes all and supported by disparity refinements like consistency checking and sub-pixel interpolation (Section II-C). Multi-baseline matching is handled by fusion of disparities (Section II-D). Further disparity refinements include peak filtering, intensity consistent disparity selection and gap interpolation (Section II-E). Previously unpublished is the extension for matching almost arbitrarily large

H. Hirschmüller is with the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR).

[1]www.middlebury.edu/stereo

Accepted on April 16, 2007 for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence

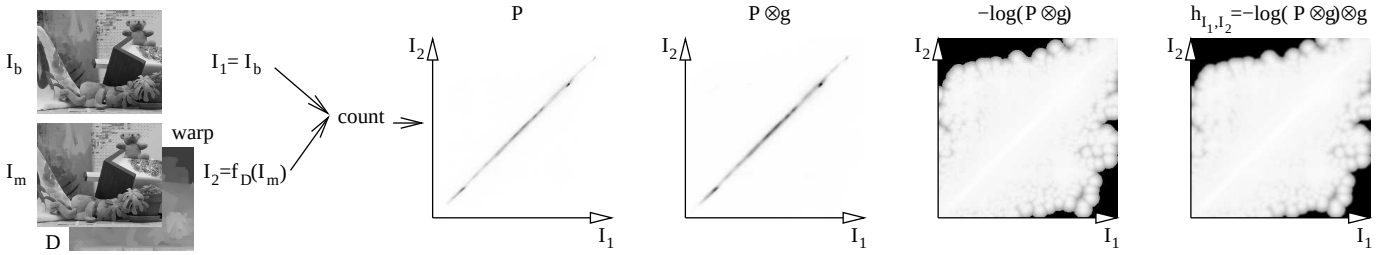


Fig. 1. Calculation of the MI based matching cost. Values are scaled linearly for visualization. Darker points have larger values than brighter points.

images (Section II-F) and the fusion of several disparity images using orthographic projection (Section II-G). Section III shows results on standard test images as well as previously unpublished extensive evaluations of the Mutual Information based matching cost. Finally, two examples of 3D reconstructions from huge aerial frame and pushbroom images are given.

II. Semi-Global Matching

The Semi-Global Matching (SGM) method is based on the idea of pixelwise matching of Mutual Information and approximating a global, 2D smoothness constraint by combining many 1D constraints. The algorithm is described in distinct processing steps. Some of them are optional, depending on the application.

A. Pixelwise Matching Cost Calculation

Input images are assumed to have a known epipolar geometry, but it is not required that they are rectified as this may not always be possible. This is the case with pushbroom images. A linear movement causes epipolar lines to be hyperbolas [21], due to parallel projection in the direction of movement and perspective projection orthogonally to it. Non-linear movements, as unavoidable in aerial imaging, causes epipolar lines to be general curves and images that cannot be rectified [22].

The matching cost is calculated for a base image pixel $\mathbf{p}$ from its intensity $I_{b\mathbf{p}}$ and the suspected correspondence $I_{m\mathbf{q}}$ with $\mathbf{q} = e_{bm}(\mathbf{p}, d)$ of the match image. The function $e_{bm}(\mathbf{p}, d)$ symbolizes the epipolar line in the match image for the base image pixel $\mathbf{p}$ with the line parameter d. For rectified images, with the match image on the right of the base image, $e_{bm}(\mathbf{p}, d) = [p_x - d, p_y]^T$ with d as disparity.

An important aspect is the size and shape of the area that is considered for matching. The robustness of matching is increased with large areas. However, the implicit assumption of constant disparity inside the area is violated at discontinuities, which leads to blurred object borders and fine structures. Certain shapes and techniques can be used to reduce blurring, but it cannot be avoided [8]. Therefore, the assumption of constant disparities in the vicinity of $\mathbf{p}$ is discarded. This means that only the intensities $I_{b\mathbf{p}}$ and $I_{m\mathbf{q}}$ itself can be used for calculating the matching cost.

One choice of pixelwise cost calculation is the sampling insensitive measure of Birchfield and Tomasi [2]. The cost $C_{BT}(\mathbf{p}, d)$ is calculated as the absolute minimum difference of intensities at $\mathbf{p}$ and $\mathbf{q} = e_{bm}(\mathbf{p}, d)$ in the range of half a pixel in each direction along the epipolar line.

Alternatively, the matching cost calculation can be based on Mutual Information (MI) [4], which is insensitive to recording and illumination changes. It is defined from the entropy H of two images (i.e. their information content) as well as their joint entropy.

$$MI_{I_1,I_2} = H_{I_1} + H_{I_2} - H_{I_1,I_2} \tag{1}$$

The entropies are calculated from the probability distributions P of intensities of the associated images.

$$H_I = -\int_0^1 P_I(i) \log P_I(i) di \tag{2}$$

$$H_{I_1,I_2} = -\int_0^1 \int_0^1 P_{I_1,I_2}(i_1, i_2) \log P_{I_1,I_2}(i_1, i_2) di_1 di_2 \tag{3}$$

For well registered images the joint entropy H_{I_1,I_2} is low, because one image can be predicted by the other, which corresponds to low information. This increases their Mutual Information. In the case of stereo matching, one image needs to be warped according to the disparity image D for matching the other image, such that corresponding pixels are at the same location in both images, i.e. $I_1 = I_b$ and $I_2 = f_D(I_m)$.

Equation (1) operates on full images and requires the disparity image a priori. Both prevent the use of MI as pixelwise matching cost. Kim et al. [6] transformed the calculation of the joint entropy H_{I_1,I_2} into a sum over pixels using Taylor expansion. It is referred to their paper for details of the derivation. As result, the joint entropy is calculated as a sum of data terms that depend on corresponding intensities of a pixel $\mathbf{p}$.

$$H_{I_1,I_2} = \sum_{\mathbf{p}} h_{I_1,I_2}(I_{1\mathbf{p}}, I_{2\mathbf{p}}) \tag{4}$$

The data term h_{I_1,I_2} is calculated from the joint probability distribution P_{I_1,I_2} of corresponding intensities. The number of corresponding pixels is n. Convolution with a 2D Gaussian (indicated by $\otimes g(i,k)$) effectively performs Parzen estimation [6].

$$h_{I_1,I_2}(i,k) = -\frac{1}{n} \log(P_{I_1,I_2}(i,k) \otimes g(i,k)) \otimes g(i,k) \tag{5}$$

The probability distribution of corresponding intensities is defined with the operator $\mathrm{T}[]$, which is 1 if its argument is true and 0 otherwise.

$$P_{I_1,I_2}(i,k) = \frac{1}{n} \sum_{\mathbf{p}} \mathrm{T}[(i,k) = (I_{1\mathbf{p}}, I_{2\mathbf{p}})] \tag{6}$$

The calculation is visualized in Fig. 1. The match image I_m is warped according to the initial disparity image D. This can be implemented by a simple lookup in image I_m with $e_{bm}(\mathbf{p}, D_{\mathbf{p}})$ for all pixels $\mathbf{p}$. However, care should to be taken to avoid

possible double mappings due to occlusions in I_m. Calculation of P according to (6) is done by counting the number of pixels of all combinations of intensities, divided by the number of all correspondences. Next, according to (5), Gaussian smoothing is applied by convolution. It has been found that using a small kernel (i.e. 7×7) gives practically the same results as larger kernels, but is calculated faster. The logarithm is computed for each element of the result. Since the logarithm of 0 is undefined, all 0 elements are replaced by a very small number. Another Gaussian smoothing effectively leads to a lookup table for the term h_{I_1,I_2}.

Kim et al. argued that the entropy H_{I_1} is constant and H_{I_2} is almost constant as the disparity image merely redistributes the intensities of I_2. Thus, $h_{I_1,I_2}(I_{1\mathbf{p}}, I_{2\mathbf{p}})$ serves as cost for matching two intensities. However, if occlusions are considered then some intensities of I_1 and I_2 do not have a correspondence. These intensities should not be included in the calculation, which results in non-constant entropies H_{I_1} and H_{I_2}. Apart from this theoretical justification, it has been found that including these entropies in the cost calculation slightly improves object borders. Therefore, it is suggested to calculate these entropies analog to the joint entropy.

$$H_I = \sum_{\mathbf{p}} h_I(I_{\mathbf{p}}) \tag{7a}$$

$$h_I(i) = -\frac{1}{n} \log(P_I(i) \otimes g(i)) \otimes g(i) \tag{7b}$$

The probability distribution P_I must not be calculated over the whole images I_1 and I_2, but only over the corresponding parts (otherwise occlusions would be ignored and H_{I_1} and H_{I_2} would be almost constant). That is easily done by just summing the corresponding rows and columns of the joint probability distribution, i.e. $P_{I_1}(i) = \sum_k P_{I_1,I_2}(i,k)$. The resulting definition of Mutual Information is,

$$MI_{I_1,I_2} = \sum_{\mathbf{p}} mi_{I_1,I_2}(I_{1\mathbf{p}}, I_{2\mathbf{p}}) \tag{8a}$$

$$mi_{I_1,I_2}(i,k) = h_{I_1}(i) + h_{I_2}(k) - h_{I_1,I_2}(i,k). \tag{8b}$$

This leads to the definition of the MI matching cost.

$$C_{MI}(\mathbf{p}, d) = -mi_{I_b, f_D(I_m)}(I_{b\mathbf{p}}, I_{m\mathbf{q}}) \tag{9a}$$

$$\mathbf{q} = e_{bm}(\mathbf{p}, d) \tag{9b}$$

The remaining problem is that the disparity image is required for warping I_m, *before* $mi()$ can be calculated. Kim et al. suggested an iterative solution, which starts with a random disparity image for calculating the cost C_{MI}. This cost is then used for matching both images and calculating a new disparity image, which serves as the base of the next iteration. The number of iterations is rather low (e.g. 3), because even wrong disparity images (e.g. random) allow a good estimation of the probability distribution P, due to a high number of pixels. This solution is well suited for iterative stereo algorithms like Graph Cuts [6], but it would increase the runtime of non-iterative algorithms unnecessarily.

Since a rough estimate of the initial disparity is sufficient for estimating P, a fast correlation base method could be used in the first iterations. In this case, only the last iteration would be done by a more accurate and time consuming method. However, this would involve the implementation of two different stereo methods. Utilizing a single method appears more elegant.

Therefore, a hierarchical calculation is suggested, which recursively uses the (up-scaled) disparity image, that has been calculated at half resolution, as initial disparity. If the overall complexity of the algorithm is $O(WHD)$ (i.e., width $\times$ height $\times$ disparity range), then the runtime at half resolution is reduced by factor $2^3 = 8$. Starting with a random disparity image at a resolution of $\frac{1}{16}$th and initially calculating 3 iterations increases the overall runtime by the factor,

$$1 + \frac{1}{2^3} + \frac{1}{4^3} + \frac{1}{8^3} + 3\frac{1}{16^3} \approx 1.14. \tag{10}$$

Thus, the theoretical runtime of the hierarchically calculated C_{MI} would be just 14% slower than that of C_{BT}, ignoring the overhead of MI calculation and image scaling. It is noteworthy that the disparity image of the lower resolution level is used only for estimating the probability distribution P and calculating the costs C_{MI} of the higher resolution level. Everything else is calculated from scratch to avoid passing errors from lower to higher resolution levels.

An implementation of the hierarchical MI computation (HMI) would collect all alleged correspondences defined by an initial disparity (i.e. up-scaled from previous hierarchical level or random in the beginning). From the correspondences the probability distribution P is calculated according to (6). The size of P is the square of the number of intensities, which is constant (e.g. 256×256). The subsequent operations consist of Gaussian convolutions of P and calculating the logarithm. The complexity depends only on the collection of alleged correspondences due to the constant size of P. Thus, $O(WH)$ with W as image width and H as image height.

B. Cost Aggregation

Pixelwise cost calculation is generally ambiguous and wrong matches can easily have a lower cost than correct ones, due to noise, etc. Therefore, an additional constraint is added that supports smoothness by penalizing changes of neighboring disparities. The pixelwise cost and the smoothness constraints are expressed by defining the energy $E(D)$ that depends on the disparity image D.

$$\begin{aligned} E(D) = \sum_{\mathbf{p}} (C(\mathbf{p}, D_{\mathbf{p}}) &+ \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] \\ &+ \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]) \end{aligned} \tag{11}$$

The first term is the sum of all pixel matching costs for the disparities of D. The second term adds a constant penalty P_1 for all pixels $\mathbf{q}$ in the neighborhood $N_{\mathbf{p}}$ of $\mathbf{p}$, for which the disparity changes a little bit (i.e. 1 pixel). The third term adds a larger constant penalty P_2, for all larger disparity changes. Using a lower penalty for small changes permits an adaptation to slanted or curved surfaces. The constant penalty for all larger changes (i.e. independent of their size) preserves discontinuities [23]. Discontinuities are often visible as intensity changes. This is exploited by adapting P_2 to the intensity gradient, i.e. $P_2 = \frac{P_2'}{|I_{b\mathbf{p}} - I_{b\mathbf{q}}|}$ for neighboring pixels $\mathbf{p}$ and $\mathbf{q}$ in the base image I_b. However, it has always to be ensured that $P_2 \geq P_1$.

The problem of stereo matching can now be formulated as finding the disparity image D that minimizes the energy $E(D)$. Unfortunately, such a global minimization, i.e., in 2D, is NP-complete for many discontinuity preserving energies [23]. In contrast, the minimization along individual image rows, i.e., in 1D, can be performed efficiently in polynomial time using Dynamic Programming [2], [15]. However, Dynamic Programming solutions easily suffer from streaking [1], due to the difficulty of relating the 1D optimizations of individual image rows to each other in a 2D image. The problem is, that very strong constraints in one direction, i.e., along image rows, are combined with none or much weaker constraints in the other direction, i.e., along image columns.

This leads to the new idea of aggregating matching costs in 1D from *all* directions equally. The aggregated (smoothed) cost $S(\mathbf{p}, d)$ for a pixel $\mathbf{p}$ and disparity d is calculated by summing the costs of all 1D minimum cost paths that end in pixel $\mathbf{p}$ at disparity d, as shown in Fig. 2. These paths through disparity space are projected as straight lines into the base image, but as non-straight lines into the corresponding match image, according to disparity changes along the paths. It is noteworthy that only the cost of the path is required and not the path itself.

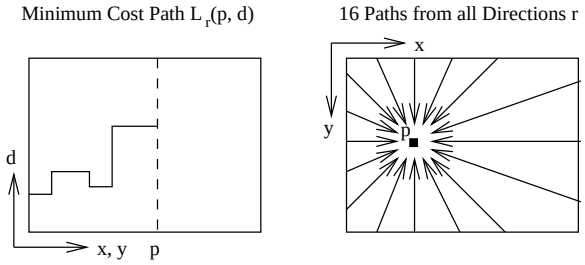


Fig. 2. Aggregation of costs in disparity space.

The cost $L'_{\mathbf{r}}(\mathbf{p}, d)$ along a path traversed in the direction $\mathbf{r}$ of the pixel $\mathbf{p}$ at disparity d is defined recursively as,

$$\begin{aligned} L'_{\mathbf{r}}(\mathbf{p}, d) =& C(\mathbf{p}, d) + \min(L'_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ & L'_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\ & L'_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, \\ & \min_i L'_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2). \end{aligned} \tag{12}$$

The pixelwise matching cost C can be either C_{BT} or C_{MI}. The remainder of the equation adds the lowest cost of the previous pixel $\mathbf{p} - \mathbf{r}$ of the path, including the appropriate penalty for discontinuities. This implements the behavior of (11) along an arbitrary 1D path. This cost does not enforce the *visibility* or *ordering* constraint, because both concepts cannot be realized for paths that are not identical to epipolar lines. Thus, the approach is more similar to *Scanline Optimization* [1] than traditional Dynamic Programming solutions.

The values of L' permanently increase along the path, which may lead to very large values. However, (12) can be modified by subtracting the minimum path cost of the previous pixel from the whole term.

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) =& C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ & L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\ & L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, \\ & \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k) \end{aligned} \tag{13}$$

This modification does not change the actual path through disparity space, since the subtracted value is constant for all disparities of a pixel $\mathbf{p}$. Thus, the position of the minimum does not change. However, the upper limit can now be given as $L \leq C_{max} + P_2$.

The costs $L_{\mathbf{r}}$ are summed over paths in all directions $\mathbf{r}$. The number of paths must be at least 8 and should be 16 for providing a good coverage of the 2D image. In the latter case, paths that are not horizontal, vertical or diagonal are implemented by going one step horizontal or vertical followed by one step diagonally.

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) \tag{14}$$

The upper limit for S is easily determined as $S \leq 16(C_{max} + P_2)$, for 16 paths.

An efficient implementation would pre-calculate the pixelwise matching costs $C(\mathbf{p}, d)$, down-scaled to 11 bit integer values, i.e., $C_{max} < 2^{11}$, by a factor s if necessary as in case of MI values. Scaling to 11 bit guarantees that the aggregated costs in subsequent calculations do not exceed the 16 bit limit. All costs are stored in a 16 bit array $C[]$ of size $W \times H \times D$. Thus, $C[\mathbf{p}, d] = sC(\mathbf{p}, d)$. A second 16 bit integer array $S[]$ of the same size is used for storing the aggregated cost values. The array is initialized by 0 values. The calculation starts for each direction $\mathbf{r}$ at all pixels $\mathbf{b}$ of the image border with $L_{\mathbf{r}}(\mathbf{b}, d) = C[\mathbf{b}, d]$. The path is traversed in forward direction according to (13). For each visited pixel $\mathbf{p}$ along the path, the costs $L_{\mathbf{r}}(\mathbf{p}, d)$ are added to the values $S[\mathbf{b}, d]$ for all disparities d.

The calculation of (13) requires $O(D)$ steps at each pixel, since the minimum cost of the previous pixel, e.g. $\min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$, is constant for all disparities of a pixel and can be pre-calculated. Each pixel is visited exactly 16 times, which results in a total complexity of $O(WHD)$. The regular structure and simple operations, i.e., additions and comparisons, permit parallel calculations using integer based SIMD[2] assembler instructions.

C. Disparity Computation

The disparity image D_b that corresponds to the base image I_b is determined as in local stereo methods by selecting for each pixel $\mathbf{p}$ the disparity d that corresponds to the minimum cost, i.e. $\min_d S[\mathbf{p}, d]$. For sub-pixel estimation, a quadratic curve is fitted through the neighboring costs, i.e., at the next higher and lower disparity, and the position of the minimum is calculated. Using a quadratic curve is theoretically justified only for correlation using the sum of squared differences. However, it is used as an approximation due to the simplicity of calculation. This supports fast computation.

The disparity image D_m that corresponds to the match image I_m can be determined from the same costs, by traversing the epipolar line, that corresponds to the pixel $\mathbf{q}$ of the match

[2]Single Instruction, Multiple Data

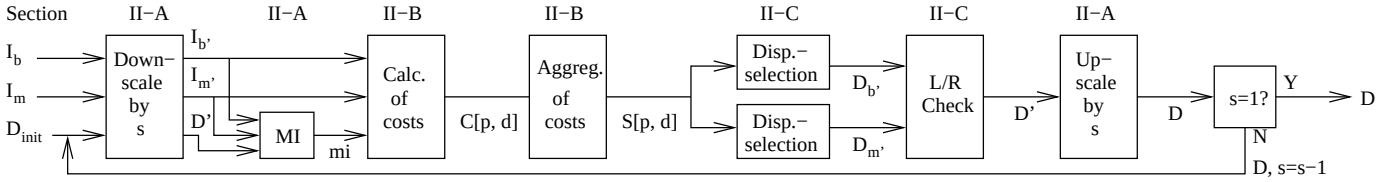


Fig. 3. Summary of processing steps of Sections II-A, II-B and II-C.

image. Again, the disparity d is selected, which corresponds to the minimum cost, i.e. $\min_d S[e_{mb}(\mathbf{q}, d), d]$. However, the cost aggregation step does not treat the base and match images symmetrically. Slightly better results can be expected, if D_m is calculated from scratch, i.e. by performing pixelwise matching and aggregation again, but with I_m as base and I_b as match image. It depends on the application whether or not an increased runtime is acceptable for slightly better object borders. Outliers are filtered from D_b and D_m, using a median filter with a small window, i.e. 3×3.

The calculation of D_b as well as D_m permits the determination of occlusions and false matches by performing a consistency check. Each disparity of D_b is compared to its corresponding disparity of D_m. The disparity is set to invalid (D_{inv}) if both differ.

$$D_{\mathbf{p}} = \begin{cases} D_{b\mathbf{p}} & \text{if } |D_{b\mathbf{p}} - D_{m\mathbf{q}}| \leq 1, \\ D_{inv} & \text{otherwise.} \end{cases} \tag{15a}$$

$$\mathbf{q} = e_{bm}(\mathbf{p}, D_{b\mathbf{p}}) \tag{15b}$$

The consistency check enforces the *uniqueness constraint*, by permitting one to one mappings only. The disparity computation and consistency check require visiting each pixel at each disparity a constant number of times. Thus, the complexity of this step is again $O(WHD)$.

A summary of all processing steps of the core SGM method including hierarchical calculation of mutual information is given in Fig. 3.

D. Multi-Baseline Matching

The algorithm could be extended for multi-baseline matching by calculating a combined pixelwise matching cost of correspondences between the base image and all match images. However, the occlusion problem would have to be solved on the pixelwise matching level, i.e. before aggregation, which is very instable. Therefore, multi-baseline matching is performed by pairwise matching between the base and all match images individually. The consistency check (Section II-C) is used after pairwise matching for eliminating wrong matches at occlusions and many other mismatches. Finally, the resulting disparity images are fused, by considering individual scalings.

Let the disparity D_k be the result of matching the base image I_b against a match image I_{mk}. The disparities of the images D_k are scaled differently, according to some factor t_k. This factor is linear to the length of the baseline between I_b and I_{mk} if all images are rectified against each other, i.e., if all images are projected onto a common plane that has the same distance to all optical centers. Thus, disparities are normalized by $\frac{D_{k\mathbf{p}}}{t_k}$.

Fusion of disparity values is performed by calculating the weighted mean of disparities using the factors t_k as weights. Possible outliers are discarded by considering only those disparities that are within a 1 pixel interval around the median of all disparity values for a certain pixel.

$$D_{\mathbf{p}} = \frac{\sum_{k \in V_{\mathbf{p}}} D_{k\mathbf{p}}}{\sum_{k \in V_{\mathbf{p}}} t_k} \tag{16a}$$

$$V_{\mathbf{p}} = \{k | \left| \frac{D_{k\mathbf{p}}}{t_k} - \operatorname*{med}_i \frac{D_{i\mathbf{p}}}{t_i} \right| \leq \frac{1}{t_k} \} \tag{16b}$$

This solution increases robustness due to the median as well as accuracy due to the weighted mean. Additionally, if enough match images are available, a certain minimum size of the set $V_{\mathbf{p}}$ can be enforced for increasing the reliability of the resulting disparities. Pixel that do not fulfill the criteria are set to invalid. If hierarchical computation is performed for MI based matching then the presented fusion of disparity images is performed within each hierarchical level for computing the disparity image of the next level.

An implementation would pairwise match the base image against all k match images and combine them by visiting each pixel once. Thus, the overall complexity of all steps that are necessary for multi-baseline matching is $O(KWHD)$ with K as the number of match images.

E. Disparity Refinement

The resulting disparity image can still contain certain kinds of errors. Furthermore, there are generally areas of invalid values that need to be recovered. Both can be handled by post processing of the disparity image.

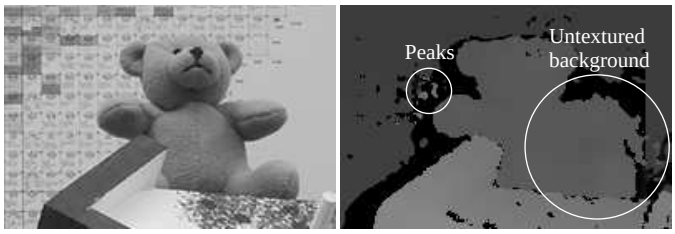


Fig. 4. Possible errors in disparity images (black is invalid).

1) Removal of Peaks: Disparity images can contain outliers, i.e., completely wrong disparities, due to low texture, reflections, noise, etc. They usually show up as small patches of disparity that is very different to the surrounding disparities, i.e. peaks, as shown in Fig. 4. It depends on the scene, what sizes of small disparity patches can also represent valid structures. Often, a threshold can be predefined on their size, such that smaller patches are unlikely to represent valid scene structure.

For identifying peaks, the disparity image is segmented [24], by allowing neighboring disparities within one segment to vary by one pixel, considering a 4-connected image grid. The disparities

of all segments below a certain size are set to invalid [17]. This kind of simple segmentation and peak filtering can be implemented in $O(WH)$ steps.

2) Intensity Consistent Disparity Selection: In structured indoor environments it often happens that foreground objects are in front of a low or untextured background, e.g. wall, as shown in Fig. 4. The energy function $E(D)$ as shown in (11) does not include a preference on the location of a disparity step. Thus, $E(D)$ does not differentiate between placing a disparity step correctly just next to a foreground object or a bit further away within an untextured background. Section II-B suggested adapting the cost P_2 according to the intensity gradient. This helps placing the disparity step correctly just next to a foreground object, because this location coincides with an intensity gradient in contrast to a location within an untextured area.

However, SGM applies the energy function not in 2D over the whole image, but along individual 1D paths from all directions, which are summed. If an untextured area is encountered along a 1D path, a disparity change is only preferred if matching of textured areas on both sides of the untextured area requires it. Untextured areas may have different shapes and sizes and can extend beyond image borders, as quite common for walls in indoor scenes (Fig. 4). Depending on the location and direction of 1D paths, they may encounter texture of foreground and background objects around an untextured part, in which case a correct disparity step would be expected. They may also encounter either foreground or background texture or leave the image with the untextured area in which cases no disparity step would be placed. Summing all those inconsistent paths may easily lead to fuzzy discontinuities around foreground objects in front of untextured background.

It is noteworthy, that this problem is a special case that only applies to certain scenes in structured environments. However, it appears important enough for presenting a solution. First, some some assumptions are made.

1) Discontinuities in the disparity image do not occur within untextured areas.
2) On the same physical surface as the untextured area is also some texture visible.
3) The surface of the untextured area can be approximated by a plane.

The first assumption is mostly correct, as depth discontinuities usually cause at least some visual change in intensities. Otherwise, the discontinuity would be undetectable. The second assumption is necessary as the disparity of an absolutely untextured background surface would be indeterminable. The third assumption is the weakest. Its justification is that untextured surfaces with varying distance usually appear with varying intensities. Thus, piecewise constant intensity can be treated as piecewise planar.

The identification of untextured areas is done by a fixed bandwidth Mean Shift Segmentation [25] on the intensity image I_b. The radiometric bandwidth σ_r is set to P_1, which is usually 4. Thus, intensity changes below the smoothness penalty are treated as noise. The spatial bandwidth σ_s is set to a rather low value for fast processing (i.e. 5). Furthermore, all segments that are smaller than a certain threshold (i.e. 100 pixels) are ignored, because small untextured areas are expected to be handled well by SGM.

As described above, the expected problem is that discontinuities are placed fuzzily within untextured areas. Thus, untextured areas are expected to contain incorrect disparities of the foreground object but also correct disparities of the background, as long as the background surface contains some texture, i.e. assumption 2. This leads to the realization that some disparities within each segment S_i should be correct. Thus, several hypotheses for the correct disparity of S_i can be identified by segmenting the disparity within each segment S_i. This is done by simple segmentation, as also discussed in Section II-E.1, i.e. by allowing neighboring disparities within one segment to vary by one pixel. This fast segmentation results in several segments S_{ik} for each segment S_i.

Next, the surface hypotheses F_{ik} are created by calculating the best fitting planes through the disparities of S_{ik}. The choice for planes is based on assumption 3. Very small segments, i.e., ≤ 12 pixel, are ignored, as it is unlikely that such small patches belong to the correct hypothesis. Then, each hypothesis is evaluated within S_i by replacing all pixel of S_i by the surface hypothesis and calculating E_{ik} as defined in (11) for all unoccluded pixel of S_i. A pixel $\mathbf{p}$ is occluded, if another pixel with higher disparity maps to the same pixel $\mathbf{q}$ in the match image. This detection is performed by first mapping $\mathbf{p}$ into the match image by $\mathbf{q} = e_{bm}(\mathbf{p}, D'_{\mathbf{p}})$. Then, the epipolar line of $\mathbf{q}$ in the base image $e_{mb}(\mathbf{q}, d)$ is followed for $d > D'_{\mathbf{p}}$. Pixel $\mathbf{p}$ is occluded if the epipolar line passes a pixel with a disparity larger than d.

For each constant intensity segment S_i the surface hypothesis F_{ik} with the minimum cost E_{ik} is chosen. All disparities within S_i are replaced by values on the chosen surface for making the disparity selection consistent to the intensities of the base image, i.e., fulfilling assumption 1.

$$F_i = F_{ik'} \text{ with } k' = \underset{k}{\operatorname{argmin}}\, E_{ik} \tag{17a}$$

$$D'_{\mathbf{p}} = \begin{cases} F_i(\mathbf{p}) & \text{if } \mathbf{p} \in S_i \\ D_{\mathbf{p}} & \text{otherwise.} \end{cases} \tag{17b}$$

The presented approach is similar to some other methods [7], [9], [11] as it uses image segmentation and plane fitting for refining an initial disparity image. In contrast to other methods, the initial disparity image is due to SGM already quite accurate so that only untextured areas above a certain size are modified. Thus, only critical areas are tackled without the danger of corrupting probably well matched areas. Another difference is that disparities of the considered areas are selected by considering a small number of hypotheses that are inherent in the initial disparity image. There is no time consuming iteration.

The complexity of fixed bandwidth Mean Shift Segmentation of the intensity image and the simple segmentation of the disparity image is linear in the number of pixels. Calculating the best fitting planes involves visiting all segmented pixels. Testing of all hypotheses requires visiting all pixels of all segments, for all hypotheses (i.e. maximum N). Additionally, the occlusion test requires going through at most D disparities for each pixel.

Thus, the upper bound of the complexity is $O(WHDN)$. However, segmented pixels are usually just a fraction of the whole image and the maximum number of hypotheses N for a segment is commonly small and often just 1. In the latter case, it is not even necessary to calculate the cost of the hypothesis.

3) Discontinuity Preserving Interpolation: The consistency check of Section II-C as well as fusion of disparity images of Section II-D or peak filtering of Section II-E.1 may invalidate some disparities. This leads to holes in the disparity image, as

shown in black in Fig. 4, which need to be interpolated for a dense result.

Invalid disparities are classified into occlusions and mismatches. The interpolation of both cases must be performed differently. Occlusions must not be interpolated from the occluder, but only from the occludee to avoid incorrect smoothing of discontinuities. Thus, an extrapolation of the background into occluded regions is necessary. In contrast, holes due to mismatches can be smoothly interpolated from all neighboring pixels.

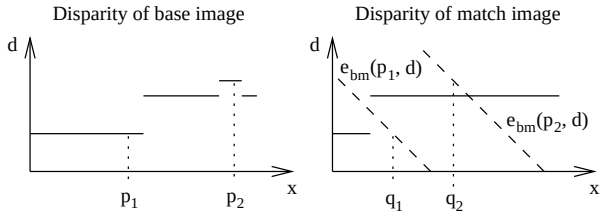


Fig. 5. Distinguishing between occluded and mismatched pixels.

Occlusions and mismatches can be distinguished as part of the left/right consistency check. Fig. 5 shows that the epipolar line of the occluded pixel $\mathbf{p}_1$ goes through the discontinuity that causes the occlusion and does not intersect the disparity function D_m. In contrast, the epipolar line of the mismatch $\mathbf{p}_2$ intersects with D_m. Thus, for each invalidated pixel, an intersection of the corresponding epipolar line with D_m is sought, for marking it as either occluded or mismatched.

For interpolation purposes, mismatched pixel areas that are direct neighbors of occluded pixels are treated as occlusions, because these pixels must also be extrapolated from valid background pixels. Interpolation is performed by propagating valid disparities through neighboring invalid disparity areas. This is done similarly to SGM along paths from 8 directions. For each invalid pixel, all 8 values $v_{\mathbf{p}i}$ are stored. The final disparity image D' is created by,

$$D'_{\mathbf{p}} = \begin{cases} \text{seclow}_i\, v_{\mathbf{p}i} & \text{if } \mathbf{p} \text{ is occluded,} \\ \text{med}_i\, v_{\mathbf{p}i} & \text{if } \mathbf{p} \text{ is mismatched,} \\ D_{\mathbf{p}} & \text{otherwise.} \end{cases} \tag{18}$$

The first case ensures that occlusions are interpolated from the lower background by selecting the second lowest value, while the second case emphasizes the use of all information without a preference to foreground or background. The median is used instead of the mean for maintaining discontinuities in cases where the mismatched area is at an object border.

The presented interpolation method has the advantage that it is independent of the used stereo matching method. The only requirements are a known epipolar geometry and the calculation of the disparity images for the base and match image for distinguishing between occlusions and mismatches.

Finally, median filtering can be useful for removing remaining irregularities and additionally smoothes the resulting disparity image. The complexity of interpolation is linear to the number of pixels, i.e. $O(WH)$, as there is a constant number of operations for each invalid pixel.

F. Processing of Huge Images

The SGM method requires temporary memory for storing pixelwise matching costs $C[]$, aggregated costs $S[]$, disparity images before fusion, etc. The size of temporary memory depends either on the image size $W * H$, the disparity range D or both as in case of $C[]$ and $S[]$. Thus, even moderate image sizes of 1 MPixel with disparity ranges of several 100 pixel require large temporary arrays that can exceed the available memory. The proposed solution is to divide the base image into tiles, computing the disparity of each tile individually as described in Sections II-A until II-C and merging the tiles together into the full disparity image before multi-baseline fusion (Section II-D).

Tiles are chosen overlapping, because the cost aggregation step (Section II-B) can only use paths from one side for pixels near tile borders, which leads to lower matching accuracy or even mismatches. This can especially be critical at low textured areas near tile borders. Merging of tiles is done by calculating a weighted mean of disparities from all tiles at overlapping areas. The weights are chosen such that pixels near the tile border are ignored and those further away are blended linearly as shown in Fig. 6. The tile size is chosen as large as possible, such that all required temporary arrays just fit into the available main memory. Thus, the available memory automatically determines the internally used tile size.

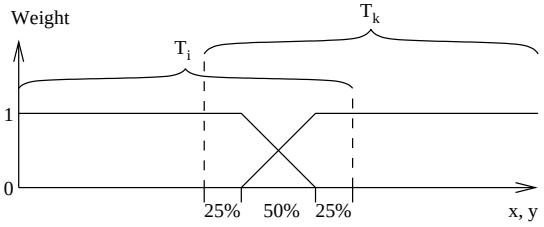


Fig. 6. Definition of weights for merging overlapping tiles T_i, T_k.

This strategy allows matching of larger images. However, there are some technologies like aerial pushbroom cameras that can produce single images of 1 billion pixel or more [22]. Thus, it may be impossible to even load two full images into main memory, not to mention matching of them. For such cases, additionally to the discussed internal tiling, an external tiling of the base image is suggested, e.g., with overlapping tiles of size 3000×3000 pixels. Every base image tile together with the disparity range and the known camera geometry immediately define the corresponding parts of the match images. All steps including multi-baseline fusion (Section II-D) and optionally post processing (Section II-E) are performed and the resulting disparity is stored for each tile individually. Merging of external tiles is done in the same way as merging of internal tiles.

Depending on the kind of scene, it is likely that the disparity range that is required for each tile is just a fraction of the disparity range of the whole images. Therefore, an automatic disparity range reduction in combination with HMI based matching is suggested. The full disparity range is applied for matching at the lowest resolution. Thereafter, a refined disparity range is determined from the resulting disparity image. The range is extended by a certain fixed amount to account for small structures that are possibly undetected while matching in low resolution. The refined, up-scaled disparity range is used for matching at the next higher resolution.

The internal and external tiling mechanism allow stereo matching of almost arbitrarily large images. Another advantage of external tiling is that all tiles can be computed in parallel on different computers.

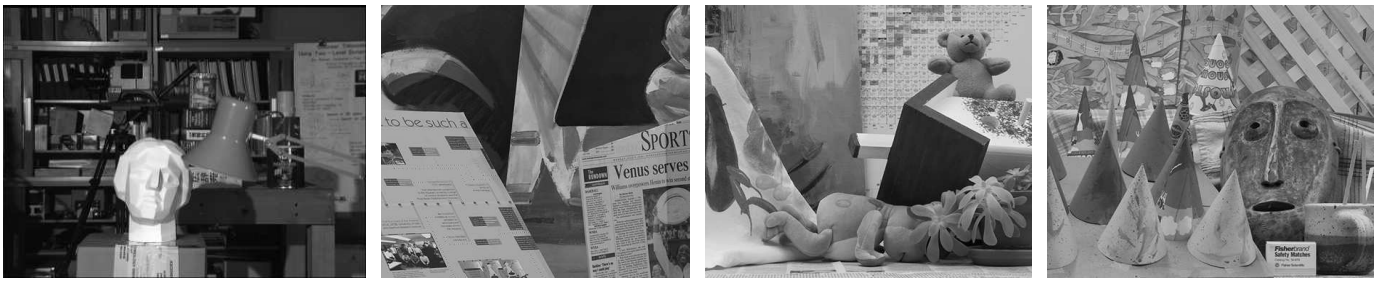


Fig. 7. The Tsukuba (384×288), Venus (484×383), Teddy (450×375) and Cones (450×375) stereo test images [1], [18].

G. Fusion of Disparity Images

Disparity images can be seen as 2.5D representations of the scene geometry. The interpretation of disparity images always requires the corresponding geometrical camera model. Furthermore, in multiple image configurations, several disparity images from different viewpoints may have been computed for representing a scene. It is often desirable to fuse the information of all disparity images into one consistent representation of the scene. The optimal scene representation depends on the locations and viewing directions of all cameras. An important special case, e.g., for aerial imaging [22], is that the optical centers of all cameras are approximately in a plane and the orientations of all cameras are approximately the same. In this case, an orthographic 2.5D projection onto a common plane can be done.

The common plane is chosen parallel to the optical centers of all cameras. A coordinate system R_o, T_o is defined such that the origin is in the plane and the z-axis is orthogonal to the plane. The x, y-plane is divided into equally spaced cells. Each disparity image is transformed separately into orthographic projection, by reconstructing all pixels, transforming them using R_o, T_o and storing the z-values in the cells in which the transformed points fall into. The change to orthographic projection can cause some points to occlude others. This is considered by always keeping the value that is closest to the camera in case of double mappings. After transforming each disparity image individually, the resulting orthographic projections are fused by selecting the median of all values that fall into each cell (Fig. 8). This is useful for eliminating remaining outliers.

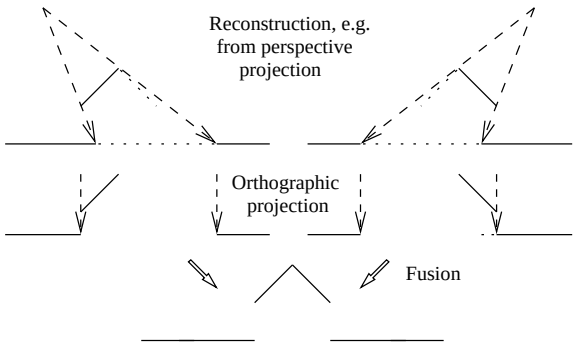


Fig. 8. Orthographic reprojection of disparity images and fusion.

It is advisable not to interpolate missing disparities in individual disparity images (Section II-E.3) before performing fusion, because missing disparities may be filled in from other views. This is expected to be more accurate than using interpolated values. Furthermore, the orthographic reprojection can lead to new holes that have to be interpolated. Thus, interpolation is required anyway. However, after orthographic projection, the information about occlusions and mismatches that is used for pathwise interpolation (Section II-E.3) is lost. Therefore, a different method is suggested for interpolating orthographic 2.5D height data.

First, the height data is segmented in the same way as described in Section II-E.1 by allowing height values of neighboring grid cells within one segment to vary by a certain predefined amount. Each segment is considered to be a physical surface. Holes can exist within or between segments. The former are filled by Inverse Distance Weighted (IDW) interpolation from all valid pixels just next to the hole. The latter case is handled by only considering valid pixels of the segment whose pixel have the lowest mean compared to the valid bordering pixel of all other segments next to the hole. This strategy performs smooth interpolation, but maintains height discontinuities by extrapolating the background. Using IDW instead of pathwise interpolation is computationally more expensive, but it is performed only once on the fused result and not on each disparity image individually.

III. Experimental Results

The SGM method has been evaluated extensively on common stereo test image sets as well as real images.

A. Evaluation on Middlebury Stereo Images

Fig. 7 shows the left images of four stereo image pairs [1], [18]. This image set is used in an ongoing comparison of stereo algorithms on the Middlebury Stereo Pages. The image sets of Venus, Teddy and Cones consist of 9 multi-baseline images. For stereo matching, the image number 2 is used as the left image and the image number 6 as the right image. This is different to an earlier publication [19], but consistent with the procedure of the new evaluation on Middlebury Stereo Pages. The disparity range is 16 pixel for the Tsukuba pair, 32 pixel for the Venus pair and 64 pixel for the Teddy and Cones pair.

Disparity images have been computed in two different configurations. The first configuration called SGM, uses the basic steps like cost calculation using HMI, cost aggregation and disparity computation (Sections II-A until II-C). Furthermore, small disparity peaks where removed (Section II-E.1) and gaps interpolated (Section II-E.3). The second configuration is called C-SGM, which uses the same steps as SGM, but additionally the intensity consistent disparity selection (Section II-E.2). All parameters have been selected for the best performance and kept constant. The threshold of the disparity peak filter has been lowered for C-SGM, because intensity consistent disparity selection helps eliminating peaks, if they are in untextured areas.

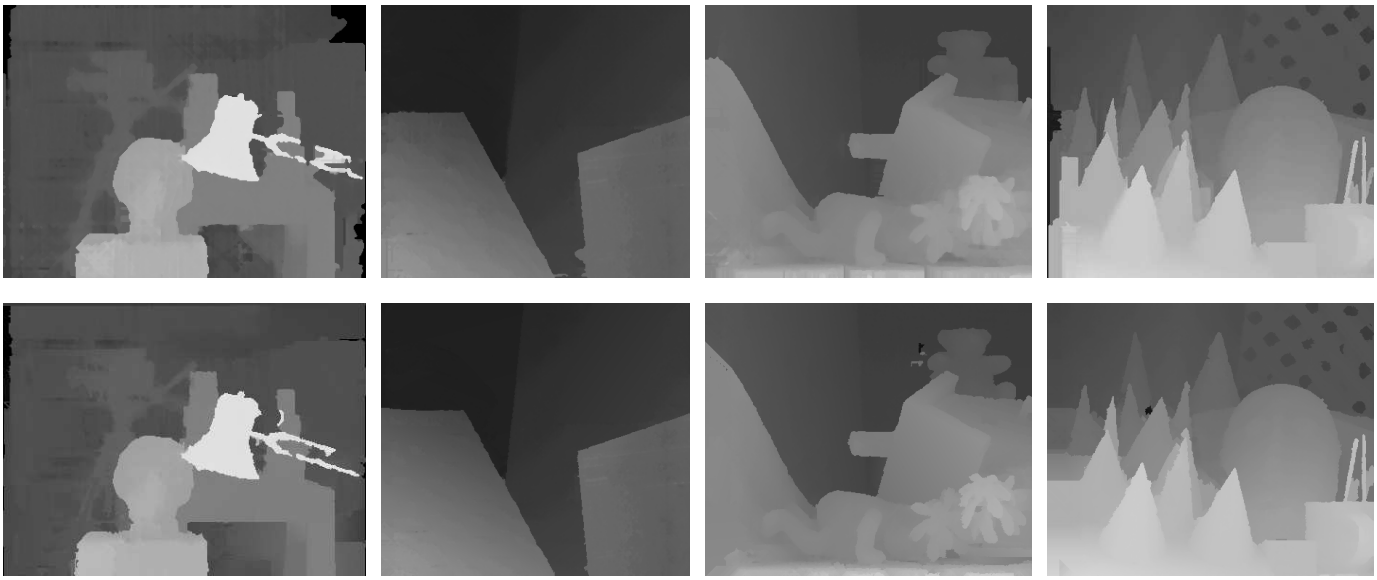


Fig. 9. Disparity images calculated by SGM (top) and C-SGM (bottom), which includes the intensity consistent disparity selection post-processing step.

TABLE I
COMPARISON USING STANDARD THRESHOLD OF 1 PIXEL (LEFT) AND 0.5 PIXEL (RIGHT), FROM OCTOBER 2006.

Algorithm	Rank	Tsuk.	Venus	Teddy	Cones
AdaptingBP [3]	1.7	1.11	*0.10*	4.22	*2.48*
DoubleBP [11]	2.3	*0.88*	0.14	*3.55*	2.90
Segm+visib [9]	5.1	1.30	0.79	5.00	3.72
SymBP+occ [14]	5.1	0.97	0.16	6.47	4.79
C-SGM	6.2	2.61	0.25	5.14	2.77
RegTreeDP [12]	7.0	1.39	0.22	7.42	6.31
AdaptWeight [10]	7.3	1.38	0.71	7.88	3.97
SGM	9.3	3.26	1.00	6.02	3.06
Currently 16 more entries ...					

Algorithm	Rank	Tsuk.	Venus	Teddy	Cones
C-SGM	3.6	13.9	3.30	*9.82*	5.37
SGM	5.0	13.4	4.55	11.0	*4.93*
AdaptingBP [3]	5.3	19.1	4.84	12.8	7.02
Segm+visib [9]	5.8	12.7	10.4	11.0	8.12
DoubleBP [11]	7.1	18.7	7.85	14.3	11.9
GenModel [26]	8.1	7.89	4.59	14.8	10.2
SymBP+occ [14]	8.8	20.7	5.96	15.7	11.4
CostRelax	9.3	26.3	*2.92*	12.3	6.33
Currently 16 more entries ...					

Fig. 9 shows the results of SGM and C-SGM. Differences can be best seen on the right side of the Teddy image. SGM produces foreground disparities between the arm the the leg of the Teddy, because there are no straight paths from this area to structured parts of the background. In contrast, C-SGM recovers the shape of the Teddy correctly. The mismatches on the left of Teddy are due to repetitive texture and are not filtered by C-SGM, because the disparity peak filter threshold had been lowered as described above, for a better overall performance.

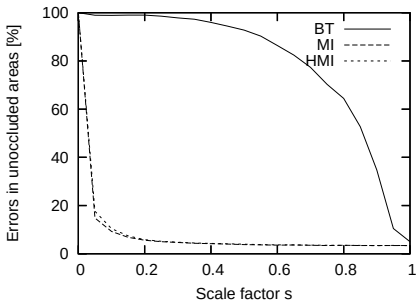
The disparity images are numerically evaluated by counting the disparities that differ by more than a certain threshold from the ground truth. Only pixels that are unoccluded according to the ground truth are compared. The result is given as percentage of erroneous pixels. Table I is a reproduction of the upper part of the new evaluation at the Middlebury Stereo Pages. A standard threshold of 1 pixel has been used for the left table. Both, SGM and C-SGM are among the best performing stereo algorithms at the upper part of the table. C-SGM performs better, because it recovers from errors at untextured background areas. Lowering the threshold to 0.5 pixel makes SGM and C-SGM the top-performing algorithms as shown in table I (right). The reason seams to be a better sub-pixel performance.

SGM and C-SGM have been prepared for working with unrectified images with known epipolar geometry, by defining a function that calculates epipolar lines point by point. This is a processing time overhead for rectified images, but permits working on pushbroom images that cannot be rectified [22]. The most time consuming cost aggregation step has been implemented using Single Instruction Multiple Data (SIMD) assembler commands, i.e. SSE2 instruction set. The processing time on the Teddy pair was 1.8s for SGM and 2.7s for C-SGM on a 2.2GHz Opteron CPU. This is much faster than most other methods of the comparison.

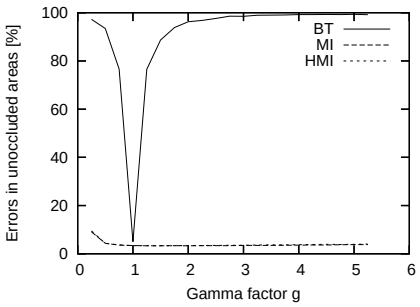
B. Evaluation of MI as Matching Cost Function

MI based matching has been discussed in Section II-A for compensating radiometric differences between the images while matching. Such differences are minimal in carefully prepared test images as those of Fig. 7, but they often occur in practice. Several transformations have been tested on the four image pairs of Fig. 7. The left image has been kept constant while the right image has been transformed. The matching cost has been calculated by sampling insensitive absolute difference of intensities (BT), iteratively calculated Mutual Information (MI) and hierarchically calculated Mutual Information (HMI). The mean error over the four pairs is used for the evaluation.

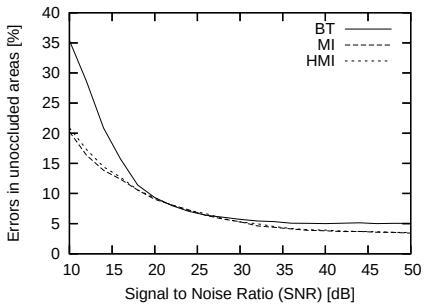
Fig. 10(a) and 10(b) show the result of globally scaling intensities linearly or non-linearly. BT breaks down very quickly,



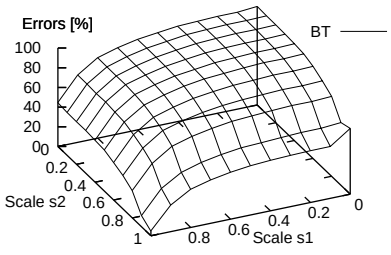
(a) Global scale change, i.e. $I' = sI$.



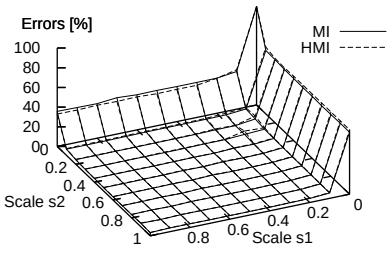
(b) Global gamma change.



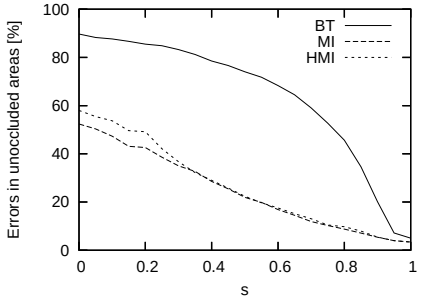
(c) Adding Gaussian noise.



(d) Different scaling of image halves.



(e) Different scaling of image halves.



(f) Linear down-scaling from center

Fig. 10. Effect of applying radiometric changes or adding noise to the right match images, using SGM with different matching cost calculations.

while the performance of MI and HMI is almost constant. They break down only due to the severe loss of image information when transformed intensities are stored into 8 bit. Fig. 10(c) shows the effect of adding Gaussian noise. MI and HMI are affected, but perform better than BT for high noise levels. 10 dB means that the noise level is about $\frac{1}{3}$rd of the signal level.

Thus, global transformations are well handled by MI and HMI. The next test scales the left and right image halves differently for simulating a more complex case with two different radiometric mappings within one image. This may happen, if the illumination changes in a part of the image. The left image of Fig. 11 demonstrates the effect. The result is shown in Fig. 10(d) and 10(e). Again, BT breaks down very quickly, while MI and HMI are almost constant. Fig. 10(f) shows the results of decreasing the intensity linearly from the image center to the border. This is a locally varying transformation, which mimics a vignetting effect that is often found in camera lenses (right image of Fig. 11). MI and HMI have more problems than in the other experiments, but compensate the effect much better than BT, especially for large s, which can be expected in practice.

Fig. 11. Examples of local scaling of intensities with $s_1 = 0.3$ and $s_2 = 0.7$ (left) and linear down-scaling from image center with $s = 0.5$ (right).

The matching costs have also been tested on the Art dataset, which is a courtesy of Daniel Scharstein. The dataset offers stereo images that have been taken with different exposures and under different illuminations, i.e. with changed position of the light source, as shown in Fig. 12(a) and 12(b). There is also a ground truth disparity available. The errors that occur when matching images of different exposures are shown in Fig. 12(c). It can be seen that BT fails completely while HMI is nearly unaffected by the severe changes of exposure. Fig. 12(d) gives the result of matching images that are taken under different illuminations. This time, also HMI is affected, but to a lower extent than BT. It should be noted that illumination changes in these images are very severe and cause many local changes.

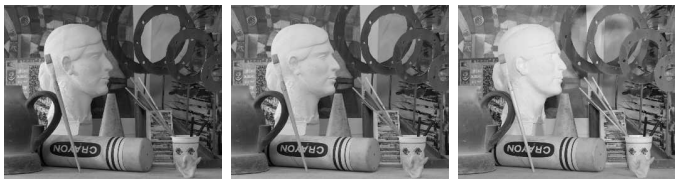
BT based matching takes 1.5s on the Teddy images, while MI base matching requires 3 iterations, which takes 4s. This is 164% slower than BT. The suggested HMI base matching needs 1.8s, which is just 18% slower than BT. The values are similar for the other image pairs.

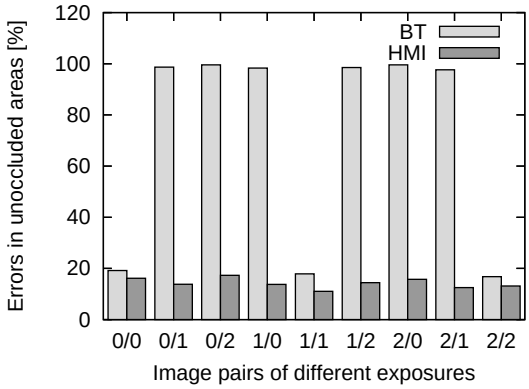
All of the experiments demonstrate that the performance of MI and HMI is almost identical. Both tolerate global changes like different exposure times without any problems. Local changes, like vignetting are also handled quite well. Changes in lighting of the scene seem to be tolerated to some extent. In contrast, BT breaks down very quickly. Thus, using BT is only advisable on images that are carefully taken under exactly the same conditions. Since HMI performed better in all experiments and just requires a small, constant fraction of the total processing time, it is always recommended for stereo matching.

C. Evaluation of Post-Processing

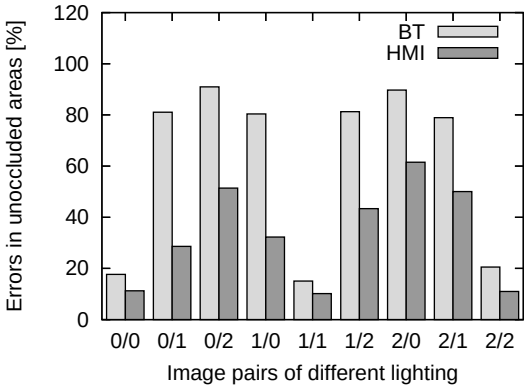
Post-processing is necessary for fixing errors that the stereo algorithm has caused and providing a dense disparity image


(a) Left images of Art dataset with varying exposure.


(b) Left images of Art dataset with varying illuminations.

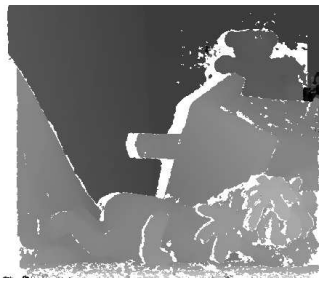


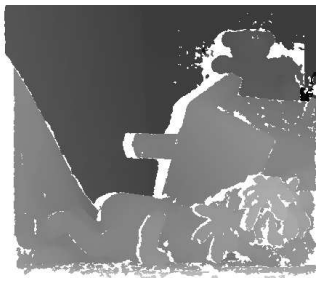
(c) Result of matching images with different exposure.



(d) Result of matching images with different illuminations.

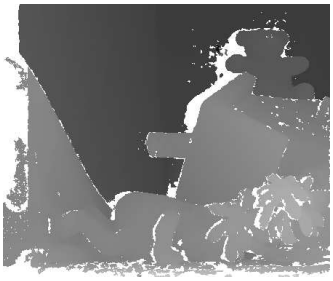
Fig. 12. Matching of images with different exposure and lighting. The Art dataset is a courtesy of Daniel Scharstein.


(a) Result after basic SGM

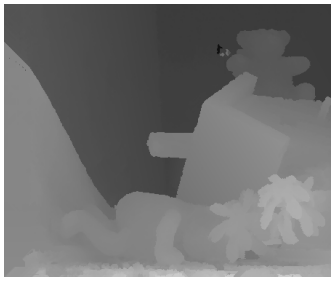

(b) Result after peak filtering


(c) Low textured segments


(d) Disparity segments


(e) Intensity Consistent Sel.


(f) Occlusions (black) and missm. (grey)


(g) Result after interpolation


(h) Errors against ground truth

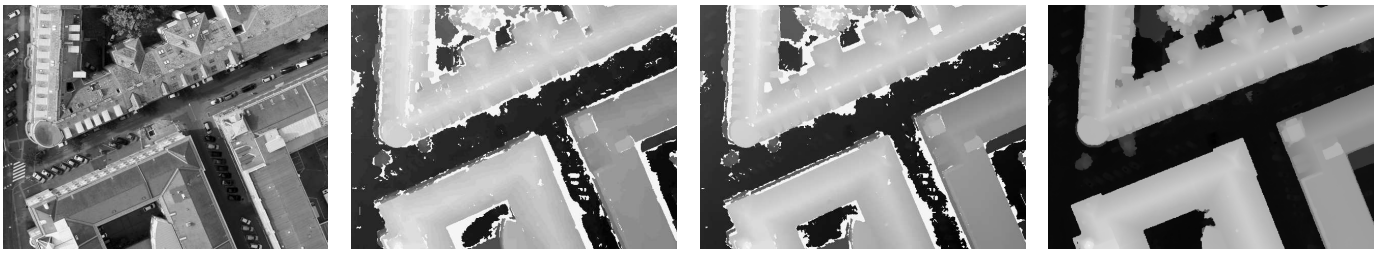
Fig. 13. Demonstration of the effect of the proposed post-processing steps on the Teddy images.

without gaps. The effects of the proposed post-processing steps are shown in Fig. 13.

Fig. 13(a) shows raw result of SGM, i.e. Sections II-A until II-C. Peak filtering, i.e. Section II-E.1, removes some isolated, small patches of different disparity as given in Fig. 13(b). Fig. 13(c) and 13(d) show the segmentation results of the intensity and disparity image that are used by the intensity consistent disparity selection method, i.e. Section II-E.2. The result in Fig. 13(e) shows that disparities in critical, untextured areas have been recovered. Fig. 13(f) gives the classification result for interpolation. Occlusions are black and other mismatches are white. The result of pathwise interpolation, i.e., Section II-E.3, is presented in Fig. 13(g). Finally, Fig. 13(h) gives the errors when comparing Fig. 13(g) against ground truth with the standard threshold of 1 pixel.

D. Example 1: Reconstruction from Aerial Full Frame Images

The SGM method has been designed for calculating accurate Digital Surface Models (DSM) from high resolution aerial images. Graz in Austria has been captured by Vexcel Imaging with an UltraCam, which has a 54° field of view and offers panchromatic images of 11500×7500 pixels, i.e. 86 MPixel. Color and infrared



(a) Small part of aerial image (b) Matching against 1 image (c) Matching against 6 images (d) Orthographic reprojection

Fig. 14. SGM matching results from aerial UltraCam images of Graz. The input image block is a courtesy of Vexcel Imaging Graz.

are captured as well, but at a lower resolution. A block of 3×15 images has been provided as courtesy of Vexcel Imaging Graz. The images were captured 900 m above ground with an overlap of approximately 85% in flight direction and 75% orthogonal to it. The ground resolution was 8 cm/pixel. The image block was photogrammetrically oriented by bundle adjustment using GPS/INS data that was recorded during the flight as well as ground control points.

The SGM method using HMI as matching cost has been applied with the same parameters as used for the comparison in Section III-A, except for post filtering. The peak filter threshold has been increased to 300 pixel. Furthermore, the intensity consistent disparity selection is not used as aerial images do typically not include any untextured background surfaces. This may also be due to the quality of images, i.e. sharpness. Finally, interpolation has not been done. The disparity range of this data set is up to 2000 pixel. The size of the images and the disparity range required internal and external tiling as well as dynamic disparity range adaptation as described in Section II-F.

A small part of one image is given in Fig. 14(a). Fig. 14(b) shows the result of matching against one image to the left and Fig. 14(c) the result of multi-baseline matching against 6 surrounding images. It can be seen that matching of two images results already in a good disparity image. Matching against all surrounding images helps to fill in gaps that are caused by occlusions and removing remaining mismatches. After matching, all images are fused into an orthographic projection and interpolated as described in Section II-G. The result can be seen in Fig. 14(d). The roof structures and boundaries appear very precise. Matching of one image against 6 neighbors took around 5.5 hours on one 2.2 GHz Opteron CPU. 22 CPU's of a processing cluster were used for parallel matching of the 45 images. The orthographic reprojection and true ortho-image generation requires a few more hours, but only on one CPU.

Fig. 15 presents 3D reconstructions from various viewpoints. The texture is taken from UltraCam images as well. Mapping texture onto all walls of buildings is possible due to the relatively large field of view of the camera and high overlap of images. The given visualizations are high quality results of fully automatic processing steps without any manual cleanup.

E. Example 2: Reconstruction from Aerial Pushbroom Images

The SGM method has also been applied to images of the High Resolution Stereo Camera (HRSC) that has been built by the Institute of Planetary Research at DLR Berlin for stereo mapping of Mars. The camera is currently operating on-board the ESA probe Mars-Express that is orbiting Mars. Another version of the camera is used on-board airplanes for mapping Earth's cities and landscapes from flight altitudes between 1500 m-5000 m above ground [27]. The camera has nine 12 bit sensor arrays with 12000 pixels, which are mounted orthogonally to the flight direction and look downwards in different angles up to 20.5°. Five of the sensor arrays are panchromatic and used for stereo matching. The other four capture red, green, blue and infrared. The position and orientation of the camera is continuously measured by a GPS/INS system. The ground resolution of the images is 15-20 cm/pixel.

The SGM method has been applied to HRSC images that have been radiometrically and geometrically corrected at the Institute of Planetary Research at DLR Berlin. The result are 2D images from the data captured by each of the nine sensor arrays. However, despite geometric rectification, epipolar lines are in general not straight, as this is not possible for aerial pushbroom images. Thus, epipolar lines are calculated during image matching as described previously [22].

SGM using HMI as matching cost has been applied again as in Section III-D with the same parameters. Matching is performed between the five panchromatic images of each flight strip individually. Each of these images can have a size of up to several GB, which requires internal and external tiling as well as dynamic disparity range adaptation as described in Section II-F. Matching between strips is not done as the overlap of strips is typically less than 50%.

The fully automatic method has been implemented on a cluster of 40 2.0 GHz and 2.2 GHz Opteron CPU's. The cluster is able to process an area of 400 km^2 in a resolution of 20 cm/pixel within three to four days, resulting in around 50 GB of height and image data. A total of more than 20000 km^2 has been processed within one year.

Fig. 16 shows a reconstruction of a small part of one scene. The visualizations were calculated fully automatically, including mapping of the wall texture from HRSC images. It should be noted that the ground resolution of the HRSC images is almost three times lower than that of the UltraCam images of Fig. 15. Nevertheless, a good quality of reconstruction, with sharp object boundaries can be achieved on huge amounts of data. This demonstrates that the proposed ideas are working very stable on practical problems.

IV. Conclusion

The SGM stereo method has been presented. Extensive tests show that it is tolerant against many radiometric changes that occur in practical situations due to a hierarchically calculated Mutual Information (HMI) based matching cost. Matching is done

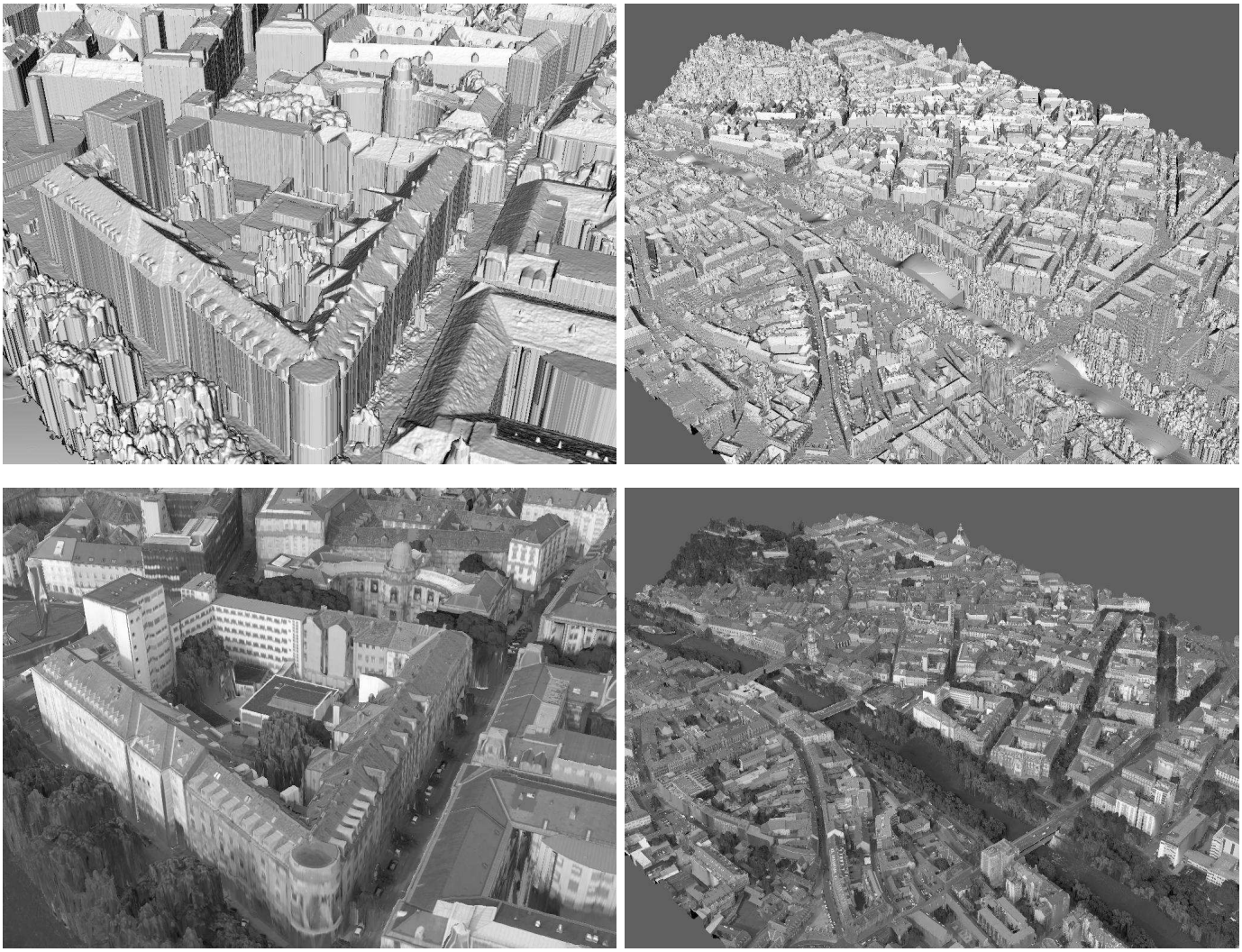


Fig. 15. Untextured and textured 3D reconstructions from aerial UltraCam images of Graz.

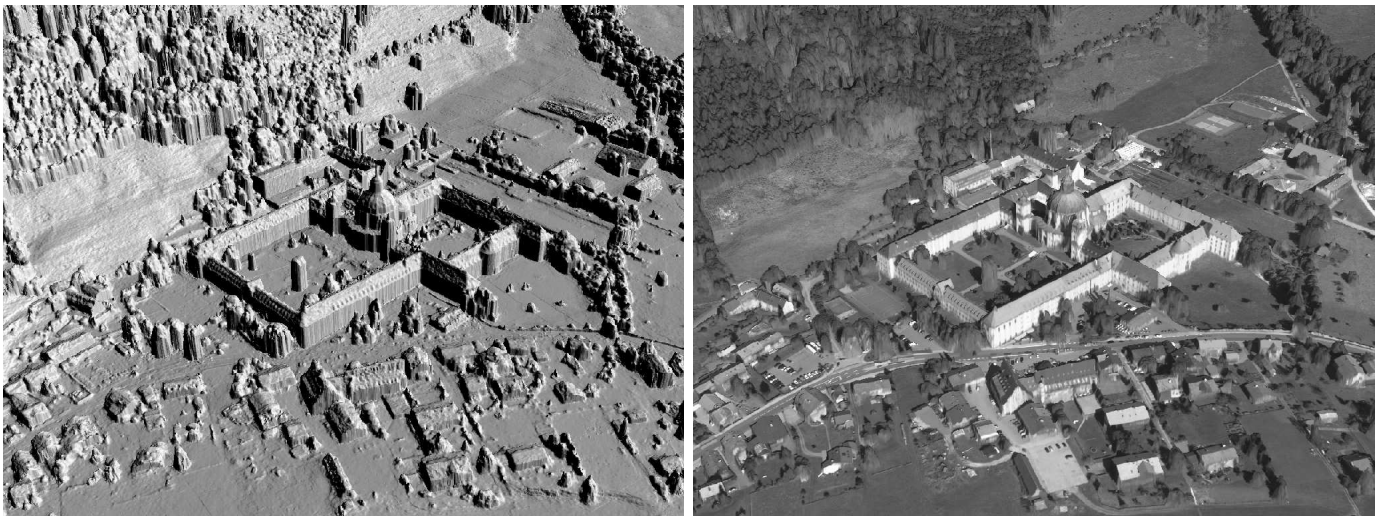


Fig. 16. Untextured and textured reconstructions from images of the DLR High Resolution Stereo Camera (HRSC) of Ettal.

accurately on a pixel level by pathwise optimization of a global cost function. The presented post filtering methods optionally help by tackling remaining individual problems. An extension for matching huge images has been presented as well as a strategy for fusing disparity images using orthographic projection.

The method has been evaluated on the Middlebury Stereo Pages. It has been shown that SGM can compete with the currently best stereo methods. It even performs superior to all other methods when the threshold for comparing the results against ground truth is lowered from 1 to 0.5 pixel, which shows an excellent sub-pixel performance. All of this is done with a complexity of $O(WHD)$ that is rather common for local methods. The runtime is just 1-2s on typical test images, which is much lower than that of most other methods with comparable results. Experiences of applying SGM on huge amounts of aerial full frame and pushbroom images demonstrate the practical applicability of all presented ideas. All of these advantages make SGM a prime choice for solving many practical stereo problems.

ACKNOWLEDGMENTS

I would like to thank Daniel Scharstein for making the Art dataset available to me, Vexcel Imaging Graz for providing the UltraCam dataset of Graz and the members of the Institute of Planetary Research at DLR Berlin for capturing and preprocessing HRSC data.

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, April-June 2002.

[2] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," in *Proceedings of the Sixth IEEE International Conference on Computer Vision*, Mumbai, India, January 1998, pp. 1073–1080.

[3] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *International Conference on Pattern Recognition*, 2006.

[4] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[5] G. Egnal, "Mutual information as a stereo correspondence measure," Computer and Information Science, University of Pennsylvania, Philadelphia, USA, Tech. Rep. MS-CIS-00-20, 2000.

[6] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," in *International Conference on Computer Vision*, October 2003.

[7] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *SIGGRAPH*, 2004.

[8] H. Hirschmüller, P. R. Innocent, and J. M. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 229–246, April-June 2002.

[9] M. Bleyer and M. Gelautz, "A layered stereo matching algorithm using image segmentation and global visibility constraints," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 3, pp. 128–150, 2005.

[10] K.-J. Yoon and I.-S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.

[11] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hirarchical belief propagation and occlusion handling," in *IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17-22 June 2006.

[12] C. Lei, J. Selzer, and Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17-22 June 2006.

[13] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *International Conference for Computer Vision*, vol. 2, 2001, pp. 508–515.

[14] J. Sun, Y. Li, S. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, USA, June 2005, pp. 399–406.

[15] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool, "A hierarchical symmetric stereo algorithm using dynamic programming," *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 275–285, April-June 2002.

[16] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, USA, June 2005, pp. 384–390.

[17] H. Hirschmüller, "Stereo vision based mapping and immediate virtual walkthroughs," Ph.D. dissertation, School of Computing, De Montfort University, Leicester, UK, June 2003.

[18] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *IEEE Conference for Computer Vision and Pattern Recognition*, vol. 1, Madison, Winsconsin, USA, June 2003, pp. 195–202.

[19] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, USA, June 2005, pp. 807–814.

[20] ——, "Stereo vision in structured environments by consistent semi-global matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17-22 June 2006.

[21] R. Gupta and R. I. Hartley, "Linear pushbroom cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 963–975, 1997.

[22] H. Hirschmüller, F. Scholten, and G. Hirzinger, "Stereo vision based reconstruction of huge urban areas from an airborne pushbroom camera (hrsc)," in *Proceedings of the 27th DAGM Symposium*, vol. LNCS 3663. Vienna, Austria: Springer, August/September 2005, pp. 58–66.

[23] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[24] E. R. Davis, *Machine Vision: Theory, Algorithms, Practicalities*, 2nd ed. Academic Press, 1997.

[25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 1–18, May 2002.

[26] C. Strecha, R. Fransens, and L. Van Gool, "Combined depth and outlier estimation in multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17-22 June 2006.

[27] F. Wewel, F. Scholten, and K. Gwinner, "High resolution stereo camera (hrsc) - multispectral 3d-data acquisition and photogrammetric data processing," *Canadian Journal of Remote Sensing*, vol. 26, no. 5, pp. 466–474, 2000.

Heiko Hirschmüller received his Dipl.-Inform. (FH) degree at Fachhochschule Mannheim, Germany in 1996, M.Sc. in Human Computer Systems at De Montfort University Leicester, UK in 1997 and Ph.D. on real time stereo vision at De Montfort University Leicester, UK in 2003. From 1997 to 2000 he worked as software engineer at Siemens Mannheim, Germany. Since 2003, he is working as computer vision researcher at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) in Oberpfaffenhofen near Munich. His research interests are stereo vision algorithms and 3D reconstruction from multiple images.