CS6476: Stereo Correspondence Final Project Report

Introduction

Stereo correspondence is widely explored in the field of computer vision due to its practicality. Its application improves the way machines interpret and interact with the world, increasing its useability in real-life scenarios. Stereo correspondence techniques can be split into two groups, local algorithms and global algorithms. With the increasing prevalence of artificial intelligence, researchers currently explore stereo correspondence using machine learning, but this project will focus on advanced traditional algorithms.

Related Work

Local algorithms match a pair of rectified stereo images by comparing a window centered on the reference pixel in the first image with a window centered on a sample pixel in the other image. Finding two features that have the highest similarity or lowest difference would define a matched pair of pixels. Common metrics that have been used include sum of absolute differences, sum of square differences, and correlation. To further improve local algorithms, features such as uniqueness preservation, order preservation, and occlusion detection can be incorporated although each have their own edge cases and challenges.

Global algorithms will define the cost or energy of a defined set of correspondence pairs and aim to minimize this energy by evaluating different sets. Techniques that evaluate a match globally include various dynamic programming methods and graph cut problems. These methods get better results than local algorithms but are expensive and difficult to run.

Boykov, Veksler, and Zabih showed two simple graph cut algorithms that focused on the energy equation:

$$E(f) = E_{data}(f) + E_{smoothness}(f)$$

which defined energy for a given disparity map configuration (f) as a combination of the difference between pixel intensity values in a corresponding pair (E_{data}) and a value indicating the conformity of the pixel with its neighbors' disparity ($E_{smoothness}$). In a basic alpha-beta swap algorithm, disparity levels are combined into every possible alpha-beta value pair. At each evaluation of a pair, a graph is formed with nodes representing pixels with disparity alpha and beta each connecting to a source and sink node which represent alpha and beta respectively. Cutting an edge represents the cost of partitioning a node to the alpha or beta side and impacting neighborhood smoothness through a disparity swap. A second interpretation of this energy equation is alpha expansion which acts similar to the alpha-beta swap but focuses on swapping all pixels to one disparity alpha value at a time. For each disparity level, nodes would be created for all pixels that were not disparity alpha and a cut would determine whether or not to swap that pixel to have a disparity of alpha or keep its original disparity.

Method

Local Feature Matching

A local algorithm using the sum of square differences (SSD) was used as the baseline model. SSD is defined as the following equation:

$$SSD(W_{ref}, W_{sample}) = \Sigma(w_{ref\ intensity} - w_{sample\ intensity})^2$$

where pixel intensities in one window (W) are compared to pixel intensities in the other. Two rectified images with matching epipolar lines were used for matching. The images are grayscale with black borders added to accommodate windows centered on the corner and edge pixels. For each pixel in the reference image, pixels along the same y-axis are evaluated as candidates based on the SSD calculated between their windows.. This step is repeated until all pixels in the reference image have a corresponding pixel in the sample image. In this implementation, the right image is always the reference image while the left image is the sample image since objects in the scene will shift to the right and make feature searching convenient. The search for a candidate pixel will begin from the same coordinates as the reference pixel. A maximum offset value (max_offset) is used to limit the search area on the right side of the starting coordinates. This baseline implementation does not detect occlusion, preserve order, or preserve uniqueness. To discover occluded or incorrect pixels, a second disparity map is built with the left image used as the reference instead. The two stereo images are flipped to allow objects in the right image to shift right when compared to the left image. This allows the left image to be used as a reference in this project's window sliding implementation. This second disparity map is then shifted to the left until it is calibrated to match the scene of the original disparity map. Disparity values that are not shared between the two are set to 0 and considered occluded. The map is now corrected.

The second local algorithm implementation is identical to the first but attempts to locally detect occlusion and preserve uniqueness. A value can be inputted to limit the SSD value of each candidate search. If the comparison yields an SSD less than the inputted value, the candidate will be considered for matching, otherwise it is ignored. If a reference pixel fails to have any matches, then it is considered occluded. To preserve uniqueness, each candidate pixel is checked to see if it already belongs with a matched pair. If it is already matched, then it will not be considered for matching.

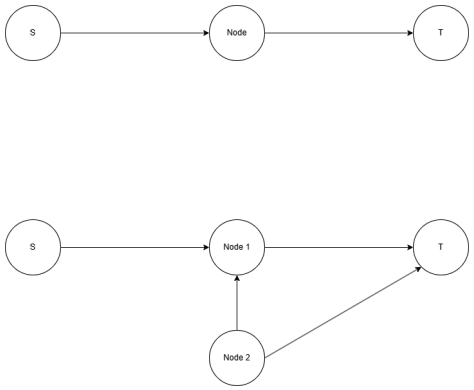
Global Energy Minimization

This project takes the alpha expansion model proposed by Boykov, Veksler, and Zabih and implements a model built by Kolmogorov and Zabih which incorporates various other features that improve disparity identification. The Kolmogorov and Zabih model focuses on the energy equation:

$$E(f) = E_{data}(f) + E_{occlusion}(f) + E_{smoothness}(f) + E_{uniqueness}(f)$$

which adds two additional energy terms to the data and smoothness equation. $E_{occlusion}$ is an energy term that applies a penalty for pixels that are occluded at the end of the current graph cut and $E_{uniqueness}$ applies an edge value of infinity between nodes sharing the same reference or sample pixel where a cut to activate both is a non solution. To

model this new energy equation, graph nodes are redefined to be pixel pairs that belong in two groups: active pairs that have a disparity of not alpha and inactive pairs of disparity alpha. The source node (S) and sink node (T) now represents changing a node state or not changing a node state respectively. To perform an alpha expansion, an active non alpha disparity node and an inactive alpha disparity node sharing the same reference pixel would both change states. This move preserves the uniqueness of the pixel in the disparity map while also updating its disparity. Nodes in the graph are structured in a unary form or pairwise form as shown in the following:

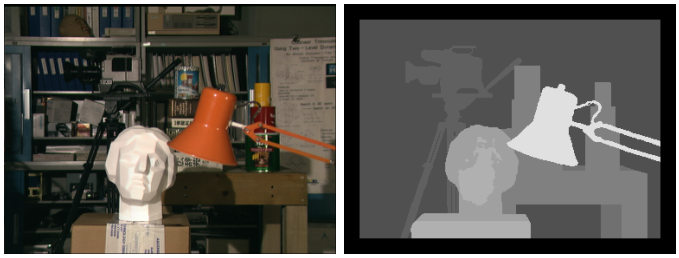


The unary form models the basic decision making structure of changing the state of a node. Partitioning to the source node means the node remains in its original state while partitioning to the sink node means the node changes its state. E_{data} and $E_{occlusion}$ are represented in this form. Nodes with active non alpha disparity pairs would have edges towards source be weighed by $E_{occlusion}$ to represent the cost of being turned inactive and have its edges towards sink be weighed by E_{data} to represent the cost of remaining active. Nodes non active alpha disparity pairs follow the same idea but reversed. Notice that the pairwise form contains the unary form but also models relationships between nodes that are not the source or the sink. $E_{smoothness}$ and $E_{uniqueness}$ take advantage of this form to model the cost of changing one of the node states or changing both of the node states. Removing a node from a neighborhood of matching disparities would cost the cut $E_{smoothness}$, but it costs nothing if the whole neighborhood is removed. Nodes that share the same reference pixel or sample pixel would have an edge cost of infinity where $E_{uniqueness}$ ensures the two are never active at the same time.

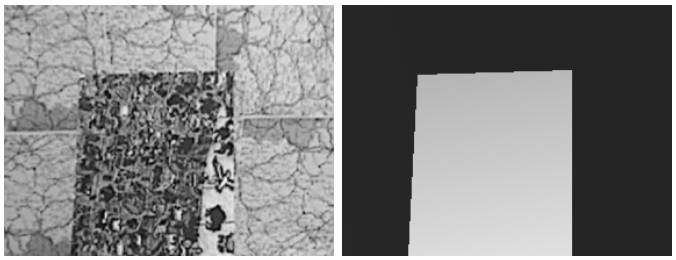
Experimentation

Results

The Tsukuba dataset and Map dataset were both used to demonstrate the models built for this project.



(Left to right: Tsukuba right stereo image, Tsukuba ground truth)



(Left to right: Map right stereo image, Map ground truth)

Disparity maps were compared to the ground truth using mean squared error (MSE):

$$MSE(I_{map}, I_{ground}) = \frac{1}{pixel\ count}\Sigma(i_{map} - i_{ground})^2$$

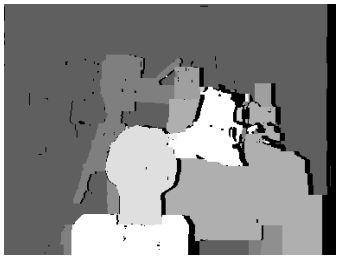
MSE was used for this project to compare and contrast the improvement between window matching and graph cut techniques. This metric provides the average square difference between pixels and keeps the number small enough to understand.

The results of the Tsukuba dataset using window matching are shown below:

(Left to right: window matching, window matching with correction, window matching with uniqueness)

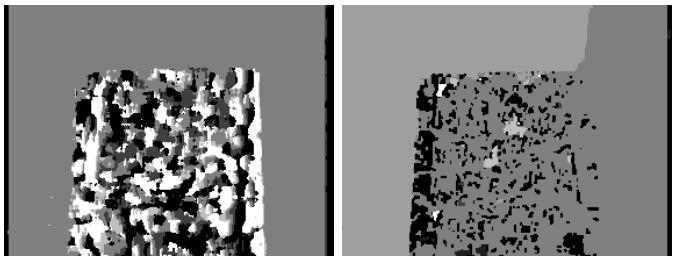
Raw window matching had an MSE of 78.5 , window matching with correction had an MSE of 121.6 and window matching with uniqueness preservation had an MSE of 126.9. While it seems that raw window matching scored very highly when compared to the other two, it is important to note that the window matching with correction is a more accurate representation of the naive window matching model since it removes occluded pixels and false positive matches. Visually the raw window matching looks better than the other two, but lacks clarity. The corrected map is missing many matches but otherwise looks interpretable. The uniqueness map contains many artifacts in an attempt to preserve uniqueness with local knowledge.

The results of the Tsukuba dataset using alpha expansion are shown below:



The MSE score of the alpha expansion disparity map is 75.2. This scores relatively close to the raw window matching model which may seem like there was not much of an improvement. However this disparity map achieves this score despite featuring occlusion, making it most accurately comparable to the window matching with correction disparity map. With this more accurate comparison, alpha expansion shows very high improvement and the map is visually very accurate to the ground truth save for certain errors like missing a part of the camera and small spots of missing pixels.

The results of the Map dataset using window matching and alpha expansion are shown below:



(Left to right: window matching, alpha expansion)

The MSE for the Map window matching (no correction) and alpha expansion are 188.6 and 175.6 respectively. Alpha expansion once again performs better window matching, but both images have relatively high error. Visually the two disparity maps roughly show the one object in the scene. However both methods get confused by the messy, multicolored texturing in the object and in the background, causing the models to have trouble differentiating the object from the scene.

Conclusion

In conclusion, global methods perform significantly better than local methods, with the ability to incorporate multiple requirements at a global evaluation to accurately form disparity maps. Specific quirks with different stereo images still confuse the advanced graph cut model, but incorporating new features or adjusting the model in detail may improve its performance.

Bibliography

Papers

Boykov, Y. & Veksler, O. & Zabih, R. N.d. “Fast Approximate Energy Minimization via Graph Cuts.” Retrieved December 5th, 2024.

Kolmogorov, V. & Monasse, P. & Tan, P. 2014. “Kolmogorov and Zabih’s Graph Cuts Stereo Matching Algorithm.” Retrieved December 5th, 2024. https://www.ipol.im/pub/art/2014/97/?utm_source=doi

Data

Middlebury. 2011. “Map.” Retrieved December 5th, 2024. https://vision.middlebury.edu/stereo/data/scenes2001/data/map/

Middlebury. 2011. “Tsukuba.” Retrieved December 5th, 2024. https://vision.middlebury.edu/stereo/data/scenes2001/data/tsukuba/