Efficient Belief Propagation in Depth Finding

Samantha Luber | Major: Computer Engineering B.S.E. | Projected Year of Graduation: 2012

Advisor: Dr. Edwin Olson | Professor in Computer Science Engineering department | College of Engineering

One of the greatest abilities of the human eye is its capacity to perceive depth, an essential skill that allows us to perform fundamental tasks, such as avoiding obstacles and retrieving objects, as well as complicated tasks, such as driving a car. As advancements in the field of robotics allow robots to successfully perform these aforementioned tasks, the need for simulated depth perception, ideally in an efficient manner, continues to grow. With the specific application of creating an efficient depth finding algorithm for robots with simple binocular cameras, various optimizations, introduced by Pedro Felzenszwalb of the University of Chicago, were applied to a naïve belief propagation algorithm to achieve more efficient belief propagation in depth finding. This paper provides an overview of the naïve belief propagation algorithm, the algorithm optimizations, and experimental results and analysis on the impact of these optimizations on algorithm performance.

I. Naïve Belief Propagation Algorithm

The general approach to accurately approximating a disparity depth map from two frame images is loopy belief propagation, a method that assigns each pixel in the disparity depth map to a value, which corresponds to the depth of the pixel in the two-frame images. Each pixel has an energy, consisting of a fixed data cost based on the pixel's value assignment and discontinuity cost, which depends on the value assignments to the pixel's neighboring pixels [4]. During an iteration, each pixel sends a message to each of its neighbors with an information vector containing the cost the neighbor will incur based on every possible value assignment to the pixel currently sending the message [2]. The algorithm loops through every possible pixel value assignment in order to find the value assignment that minimizes the energy of each pixel, based on the messages received from its neighbors. Over each successive iteration, the pixels receive discontinuity cost information from more distant pixels and are reassigned accordingly, decreasing the total energy of the graph from the previous iteration [2]. When run over many iterations, the naïve belief propagation algorithm's total energy converges, producing a reasonably accurate disparity depth map approximation.

II. Felzenszwalb's Optimized Belief Propagation Algorithm

The Felzenszwalb algorithm uses three optimization techniques, including fast message updates, grid graph, and multi-grid, on the naïve belief propagation algorithm in its belief propagation approach.

A. Optimization 1: Fast Message Updates

The fast message updates technique computes message updates in linear time by expressing these updates as min convolutions [2]. Instead of computing the optimal value assignment of the current pixels with the optimal value assignment of each neighboring pixel together, the optimal value assignment of the neighboring pixels can be computed independently of the current pixel's values. The algorithm needs only to iterate over the possible pixel values for each neighboring pixel twice; thus, message updates can be computed in linear, in contrast to the standard quadratic, time.

B. Optimization 2: Grid Graph

The grid graphtechnique computes messages in linear time by passing messages to every other pixel on even iterations and vice versa on odd iterations. This technique eliminates the need to store messages from the previous iteration for calculating and updating current messages.

C. Optimization 3: Multi-Grid

The multi-grid technique involves running the belief propagation algorithm in a coarse-to-fine manner, which increases the efficiency of passing messages over long-range distances by building a data pyramid of message updates [2]. At the highest level of the data pyramid, the algorithm runs in a coarse manner and passes messages over a large number of iterations. Running in a progressively finer manner, the number of message-passing iterations is reduced at each successive level. The Felzenszwalb algorithm is approximately the application of these techniques to the naïve belief propagation algorithm.

III. Algorithm Efficiency Experiments

In order to compare and evaluate the efficiency and accuracy of these algorithms, the naïve belief propagation algorithm and the Felzenszwalb optimized belief propagation algorithm were used to generate disparity depth maps of two-frame images from the Middlebury Stereo Datasets [1]. Both the naïve and optimized algorithms were implemented in Java for the experiments. The comparison data consists of the runtime and total energy for each algorithm recorded over a range of belief propagation iterations values. Because all of the algorithms attempt to minimize the overall graph energy, an algorithm that minimizes its total energy over less iteration is considered more efficient than an algorithm whose total energy reduces more slowly. Furthermore, as the correct disparity depth map is reached, the total energy of the graph will converge to approximately its minimum value. Naturally, an algorithm whose total energy converges over less iteration is preferred.

A. Resulting Disparity Depth Map Images

Figure 1: Disparity Depth Map resulting from naïve belief propagation algorithm for 5000 iterations

Figure 2: Disparity Depth Map resulting from naïve belief propagation algorithm with Felzenszwalboptimizations for 5000 iterations

Figure 3: Disparity Depth Map resulting from Felzenszwalb belief propagation algorithm with 5000 iterations

B. Algorithm Performance Graphs

Figure 4: Iterations vs. Performance of the naïve belief propagation algorithm, the naïve belief propagation algorithm with Felzenszwalb optimizations, and the Felzenszwalb belief propagation algorithm.

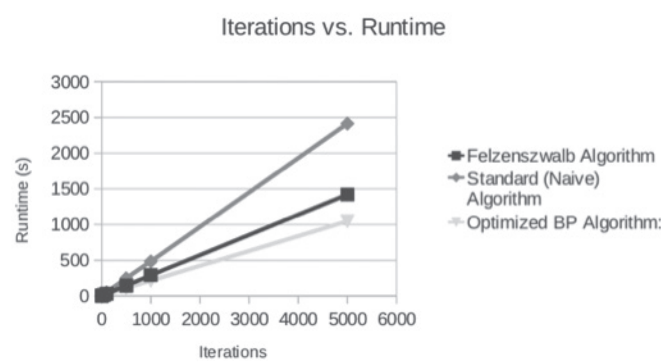
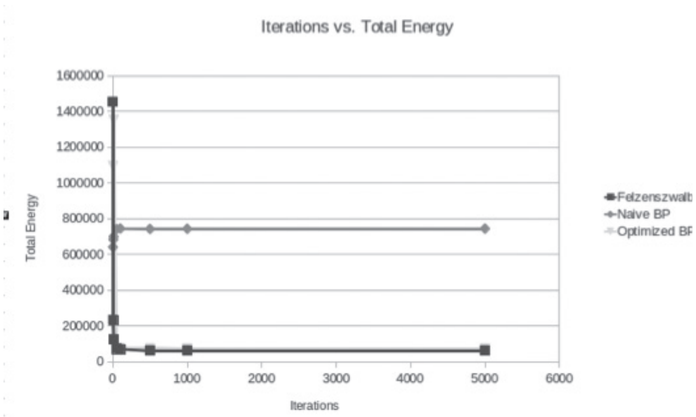


Figure 5: Iterations vs. Total Energy of the naïve belief propagation algorithm, the naïve belief propagation algorithm with Felzenszwalb optimizations, and the Felzenszwalb belief propagation algorithm.



From the data, it can be observed that the Felzenszwalb algorithm converges more quickly and has a shorter runtime than the naïve belief propagation algorithm. Furthermore, the Felzenszwalb algorithm and naïve belief propagation incorporating the Felzenszwalb optimization techniques have roughly the same runtime and convergence time; thus, by applying these optimization techniques to a naïve belief propagation algorithm, the Felzenszwalb algorithm's performance can be achieved.

IV. Conclusions

Based on the comparisons of the Felzenszwalb algorithm and the naïve belief propagation algorithm using the criteria of belief propagation iterations versus runtime and belief propagation iterations versus total graph energy, it can be reasonably confirmed that the Felzenszwalb optimizations increase both the runtime and accuracy of the belief propagation algorithm over a fixed iteration period. Furthermore, as the Java implementation of the Felzenszwalb algorithm and the naïve belief propagation algorithm incorporating the Felzenszwalb optimization techniques have reasonably close accuracy and runtime performances, it can be confirmed that Felzenszwalb's proposed techniques improve the runtime and accuracy of naïve belief propagation algorithms. Future optimizations to be explored include implementing the Felzenszwalb optimized belief propagation algorithm with parallelization.

Efficient computer vision techniques have strong applications in robotics. This belief propagation algorithm is efficient enough to run on even a small processor with little memory, allowing simple, inexpensive robotsto calculate consistent depth information with comparable accuracy to their more expensive robot counterparts. Inexpensive robots with increased capability make robotics research more accessible, thus enabling the field to continue growing and improving.

V. Acknowledgements

Particular thanks to Johannes Strom for meeting with me on a weekly basis to help with the implementation and debugging of these algorithms and to Dr. Edwin Olson for overseeing the project.

VI. References

[1] Middlebury Stereo Datasets. <http://vision.middlebury.edu/stereo/data/>.

[2] Felzenszwalb, Pedro. Efficient Belief Propagation for Early Vision. <http://people.cs.uchicago.edu/~pff/papers/bp-long.pdf>.

[3] Luber, Samantha. Public Repository of BP Algorithms. </afs/umich.edu/user/s/a/saluber/Public/stereo.git>.

[4] Yedidia, J.S.; Freeman, W.T.; Weiss, Y.; Y. (July 2005). "Constructing free-energy approximations and generalized belief propagation algorithms". IEEE Transactions on Information Theory. <http://www.merl.com/reports/docs/TR2004-040.pdf>.

[5] Bishop, Christopher M (2006). "Chapter 8: Graphical models". Pattern Recognition and Machine Learning.Springer. pp. 359–418. <http://research.microsoft.com/en- us/um/people/cmbishop/PRML/Bishop-PRML-sample.pdf>.