An In-Depth Robustness Evaluation of Stereo Algorithms on Long Stereo Sequences

Sandino Morales, Tobi Vaudrey, and Reinhard Klette
The *.enpeda..* Project, Department of Computer Science, University of Auckland, New Zealand

***Abstract*—This paper presents an approach to test stereo algorithms against long stereo sequences (100+ image pairs). Stereo sequences of this length have not been quantitatively evaluated in the past. Using stereo sequences allows one to exploit the temporal information, which is in general not well used currently. Furthermore, the presented approach focuses on evaluating the robustness of algorithms against differing noise parameters (Gaussian noise, brightness differences, and blurring).**

I. Introduction

With all the stereo algorithms in the computer vision community, it is hard to decide which algorithms should be used for different applications, or to decide which algorithm performs the best under certain situations [1]. Some of the earliest work done in comparisons of stereo algorithms was performed in [2]; this publication compared different cost functions for line-scan stereo algorithms (such as correlation-based or dynamic programming stereo). The metrics used were % errors for various regions such as % correct edges matched and total % error. The test data used was a stereo aerial view of the Pentagon, made available on [3]. For more recent stereo test data, see, for example, [4].

Stereo evaluations have also been performed in [6] by testing different algorithms with several cost functions against a dataset [5] with known ground truth; images are either computed by rendering 3D scenes (thus knowing ground truth via ray-tracing), or show indoor scenes where ground truth is measured (e.g., by using structured light). Ground truth is then compared against disparities, estimated by various algorithms, by Bad Pixel % (using differing error thresholds).

These evaluations focus on accuracy of algorithms, and for a relatively small dataset. [2] used only one image pair; evaluations on [5] test against several image pairs, and split each image pair into three sections: non-occluded areas, discontinuity areas, and all areas. The top performing algorithms have exploited the static nature of the given image pairs, but do not take into consideration temporal integration. Nor does any evaluation define or test robustness of stereo algorithms.

Our paper presents an approach to test stereo algorithms against long stereo sequences (100+ image pairs), which are also publicly available on the *.enpeda..* Image Sequence Analysis Test Site [7]. Stereo sequences of this length have not been quantitatively evaluated in the past. Using stereo sequences allows one to exploit the temporal information (e.g., [8]), which is in general not well used currently. Furthermore, our approach focuses on evaluating the robustness of algorithms against differing noise parameters; Gaussian noise, brightness differences, and blurring.

The next section introduces our approach for evaluating stereo algorithms against long stereo sequences, focusing on robustness of the algorithms. This section also introduces the dataset and error metrics we use. Section III provides a summary of the stereo algorithms used in this paper. Sections IV and V provide the results to our findings. Conclusions and future work end our paper.

II. Approach for Evaluation

Figure 1 highlights the outline of our robustness evaluation. The left and right images used are identified in Section II-A. The different types of noise we add are detailed in Section II-B. The error metrics used are detailed in Section II-D. This approach is used over an entire sequence of images, so the results can be shown over time. This is shown in the results Sections IV and V.

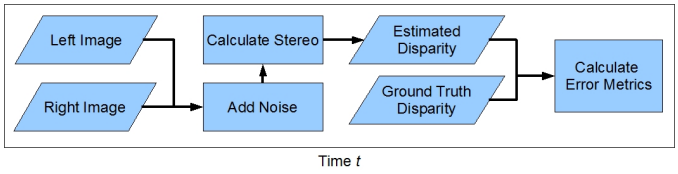


Fig. 1. Approach for robustness evaluation outlined in this paper.

A. Dataset

Synthetic sequences have been used for evaluation over the recent history. This is because the ground truth (GT) is easily obtained, and comparisons can be done using metrics such as Root Mean Square (RMS) for stereo.

In this paper we perform our evaluation on test data of Set 2 from the *.enpeda..* website [7], a synthetic sequence as introduced in [9]. This is a long synthetic stereo sequence, with 100 stereo rectified image pairs. We use the grey-scale images for our evaluation. (The sequence is also provided in color.) Ground truth data is available and is what we use for our calculating our metrics. Examples from the sequence can be seen in Figure 2.

B. Noise Added to Images

For our robustness approach we chose three different noise functionals to add to the original images $I_{\text{in}}(\mathbf{x}, t)$, where $\mathbf{x}$ is pixel position and t is time (frame #). The three noise functions

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

Fig. 2. Sample image pair (number 44) from the synthetic stereo sequence in Set 2 of [7]. Ground truth disparity colour encoding; light = close, dark = far, white = occlusion.

we chose were Gaussian white-noise, Gaussian blurring, and constant intensity change. This list of noise types is not exhaustive and there are many more variations of noise. This evaluation is to analyse the noise over the sequence.

Furthermore, we vary the amount of noise throughout the sequence. From low levels of noise, to very high levels. The noise is also varied so that only one image is affected by noise and the other one is not. This is explained in detail below. For the following subsections, the output image is mapped into the range $0 \leq I_{\text{out}}(\mathbf{x}, t) \leq I_{\max}$ (here, $I_{\max} = 255$, i.e., 8-bit), where $I_{\max}$ is the allowed maximum.

1) Constant Brightness: Differences in intensity between different cameras is a very common issue in real-world imagery. For driver assistance systems, this is even more obvious as a shadow can cover one camera and not the other. We simulate this difference in brightness with a simple addition of constant brightness to an image:

$$\forall\ \mathbf{x} : I_{\text{out}}(\mathbf{x}, t) = I_{\text{in}}(\mathbf{x}, t) + c \tag{1}$$

2) Gaussian White-Noise: Small amounts of Gaussian noise are present in real imagery. The amount of noise has been and continues to be reduced over time as sensor technology improves. We test from small amounts to large amounts of Gaussian noise [10]. At each pixel in an image, we add random Gaussian (normal distribution) white-noise $\mathcal{N}(\mu, \sigma)$; with a mean μ of zero, and a varying standard deviation σ. This is defined as:

$$\forall\ \mathbf{x} : I_{\text{out}}(\mathbf{x}, t) = I_{\text{in}}(\mathbf{x}, t) + \mathcal{N}(\mu = 0, \sigma) \tag{2}$$

3) Gaussian Blur: In real imagery, if a camera is slightly out of focus, then minor blurring can occur. For good optical lenses, this is reduced dramatically, and visual inspection can reduce this effect. We approximate this blurring effect with a Gaussian blurring convolution (as known from scale space [11]). For an entire image, we convolve the image using differing Gaussian smoothing kernel sizes:

$$I_{\text{out}}(\mathbf{x}, t) = I_{\text{in}}(\mathbf{x}, t) * G(k) \tag{3}$$

where $G(k)$ represents a $k \times k$ Gaussian smoothing kernel.

C. Noise Parameters

For our evaluation, we chose to vary the noise throughout the sequence. For the brightness noise, we chose to start the left image at a low brightness, and increase the brightness over the sequence. For the right image we did the opposite (i.e., start high and then decrease). For both the Gaussian noise and Gaussian blur, we increase the parameters through the first half of the sequence, for both the left and right image. In the second half of the sequence, the left image has no noise, and the right image decreases in noise. The specific parameters are detailed in the table below:

Noise Method	Image	$1 \leq t \leq 50$	$51 \leq t \leq 100$
Brightness	Left	$c = t - 50$	
	Right	$c = 50 - t$	
Gaussian Noise	Left	$\sigma = t$	No noise
	Right	$\sigma = t$	$\sigma = 101 - t$
Gaussian Blur	Left	$k = 2\,t - 1$	No noise
	Right	$k = 2\,t - 1$	$k = 203 - 2\,t$

D. Evaluation Methodology

For our evaluation, we measure two quality metrics for each image, over the entire sequence of images. These results can then be evaluated over an entire image sequence. We use some sample metrics from [6]. These metrics are not exhaustive, but will allow us to compare robustness.

1) RMS (root mean squared): This is the difference in computed disparity $d(\mathbf{x}, t)$, from one of the algorithms, and the ground truth disparity $d^*(\mathbf{x}, t)$. RMS is defined as

$$R(t) = \sqrt{\frac{1}{N} \sum_{\Omega} \Big(d(\mathbf{x}, t) - d^*(\mathbf{x}, t) \Big)^2} \tag{4}$$

where N is the number of pixels in the image domain Ω.

2) % Bad Pixels: This is the number of badly estimated disparities in the image domain, defined as

$$B(t) = \frac{1}{N} \left(\sum_{\Omega} \Big(|d(\mathbf{x}, t) - d^*(\mathbf{x}, t)| > \delta_d \Big) \right) \times 100\% \tag{5}$$

where δ_d is a threshold for the allowed disparity error. We use thresholds $\delta_d = 1$ or $= 2$ to determine robustness.

Since we are evaluating over the entire sequence, we can make statistical inference on the sequence data, including variances, medians, means, and range of error. In this paper, we will calculate the mean, variance, maximum and minimum for each error metric over the entire sequence.

III. Stereo Algorithms

The stereo algorithms compared are briefly identified below. All our approaches are single pixel accurate. They can be extended to sub-pixel accuracy, but the sub-pixel accurate algorithms are in scope for future work, not this brief report.

For *dynamic programming stereo*, we compare a standard algorithm [12] (DP), against one with temporal (DPt), spatial (DPs), or temporal and spatial (DPts) propagation; see [13] for propagation details.

For *belief propagation stereo* (BP), we use a coarse-to-fine algorithm [14] with quadratic cost function, as reported in [15] for Set 1 of [7].

Semi-global matching [16] (SGM) characterizes one of the top performing stereo strategies, see [5]; it is also used in commercial applications (e.g., [17]). We chose two cost

functions to contrast and compare effects of noise, mutual information (SGM MI) or Birchfield-Tomasi (SGM BT).

IV. Evaluation of Noise Across Sequence

This section shows the details of our experimental results. This includes the evaluation of the algorithms over a long stereo sequence, along with the robustness evaluation using noise corrupted input images.

A. Noise-Free Results

Here we present our results using the three metrics identified in Section II. This is the starting point for our robustness test.

For RMS (Figure 3), it is easy to see that most of the algorithms follow a similar shape, peaking in the same points and having small ranges. However, the algorithms show different magnitudes in RMS. Clearly, SGM outperforms all other algorithms, with BT winning over MI, BP follows, and all DP algorithms are much worse.

Looking at the Bad Pixel % graphs (Figures 21 and 22) it appears that the graphs both have similar shapes, with a slightly different magnitude. This indicates that whole pixel thresholding does not really differentiate algorithms. If an algorithm is inaccurate by more than a pixel, then it will be inaccurate by more than two pixels as well. This may not be true for sub-pixel thresholds, but is outside the scope of this report.

For all graphs, the rank of accuracy for the algorithms is clear, and also the same. The Bad Pixel % seems to more clearly separate the algorithms. This is further highlighted in Tables I and VII, where the results clearly show that SGM-BT performs the best. In these tables, and in all of the following ones, Standard Deviation (St. Dev.) is the zero mean standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{t=1}^{N} x(t)^2},$$

where $x(t)$ is the error at time t and N is the number of frames. The increase in St. Dev. from the mean show how stable is the algorithm. The interesting note is that the two best algorithms are the ones whose St. Dev. values are farther away from its respective means.

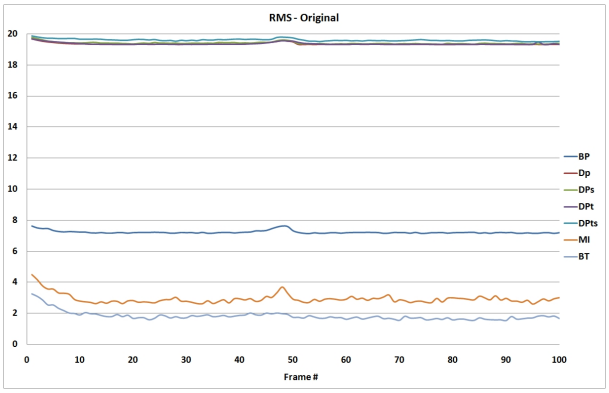


Fig. 3. RMS; on original images.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-BT	1.81	1.84	1.52	3.24
SGM-MI	2.93	2.95	2.61	4.50
BP	7.22	7.22	7.13	7.61
DP	19.34	19.34	19.30	19.68
DPt	19.35	19.35	19.30	19.68
DPs	19.39	19.39	19.32	19.76
DPts	19.61	19.61	19.48	19.87

TABLE I
RMS results using original sequence.

B. Brightness Difference Results

The tests above were performed on the brightness altered images. For RMS (Figure 4) it becomes clear there are very different shaped curves for each algorithm. All the DP algorithms are not affected much, and also the SGM-MI has very little degradation. In this case, SGM-MI performs the best overall, except around the mid-point of the sequence, where SGM-BT wins. This can be explained because the brightness difference drops to near zero around this point. SGM-BT shows very bad results for the rest of the sequence highlighting that it is not robust to brightness changes at all. The other main point is that BP has a similar effect to brightness noise as SGM-BT. This shows that they can both handle brightness differences of around ± 10.

Looking at the Bad Pixel % graphs (Figures 23 and 24) it appears that the graphs both have similar shapes to the RMS graph. There is no new information given by this metric.

Tables II and VIII confirm the subjective view. SGM-MI ranks highest in both cases. DP and DPt rank slightly higher than the other DP algorithms. SGM-BT and BP presented the highest difference between their St. Dev and mean, which is expected from the massive change in error due to brightness.

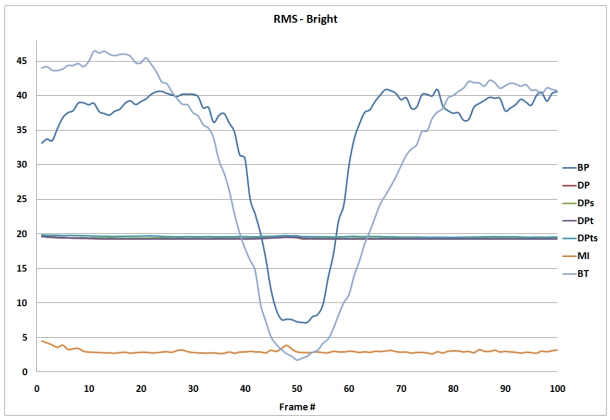


Fig. 4. RMS; on brightness altered images.

C. Gauss Noise Results

The tests above were performed on the Gaussian noise added images. For RMS (Figure 5) the shape is as expected; as the noise increases, the results get worse. At $t = 50$ the results get better as the left image no longer has noise. This holds true for all algorithms except BP, which is not affected much

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-MI	2.99	3.00	2.62	4.54
DP	19.34	19.34	19.30	19.67
DPt	19.34	19.34	19.30	19.67
DPs	19.40	19.40	19.33	19.78
DPts	19.63	19.63	19.49	19.89
SGM-BT	31.35	34.64	1.73	46.45
BP	33.52	35.08	7.22	40.92

TABLE II
RMS RESULTS USING BRIGHTNESS-ALTERED SEQUENCE.

Algorithm	Mean	St. Dev.	Min	Max
SGM-BT	3.33	3.57	1.70	6.61
SGM-MI	5.50	6.16	2.68	12.74
BP	7.73	7.75	7.17	9.39
DPt	20.30	20.31	19.34	22.32
DPts	21.90	21.94	19.80	25.40
DPs	22.40	22.54	19.39	28.09
DP	22.79	22.62	20.34	27.32

TABLE III
RMS RESULTS USING GAUSSIAN-NOISE SEQUENCE.

by the Gaussian noise. There are clear differences between the DP algorithms. DPt outperforms the other cost functions except near the end of the sequence where the noise drops to very little and then DPs works slightly better. Observe that DP is the most affected. BP performs poorly, on average being slightly higher than the DP results. Again SGM using MI and BT perform the best, with BT being the best in this situation.

The Bad Pixel % graphs (Figures 25 and 26) have slightly different results to the RMS graphs. On average, the rankings are similar, but there is more overlap between BP, MI and BT. However, BT still beats MI for more frames.

The rankings suggested above are backed up by the data in both Tables III and IX.

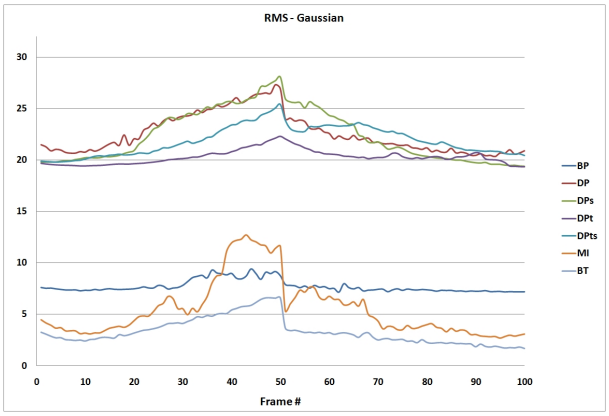


Fig. 5. RMS; on Gaussian noise-added images.

D. Gauss Blur Results

The tests above were performed on the Gaussian blurred images. For RMS (Figure 6) the results for $1 \leq t \leq 50$ are surprising. As the Gauss blurring kernel increases, the results do not get much worse for most algorithms. MI seems to be the most affected. (Can this be explained due to the cost function?) However, after $t = 50$ when the blurring effect is removed from the left image, the results get worse for all algorithms. This type of noise is more realistic as the cameras may have slightly different blurring effects between them. Again SGM-BT is the best, followed by BP/MI. MI seems to have the same problem if both images are blurred, or just one.

The Bad Pixel % graphs (Figures 27 and 28) show a much higher overlap than the previous noise levels. BT still ranks the best followed by BP. However, DP and DPt rank close to BT and are even better than BP for $50 < t < 80$ when the blurring is large in only the right image. The problems with MI is highlighted even more in this situation.

Tables IV and X further add to the comments above. SGM-BT wins in both cases, followed by BP. The interesting point is that, when looking at Table X, both DP and DPt are higher than SGM-MI. Stating that SGM-MI has more pixels that are inaccurate, but the inaccurate pixels in DP/DPt are further from the true value.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-BT	8.34	10.83	1.84	22.03
BP	13.19	15.36	6.99	28.40
SGM-MI	17.52	19.04	3.19	23.95
DP	22.79	23.14	19.34	30.28
DPt	23.26	23.64	19.32	29.49
DPs	24.38	24.89	19.47	32.63
DPts	25.02	25.47	19.74	32.69

TABLE IV
RMS RESULTS USING BLURRED SEQUENCE.

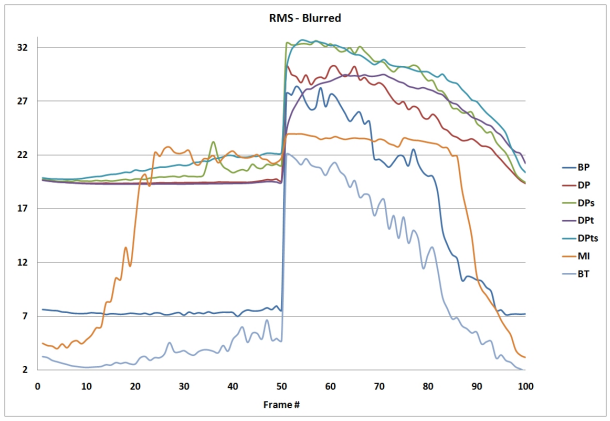


Fig. 6. RMS; on blurred images.

E. Summary

Table V shows the summary of RMS results using all data from the four sequences (original and three noise added). It highlights the points made in the previous sections. SGM-MI gets the highest robustness ranking. This may be a little bias as SGM-BT and BP were both incredibly bad on brightness altered images.

Table VI shows the summary of Bad Pixel % results. This table is more interesting, ranking DPt second highest

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-MI	7.23	20.45	2.61	23.95
SGM-BT	11.21	36.51	1.52	46.45
BP	15.41	39.73	6.99	40.92
DPt	20.56	41.47	19.30	29.49
DP	21.00	42.37	19.30	30.28
DPs	21.40	43.37	19.32	32.63
DPts	21.54	43.58	19.48	32.69

TABLE V
RMS RESULTS USING ALL FOUR SEQUENCES.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-MI	3.36	11.67	0.59	15.10
DPt	4.57	11.19	2.78	15.29
DP	5.18	12.76	2.78	17.78
SGM-BT	6.07	22.16	0.17	30.23
DPs	6.07	15.75	2.93	22.40
DPts	6.21	15.41	3.21	20.87
BP	9.70	32.13	1.18	37.27

TABLE VI
BAD PIXEL %, $\delta = 1$ RESULTS USING ALL FOUR SEQUENCES.

after SGM-MI. This indicates that the errors in DP are high when the pixel is incorrectly estimated, but the number of poorly estimated pixels is still on similar standing to the better algorithms. DPt gets the added bonus of using temporal consistency, so acquires better results over DP. If the scene was not temporally consistent, this would not be the case.

V. ALGORITHM RESULTS

In this section, we show how each algorithm is affected by the noise components. This more clearly shows how robust each algorithm is to the different types of noise.

As shown in the previous section, there is no major difference between Bad Pixel % when using $\delta = 1$ or 2. Therefore we have omitted $\delta = 2$ from this section. Example disparity images are shown using each noise type.

A. Belief Propagation - BP

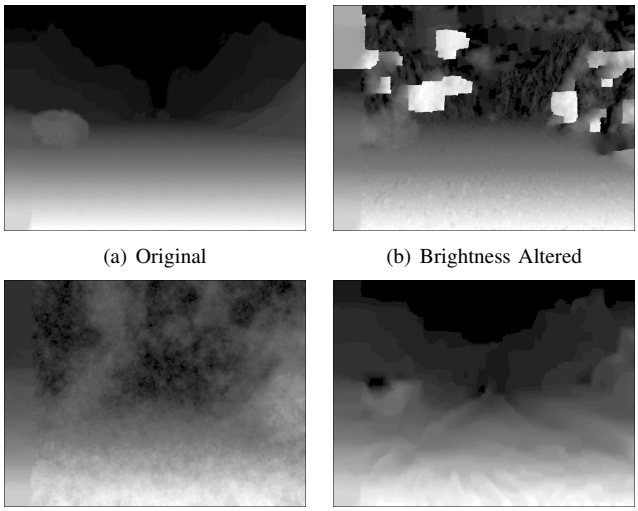
In Figures 7 and 8 it is clear which noise types affect BP more than the other. Differences in brightness cause the major issues, but Gaussian Noise seems to be mis-represented looking at the subjective result.

B. Dynamic Programming - DP

Figures 9 and 10 show clearly that the Gaussian noise makes incredibly bad results. Even small amounts of Gaussian noise make the results much worse (looking at the start/end of the graph). Brightness does not seem to affect the result that much.

C. DP Spatial Propagation - DPs

Figures 11 and 12 show clearly that the Gaussian noise makes even worse results that DP. This is expected as the errors will be propagated spatially throughout the image. The only exception is near the start/end of the graph when the noise is low. This may make the spatial propagation useful for real images where the noise content is low. However, blurring makes the results much worse.



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 7. Example disparity results for BP using the different noised images. Original images and colour encoding as in Figure 2.

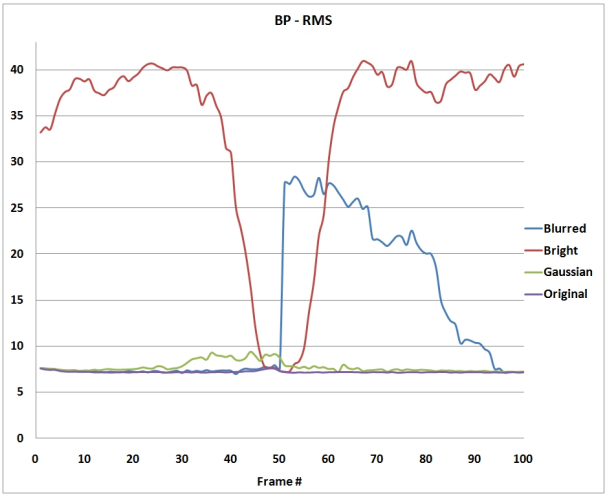


Fig. 8. RMS for BP on all images.

D. DP Temporal Propagation - DPt

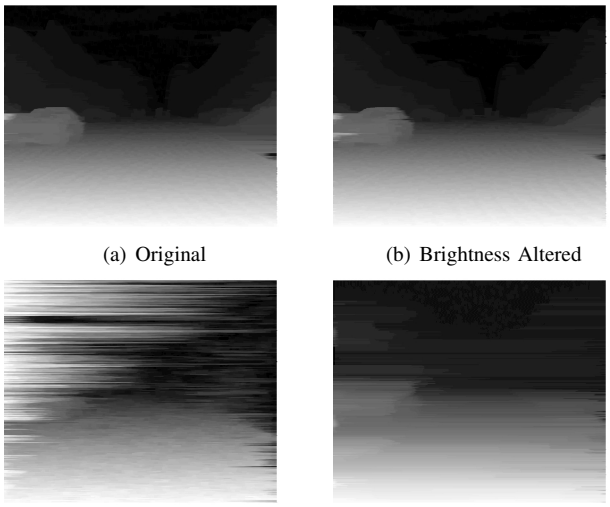
Figures 13 and 14 indicate that DPt may be the most robust of all the DP algorithms in this paper. The temporal propagation helps reduce the effect of Gaussian noise and also is not affected if both images are blurred at the same time, or if there are brightness differences. Even when the blurring effects are only on the right image, DPt is less affected than both DP and DPs.

E. DP Temporal-Spatial Propagation - DPts

From Figures 15 and 16, it appears that DPts takes both DPt and DPs into account, creating a worse overall result. This is not the desired effect, but appears that is what is happening.

F. SGM Mutual Information - MI

From Figures 17 and 18, it shows that the algorithm is pretty robust in appearance to brightness changes and Gaussian noise.



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 9. Example disparity results for DP using the different noised images. Original images and colour encoding as in Figure 2.

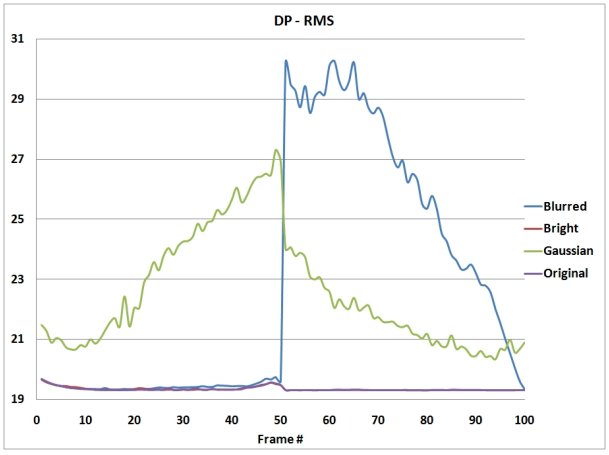
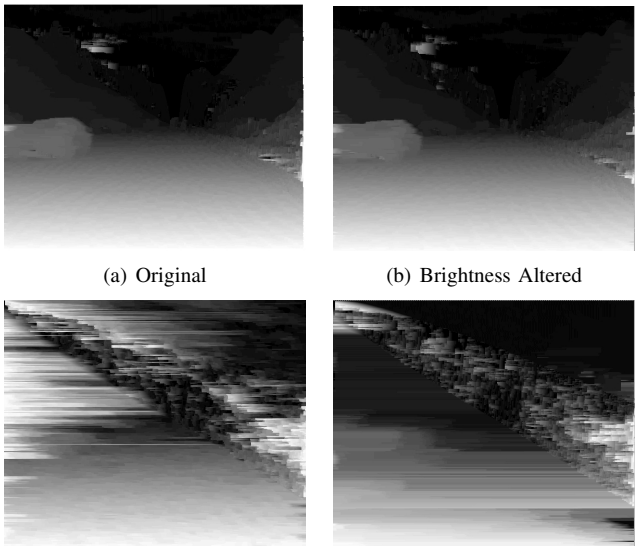


Fig. 10. RMS for DP on all images.



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 11. Example disparity results for DPs using the different noised images. Original images and colour encoding as in Figure 2.

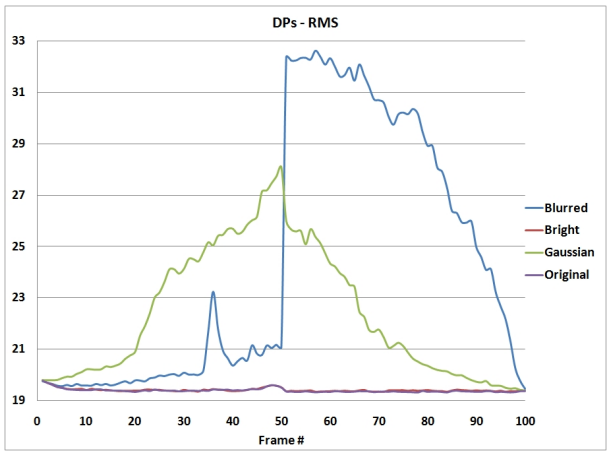


Fig. 12. RMS for DPs on all images.

However, for medium amounts of blur, the results are degraded dramatically.

G. SGM Birchfield-Tomasi - BT

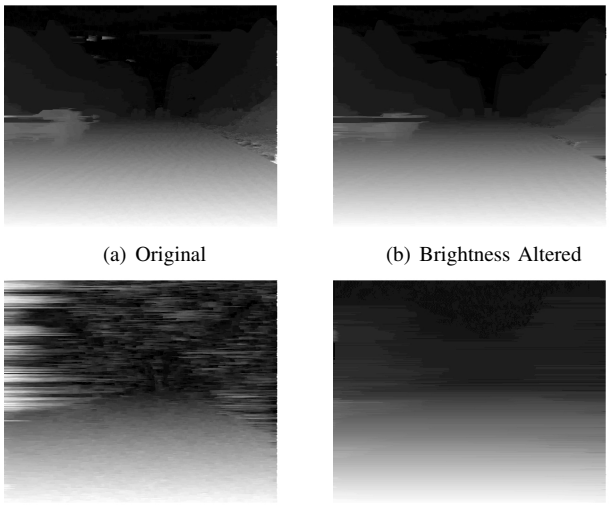
Figures 19 and 20 show that the algorithm is robust in appearance to all noise except brightness change and single image blurring. If you ignore the brightness change then this is the best algorithm by far! However, brightness change is one of the most common problems in real world situations, so it very important in this evaluation.

VI. Conclusions and Future Work

In this paper we evaluated different stereo algorithms, with different cost functions against long synthetic stereo sequence. Our approach focused on robustness of the algorithms using three different noise operators (i.e., brightness difference, Gaussian noise, and Gaussian blurring).

From our evaluation, we can see that SGM-BT is the best algorithm on average. However, it is one of the worst algorithms when there are brightness differences. In this case, SGM-MI is the best algorithm, and actually gets the top ranking using the mean over all images. BP is a good algorithm on most images, but as soon as any type of brightness change is introduced, the algorithm becomes unusable. DP is the least accurate of the algorithms tested, but it proves to be relatively robust to all types of noise. The DPt is the best of the DP cost functions, and highlights the importance of using temporal propagation for stereo algorithms to increase robustness.

Another important finding is that there is little to no difference between the Bad Pixel % when using a threshold δ of 1 or 2. Using the RMS value sometimes highlights differences in algorithm performance more clearly than Bad Pixel %, and vice-versa. In saying this, the metrics do provide different rankings overall, with DPt and DP ranking higher than SGM-BT and BP. This shows that although the algorithm has pixels with large errors, overall, the number of pixels that



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 13. Example disparity results for DPt using the different noised images. Original images and colour encoding as in Figure 2.

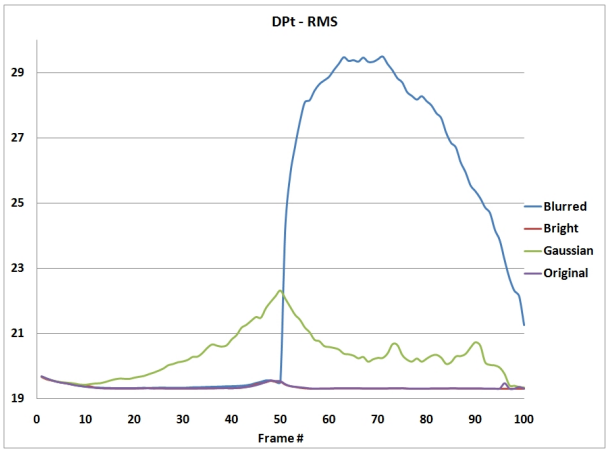
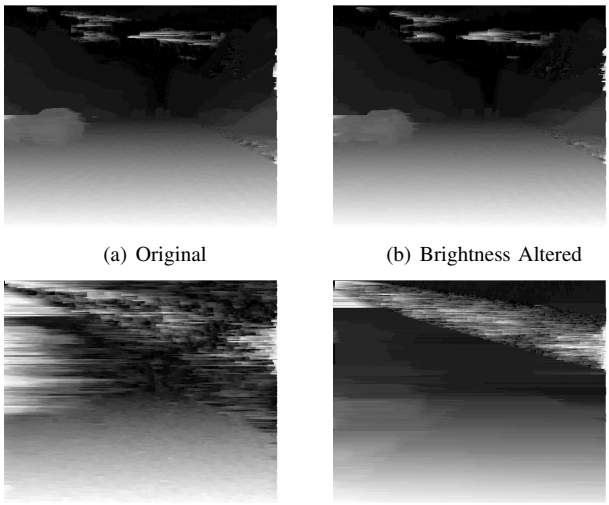


Fig. 14. RMS for DPt on all images.



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 15. Example disparity results for DPts using the different noised images. Original images and colour encoding as in Figure 2.

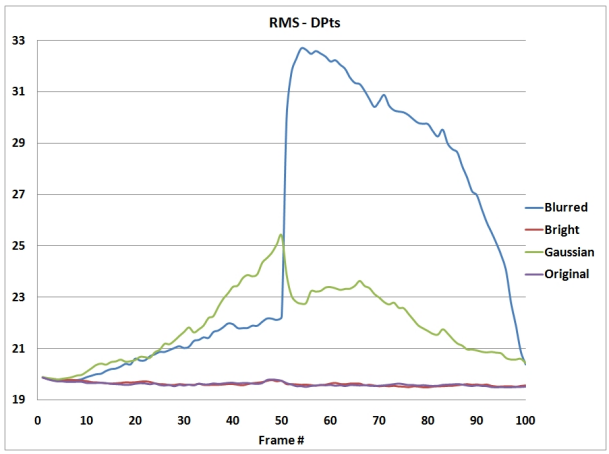


Fig. 16. RMS for DPts on all images.

are incorrect is less.

This brief report highlights the importance of using long stereo sequences for evaluating algorithms. Future work will include testing against differing noise models, analysing the results over time (e.g., moving means, standard deviations), increasing the size of our data set to test against more/different data, and analysing the metrics more clearly to help differentiate algorithms.

ACKNOWLEDGMENT

The authors would like to thank Stefan Gehrig for his implementations of SGM , Shushi Guan for his implementation of BP, and Darren Troy for his implementation of DP.

REFERENCES

[1] N.A. Thacker, A.F. Clark, J.L. Barron, J.R. Beveridged, P. Courtneye, W.R. Crum, V. Ramesh, and C. Clark, "Performance characterization in computer vision: A guide to best practices," *Computer Vision and Image Understanding*, vol. 109, pp. 305–334, 2008.

[2] R. Mohan, G. Medioni, and R. Nevatia, "Stereo error detection, correction, and evaluation," *Pattern Analysis and Machine Intelligence*, vol. 11, pp. 113–120, 1989.

[3] ——. CMU image data base. [Online]. Available: http://vasc.ri.cmu.edu/idb/html/stereo/

[4] A. Bellmann, O. Hellwich, V. Rodehorst, and U. Yilmaz, "A benchmarking dataset for performance evaluation of automatic surface reconstruction algorithms," in *CVPR*, 2007.

[5] ——. Middlebury stereo vision evaluation. [Online]. Available: http://vision.middlebury.edu/stereo/

[6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. of Computer Vision*, vol. 47, pp. 7–42, 2002.

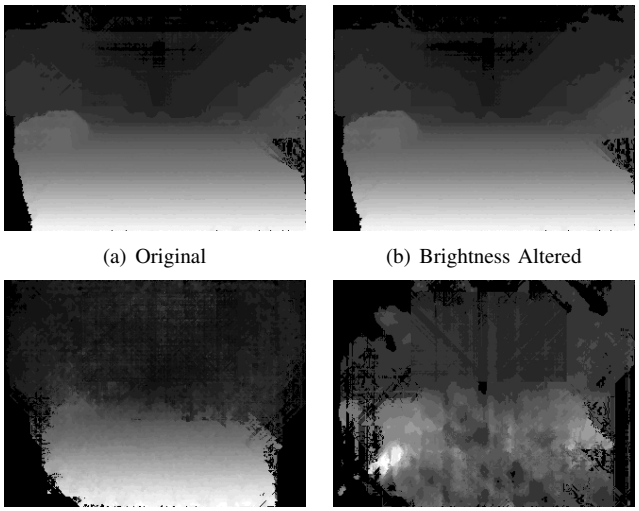
[7] *.enpeda..* image sequence analysis test site (eisats). [Online]. Available: http://www.mi.auckland.ac.nz/EISATS/

[8] T. Vaudrey, H. Badino, and S. Gehrig, "Integrating disparity images by incorporating disparity rate," in *Robot Vision*, pp. 29–42, 2008.

[9] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *Proc. Int. Conf. Image and Vision Computing New Zealand*, 2008.

[10] K. Kafadar, "Gaussian white-noise generation for digital signal synthesis," *IEEE Trans. Instr. and Meas.*, vol. 35, pp. 492–495, 1986.

[11] L. Florack, B.M. ter Haar Romeny, M.A. Viergever, and J.J. Koenderink,



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 17. Example disparity results for SGM-MI using the different noised images. Original images and colour encoding as in Figure 2.

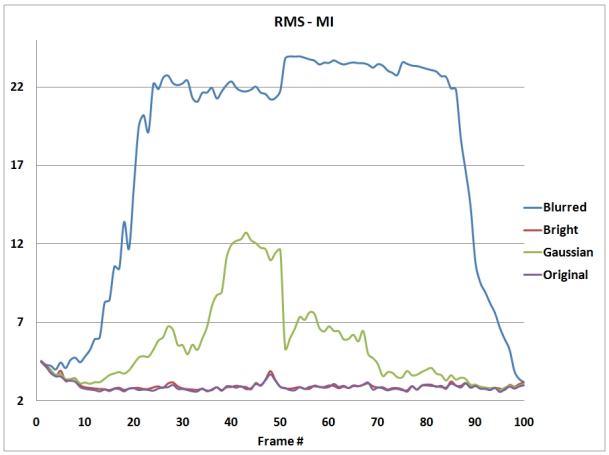


Fig. 18. RMS for MI on all images.

"The Gaussian scale-space paradigm and the multiscale local jet," *Int. J. Computer Vision*, vol. 18, pp. 61–75, 1996.

[12] Y. Ohta and T. Kanade, "Stereo by two-level dynamic programming," in *Proc. IJCAI*, pp. 1120–1126, 1985.

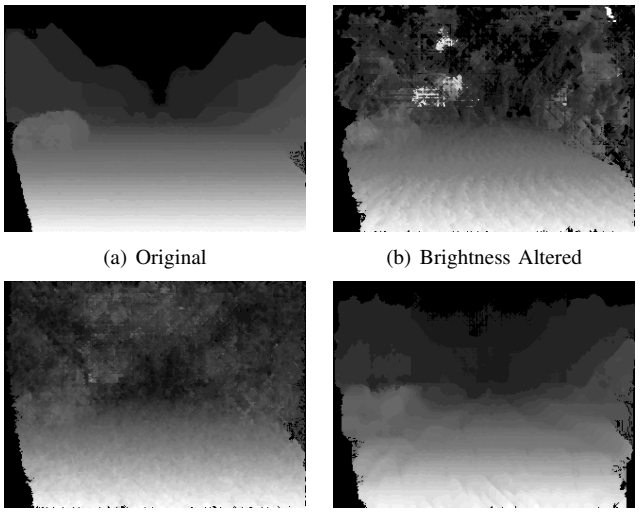
[13] Z. Liu and R. Klette, "Dynamic programming stereo on real-world sequences", In *Proc. ICONIP*, LNCS, 2008.

[14] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient belief propagation for early vision", *Int. J. Computer Vision*, vol. 70, pp. 261–268, 2006.

[15] S. Guan, R. Klette, and Y.W. Woo, "Belief propagation for stereo analysis of night-vision sequences", in *Proc. PSIVT*, LNCS 5414, pp. 932–943, 2009.

[16] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 807–814, 2005.

[17] 3D Reality Maps. [Online]. Available: http://www.realitymaps.de/



(a) Original (b) Brightness Altered

(c) Gaussian Noise (d) Blurred

Fig. 19. Example disparity results for SGM-BT using the different noised images. Original images and colour encoding as in Figure 2.

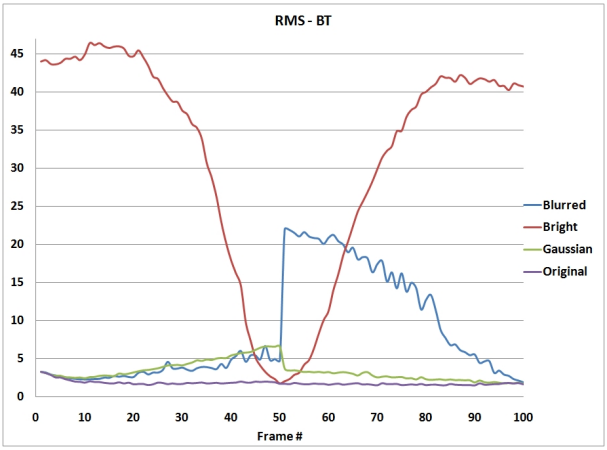


Fig. 20. RMS for BT on all images.

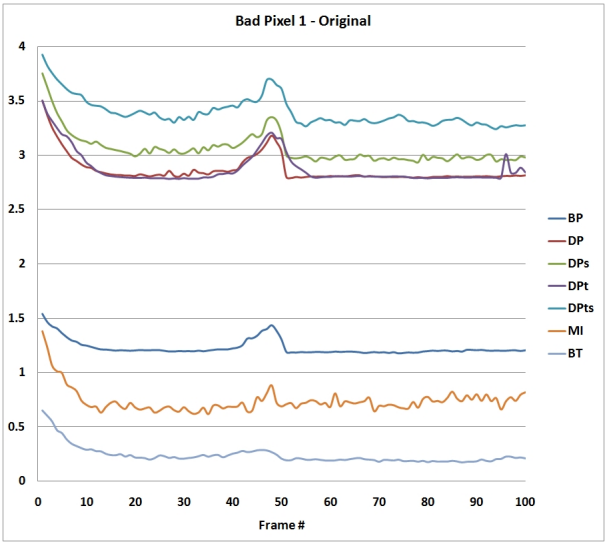


Fig. 21. Bad pixel %, $\delta = 1$; on original images.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-BT	0.24	0.25	0.17	0.65
SGM-MI	0.74	0.74	0.62	1.38
BP	1.23	1.23	1.18	1.54
DP	2.87	2.87	2.79	3.50
DPt	2.87	2.88	2.78	3.50
DPs	3.06	3.06	2.93	3.75
DPts	3.39	3.39	3.24	3.92

TABLE VII
BAD PIXEL %, $\delta = 1$ RESULTS USING ORIGINAL SEQUENCE.

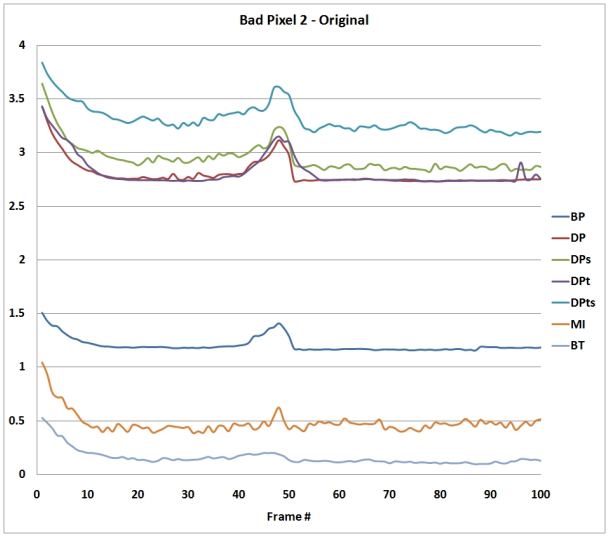


Fig. 22. Bad pixel %, $\delta = 2$; on original images.

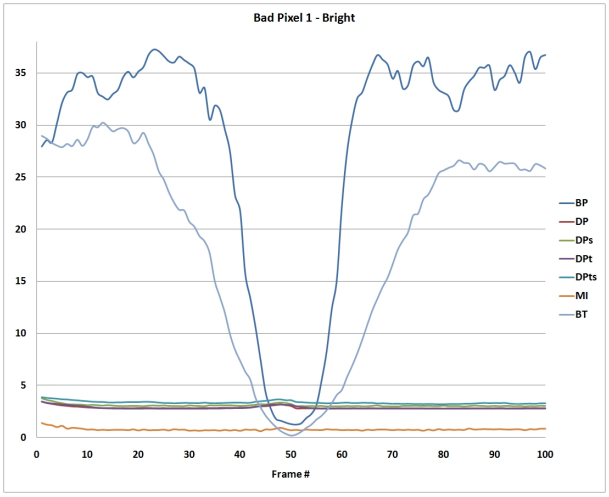


Fig. 23. Bad pixel %, $\delta = 1$; on brightness altered images.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-MI	0.76	0.77	0.59	1.42
DP	2.86	2.86	2.78	3.46
DPt	2.87	2.87	2.78	3.46
DPs	3.07	3.07	2.96	3.77
DPts	3.38	3.38	3.21	3.88
SGM-BT	18.68	21.26	0.21	30.23
BP	28.32	30.55	1.26	37.27

TABLE VIII
BAD PIXEL %, $\delta = 1$ RESULTS USING BRIGHTNESS-ALTERED SEQUENCE.

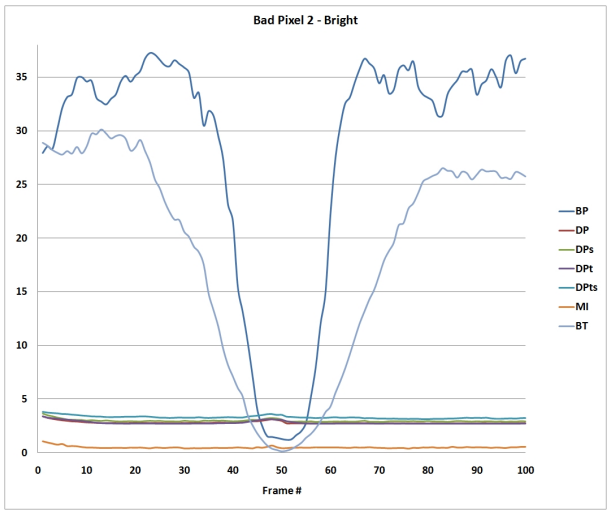


Fig. 24. Bad pixel %, $\delta = 2$; on brightness altered images.

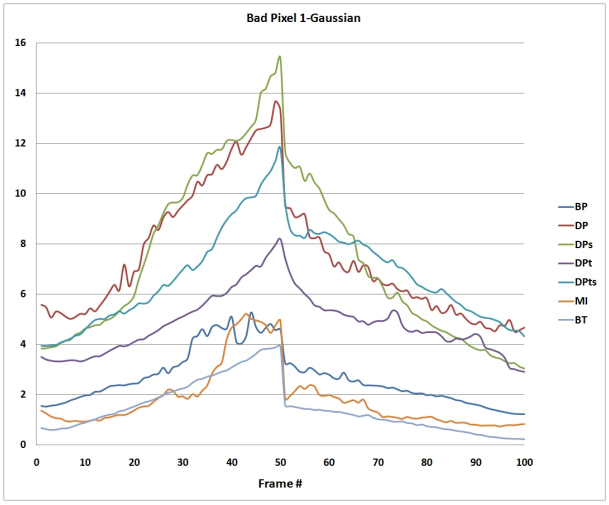


Fig. 25. Bad pixel %, $\delta = 1$; on Gaussian noise-added images.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-BT	1.46	1.77	0.21	3.92
SGM-MI	1.83	2.21	0.74	5.22
BP	2.67	2.87	1.21	5.27
DPt	4.85	5.00	2.91	8.20
DPts	6.77	7.03	3.94	11.80
DPs	7.47	8.21	3.05	15.41
DP	7.49	7.90	4.51	13.69

TABLE IX
BAD PIXEL %, $\delta = 1$ RESULTS USING GAUSSIAN-NOISE SEQUENCE.

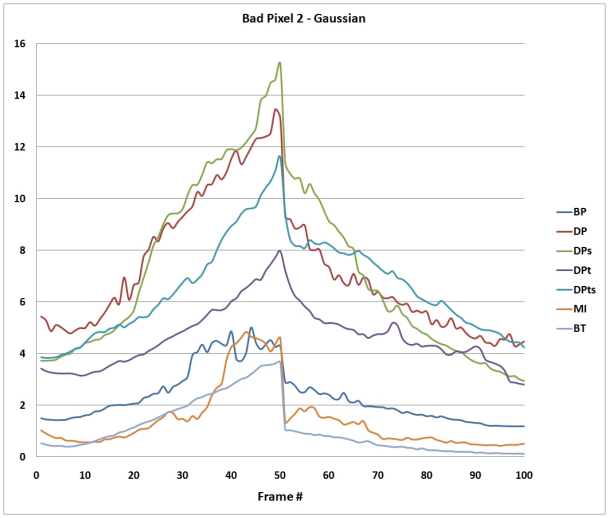


Fig. 26. Bad pixel %, $\delta = 2$; on Gaussian noise-added images.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM-BT	3.91	6.00	0.26	13.58
BP	6.59	9.44	1.25	19.89
DP	7.49	9.16	2.86	17.78
DPt	7.68	9.15	2.86	15.29
SGM-MI	10.12	11.41	0.91	15.10
DPs	10.70	12.73	3.20	22.40
DPts	11.28	12.85	3.78	20.87

TABLE X
BAD PIXEL %, $\delta = 1$ RESULTS USING BLURRED SEQUENCE.

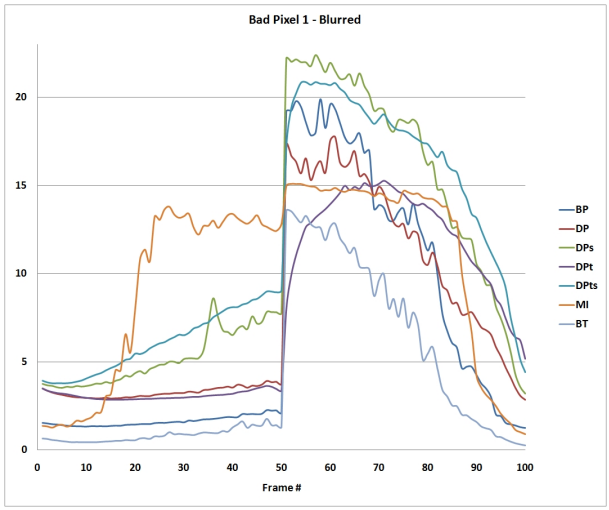


Fig. 27. Bad pixel %, $\delta = 1$; on blurred images.

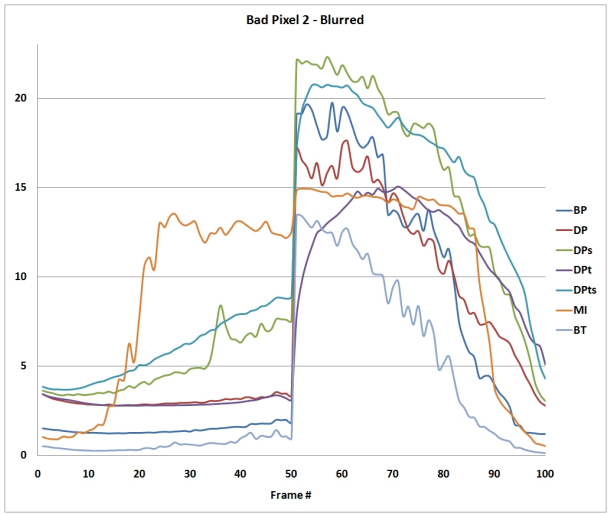


Fig. 28. Bad pixel %, $\delta = 2$; on blurred images.

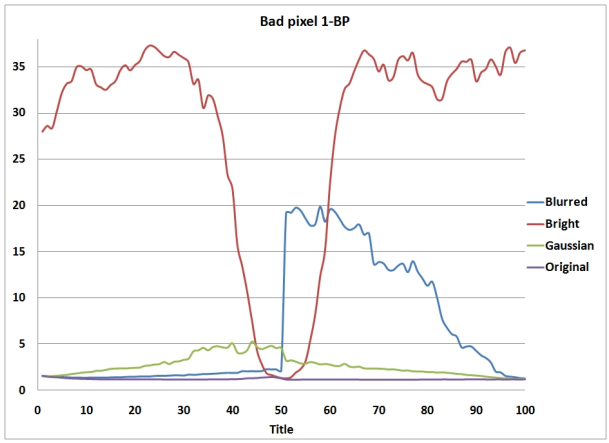


Fig. 29. Bad pixel %, $\delta = 1$; for BP on all images.

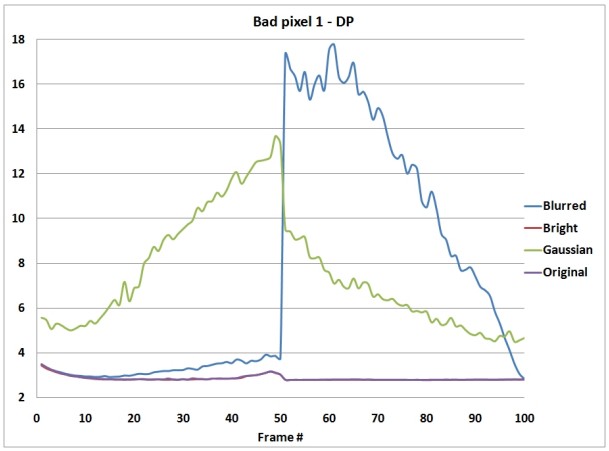


Fig. 30. Bad pixel %, $\delta = 1$; for DP on all images.

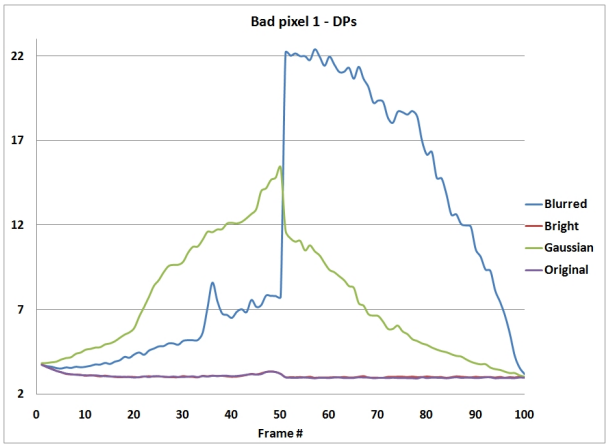


Fig. 31. Bad pixel %, $\delta = 1$; for DPs on all images.

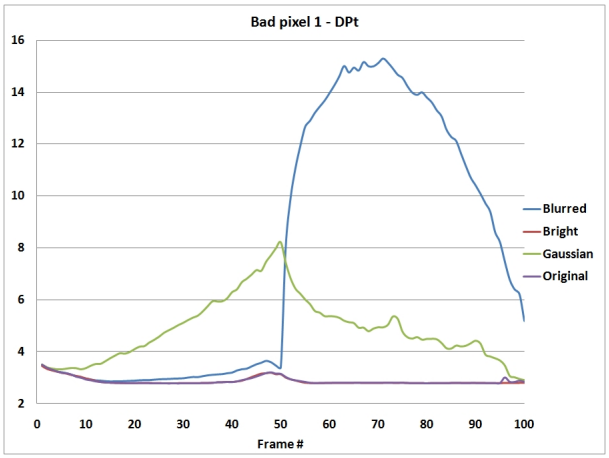


Fig. 32. Bad pixel %, $\delta = 1$; for DPt on all images.

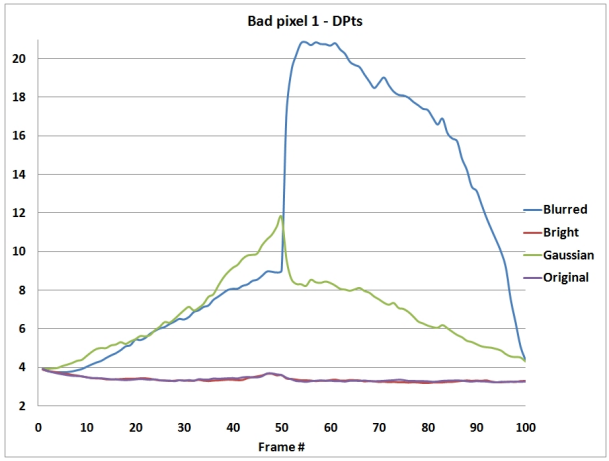


Fig. 33. Bad pixel %, $\delta = 1$; for DPts on all images.

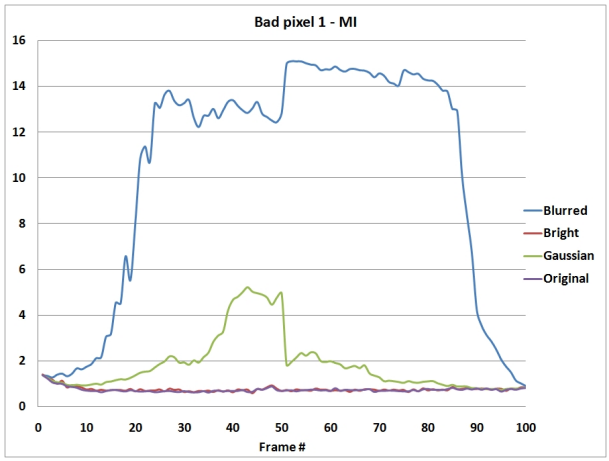


Fig. 34. Bad pixel %, $\delta = 1$; for MI on all images.

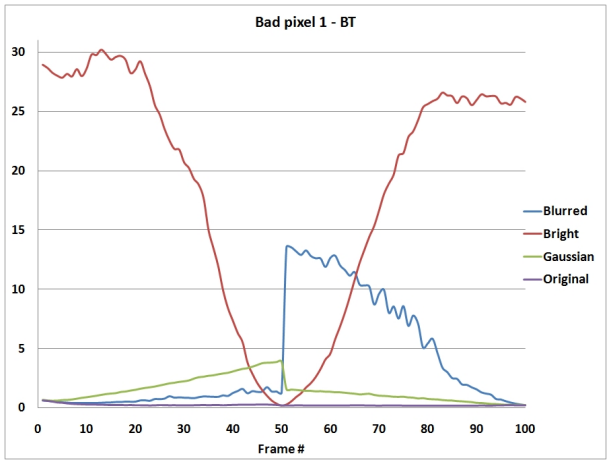


Fig. 35. Bad pixel %, $\delta = 1$; for BT on all images.