A Comparative Evaluation of SGM Variants (including a New Variant, tMGM) for Dense Stereo Matching

Sonali Patil, Tanmay Prakash, Bharath Comandur, and Avinash Kak

Abstract—Our goal here is threefold: [1] To present a new dense-stereo matching algorithm, tMGM, that by combining the hierarchical logic of tSGM with the support structure of MGM achieves 6-8% performance improvement over the baseline SGM (these performance numbers are posted under tMGM-16 in the Middlebury Benchmark V3); and [2] Through an exhaustive quantitative and qualitative comparative study, to compare how the major variants of the SGM approach to dense stereo matching, including the new tMGM, perform in the presence of: (a) illumination variations and shadows, (b) untextured or weakly textured regions, (c) repetitive patterns in the scene in the presence of large stereo rectification errors. [3] To present a novel DEM-Sculpting approach for estimating initial disparity search bounds for multi-date satellite stereo pairs. Based on our study, we have found that tMGM generally performs best with respect to all these data conditions. Both tSGM and MGM improve the density of stereo disparity maps and combining the two in tMGM makes it possible to accurately estimate the disparities at a significant number of pixels that would otherwise be declared invalid by SGM. The datasets we have used in our comparative evaluation include the Middlebury2014, KITTI2015, and ETH3D datasets and the satellite images over the San Fernando area from the MVS Challenge dataset.

1 INTRODUCTION

Many modern matching methods for calculating dense disparity maps from stereo pairs are rooted in Markov Random Field (MRF) modeling of the disparity maps, which allows for the joint probability distribution of the disparity values over a reference image to be expressed as a product of the potentials over local neighborhoods. This simplified representation of the disparity probability distributions allows an “energy” function to be defined whose minimization leads to a MAP estimate of the disparities. The energy function consists of two “costs”: the first represents the cost associated with the assignment of disparities to the individual pixel positions in the reference image and the second represents the cost associated with the assignment of two different disparities at two neighboring pixel positions. The first cost is frequently referred to as the Data Cost and the second as the Discontinuity Cost. An immediate consequence of MRF modeling is that the second cost only involves local neighborhoods around each pixel position in the reference image.

While it is relatively straightforward to construct a theoretical formalism along the lines indicated above and to end up with an energy function whose minimization would lead to a solution for the disparities, it is an entirely different matter to come up with a viable computational approach for the minimization of the energy function — especially in light of the fact that solving the energy minimization problem exactly can be shown to be NP-Hard [8]. What that means is that we can only construct approximate solutions to the minimization of the energy function. The question then becomes as to whether the quality of the resulting solutions is acceptable.

Fortunately, with the advent of SGM (Semi-Global Matching) based solutions, as first advanced by Hirschmüller [14], we now have approximate solutions that are not only computationally efficient but that also produce high quality disparity maps. SGM aggregates the Data and the Discontinuity costs along several symmetric 1D paths for each possible disparity at each pixel position in the reference image. Subsequently, the summed energy at each pixel position is calculated by simply adding up the contributions calculated along each of the paths. Finally, in a
winner-take-all strategy, we retain at each pixel that disparity which has associated with it the minimum summed energy.

Over the years, several authors have proposed modifications to SGM to improve the quality of the results obtained. Most of the variants include one or more of the following:

1) Using different data cost terms 
2) Adapting the penalty coefficients in the discontinuity-preserving smoothness term to the gradients, or learning them directly from the data 
3) Using different aggregation strategies 
4) Truncating the disparity search range using hierarchical approaches

Of the several variants that now exist, MGM by Facciolo et al. and tSGM by Rothermel are arguably among the most prominent. MGM was motivated by the perceived shortcomings associated with scanline based support regions in SGM whereas tSGM was developed to reduce the memory and runtime demands of SGM. In MGM, the scanline based star-shaped support regions are replaced by “quad-area” based support regions that allow neighboring pixel positions to exert greater influence on one another’s disparity values. In tSGM, the processing is performed in a coarse-to-fine manner that allows a coarse level to constrain the disparity range that must be searched at the next finer level — which results in substantial savings in computation. Additionally, in tSGM, using the same smoothness parameters and the same matching criterion for the corresponding pixels (specifically the Census Filter) at all levels in the hierarchy enables the coarser levels to act as “mediators” for suppressing the noise and the artifacts in the finer levels.

It should therefore be no surprise that we would want to pull together the benefits of MGM and tSGM, which is exactly what we have done in a new dense stereo matching algorithm that, out of respect for its origins, we have named tMGM.

That has dictated the following two goals for the present submission: First, to propose the tMGM alternative for dense stereo matching. And, second, to provide comparative insights into how the different SGM variants, including the new tMGM, differ with regard to the matching difficulties created by the different types of scene conditions: illumination differences between the images, the presence of untextured or weakly textured regions in the images, and the presence of repetitive patterns especially when there exist large rectification errors in the regions with repetitive patterns.

Our overall conclusion is that the hierarchical variants of the SGM algorithm perform better than their non-hierarchical counterparts for all cases represented by the stereo pairs in all four datasets we have used in this study. As we will show, the sensitivity of the SGM variants to important confounding factors varies both as a function of the variant itself and as a function of the support structure. By confounding-factors we obviously mean the illumination differences between the images, the presence of untextured or weakly textured regions in the images, and the presence of repetitive patterns (especially when there exist large rectification errors in the regions with repetitive patterns).

We now list here some of the specifics in the conclusions we have drawn from our study. In the listing shown below, we have used the notation xGM-N, where x stands for “S”, “M”, “tS”, or “tM” and the integer N for the value of the main parameter that controls the shape of the support region.

- In the presence of significant illumination differences in a stereo pair, the tMG-M-8 and the tMGM-16 algorithms are the best to use for disparity calculations. Illumination differences may be caused by the shadows cast by the objects in a scene, by the presence of pixels that correspond to specularly reflecting scene points, and, even more ordinarily, by the different lighting conditions under which the two images of a stereo pair were recorded (something that is likely to be the case on account of the different sun angles for out-of-date satellite images).
- If untextured or weakly textured regions dominate the scene, we are likely to get the most accurate disparity maps with the proposed tMGM-16 algorithm.
- If a scene is rich in repetitive structures and also has large rectification errors in such regions, we are likely to get the best disparity results with either the tSGM-8 or the tMGM-8 algorithm.

The rest of the paper is organized as follows: In Section 2 we quickly review the related work. Section 3 then briefly describes each of the main algorithms — SGM, MGM, tSGM, and tMGM. In Section 4, we present our comparative results for the four different algorithms and for different parameter choices related to the shape of the support region. In Section 5, we discuss further refinements to the tMGM algorithm and evaluation on the Middlebury Benchmark V3. We finally conclude in Section 6.

2 Related Work

The dense stereo correspondence problem is fundamental to many larger problems in robotics, remote sensing, virtual/augmented reality, etc., and as such has inspired a diverse set of algorithms. Scharstein and Szeliski created a taxonomy by which to understand them all, dividing the approaches into global, local, cooperative, and dynamic programming approaches.

Global methods, which model the entire disparity map with a 2D Markov Random Field as described in the introduction, are the primary interest of this paper. As mentioned earlier, though calculating the MAP estimate of a 2D MRF is NP-Hard, approximate algorithms have produced promising results. The work presented in uses graph-cut to find a local minimum within a known factor of the global minimum. Drory et al. show that SGM can be understood as a belief propagation based approach to optimizing the joint probability defined by the 2D MRF. In this interpretation however, it is shown that the cost term is overcounted and proposes an overcounting correction.
There is a significant body of recent work that attempts to use machine learning, often deep-learning, to solve the dense stereo correspondence problem. Many contributions, however, still utilize semi-global matching in some form, for instance as a refinement of initial disparities \cite{28} or as the model for which the parameters are learned \cite{25}. As such, a study of the underlying parameters of SGM and its variants may further improve these approaches.

The other approaches in the taxonomy presented by Scharstein and Szeliski \cite{23} generally avoid directly addressing the global distribution of disparity. Local methods consider the neighborhood of intensity values, rather than disparity values, simplifying computation \cite{15, 27}. Dynamic programming approaches consider individual scanlines, since 1D MRFs can be optimized in polynomial time \cite{6}, and often rely on post-processing to reduce the inconsistency between adjacent scanlines. Cooperative algorithms rely on iterating over local operations until convergence produces global order \cite{17}.

Public challenges like the Middlebury Stereo Challenge \cite{23} and the KITTI Vision Benchmark \cite{19} provide comparisons of a number of different stereo algorithms. While these challenges serve to determine which algorithms perform the best on a set of metrics, this paper attempts to delve deeper into how varying the underlying parameters affects performance. Note that the paper by D’Angelo \cite{9} also presents a comparison of SGM-8, SGM-16, ocSGM-16 (oc - overcounting corrected), and MGM-8 on satellite image data, but does not include MGM-16, tSGM, or tMGM.

### 3 SGM and Its Variants

The MRF based approach to the estimation of a univalued disparity over the pixel positions in the reference image of a rectified stereo pair is stated through the minimization of the following expression for “energy”:

$$E(D) = \sum_{p} \left( C(p, D_p) + \sum_{q \in N_p} V(D_p, D_q) \right)$$

where \( p = [p_x, p_y]^T \) represents a pixel position in the reference image of a stereo pair and \( D_p \) the disparity assigned to that pixel position.

The notation \( C(p, D_p) \), referred to as the Data Cost, involves comparing the reference image pixel at \( p \) with the other image pixel at \( [p_x + D_p, p_y]^T \). The notation \( V(D_p, D_q) \), called the Discontinuity Cost, denotes the cost of assigning two different disparities to two pixel positions \( p \) and \( q \) where the latter is in the neighborhood \( N_p \) associated with the former.

We want to find the disparity map that minimizes the energy function shown above.

For stereo matching, the expression for energy is expressed in the following manner \cite{14}:

$$E(D) = \sum_{p} \left( C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] \right.$$  

$$\quad + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \right)$$

Now the Discontinuity Cost, at pixel \( p \), is broken into two separate parts, one for the case when the disparity values at \( p \) and \( q \) differ by exactly 1 and, two, when they differ by more than 1. The notation \( T[\cdot] \) evaluates the truthvalue of the predicate. The two portions of the Discontinuity Cost carry the user-supplied weights \( P_1 \) and \( P_2 \). Since the minimization of the Discontinuity Cost is to ensure local smoothness of the reconstructed surfaces, we assign a small penalty \( P_1 \) to the case when the disparity varies by 1 for pixels in the neighborhood \( N_p \) of pixel \( p \). This term takes care of slanted or gently curved surfaces. And we assign a larger penalty \( P_2 \) when the disparity changes more rapidly.

### 3.1 Baseline SGM

Unfortunately, the 2D global energy minimization as given in Eq. (2) is NP-Hard \cite{6}. In the baseline SGM implementation \cite{14}, we aggregate the cost recursively along several 1D directions. The aggregated cost along a single direction \( r \) is given as follows:

$$L_r(p, d) = C(p, d) + \min \left( L_r(p - r, d), L_r(p - r, d - 1) + P_1, L_r(p - r, d + 1) + P_1, \min \left( L_r(p - r, i) + P_2 \right) \right)$$

(3)

where the notation \( L_r(p - r, d) \) means that we are now calculating a line-based estimation of the energy with the direction of the line given by the vector \( r \). Note that \( L_r(p, d) \) is initialized with the data cost \( C(p, d) \) for all pixel positions \( p \) and all possible disparities \( d \). Also, the range of values spanned by the variable \( i \) in the inner minimization in Eq. (3) is over all possible disparities. Comparing with Eq. (2), the terms \( L_r(p - r, d \pm 1) + P_1 \) are for the case of locally smooth disparity (i.e., when the disparity difference is \( \pm 1 \) along the direction \( r \)). In this case, we use the penalty \( P_1 \) which is relatively small. On the other hand, when the disparity differences are locally large, the last term, \( \min \left( L_r(p - r, i) + P_2 \right) \), kicks in. In this case, the penalty coefficient \( P_2 \) is relatively large. To allow disparity discontinuities along the edges in an image, \( P_2 \) can be adapted to the response of an edge operator. When a disparity map is not locally smooth, that is, when the disparity differences exceed \( \pm 1 \), the outer minimization in Eq. (3) will be dominated by what is returned by the inner minimization that depends on the penalty \( P_2 \).

#### Algorithm 1 Core SGM Algorithm

**Input:** \( I, I_m, d_{min}, d_{max}, r \)  
**Output:** \( D, D_m \)

1. procedure CORE SGM  
2. \( C = \text{census	extunderscore cost}(I, I_m, d_{min}, d_{max}) \)  
3. \( \text{canny} = \text{canny	extunderscore filter}(I_m) \)  
4. \( S = \text{aggregate	extunderscore cost}(C, \text{canny}) \)  
5. \( D'_s = \text{Compute	extunderscore disp}(S) \)  
6. \( \text{Median	extunderscore filter}(D'_s, 3) \)  
7. \( C = \text{census	extunderscore cost}(I, I_m, -d_{max}, -d_{min}) \)  
8. \( \text{canny} = \text{canny	extunderscore filter}(I_m) \)  
9. \( S = \text{aggregate	extunderscore cost}(C, \text{canny}) \)  
10. \( D'_m = \text{Compute	extunderscore disp}(S) \)  
11. \( \text{Median	extunderscore filter}(D'_m, 3) \)

The total aggregated cost along all the directions is given as

$$S(p, d) = \sum_r L_r(p, d)$$

(4)

A disparity map is computed from the \( S \) values by taking the argmin of \( S \) at each pixel over all possible disparities. Fig. \cite{3} and Algorithm \cite{3} summarize the steps for the core SGM algorithm.

The data term \( C \) is pre-calculated for the given disparity range. Since the data term is prone to noise, which may be due to
the presence of untextured or weakly textured regions, different radiometric characteristics of the two images, and so on, some smoothing of the data term is required. Therefore, the aggregation step includes some smoothing. Aggregation is carried out along each horizontal slice of C, with each such slice corresponding to one value of disparity, going from bottom to top in the disparity volume. Note that, on account of the vertical dependencies, the horizontal slices cannot be processed independently and the bottom-to-top order within the disparity volume is important. Aggregation along each direction \( L \) results in a volume and summing all these volumes gives us the final sum volume \( S \). Disparity map \( D_b \) is calculated by taking argmin over the sum volume \( S \). Since we are aggregating along several directions and not only along epipolar lines, we cannot detect occluded areas in this framework. Therefore, a consistency check in the form of a Left-Right-Right-Left (LRRL) check is carried out on the disparity map over the base image \( D_b \) and on the match image \( D_m \) in order to detect the invalid pixels in the base image. Note that a pixel may be invalid because of occlusion or because of mismatch.

\[
D_b p = \begin{cases} 
D_{b_p} & \text{if } |D_{b_p} - D_{m_q}| \leq 1 \\
\text{invalid} & \text{otherwise}
\end{cases} \tag{5}
\]

where \( q = [p_x + D_b(p, p_y)]^T \). The same thing is done for finding the invalid pixels in the match image simply by reversing the subscripts \( b \) and \( m \) in Eq. 5.

### 3.2 More Global Matching (MGM)

Facciolo et al. [12] observed that the star-shaped support region of SGM was not always sufficient for the neighboring pixel disparities to sufficiently constrain each other for the minimization of the Discontinuity Cost. As a result, one could sometimes see streaking artifacts in disparity maps. They got around the problem by an alternative formulation of the aggregation step which results in what may be referred to as “quad-based” support regions shown in Fig. 2.

The remarkable thing about the aggregation strategy proposed by Facciolo et al. [12] is that it requires only one more lookup in order to create area-based support regions of the sort shown in Fig. 2. This additional lookup is denoted by the symbol \( r^\perp \) in the following energy expression that is minimized in MGM:

\[
L_r(p, d) = C(p, d) + \frac{1}{2} \sum_{x \in r^\perp} \min L_x(p - x, d), \tag{6}
\]

\[
L_x(p - x, d - 1) + P_1, L_x(p - x, d + 1) + P_1, \\
\min L_x(p - x, i) + P_2
\]

Fig. 2: This additional lookup is denoted by the symbol \( r^\perp \) in the following energy expression that is minimized in MGM:

\[
L_r(p, d) = C(p, d) + \frac{1}{2} \sum_{x \in r^\perp} \min L_x(p - x, d), \tag{6}
\]

Now instead of just aggregating the costs along a given direction \( r \), we take the average of the value supplied by the direction \( r \) and a direction \( r^\perp \) which is perpendicular to \( r \). Fig 2 shows the example of 8 scanning paths in SGM versus those in MGM. Since MGM recursively aggregates the cost from the original SGM direction and a direction perpendicular to it, it covers an area as shown for each direction. Note that for each SGM direction a perpendicular along the anti-clockwise direction is selected to produce the quad-area based coverage. Algorithm 2 summarizes the steps of MGM.

#### Algorithm 2 MGM Algorithm

**Input:** \( I_b, I_m, d_{min}, d_{max}, r, r^\perp \)

**Output:** \( D_b, D_m \)

1: procedure MGM
2: \( C = \text{census}_\text{cost}(I_b, I_m, d_{min}, d_{max}) \)
3: \( \text{canny} = \text{canny}_\text{filter}(I_b) \)
4: \( S = \text{aggregate}_\text{cost}(C, r, r^\perp, \text{canny}) \)
5: \( D_b = \text{Compute}_\text{disp}(S) \)
6: \( \text{Median}_\text{filter}(D_b, 3) \)
7: \( C = \text{census}_\text{cost}(I_m, I_b, -d_{max}, -d_{min}) \)
8: \( \text{canny} = \text{canny}_\text{filter}(I_m) \)
9: \( S = \text{aggregate}_\text{cost}(C, r, r^\perp, \text{canny}) \)
10: \( D_m = \text{Compute}_\text{disp}(S) \)
11: \( \text{Median}_\text{filter}(D_m, 3) \)

### 3.3 tSGM and tMGM

A great deal of SGM computation is related to Eq. 3 being iterated over all possible disparities at every pixel. This is made computationally more efficient in tSGM by Rothermel [21] by setting dynamically calculated bounds for possible disparities in a hierarchical examination of the image. At the coarsest level,
Rothermel set the disparity search bound to \( [0, W] \) where \( W \) is the image width. Descending down the hierarchy, the disparity search bound at each pixel at level \( l \) is based on the min and the max of the disparities estimated in a window of a specified size at level \( l - 1 \). We are using index 0 for the coarsest level. This logic is used for valid pixels only. As far as invalid pixels are concerned, for each such pixel at the current level, you select the median of the disparities at all the valid pixels in a \( 31 \times 31 \) window at the next coarser level. If we denote this median value by \( m \), the disparity search bound at the current pixel is set to \( [m - \tau, m + \tau] \) for a user-specified \( \tau \). Should such a median not be available, you replace \( m \) by the average of the disparities calculated over the entire image at the next coarser level.

Rothermel has applied his implementation of tSGM to aerial images; the qualitative results obtained by the author are reported in [21]. Our experience with tSGM based on the implementation described in [21] was that it did not generalize well to the Middlebury dataset we have used in the comparative results reported in this paper. In order to rectify this issue, we have had to make certain changes to the algorithm itself[1]—while preserving the hierarchical calculation of the disparity maps. We made two key changes: (1) initialization of the disparity bounds; and (2) propagation of the disparity bounds through the hierarchy. Our initialization logic for the disparity search bounds is described in Table 1 and our strategy to propagate the disparity bounds is explained below.

For valid pixels, we have modified the disparity bound value at the current pixel at a given level by enforcing the constraint that both the upper and the lower bounds do not violate a user-specified global constraint, which for public datasets are supplied by the dataset creators. More specifically, using the estimated disparity map at the previous level in the resolution hierarchy, the disparity range at the current level is estimated at each valid pixel using the local min and max disparity values in a \( 7 \times 7 \) window around the pixel, denoted as \( w_{\text{min}} \) and \( w_{\text{max}} \), respectively. Then \( T_{\text{p}}^{\text{min}} \) and \( T_{\text{p}}^{\text{max}} \) are updated as follows:

\[
T_{\text{p}}^{\text{min}} = \max(u_{\text{p}, \text{min}}^L - \epsilon, [d_{\text{min}}/s] - \epsilon) \\
T_{\text{p}}^{\text{max}} = \min(u_{\text{p}, \text{max}}^L + \epsilon, [d_{\text{max}}/s] + \epsilon) \tag{7}
\]

where \( \epsilon \) is a relaxation parameter and \( s \) is the scale factor at the current level. The parameters \( d_{\text{min}} \) and \( d_{\text{max}} \) are supplied with the dataset images for the min and the max bounds for disparity search.

For an invalid pixel \( \text{p} \), the values of \( T_{\text{p}}^{\text{min}} \) and \( T_{\text{p}}^{\text{max}} \) are set to \( [d_{\text{min}}/s] - \epsilon \) and \( [d_{\text{max}}/s] + \epsilon \) respectively. This amounts to searching the complete range \( [d_{\text{min}}/s, d_{\text{max}}/s] \) for invalid pixels, thereby reducing artifacts and creating denser disparity maps.

Setting the disparity search range as indicated above does affect the overall disparity estimation accuracy. If the range is set too high (e.g. the image width) then we have a higher chance of getting false positives, and if it is set too low, we may lose some important scene details. More precisely, the data cost term will have greater ambiguities in a larger disparity search range, since a pixel can be matched to multiple pixels along the epipolar line. On the other hand, if it is set too low for a pixel, then the corresponding matching pixel in the second image may be outside the search bound — resulting in a mismatch or getting marked as an occluded pixel, when it’s not occluded.

As we will show in Section 3.3, implementing the SGM logic in a hierarchy not only improves the time performance for stereo matching, but also results in a disparity map that has fewer errors. That leads to the following questions: If a hierarchical implementation improves both the time and matching performance of the basic SGM algorithm, why not do the same with MGM? Why not create a new version of MGM that can benefit from the overall speedup that can be achieved when its logic is implemented in a hierarchy? Additionally, as we see in tSGM, why not boost the matching performance of the basic MGM logic by the multi-scale census transform that would get used in a hierarchical implementation?

Our implementation of tMGM is the answer to these questions. The new algorithm tMGM initializes the disparity search bounds in the same manner as in tSGM (ref. Table 1). Since the matching logic operates in a hierarchy in tMGM, the matching precision can benefit from the census transform being applied automatically on a multi-scale basis. When this is combined with the power that tMGM derives from the “quad-area” based support structure, we end up with an algorithm that is superior to all others.

The reader may think while the precision in stereo matching may be expected to go up because of a combination of “quad-area” based support and the multi-scale census transform, the price to pay for that would the computational time — on account of the same “quad-area” based support structure. As we report in Section 4.2, the time performance of tMGM is only marginally slower than that of tSGM. As we will show in that section, comparing just SGM and MGM, the latter is slower by around 8-13%. And comparing tSGM and tMGM, the latter is slower by around 6-10%. To connect these two comparisons, when comparing SGM with tSGM, the latter is faster by around 50-370%.

Fig. 3 illustrates a comparison between the search space in SGM and MGM vis-a-vis the search space for tSGM and tMGM. For SGM and MGM, the search space is a full cuboidal volume whose width and height are the same as for the images and whose depth equals the search bound for the disparities. On the other hand, for tSGM and tMGM, the width and the height remain the same except for the effect of downsampling at the coarser levels of the hierarchy, and the depth varies from pixel to pixel since,

---

1. To the best of our knowledge, there has not yet been a quantitative assessment of the performance of this algorithm.

2. We did try to download the executable that is made available by Rothermel [1], but it requires camera projection matrices that are not available for the Middlebury dataset. In addition, the executable produces the final point cloud that is a result of stereo fusion. It is not clear how to change the behavior of the executable in order to produce the disparity maps.

| Initialization | Parameter Definition |
|----------------|----------------------|
| \( D_{\text{P}} = 0 \) | Disparity at position \( \text{p} \) in base image |
| \( D_{\text{P}} = 0 \) | Disparity at position \( \text{p} \) in match image |
| \( T_{\text{P}}^{\text{min}} = [d_{\text{min}}/s] - \epsilon \) | Lower search bound at pixel \( \text{p} \) in base image |
| \( T_{\text{P}}^{\text{max}} = [d_{\text{max}}/s] + \epsilon \) | Upper search bound at pixel \( \text{p} \) in base image |
| \( T_{\text{P}}^{\text{min}} = [-d_{\text{max}}/s] - \epsilon \) | Lower search bound at pixel \( \text{p} \) in match image |
| \( T_{\text{P}}^{\text{max}} = [-d_{\text{min}}/s] + \epsilon \) | Upper search bound at pixel \( \text{p} \) in match image |

**TABLE 1.** tSGM and tMGM initialization. \( s, d_{\text{min}} \) and \( d_{\text{max}} \) are input parameters to tSGM and tMGM algorithms (see Algorithms 3 and 4) and \( \epsilon \) is a relaxation parameter.
TABLE 2: Errors averaged over all the image pairs in each dataset. MB stands for Middlebury. For the Middlebury 2014 dataset, this table only considers the image pairs with perfect rectification and with ground truth. (There are 23 such image pairs out of a total of 33.) The notation “P_E” stands for the image pairs with exposure variation in views, and the notation “P_L” is for the image pairs with illumination variations. Note that bad pixel errors only consider pixels that are valid in both D_{gt} and D_{b}. Note that for the Kitti2015 and ETH3D we used only the training images.

**Algorithm 3** tSGM Algorithm

**Input:** I_{0}, I_{m}, T_{b}^{max}, T_{b}^{min}, T_{m}^{max}, T_{m}^{min}, d_{min}, d_{max}, scale factor s (All inputs Downsampled by s)

**Output:** D_{b}, D_{m}

1: procedure tSGM ITERATIONS
2: while s ≠ 0 do
3: \[ D_{b} = SGM(I_{0}, I_{m}, T_{b}^{max}, T_{b}^{min}) \] \hspace{1cm} \( \triangleright \) For all r
4: \[ D_{m} = SGM(I_{0}, I_{m}, T_{m}^{max}, T_{m}^{min}) \] \hspace{1cm} \( \triangleright \) For all r
5: \[ D_{b} = LRRL(D_{b}, D_{m}) \]
6: \[ D_{m} = LRRL(D_{m}, D_{b}) \]
7: s = [s/2]
8: D_{b} = 2 * D_{b}
9: D_{m} = 2 * D_{m}
10: by 2
11: \[ T_{b}^{max}, T_{b}^{min} = Update(D_{b}, d_{min}, d_{max}) \] \hspace{1cm} \( \triangleright \) Update limits
12: \[ T_{m}^{max}, T_{m}^{min} = Update(D_{m}, -d_{max}, -d_{min}) \] \hspace{1cm} \( \triangleright \) Update limits

**Algorithm 4** tMGM Algorithm

**Input:** I_{0}, I_{m}, T_{b}^{max}, T_{b}^{min}, T_{m}^{max}, T_{m}^{min}, d_{min}, d_{max}, scale factor s (All inputs Downsampled by s)

**Output:** D_{b}, D_{m}

1: procedure tMGM ITERATIONS
2: while s ≠ 0 do
3: \[ D_{b} = MGM(I_{0}, I_{m}, T_{b}^{max}, T_{b}^{min}) \] \hspace{1cm} \( \triangleright \) For all r and r_{⊥}
4: \[ D_{m} = MGM(I_{0}, I_{m}, T_{m}^{max}, T_{m}^{min}) \] \hspace{1cm} \( \triangleright \) For all r and r_{⊥}
5: \[ D_{b} = LRRL(D_{b}, D_{m}) \]
6: \[ D_{m} = LRRL(D_{m}, D_{b}) \]
7: s = [s/2]
8: \[ D_{b} = 2 * D_{b} \]
9: \[ D_{m} = 2 * D_{m} \]
10: by 2
11: \[ T_{b}^{max}, T_{b}^{min} = Update(D_{b}, d_{min}, d_{max}) \] \hspace{1cm} \( \triangleright \) Update limits
12: \[ T_{m}^{max}, T_{m}^{min} = Update(D_{m}, -d_{max}, -d_{min}) \] \hspace{1cm} \( \triangleright \) Update limits
at each level, it is set according to the disparity map at the next
coarser level. At the coarsest level, the depth is constant and set
according to Table 1.

4 COMPARATIVE EVALUATION

This section is organized as follows. We first describe the datasets
that we use for our exhaustive evaluation. We then present the
results of a comparative evaluation of the different algorithms on
the Middlebury2014 [22]. KITTI2015 [18] and ETH3D [24] two-
view stereo datasets using the Middlebury V3 benchmark. This
is followed by a separate subsection on our experiments using
satellite images from the MVS Challenge dataset [7].

4.1 Datasets and Implementation

For the Middlebury2014 dataset, we use the 23 full resolution (5-
6 MP) image pairs for which the groundtruth is available. These
images mainly feature indoor scenes. For each image pair, there
are six ways of pairing the two images: (1) with perfect recti-


Fig. 3: Search volume in SGM and MGM vis-a-vis that in tSGM
and tMGM. Whereas SGM and MGM use images at their original
resolution, tSGM and tMGM use image resolution hierarchies in
which each coarse level is used to estimate the disparity bounds to
be used for the search in the next finer level.

fication; (2) with imperfect rectification; (3) perfect rectification
with differences in illumination; (4) imperfect rectification with
differences in illumination; (5) perfect rectification with exposure
differences; and, finally, (6) imperfect rectification with exposure
differences. Imperfect rectification means that the rows are not
aligned correctly due to, say, camera calibration errors.

The KITTI2015 dataset consists of 200 image pairs of outdoor
scenes for which we have the groundtruth. This dataset provides
semi-dense groundtruth disparity maps over roughly 30% of all the
pixels. The pixels where the groundtruth is available are generally
in the bottom half of the images on account of the fact that images
are generated with vehicle mounted cameras.

The ETH3D dataset consists of 27 image pairs with dense
ground truth disparity maps, featuring both indoor and outdoor
scenes. All 27 image pairs that we have used are from the training
dataset for which the groundtruth is available.

The comparative results we show in this section are all based
on our own implementations of all four matching algorithms.
Having our own implementation enables us to use consistent code
for the cross-comparison evaluation and to easily experiment with
the different parameters that specify the shape of the support
structure. For sanity check, we have compared the results we
obtain with our implementation with those produced by the code
provided by Faccio for SGM-8 and MGM-8 [2]. For SGM-8 and
MGM-8, we obtain similar results in both cases.

We use the overcounting correction proposed by Drory et al.
in all the algorithms in our evaluation. Referring to Eq. 4 in the
baseline SGM algorithm, the data cost is overcounted in the
sum volume $S$ as it is a part of each $L_r$. The correction consists
of updating the volume $S$ as follows

$$S(p, d) = S(p, d) - (N - 1)C(p, d)$$

where $N$ is the number of directions, 8 or 16 in our case.

All algorithms use the Hamming distance based on the census
transform over $5 \times 5$ neighborhoods after it is normalized by the
number of image color channels. For all implementations of SGM
and MGM, a median filter of size $3 \times 3$ is applied to $D_r'$ and $D_m'$.
For tSGM and tMGM, the median filter is applied at each level.

Referring to Section 3, we notice that these algorithms have
additional parameters that need to be chosen with some care and
intuition, particularly $P_1$ and $P_2$. We set $P_1 = 24$ and use the
Canny filter response to adaptively update $P_2$. When the filter
response is 1, we use $P_2 = 27$ and when there is no edge, we
use $P_2 = 96$. These values were chosen empirically to get best
results for the baseline SGM-N algorithm. Additionally, for tSGM
and tMGM, we set $s = 8$ and $\epsilon = 4$ where $s$ is the scale factor
defined in Algorithms 3 and 4 and $\epsilon$ is the relaxation parameter
used in Eq. 7.

4.2 Results

In this section, we will first present the metrics we have used for
a quantitative assessment of the disparities as calculated by the
different SGM variants. That will be followed by a presentation
of the quantitative results obtained with the metrics; these will be
conveyed through the numbers shown in Table 2.

Since gross numbers never tell the whole story about how an
algorithm may have performed with respect to the scene and
illumination conditions, the subsections that follow take up each
of those conditions and show the artifacts and the errors given rise
to by those conditions.

For quantitative evaluation, we use the following metrics:
invalid pixel error, bad pixel error, total error, and average error
as defined in the Middlebury V3 Benchmark [7]. The invalid pixel
error is defined as the percentage of the valid pixels in the ground
truth disparity map $D_{gt}$, that are marked as invalid in the estimated
disparity map $D_h$. The bad pixel error is defined as the percentage
of the valid pixels in $D_{gt}$ such that $|D_{hp} - D_{gtp}| > \delta$ for a given
threshold $\delta$. Note that a pixel that is marked invalid in $D_h$
but valid in $D_{gt}$ is not counted in the bad pixel error because it has
already been counted in the invalid pixel error. The total error is
simply the sum of the invalid pixel error and the bad pixel error.
The average error is defined as $\frac{1}{V} \sum_{p} |D_{hp} - D_{gtp}|$ where $V$ is
the set of pixels that are valid in both $D_h$ and $D_{gt}$.

Table 2 reports the errors defined as above, for $\delta \in \{1, 2, 3\}$
pixels, for the Middlebury2014 dataset with perfect rectification
(with and without exposure or illumination variations), the
KITTI2015 and ETH3D datasets. The imperfect rectification case
for the Middlebury2014 dataset is discussed separately in Subsec-

4.2.3.

With regard to the invalid pixel error, as shown in Table 2 tMGM-16 produces the best results in all cases. Note that although SGM-8 produces a smaller bad-pixel-error than the other algorithms, this error does not account for invalid pixels. With regard to the total error, which does account for the invalid pixels, tMGM-16 is superior for all cases except for the KITTI2015
dataset. The KITTI2015 dataset differs from the other datasets in certain aspects: the groundtruth disparity maps are semi-dense and the images are smaller and contain only outdoor scenes. Another possible reason could be that the values we chose for P1 and P2 parameters using the Middlebury2014 dataset. It could very well be that tuning P1 and P2 to the KITTI2015 dataset will bridge the gap between tMGM-16 and tSGM-8 for that dataset.

Overall, tMGM-16 does indeed benefit from the combined use of the hierarchical approach, “quad-area” based support structures, and 16 directions along which the costs as given by Eq. 6 are aggregated. For all the datasets, the hierarchical versions tSGM-N and tMGM-N achieve lower total error when compared to their non-hierarchical counterparts. Interestingly, tSGM-8 produces the lowest average disparity errors in all cases except for the ETH3D dataset for which it performs comparably to the best. The average errors for tMGM-16 are quite close to tSGM-8.

| Dataset      | SGM-8 | MGM-8 | tSGM-8 | tMGM-8 |
|--------------|-------|-------|--------|--------|
| Perfect (M)  | 104.1 | 113.3 | 69.9   | 74.5   |
| KITTI2015    | 7.5   | 8.6   | 4.0    | 4.3    |
| ETH3D        | 16.4  | 18.3  | 4.4    | 4.9    |

TABLE 3: Average runtime in seconds

Table 3 presents the run time for each algorithm. The time shown is averaged over 10 image pairs for each dataset. As expected, MGM-8 is slower than SGM-8, and, due to its hierarchical nature, tSGM-8 is faster than SGM-8. Correspondingly, tMGM-8 is slightly slower than tSGM-8. Similar observations can be expected for the xGM-16 variants. For all algorithms mentioned in this paper, we have C++ based implementations with OpenMP support for CPU parallelization. The runtimes we report are recorded on a machine with 60GB RAM, and the Intel Broadwell-IBRS @ 2.60 GHz processor with 13 physical cores.

In the subsections that follow, we take up each of the confounding conditions and show the artifacts that are caused by each condition for the different variants of SGM. As mentioned earlier, these confounding conditions are:

- illumination variations between the two views, scenes with untextured or weakly textured regions and repetitive patterns in the presence of imperfect rectification. We illustrate each case with examples.

(a) Right Image  (b) Left Image  (c) GT Disparity

Fig. 4: An example from the Middlebury2014 dataset for scenes with illumination variations in two views. See Figs. 5 and 6 for the xGM-8 and xGM-16 results.

4.2.1 Illumination Variations

In Table 2, “MB2014 (P, L)” represents the image pairs with perfect rectification in the Middlebury2014 dataset that exhibit illumination differences caused by strong shadows and by the presence of specularly reflecting surfaces. As mentioned earlier in Section 4.2, tMGM-16 produces the lowest total error for this case. In fact, we observe that all hierarchical variants, namely tSGM-N and tMGM-N benefit from using the multi-scale census data cost when compared to their corresponding non-hierarchical counterparts. We support this observation with comparisons in Figs. 5 and 6. Upon inspecting the highlighted region that contains strong shadows, we see that the errors are sparser for tMGM-16 which also benefits from the “quad-based” support structure.

4.2.2 Untextured or Weakly Textured Regions

The ETH3D dataset mostly contains scenes with untextured or weakly textured regions and Fig. 7 shows one such example. The estimated disparity maps and error masks obtained with the different algorithms are shown in Figs. 9 and 10. In this example, the weakly textured regions correspond to the walls and are highlighted using green boxes. The benefits of the multi-scale census data cost are once again observed for the tSGM and tMGM algorithms. Moreover since the Discontinuity Cost term is used at every level, we get denser disparity maps using the hierarchical algorithms. The cumulative effect is that tSGM and tMGM produce denser disparity maps compared to SGM and MGM for untextured or weakly textured regions. For strongly textured areas, all algorithms are comparable.

4.2.3 Imperfect Rectification

In this section, we focus on the image pairs from the Middlebury2014 dataset with imperfect rectification. Table 4 reports the disparity errors averaged over all the image pairs with imperfect rectification (with and without illumination or exposure variations). In this table, MB2014(I) corresponds to image pairs with no exposure and illumination variations, MB2014(I_L) corresponds to image pairs with illumination differences and MB2014(I_E) corresponds to image pairs with exposure variations. We observe that just like in the case with perfect rectification, tMGM-16 produces the lowest total errors in all cases. It’s interesting to note that tMGM-8 gives the lowest average error for the MB2014(I) and MB2014(I_E) cases, whereas tSGM-16 gives the lowest average error for the remaining case of MB2014(I_L). Nevertheless, similar to the perfect rectification case, the average error for tSGM-8 and tMGM-16 remain close to the lowest average errors in all cases.

The aforementioned observations for the cases of untextured and illumination variations with perfect rectification hold true for the cases with imperfect rectification as well. However, special attention must be paid to scenes with repetitive patterns. With perfect rectification, all algorithms perform comparably in regions with repetitive patterns. However, rectification errors in such regions can result in ambiguities in disparity estimation. Repetitive patterns in images can occur at both the pixel level and structural level. An example of the latter would be a grid of roads in an aerial image of an urban area. We are concerned with pixel level repetitive patterns as shown in Fig. 13 — the patterns are in the carpet on the floor as should be discernible in the magnified view of a small portion of the carpet.

The corresponding results for xGM-8 and xGM-16 are shown in Figs. 14 and 15 respectively. We can see that with imperfect rectification, the hierarchical approaches are more tolerant to stereo rectification errors in regions with repetitive patterns.
Fig. 5: Middlebury2014 results using xGM-8 in the presence of illumination differences. A region with strong shadows is highlighted in magenta. The top row shows the estimated disparity maps and the bottom row shows the corresponding binary error masks, where a pixel is set to foreground if the disparity error (> 1) or if it is invalid in $D_b$ and valid in $D_{gt}$.

Fig. 6: Middlebury2014 results using xGM-16 in the presence of illumination differences. A region with strong shadows is highlighted in magenta. The top row shows the estimated disparity maps and the bottom row shows the corresponding binary error masks, where a pixel is set to foreground if the disparity error (> 1) or if it is invalid in $D_b$ and valid in $D_{gt}$.

Fig. 7: An example for scenes with weakly textured regions from the ETH3D dataset.

Fig. 8: An example of an out-of-date image pair from the MVS challenge dataset. The yellow box highlights a region with strong shadows, varying scene contents such as parked cars and noise due to trees.

4.3 Results on Satellite Images

3D stereo reconstruction from multi-date satellite images is quite challenging because out-of-date stereo pairs can vary significantly in illumination, scene content, sun angle, weather conditions etc. To understand how the different SGM variants perform under such conditions, we evaluate our comparison on 50 satellite images collected over the San Fernando area. These WorldView-3 images were originally provided as part of the MVS Challenge [7] and are now available from the SpaceNet dataset [3]. An example pair of images is shown in Fig. 8. For evaluation purposes, we focus on four areas of interest – Explorer, MasterProvisional1, MasterProvisional2 and MasterProvisional3 covering areas of 0.45 sq.km, 0.13 sq.km, 0.14 sq.km and 0.1 sq.km respectively.

Since this dataset was intended for 3D reconstruction, ground truth disparity maps are unavailable. However, LiDAR ground truth is provided. Therefore for a quantitative evaluation, we generate DSMs for each SGM variant and then use the completeness, median error and RMSE metrics defined in [7]. We borrowed modules for rectification and triangulation from the s2p pipeline [13] and plugged in our own implementations of the SGM variants to construct DSMs. Table 5 reports errors using three metrics.
Fig. 9: Stereo matching results with xGM-8 algorithms. Green boxes highlight weakly textured regions. The top row shows the estimated disparity maps and the bottom row shows the corresponding binary error masks, where a pixel is set to foreground if the disparity error (> 1) or if it is invalid in $D_b$ and valid in $D_{gt}$.

![Stereo matching results with xGM-8 algorithms](image1)

Fig. 10: Stereo matching results with xGM-16 algorithms. Green boxes highlight weakly textured regions. The top row shows the estimated disparity maps and the bottom row shows the corresponding binary error masks, where a pixel is set to foreground if the disparity error (> 1) or if it is invalid in $D_b$ and valid in $D_{gt}$.

![Stereo matching results with xGM-16 algorithms](image2)

| Algo  | Img         | Avg Disp Err (pix) | InvDispErr (%) | > 1 pixel error | > 2 pixel error | > 3 pixel error |
|-------|-------------|--------------------|----------------|-----------------|-----------------|-----------------|
|       |             |                    |                | Bad Pix Err (%) | Bad Pix Err (%) | Bad Pix Err (%) |
|       |             |                    |                |                 |                 |                 |
| SGM-8 | MB2014 (I)  | 11.47              | 42.44          | 16.71           | 59.25           | 10.73           |
|       | MB2014 (I E)| 13.09              | 46.18          | 17.26           | 62.44           | 10.44           |
|       | MB2014 (I L)| 23.33              | 58.59          | 17.23           | 75.82           | 11.90           |
| MGM-8 | MB2014 (I)  | 14.17              | 39.74          | 18.03           | 57.77           | 12.23           |
|       | MB2014 (I E)| 16.51              | 42.50          | 18.69           | 61.20           | 14.38           |
|       | MB2014 (I L)| 28.95              | 55.52          | 19.64           | 75.16           | 14.38           |
| SGM-16| MB2014 (I)  | 10.20              | 36.51          | 19.34           | 53.85           | 11.97           |
|       | MB2014 (I E)| 11.71              | 39.16          | 19.97           | 59.13           | 11.74           |
|       | MB2014 (I L)| 20.59              | 52.04          | 20.01           | 72.05           | 13.25           |
| MGM-16| MB2014 (I)  | 10.37              | 34.01          | 19.10           | 53.11           | 12.01           |
|       | MB2014 (I E)| 12.25              | 36.75          | 20.03           | 56.78           | 11.92           |
|       | MB2014 (I L)| 22.59              | 49.35          | 20.94           | 70.29           | 14.28           |
| tSGM-8| MB2014 (I)  | 6.83               | 25.20          | 24.19           | 49.40           | 14.60           |
|       | MB2014 (I E)| 7.76               | 27.01          | 25.50           | 52.51           | 14.75           |
|       | MB2014 (I L)| 17.50              | 37.76          | 28.60           | 66.35           | 19.47           |
| tMGM-8| MB2014 (I)  | 6.67               | 23.26          | 24.43           | 47.69           | 14.87           |
|       | MB2014 (I E)| 7.73               | 25.02          | 25.76           | 50.78           | 14.97           |
|       | MB2014 (I L)| 19.13              | 36.35          | 29.28           | 65.64           | 20.27           |
| tSGM-16| MB2014 (I) | 7.91               | 22.19          | 26.66           | 48.86           | 16.56           |
|       | MB2014 (I E)| 8.90               | 23.62          | 28.17           | 51.79           | 16.89           |
|       | MB2014 (I L)| 17.41              | 32.39          | 32.16           | 64.56           | 22.15           |
| tMGM-16| MB2014 (I) | 7.06               | 20.50          | 25.85           | 46.35           | 15.80           |
|       | MB2014 (I E)| 7.95               | 21.87          | 27.46           | 49.32           | 16.08           |
|       | MB2014 (I L)| 17.80              | 31.23          | 31.87           | 63.10           | 21.89           |

TABLE 4: Errors averaged over the 23 image pairs with groundtruth and with imperfect rectification in the Middlebury2014 dataset. M stands for Middlebury2014. The suffix "_E" is for the image pairs with exposure variation between views, and the suffix "_L" is for the image pairs with illumination variations. Note that bad pixel errors only consider pixels that are valid in both $D_{gt}$ and $D_b$. 


Table 1 shows the comparison of different algorithms. SGM-8, MGM-8, tSGM-8, tMGM-8, and GT DSM are compared. The white box in each figure corresponds to the yellow box in Fig. 8.

Table 2 shows the comparison of different algorithms. SGM-16, MGM-16, tSGM-16, tMGM-16, and GT DSM are compared. The white box in each figure corresponds to the yellow box in Fig. 8.

Fig. 11: DSM generated using xGM-8 algorithms for the satellite image pair from Fig. 8. The white box corresponds to the yellow box in Fig. 8.

Fig. 12: DSM generated using xGM-16 algorithms for the satellite image pair from Fig. 8. The white box corresponds to the yellow box in Fig. 8.

Fig. 13: An example for scenes with repetitive patterns. Figs. 14 and 15 show results using the xGM-8 and xGM-16 algorithms, respectively. Large y-alignment errors are highlighted with magenta boxes in 13c. The y-alignment errors in these regions vary approximately from 0.8 to 2.6 pixels.

namely completeness, median error and RMSE. The completeness metric measures the fraction of 3D points with Z (height) error less than 1 meter. Higher completeness score is desirable. For reporting median and RMSE errors, we only consider points that are valid in both the ground truth and generated DSMs.

From Table 5, we observe that tMGM-16 and tSGM-16 outperform the other algorithms on all regions in terms of the median error and RMSE. With regard to completeness, tMGM-16 has the best performance on MasterProvisional2 region and produces results comparable to the maximum on the other regions. Note that we do not fuse the pairwise DSMs but rather compare each pairwise DSM with LiDAR ground truth. The metrics are averaged over 25 pairs. Therefore one cannot expect our metrics to match the results in [7] which are computed using dense fused DSMs. As a future work, we would like to investigate how these metrics relate to the Middlebury V3 metrics that directly measure error in disparities. Figs. 11 and 12 show reconstructed DSMs for the image pair in Fig. 8. While the DSMs produced by SGM and MGM are noisy, tSGM-16 and tMGM-16 produce denser and cleaner DSMs even in the untextured and shadowy regions. In order to get denser and more accurate DSMs, one would need to fuse multiple such DSMs obtained from different stereo pairs.

5 Enhancements to tMGM-16 for the Middlebury Benchmark V3 Evaluation

The discussion so far has focused on comparing the different variants of SGM on the basis of whether or not the implementation
was hierarchical, and the type of the support structure used. For this comparison, we intentionally did not apply any special post-processing to the output disparities. Our concern was that post-processing of any kind could mask the consequences of the choices made for the comparisons.

Having established the superiority of tMGM-16 for the Middlebury2014 and the ETH3D datasets, our goal in this section is to add enhancements to this algorithm in order to further improve its performance. These improvements include:

- post-processing for peak removal and sub-pixel refinement at every level of the hierarchy
- replacing the median filter with a joint bilateral filter
- a more robust way to set the disparity search range for valid pixels
- new formulas for the weights $P_1$ and $P_2$
- and, finally, dealing with the holes in the final disparity map.

For the sub-pixel refinement we fit a parabola to the aggregated cost function at $\{-1, 0, +1\}$ disparity values and the location of the minima for this parabola yields a sub-pixel shift. The peak removal is carried out with the same logic as presented in [14].

For updating the weights $P_1$ and $P_2$ we have extended the approach presented in [29] as we now explain. We first define the intensity differences $D_1$ and $D_2$ in the base and the match images as follows:

$$D_1 = |I_b(p) - I_b(p-r)| \quad \text{and} \quad D_2 = |I_m(p+d) - I_m(p + d - r)|$$

These are then used to set the values for the weights $P_1$ and $P_2$ according to the following conditional logic that depends on the direction of the path in the support structure. For pixels in the horizontal paths, $P_1$ and $P_2$ are changed using the formula:

$$(P_1, P_2) = \begin{cases} (P_1, P_2) & \text{if } D_1 \leq \tau \text{ and } D_2 \leq \tau \\ \left(\frac{P_1}{Q_1}, \frac{P_2}{Q_2}\right) & \text{if } D_1 \geq \tau \text{ and } D_2 \geq \tau \\ \left(\frac{P_1}{Q_1}, \frac{P_2}{Q_2}\right) & \text{otherwise} \end{cases}$$

For updating the weights $P_1$ and $P_2$ with and without the improvements described in Section 5.
where \( Q_2 > Q_1 \). On the other hand, for pixels in the vertical paths, following the recommendation made in \( \frac{29}{29} \), after computing \( P_1 \) and \( P_2 \) according to the formula shown above, we set \( P_1 = P_2 / V \). And, when a pixel is on a diagonal path in the support structure, we set \( P_1 = P_1 \times \sqrt{1 + \frac{1}{\tau^2}} \). In all our experiments we have set \( V = 1.25 \), \( \tau = 20 \), \( Q_1 = 3 \), and \( Q_2 = 6 \).

That brings us to the issue of how to set the search bounds robustly for disparity estimation. Since disparity estimation at coarser levels can be noisy, instead of taking the maximum and the maximum as the search bounds at valid pixels as explained in Eq. \( \frac{7}{7} \), we take the 5th percentile as the lower bound and the 95th percentile as the upper bound within a 41 x 41 window centered at each valid pixel.

Finally, we use discontinuity preserving interpolation \( \frac{14}{14} \) to fill holes in the final disparity map.

In addition to the post-processing refinements described above, we also correct for the stereo rectification errors since, according to the Middlebury benchmark website, a majority of the image pairs in the test dataset possess stereo rectification errors\( \frac{1}{1} \). The stereo rectification errors are corrected by using RANSAC to fit a linear model to the coordinates of the correspondences between the key points extracted from the two images. The linear model, in turn, yields the offset that brings the two images into vertical alignment. This step is carried out as a pre-processing step.

Figure 16 shows the effect of all the above mentioned improvements to tMGM-16 with respect to just one metric — the total error for \( \delta > 1 \) as defined in Section 4.2.

### 5.1 Baseline SGM versus tMGM-16 in the Middlebury Benchmark V3 Evaluation

The tMGM-16, refined as discussed above, was submitted for evaluation to the Middlebury Benchmark V3. The benchmark returned the results shown in Table 6. The column headings are the \( \delta \) values for which the weighted average errors were computed using 15 test image pairs. The benchmark computes a large number of evaluation results using different metrics. Table 6 shows the results for the total errors at the non-occluded pixels. The first row of the results are for the baseline SGM as submitted by Hirschmuller \( \frac{14}{14} \). As can be seen in the table, tMGM-16 achieves an improvement of 6-8% on the average over the baseline SGM.

| Method       | 0.5px | 1px  | 2px  | 4px  |
|--------------|-------|------|------|------|
| SGM-HH       | 35.5  | 35.5 | 25.1 | 19.0 |
| tMGM-16      | 48.2  | 27.8 | 17.3 | 11.5 |

**TABLE 6**: Official results for non-occluded pixels from the Middlebury benchmark. SGM-HH results are for the optimized baseline SGM algorithm as submitted by \( \frac{14}{14} \).

### 6 DEM-Sculpting for Satellite Stereo Matching

In this Section, we present a case study of stereo reconstruction using a few multi-date satellite images that are acquired under the limited range of azimuth and obliquity angles of a satellite as well as the sun. We fixed the stereo matching algorithm as tSGM-8. Fig. 17 shows the plots for satellite and sun angle distribution of the available number of WV3 images. The outer circle marked with E, NE and so on represents azimuth angle and the concentric circles represent the obliquity angle. Note that from the plot of sun angle distribution, all the images are captured around the sun rise time and the obliquity angles are very large that means the elevation angles are very low. Additionally, the region contains a tall mountain with small mud houses on its elevation which introduces additional stereo matching challenges due to large disparity search range and untextured regions (see Fig. 18). Out of thirteen unique WV3 images two were heavily occluded due to clouds. Its widely known that a wider baseline yields poor results in stereo matching and a very narrow baseline yields poor results in triangulation. Therefore, additional thresholds are applied to filter out stereo pairs based on view-angle and acquisition time differences. After all the pre-processing, only 17 stereo pairs were available to generate the fused DSM for the Area of Interest (AOI) in Kabul.

A common practice to estimate the initial global disparity search bound is to use a set of world points sampled either on a 3D grid generated using the extents available in Rational Polynomial Coefficient (RPC) \( \frac{10}{10} \) camera model or using a low resolution SRTM (Shuttle Radar Tomography Mission) DEM (Digital Elevation Model), e.g., \( \frac{13}{13} \). For the rest of the discussion, we assume the RPC parameters are corrected using

5. In general, satellite or pushbroom camera parameters are represented as RPC model.

---

**TABLE 5**: Completeness, median error and RMSE averaged over 25 satellite image pairs from each region

| Algo   | Dataset                  | Completeness (%) | Median (m) | RMSE (m) |
|--------|--------------------------|------------------| -----------|----------|
| SGM-8  | Explorer                 | 66.80            | 0.59       | 11.24    |
|        | Master Provisional1      | 65.64            | 1.22       | 7.86     |
|        | Master Provisional2      | 51.33            | 2.23       | 10.66    |
|        | Master Provisional3      | 35.47            | 5.90       | 20.31    |
| MGM-8  | Explorer                 | 65.31            | 0.72       | 11.71    |
|        | Master Provisional1      | 65.20            | 1.22       | 8.18     |
|        | Master Provisional2      | 49.16            | 3.10       | 11.36    |
|        | Master Provisional3      | 34.54            | 6.09       | 20.84    |
| SGM-16 | Explorer                 | 69.49            | 0.48       | 10.41    |
|        | Master Provisional1      | 67.30            | 0.62       | 7.14     |
|        | Master Provisional2      | 54.60            | 1.68       | 9.80     |
|        | Master Provisional3      | 37.22            | 5.61       | 19.55    |
| MGM-16 | Explorer                 | 68.58            | 0.50       | 10.76    |
|        | Master Provisional1      | 67.07            | 0.62       | 7.38     |
|        | Master Provisional2      | 53.68            | 1.89       | 10.17    |
|        | Master Provisional3      | 36.70            | 5.74       | 19.97    |
| tSGM-8 | Explorer                 | 70.93            | 0.46       | 8.66     |
|        | Master Provisional1      | 67.38            | 0.51       | 6.11     |
|        | Master Provisional2      | 58.43            | 1.08       | 7.22     |
|        | Master Provisional3      | 38.47            | 5.11       | 18.14    |
| tMGM-8 | Explorer                 | 70.57            | 0.46       | 8.86     |
|        | Master Provisional1      | 67.27            | 0.51       | 6.32     |
|        | Master Provisional2      | 58.23            | 1.08       | 7.41     |
|        | Master Provisional3      | 38.09            | 5.21       | 18.54    |
| tSGM-16| Explorer                 | 70.90            | 0.46       | 8.53     |
|        | Master Provisional1      | 67.23            | 0.51       | 6.11     |
|        | Master Provisional2      | 58.74            | 1.06       | 7.14     |
|        | Master Provisional3      | 38.53            | 5.09       | 18.06    |
| tMGM-16| Explorer                 | 70.89            | 0.46       | 8.43     |
|        | Master Provisional1      | 67.17            | 0.51       | 5.97     |
|        | Master Provisional2      | 58.60            | 1.06       | 7.08     |
|        | Master Provisional3      | 38.79            | 5.03       | 17.78    |
a commonly used approach, based on bundle adjustment \[5\]. In other words, image-to-image alignment is performed with sub-pixel accuracy. Fig. 19a shows the intuition behind estimating global disparity search bounds using DEM and the fused DSM output using 17 stereo pairs. Clearly, the global search bound is too large in flat region as compared to the mountain region causing very large spikes in the flat region.

We propose a novel DEM-sculpting logic to estimate tighter per-pixel search bounds to alleviate the aforementioned problem. We now summarize the algorithm in the following steps.

1) Add upper and lower offsets to DEM, estimated per RPC camera model. An upper offset $u_o$ is estimated as

$$u_o = 0.5(h_{\text{RPC}}^{\text{max}} - h_{\text{DEM}}^{\text{max}})$$

where $h_{\text{DEM}}^{\text{max}}$ is the maximum DEM height and $h_{\text{RPC}}^{\text{max}}$ is the maximum height for a given RPC camera model. Similarly, a lower offset is estimated as

$$l_o = 0.15(h_{\text{RPC}}^{\text{min}} - h_{\text{DEM}}^{\text{min}})$$

where $h_{\text{RPC}}^{\text{min}}$ and $h_{\text{DEM}}^{\text{min}}$ are minimum heights in RPC and DEM, respectively. We obtain the two DEMs after adding $u_o$ and $l_o$ to the original DEM (see Fig. 19d).

2) Generate a set of sparse points from a 2D grid in stereo rectified reference image.

3) Map the sampled points into the corresponding reference unrectified view. Note that we use the approach proposed by Oh et al. \[20\] to stereo rectify the satellite images. The approach by Oh et al. produces a rectification $xy$-map which maps point locations in rectified view to the corresponding unrectified view. We approximate the inverse mapping using a KDTree-based nearest neighbor search and inverse distance weighted interpolation.

4) Backproject the sampled points from the unrectified reference view to the DEMs obtained by adding $u_o$ and $l_o$ to obtain two sets of world points.

5) Forward project the world points to the unrectified secondary view and then mapped to the rectified secondary view using the inverse mapping as explained in Step (3).

6) Steps 1-5 give us the sparse correspondences in a reference image.
and the secondary rectified views. By applying the definition of disparity we get lower and upper disparity search bounds.

7) The holes in sparse search bound maps are then filled using the nearest neighbor interpolation.

Note that in theory, one could start with all the points in a reference rectified views, however the backprojection using RPC is computationally expensive. We noticed that for stereo matching purpose coarse search bound estimation offers a good compromise between speed and accuracy. Fig. 19d shows the fused DSM result using DEM-sculpting logic for the initial disparity search bounds.

7 CONCLUSIONS

Our results show that hierarchical approaches are not just faster but are also more accurate than their non-hierarchical counterparts for all datasets. This improvement stems from using the multi-scale census transform and the Discontinuity Cost term at every level in the hierarchy. The performance of tMGM-16 is the best or close to the best for all datasets except the KITTI2015 dataset. Possible reasons and solutions to improve the performance for the KITTI2015 dataset have been discussed in Section 4.2. In general, even in the presence of significant illumination differences, untextured or weakly textured regions, repetitive structures and imperfect rectification, the tMGM-16 algorithm is the best to use for disparity calculations. The combination of the hierarchical approach, “quad-area” based support regions and 16 directions in tMGM-16 produces dense and accurate disparity maps. We present a novel DEM-sculpting approach to estimate initial disparity search bounds which further supports our observations about the effects of tighter search bounds on the overall matching accuracy.

ACKNOWLEDGMENTS

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior / Interior Business Center (DOI/IBC) contract number D17PC00280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

REFERENCES

[1] Middlebury stereo evaluation - version 3. http://vision.middlebury.edu/stereo/submit3/

[2] More global matching on github. https://github.com/gfacciolo/mgm

[3] Spacenet on amazon web services (aws) datasets the spacenet catalog. last modified april 30, 2018. https://spacenetchallenge.github.io/datasets/datasetHomePage.html. Accessed on October 25, 2018.

[4] Sure- photogrammetric surface reconstruction from imagery. https://ifpwww.ifp.uni-stuttgart.de/publications/software/sure/index.en.html

[5] Bundle Adjustment-A Modern Synthesis. Vision Algorithms, 54099:298–372, 2000.

[6] S. Birchfield and C. Tomasi. Depth Discontinuities by Pixel to Pixel Stereo. International Journal of Computer Vision, 35(3):1–58, 1999.

[7] M. Bosch, Z. Kurtz, S. Hagstrom, and M. Brown. A multiple view stereo benchmark for satellite imagery. In 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pages 1–9, Oct 2016.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on pattern analysis and machine intelligence, 23(11):1222–1239, 2001.

[9] P. D’Angelo. Improving semi-global matching: cost aggregation and confidence measure. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, XLII(1):299–304, 2016.

[10] G. Dial and J. Grodecki. RPC replacement camera models. Proc. ASPRS Annual Conference, Baltimore, pages 1–5, 2005.

[11] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht. Semi-global matching: a principled derivation in terms of message passing. In German Conference on Pattern Recognition, pages 43–53. Springer, 2014.

[12] G. Facciolo, C. De Franchis, and E. Meinhardt. MGM: A significantly more global matching for stereovision. In BMVC, pages 90–1, 2015.

[13] G. Facciolo, C. D. Franchis, and E. Meinhardt-Llopis. Automatic 3d reconstruction from multi-date satellite images. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1542–1551, July 2017.
[14] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[15] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, Sep 1994.

[16] C. LeGendre, K. Batsos, and P. Mordohai. High-resolution stereo matching based on sampled photoconsistency computation. In *British Machine Vision Conference*, 2017.

[17] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[18] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[19] M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[20] J. Oh, W. H. Lee, C. K. Toth, D. A. Grejner-Brzezinska, and C. Lee. A piecewise approach to epipolar resampling of pushbroom satellite images based on rpc. *Photogrammetric Engineering & Remote Sensing*, 76(12):1353–1363, 2010.

[21] M. Rothermel. Development of a SGM-based multi-view reconstruction framework for aerial imagery. Dissertation, University of Stuttgart, 2016.

[22] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[23] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[24] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] A. Seki and M. Pollefeys. SGM-Nets: Semi-global matching with neural networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, pages 6640–6649, 2017.

[26] S. N. Sinha, D. Scharstein, and R. Szeliski. Efficient high-resolution stereo matching using local plane sweeps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1582–1589, 2014.

[27] K. J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006.

[28] J. Zbontar and Y. Lecun. Stereo Matching by Training a Convolutional Neural. *Journal of Machine Learning Research*, 17:1–32, 2016.

[29] J. Zbontar, Y. LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.