Preprocessing for Traditional Stereo Correspondence

Michael Hayashi
Stanford University
mikehaya@stanford.edu
Github: https://github.com/mikephayashi/stereo-research

1. Abstract

Stereo correspondence has recently been solved more with deep learning, but this experiment revisits once-popular traditional methods. In particular, this focuses on preprocessing images to enhance quality to provide better input to these methods. The preprocessing pipeline starts with denoising, restore the image with deconvolution, and enhances the final image with histogram equalization. The Midleburry benchmarks provide high-quality images, performance metrics, and leader board to evaluate the effectiveness of different methodologies. These metrics include Root Mean Square (RMS), bad pixel, and percentiles. The preprocessing filters on this dataset showed marginal improvement and even degraded performance. In the future, a worst quality dataset may prove to be better for this preprocessing pipeline.

2. Introduction

This paper researches the stereo correspondence problem. As described in class, this problem is concerned with matching equivalent points in different images of the same scene. Professor Tomasi in his paper describes this is "one of the most thoroughly studied problems in computer science", and it's "relatively easy to formulate, but hard to solve" [1].

In his paper, he outlines three primary approaches: global problem, local point comparison, or corresponding epipolars. For epipiloar comparison, dynamic programming is feasible but has its limits. For the entire image comparison, max flow/ min cut wins out.

This problem has been researched heavily in the past few decades and has been at the forefront of computer vision research. From the beginning and for most of the time, traditional methods has been used. However, correspondence research has followed the trend of deep learning methods. This paper revisits traditional methods and the impact of pre-processing on the results.

3. Background/Related Work

In this research, the paper revisits traditional methods and evaluates the performance of pre-processing the input images. In the taxonomy of traditional correspondence stereo algorithms, there are four key phases: 1) matching cost computation, 2) cost aggregation, 3) disparity computation, and 4) disparity refinement. Research in traditional correspondence often addresses one or two parts of this phase [2]. The Middlebury website hosts a leaderboard of the top performing research papers. Filtering out deep learning approaches, this paper will evaluate the combination of different traditional approaches and see how effective they are.

Chapter 2 in the book Computer Vision Metrics catalogs pre-processing approaches divided into discreet categories [3]. These filters fall under spatial filtering, the Fourier family, edge detection, segmentation/morphology, and thresholding. This proposed pipeline involves spatial filtering and thresholding.

Similar to the pipeline in the paper "A robust and efficient pre-processing techniques for stereo images" [4], the sequence of pre-processing steps will follow the same order: noise removal, deblurring, and color enhancement. Through the perspective of Computer Vision Metrics, noise removal can be either spatial filtering from the Fourier family. In this case, it is spatial filtering. Deblurring is not categorized from the previous list but is considered a deconvolution of the original image; in other words, it's a restoration from the blurring caused in the previous step. Color enhancement falls under thresholding because of the specific method used in the paper.

Other related work includes experimentation with different pipelines and filters. In Preprocessing for stereo vision based on LOG filter, they experimented with a slightly different pipeline. They removed noicse with both mean and Gaussian filter. Instead of the deconvolution step, they skipped to hisetogram equalization. They finished with edge sharpening by using a LOG (Laplacian of Gaussian). [5] In a New Preprocessing Technique for Computational of Stereo Matching Algoirthm, the paper used CLAHE,

AGCWD, and guided filter [6].

Spatial filtering, also known as domain filtering, operates on discrete pixel arrays. Subsections of these include convolutional filtering and detection, such as Gaussian, median, and mean filtering. Both Gaussian and median filtering work on a window of pixels. The median pixel filtering sorts the pixels in order of intensity and chooses the median of the rank. This value is then assigned to each pixel in the window. This is particularly effective against speckle, random noise. Alternatively, the Gaussian filter takes a similar approach but does a weighted averaging of the neighboring pixels based on a Gaussian distribution around the center pixel. Mean filtering can be broken down into three types: local, percentile, and bilateral mean. The local mean is like Gaussian mean but with equal weighting. Percentile mean sorts the pixels and retains the values between the 10th and 90th percentile. bilateral mean retains pixels in a range given some g with some threshold offset such as negative and positive 500. Percentile and local mean are comparable that is indiscriminate to the foreground details and background. On the other hand, bilateral mean targets the background more.

Image restoration is achieved by applying a deconvolution filter on the image. This is based on the Point Spread Function (PSF) by iteratively tuning the image. Two methods include the Wiener filter and Richardson-Lucy filter. Wiener filter is an inverse filter with a penalization on high frequency.

Thresholding segments the image based on intensity levels to binarize the image. Some of these techniques include floor, ceiling, ramp, and point. These can be bucketed into global and local thresholding. Specifically, for this case, histogram equalization uses both a floor and ceiling filter to spread the pixels between equal bins. This can be further divided into contrast stretching, equalization, and adaptive equalization.

4. Approach

The performance of the algorithms are analyzed using Middlebury Stereo Evaluation v.3 for datasets and evaluation metrics [2]. This page reflects the taxonomy and evaluation of existing correspondence algorithms. In the first paper listed above, Tomasi described this website as a "precious resource for comparing approaches."

As of 2015, the newest dataset available includes datasets of 15 image pairs. At full resolution, disparities in the image range from 200-800 pixels. These images come in 3 resolution options (full, half, and quarter) Additionally, there are versions with perfect rectification and changed exposure or lighting values for radiometric changes. Because of limited computational resources, this reduced the set to four images: Adirondeck, Jade, Piano, and Playroom.

Version 2 used a vanilla average rank, but this hid away the effectiveness of each method. Now, it uses a weighted average, and the actual weights change from year to year. Results are evaluated both with non-occluded pixels and all pixels where the former is the default. For the metrics itself:

RMS (root-mean-squared) disparity error
$R = (\frac{1}{N}\Sigma|d_C(x,y) - d_T(x,y)|^2)^{1/2}$
Average between the ground truth and computed disparity values.

Bad pixel percentage for threshold δ_d, where $\delta_d = 0.5, 1, 2, 4$
$B = \frac{1}{N}\Sigma|d_C(x,y) - d_T(x,y)| > \delta_d$
Percentage of the difference between ground truth and computed disparity values are greater than a certain threshold

Error quantiles at $A = 50, 90, 95, 99$
Where the quantile describes the value at which A percentage of pixels. For example, A90 is the value at which 90% of pixels are less than.

For this research, we have 3 steps. For the first step, spatial filters include median, gaussian, percentile mean, bilateral mean, and normal mean. For the second, deconvolution methods include Wiener filter and Richardson-lucy filter. For the last step, this only uses histogram equalization but three different variants: 1) contrast stretching, equalization, and adaptive equalization.

Several filters can be used in each of the 3 steps of this pipeline. Each step is also dependent on the output of the previous step. It's unclear which set of filters work best with each other. Because of this, we link each filter from each step with the following steps producing an exhaustive combination of all filters.

After completing the pre-processing step, this implements a simple traditional stereo algorithm: cross-correlational (dot product)sliding window. I experimented with other computation cost functions besides dot product like RMS error and absolute difference, but it's not as efficient as dot product. This is important because an optimized approach can be far too slow even on a small dataset.

5. Experiment/Analysis

Originally, I had a triple for loop - 2 for the image and one for the maximum number of disparities. Python optimizations like the just-in-time library Numba sped up the for loops but were still rather slow. I considered Cython, but pure Numpy was the best solution. For dot product, we can leverage Numpy's einsum function that computes very quickly.

First, only the sliding window was applied to the raw images without any preprocessing. This experiment evaluated

the performance from a minimum of 2 of 50 for maximum disparity. The metric of the performance versus maximum disparity level is plotted below in the Appendix. Most metrics indicated a plateau in performance increase around a maximum disparity of 30. This value is used as the baseline and sets the maximum disparity for preprocessing images.

Looping over the subset of images, each combination of filter was applied to the image and evaluated based on the performance metric. There are 5 denoising filters, 2 deconvolution filters, and 3 histogram equalization filters. In total, there are 30 combinations of filters.

The results of the experiment are below. Combinations including Richardson-lucy were omitted because they performed the same as the Wiener filter.

Compared to the baseline, most filter combinations performed about the same. Some performed marginally better, and others even performed worse. The lack of improvement can be attributed to the high-quality images chosen for computation.

RMS error ranged from 13.6 to 14.752. Bad pixel thresholds ranged from 0.8-0.99 across varying thresholds. For each quantile, pixels deviated around 2 pixels. For the 50th epercentile, it ranges from about 10-13 pixels. For the 99th percentile, it levels out around 27. Between each filter, they performed around the same with no significant performance gains.

Normal mean denoising, Wiener, deconvolution, and contrast stretching performed the worse across RMS error and bad pixel thresholds, albeit not as bad on the quantile metric. Gaussian denoising, Wiener deconvolution, and adaptive equalization performed the best on RMS error. Both the gains and loss in each of these performances are around 1-10%, hardly anything significant. This can be attribute to stochastic variations, and don't prove anything.

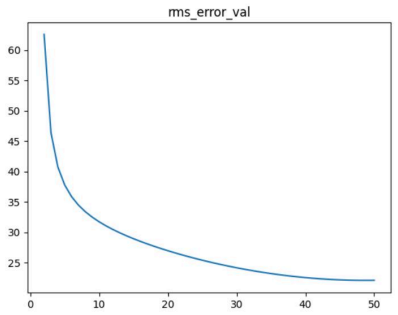
6. Conclusion

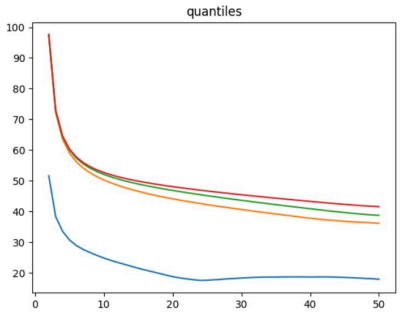
This particular pipeline of preprocessing filters does not improve performance eon high quality images in the Middleburry benchmark.

Future directions of this research include using different stereo methods like global or DP methods. Only a local matching algorithm was used. The locality of the filters may deteriorate the performance the locality of the stereo matching. The window size and disks of each may also matter as either being bigger or smaller may be impactful. Also, using a bigger and more varied dateset may show improvement. These pre-processing methods could be more applicable to a quality dataset that would benefit more from these enhancements. The images themselves were varied, but the quality of the images were consistently high quality.

7. Appendix

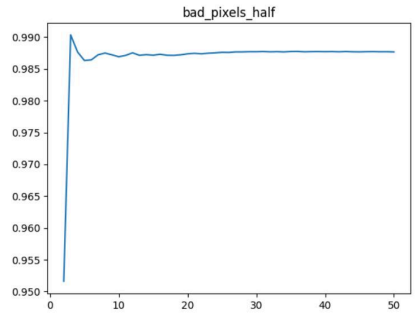
On each of the graphs, the x-axis represents the disparity level, starting at 2 and ending at 50. The y-value on RMS and quantiles represents the difference in dot product whereas the y-axis for bad pixels represents the percentage. Based on the graphs, each metric demonstrates that the disparity level with the biggest gains in error is around 10 to 20. After increasing disparity beyond this, the error is still improving but not by as much. RMS and quantile graphs show that they are approximately monotonically decreasing with the disparity level. This shows that they can be improved by increasing maximum disparity, albeit not as significant as lower disparity levels. For bad pixel graphs, the improvement flatlines, so there's a clear disparity level at which is more effective for these error metrics. For a threshold of 4 bad pixels, around 22 seems most effective. For 2 bad pixels, around 12 seems most effective. This For these graphs, it's also not monotonically increasing. Therefore, a lower disparity level doesn't necessarily indicate a worse performance.

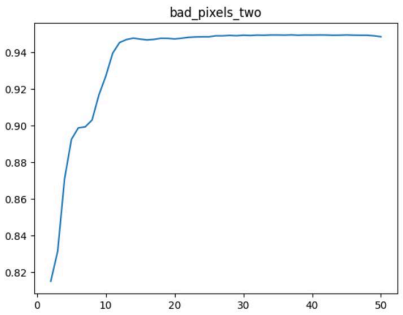


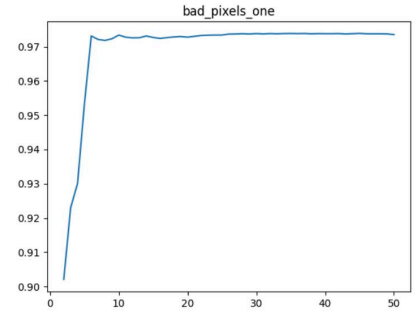


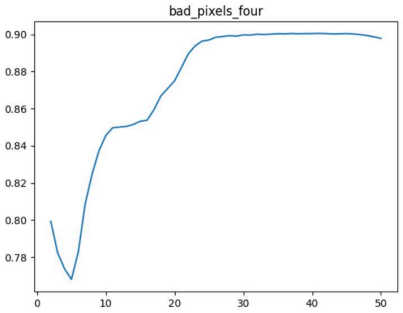
	RMS Error	Bad 0.5	Bad 1	Bad 2	Bad 4	Q 0.5	Q 0.9	Q 0.95	Q 0.99
baseline	13.785	0.980	0.951	0.911	0.821	11.114	22.395	24.296	27.949
gaussian-wiener-constrastStretching	13.749	0.980	0.950	0.910	0.820	11.112	22.326	24.222	27.945
gaussian-wiener-equalization	13.757	0.980	0.951	0.910	0.821	11.097	22.325	24.253	27.943
gaussian-wiener-adaptiveEqualization	13.600	0.980	0.949	0.908	0.817	10.880	22.208	24.138	27.973
median-wiener-constrastStretching	13.749	0.980	0.950	0.910	0.820	11.112	22.326	24.222	27.945
median-wiener-equalization	13.757	0.980	0.951	0.910	0.821	11.097	22.325	24.253	27.943
median-wiener-adaptiveEqualization	13.600	0.980	0.949	0.908	0.817	10.880	22.208	24.138	27.973
percentile-wiener-constrastStretching	14.752	0.999	0.963	0.960	0.891	13.094	21.941	24.151	26.766
percentile-wiener-equalization	13.748	0.980	0.951	0.910	0.821	11.088	22.310	24.238	27.942
percentile-wiener-adaptiveEqualization	13.605	0.980	0.949	0.908	0.817	10.885	22.217	24.134	27.973
bilateral-wiener-constrastStretching	14.752	0.999	0.963	0.960	0.891	13.094	21.941	24.151	26.766
bilateral-wiener-equalization	13.748	0.980	0.951	0.910	0.821	11.088	22.310	24.238	27.942
bilateral-wiener-adaptiveEqualization	13.605	0.980	0.949	0.908	0.817	10.885	22.217	24.134	27.973
normal-wiener-constrastStretching	14.752	0.999	0.963	0.960	0.891	13.094	21.941	24.151	26.766
normal-wiener-equalization	13.748	0.980	0.951	0.910	0.821	11.088	22.310	24.238	27.942
normal-wiener-adaptiveEqualization	13.605	0.980	0.949	0.908	0.817	10.885	22.217	24.134	27.973

Figure 1. Results









References

[1] C. tomasi. Global Stereo in Polynomial Time. Duke University.

[2] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002. Microsoft Research Technical Report MSR-TR-2001-81, November 2001.

[3] Krig, S. (2016). Computer Vision Metrics. Springer International Publishing. https://doi.org/10.1007/978-3-319-33762-3

[4]Deepa and K. Jyothi, "A robust and efficient pre processing techniques for stereo images," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 2017, pp. 89-92, doi: 10.1109/ICEECCOT.2017.8284645. keywords: Wiener filters;Histograms;Gabor filters;Image restoration;Conferences;Robustness;Deblurring;Histogram Equalization;Median Filter,

[5] Yu Shuchun, Yu Xiaoyang, Hu Lijuan and Wang Jue, "Preprocessing for stereo vision based on LOG filter," Proceedings of 2011 6th International Forum on Strategic Technology, Harbin, Heilongjiang, 2011, pp. 1074-1077, doi: 10.1109/IFOST.2011.6021206. keywords: Brightness;Histograms;Image edge detection;Matched filters;Gaussian noise;stereo vision;preprocessing;LOG filter;histogram equalization

[6] Advances in Mathematics: Scientific Journal 10 (2021), no.2, 743–758 ISSN: 1857-8365 (printed); 1857-8438 (electronic) https: //doi. org/10. 37418/amsj. 10. 2. 6