the matching cost by performing two-pass aggregation using two orthogonal 1D windows [5], [6], [8]. The two-pass method first aggregates matching costs in the vertical direction, and then computes a weighted sum of the aggregated costs in the horizontal direction. Given that support regions are of size $\omega \times \omega$, the two-pass method reduces the complexity of cost aggregation from $\mathcal{O}(\omega^2)$ to $\mathcal{O}(\omega)$.

### *B. Temporal cost aggregation*

Once aggregated costs $C(p, \bar{p})$ have been computed for all pixels $p$ in the reference image and their respective matching candidates $\bar{p}$ in the target image, a single-pass temporal aggregation routine is exectuted. At each time instance, the algorithm stores an auxiliary cost $C_a(p, \bar{p})$ which holds a weighted summation of costs obtained in the previous frames. During temporal aggregation, the auxiliary cost is merged with the cost obtained from the current frame using

$$C(p,\bar{p}) \leftarrow \frac{(1-\lambda)\cdot C(p,\bar{p}) + \lambda \cdot w_t(p,p_{t\text{-}1})\cdot C_a(p,\bar{p})}{(1-\lambda) + \lambda \cdot w_t(p,p_{t\text{-}1})}, \quad (4)$$

where the feedback coefficient $\lambda$ controls the amount of cost smoothing and $w_t(p, p_{t\text{-}1})$ enforces color similarity in the temporal domain. The temporal adaptive weight computed between the pixel of interest $p$ in the current frame and pixel $p_{t\text{-}1}$, located at the same spatial coordinate in the prior frame, is given by

$$w_t(p, p_{t\text{-}1}) = \exp\left(-\frac{\Delta_c(p, p_{t\text{-}1})}{\gamma_t}\right), \quad (5)$$

where $\gamma_t$ regulates the strength of grouping by color similarity in the temporal dimension. The temporal adaptive weight has the effect of preserving edges in the temporal domain, such that when a pixel coordinate transitions from one side of an edge to another in subsequent frames, the auxiliary cost is assigned a small weight and the majority of the cost is derived from the current frame.

### *C. Disparity Selection and Confidence Assessment*

Having performed temporal cost aggregation, matches are determined using the Winner-Takes-All (WTA) match selection criteria. The match for $p$, denoted as $m(p)$, is the candidate pixel $\bar{p} \in S_p$ characterized by the minimum matching cost, and is given by

$$m(p) = \underset{\bar{p} \in S_p}{\operatorname{argmin}}\, C(p, \bar{p})\,. \quad (6)$$

To asses the level of confidence associated with selecting minimum cost matches, the algorithm determines another set of matches, this time from the target to reference image, and verifies if the results agree. Given that $\bar{p} = m(p)$, i.e. pixel $\bar{p}$ in the right image is the match for pixel $p$ in the left image, and $p' = m(\bar{p})$, the confidence measure $F_p$ is computed as

$$F_p = \begin{cases} \dfrac{\min\limits_{\bar{p} \in S_p \setminus m(p)} C(p,\bar{p}) - \min\limits_{\bar{p} \in S_p} C(p,\bar{p})}{\min\limits_{\bar{p} \in S_p \setminus m(p)} C(p,\bar{p})}, & |d_p - d_{p'}| \leq 1 \\ 0, & \text{otherwise} \end{cases} . \quad (7)$$

### *D. Iterative Disparity Refinement*

Once the first iteration of stereo matching is complete, disparity estimates $D_p^i$ can be used to guide matching in subsequent iterations. This is done by penalizing disparities that deviate from their expected values. The penalty function is given by

$$\Lambda^i(p, \bar{p}) = \alpha \times \sum_{q \in \Omega_p} w(p,q) F_q^{i\text{-}1} \left| D_q^{i\text{-}1} - d_p \right| , \quad (8)$$

where the value of $\alpha$ is chosen empirically. Next, the penalty values are incorporated into the matching cost as

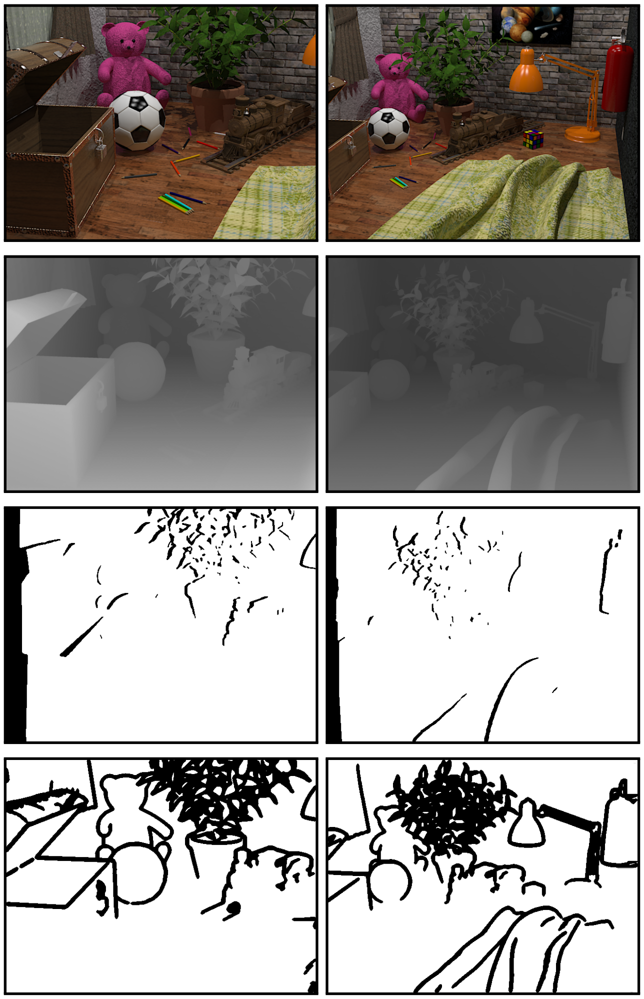
$$C^i(p, \bar{p}) = C^0(p, \bar{p}) + \Lambda^i(p, \bar{p})\,, \quad (9)$$

and the matches are reselected using the WTA match selection criteria. The resulting disparity maps are then post-processed using a combination of median filtering and occlusion filling. Finally, the current cost becomes the auxiliary cost for the next pair of frames in the video sequence, i.e., $C_a(p, \bar{p}) \leftarrow C(p, \bar{p})$ for all pixels $p$ in the and their matching candidates $\bar{p}$.

## IV. Results

The speed and accuracy of real-time stereo matching algorithms are traditionally demonstrated using still-frame images from the Middlebury stereo benchmark [1], [2]. Still frames, however, are insufficient for evaluating stereo matching algorithms that incorporate frame-to-frame prediction to enhance matching accuracy. An alternative approach is to use a stereo video sequence with a ground truth disparity for each frame. Obtaining the ground truth disparity of real world video sequences is a difficult undertaking due to the high frame rate of video and limitations in depth sensing-technology. To address the need for stereo video with ground truth disparities, five pairs of synthetic stereo video sequences of a computer-generated scene were given in [19]. While these videos incorporate a sufficient amount of movement variation, they were generated from relatively simple models using low-resolution rendering, and they do not provide occlusion or discontinuity maps.

To evaluate the performance of temporal aggregation, a new synthetic stereo video sequence is introduced along with corresponding disparity maps, occlusion maps, and discontinuity maps for evaluating the performance of temporal stereo matching algorithms. To create the video sequence, a complex scene was constructed using Google Sketchup and a pair of animated paths were rendered photorealistically using the Kerkythea rendering software. Realistic material properties were used to give surfaces a natural-looking appearance by adjusting their specularity, reflectance, and diffusion. The video sequence has a resolution of 640 × 480 pixels, a frame rate of 30 frames per second, and a duration of 4 seconds. In addition to performing photorealistic rendering, depth renders of both video sequences were also generated and converted to ground truth disparity for the stereo video. The video sequences and ground truth data have been made available at `http://mc2.unl.edu/current-research/image-processing/`. Figure 2 shows two sample frames of the synthetic stereo scene from a single camera perspective, along with the ground truth disparity, occlusion map, and discontinuity map.



Frame 30

Frame 90

Figure 2: Two sample frames from the synthetic video sequence (1st row), along with their corresponding ground truth disparity (2nd row), occlusion map (3rd row), and discontinuity map (4th row).

The results of temporal stereo matching are given in Figure 3 for uniform additive noise confined to the ranges of ±0, ±20, and ±40. Each performance plot is given as a function of the feedback coefficient $\lambda$. As with the majority of temporal stereo matching methods, improvements are negligible when no noise is added to the images [10], [19]. This is largely due to the fact that the video used to evaluate these methods is computer generated with very little noise to start with, thus the noise suppression achieved with temporal stereo matching shows little to no improvement over methods that operate on pairs of images.

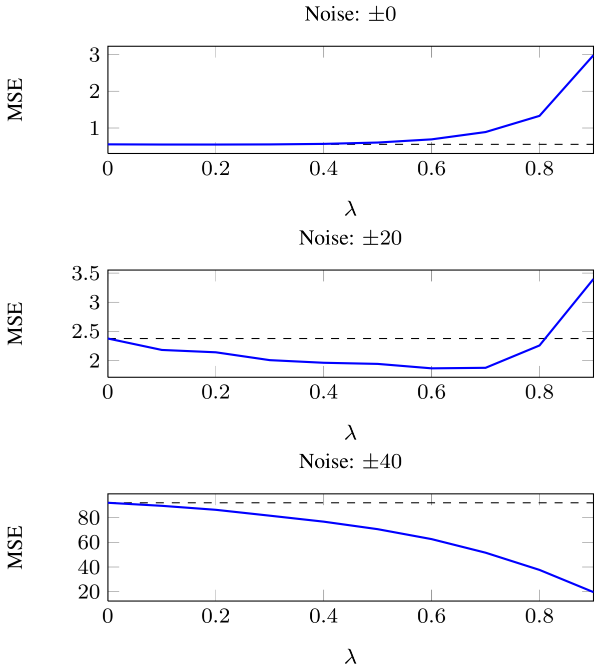


Figure 3: Performance of temporal matching at different levels of uniformly distributed image noise $\{\pm0, \pm20, \pm40\}$. Mean squared error (MSE) of disparities is plotted versus the values of the feedback coefficient $\lambda$. Dashed lines correspond to the values of MSE obtained without temporal aggregation.

Table I: Parameters used in the evaluation of real-time temporal stereo matching.

| Symbol | Description | Value |
|---|---|---|
| $\omega$ | Window size for cost aggregation | 33 |
| $\tau$ | Color difference truncation value | 40 |
| $\gamma_c$ | Strength of grouping by color similarity [1] | 0.03 |
| $\gamma_g$ | Strength of grouping by proximity [1] | 0.03 |
| $\lambda$ | Temporal feedback coefficient | varied |
| $\gamma_t$ | Strength of temporal grouping | 0.01 |
| $k$ | Number of iterations in refinement stage | 3 |
| $\alpha$ | Disparity difference penalty | 0.08 |

[1] To enable propagation of disparity information in the iterative refinement stage, the values of $\gamma_c$ and $\gamma_g$ were set to 0.09 and 0.01, respectively.

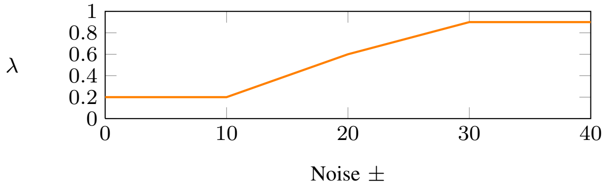


Figure 4: Optimal values of the feedback coefficient $\lambda$ corresponding to the smallest mean squared error (MSE) of the disparity estimates for a range of noise strengths.

Significant improvements in accuracy can be seen in Figure 3 when the noise has ranges of ±20, and ±40. In this scenario, the effect of noise in the current frame is reduced by increasing the feedback coefficient $\lambda$. This increasing of $\lambda$ has the effect of averaging out noise in the per-pixel costs by selecting matches based more heavily upon the auxiliary cost, which is essentially a much more stable running average of the cost over the most recent frames. By maintaining a reasonably high value of $\gamma_t$, the auxiliary cost also preserves temporal edges, essentially reducing over-smoothing of a pixel's disparity when a pixel transitions from one depth to another in subsequent frames.

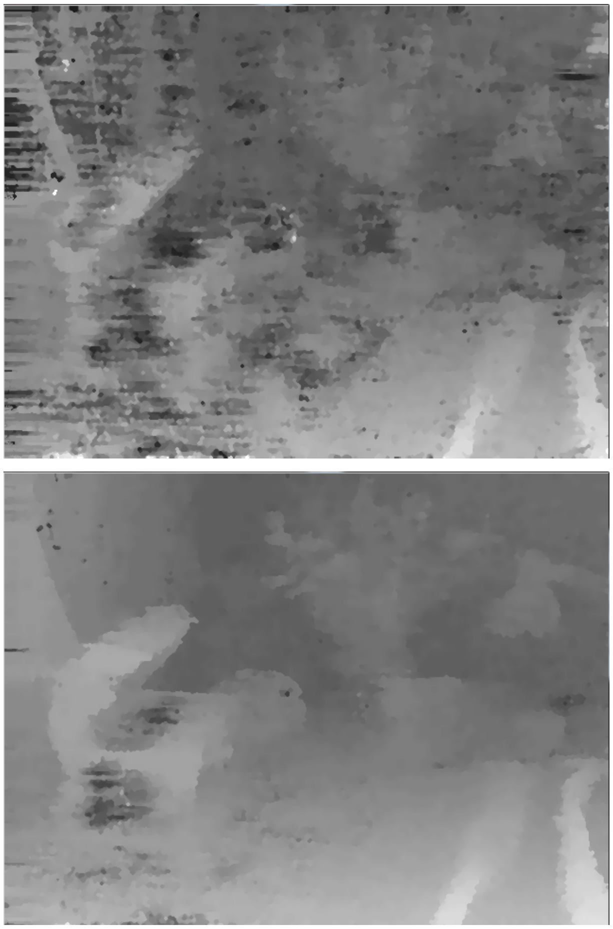


Figure 5: A comparison of stereo matching without temporal cost aggregation (top) and with temporal cost aggregation (bottom) for a single frame in the synthetic video sequence where the noise is ±30 and the feedback coefficient is $\lambda = 0.8$.

The optimal value of the feedback coefficient is largely dependent on the noise being added to the image. Figure 4 shows the optimal values of $\lambda$ for noise ranging between $\pm 0$ to $\pm 40$. As intuition would suggest, it is more beneficial to rely on the auxiliary cost when noise is high and it is more beneficial to rely on the current cost when noise is low. Figure 5 illustrates the improvements that are achieved when applying temporal stereo matching to a particular pair of frames in the synthetic video sequence. Clearly, the noise in the disparity map is drastically reduced when temporal stereo matching is used.

The algorithm was implement using NVIDIA's Compute Unified Device Architecture (CUDA). The details of the implementation are similar to those given in [3]. When compared to other existing real-time stereo matching implementations, the proposed implementation achieves the highest speed of operation measured by the number of disparity hypotheses evaluated per second, as shown in Table II. It is also the second most accurate real-time method in terms of error rate, as measured using the Middlebury stereo evaluation benchmark. It should be noted that it is difficult to establish an unbiased metric for speed comparisons, as the architecture, number of cores, and clock speed of graphics hardware used are not consistent across implementations.

Table II: A comparison of speed and accuracy for the implementations of many leading real-time stereo matching methods.

| Method | GPU | MDE/s[1] | FPS[2] | Error[3] |
|---|---|---|---|---|
| Our Method | GeForce GTX 680 | 215.7 | 90 | 6.20 |
| CostFilter [10] | GeForce GTX 480 | 57.9 | 24 | 5.55 |
| FastBilateral [7] | Tesla C2070 | 50.6 | 21 | 7.31 |
| RealtimeBFV [8] | GeForce 8800 GTX | 114.3 | 46 | 7.65 |
| RealtimeBP [21] | GeForce 7900 GTX | 20.9 | 8 | 7.69 |
| ESAW [6] | GeForce 8800 GTX | 194.8 | 79 | 8.21 |
| RealTimeGPU [5] | Radeon XL1800 | 52.8 | 21 | 9.82 |
| DCBGrid [19] | Quadro FX 5800 | 25.1 | 10 | 10.90 |

[1] Millions of Disparity Estimates per Second.
[2] Assumes $320 \times 240$ images with 32 disparity levels.
[3] As measured by the Middlebury stereo performance benchmark using the avgerage % of bad pixels.

## V. Conclusion

While the majority of stereo matching algorithms focus on achieving high accuracy on still images, the volume of research aimed at recovery of temporally consistent disparity maps remains disproportionally small. This paper introduces an efficient temporal cost aggregation scheme that can easily be combined with conventional spatial cost aggregation to improve the accuracy of stereo matching when operating on video sequences. A synthetic video sequence, along with ground truth disparity data, was generated to evaluate the performance of the proposed method. It was shown that temporal aggregation is significantly more robust to noise than a method that only considers the current stereo frames.

## 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, pp. 7–42, April-June 2002.

[2] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 195–202, June 2003.

[3] J. Kowalczuk, E. Psota, and L. Perez, "Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, pp. 94 –104, Jan. 2013.

[4] K.-J. Yoon and I.-S. Kweon, "Locally adaptive support-weight approach for visual correspondence search," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, (Washington, DC, USA), pp. 924–931, IEEE Computer Society, 2005.

[5] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," in *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, (Washington, DC, USA), pp. 798–805, IEEE Computer Society, 2006.