An Exploration of Surface Detection in Stereo Vision

Matt Blanchard, Cody Gronseth, Jon Sandness
Computer Science
St. Olaf College
1500 St. Olaf Ave. Northfield MN 55057
blanchar@stolaf.edu, gronseth@stolaf.edu, sandnesj@stolaf.edu

Our team took an image broken up into miniscule polygons and grouped these polygons together into larger regions, each representing a surface in the image. The polygons are read into an iterative connected-component labeling, or blob-recognition, algorithm, which merges the small polygons and creates the larger objects that make up the scene. To achieve this, we tested adjacent polygons, searching for similarities in color and shared geometric primitives. These tests, which borrow from the Gestalt Principles of Similarity, Proximity, and Continuity, are used iteratively in order to build up accurate, large polygons. Towards the end of our research we experimented with adding geometric checks to the merging process that would add global hard lines that would stop polygons from merging over object boundaries.

[1] Background

It is commonly known that the human brain interprets visual information by extracting significant features, such as lines, object boundaries, shapes and patterns. Using these features, along with the two distinct 2-dimensional representations gathered by the eyes, a 3-dimensional model of one's surroundings is created. While it is evident that these processes are occurring, we know very little about how the human brain handles this task.

Computer vision is capable of processing visual details with an exactness and speed far beyond that of human vision. However, computer vision often falls short of human vision, particularly in tasks involving object recognition, because where the human brain appears to interpret visual stimuli in terms of larger shapes and patterns, a computer processes images in terms of colors and edges. The human brain also has the capacity to speculate and make assumptions about its surroundings. For example, the human visual system can see a photograph and from this 2-dimensional representation extrapolate information about the depicted 3-dimensional scene, while still maintaining that the photograph itself has no depth. If we better understood how the human brain was capable of processing visual information at such an abstract level, it may be possible to create machines capable of seeing the world in much the same way we do.

Gestalt theory, more specifically the Gestalt Principles of Perception, describes why human beings tend to group certain visual elements together. While the Gestalt Principles do not attempt to explain exactly how these groupings are made, the underlying theories on organization and unified wholes can readily be applied to computer stereo vision. We believe that by designing a connected-component labeling algorithm that integrates these Gestalt Principles of Perception as much as possible, we will receive vastly improved object recognition results. We applied this algorithm to images split up into many small polygons, in the hopes of generating polygon groups which represent the surfaces of various objects in the image.

[2] Introduction

As one team in a larger, computer stereo vision and 3D modeling, project, we were given the task of taking an image broken up into very small polygons and group these together into larger regions each of which represents a surface of an object in the image. To achieve this, we took each of the small polygons and attempted to merge it with adjacent polygons, searching for similar attributes, namely, similarities in color and shared local and global geometric primitives. These tests, which borrow from Gestalt Principles of Similarity, Proximity, and Continuity, are used iteratively in order to build up accurate large polygons.

The merging process starts with brief image pre-processing, using the built-in despeckle command in the image viewing program Image Magick. This despeckling process removes excess noise in the image and allows for better initial polygons to be made. Once this step is finished, the actual algorithm begins by running these images through a polygonal segmentation program. This program uses color distortion to break the image into very small polygons. These polygons are then read into an iterative connected-component labeling, or blob-recognition, algorithm designed to merge the small polygons and create the larger objects that make up the scene. The main focus of our research was on the application of Gestalt theory in the comparison of color. To this end, three tests were created to evaluate the color similarity of two different polygons and decide whether they should be merged. [Sections 4.1, 4.2, and 4.3] Towards the end of our research we experimented with adding geometric checks to the merging process that aid in the formation of shapes. This ended up yielding varying results.[Section 5]

[3] Algorithm Overview

After the images are despeckled and small initial polygons are created, via a process we will refer to as "firstOFF", these polygons are analyzed in a graph. The technique that we used for this analysis is largely based off of the research done in "GlobFit:Consistently Fitting Primitives by Discovering Global Relations". Niloy Mital Et al. modeled their graph with primitive shapes as the nodes and edges between all primitive shapes. They gave edges weight based on how well they performed in different tests and used their scores to modify the primitives in the graph. Similarly, the nodes in our graph are polygons, with edges being created between adjacent polygons. These edges are then weighted based on four tests and the edges that score the best are merged. It was discovered that these tests alone would often merge polygons that didn't belong together, so an initial global line search was experimented with late into the development process, in an attempt to prevent incorrect merges.

[4] Initial Polygon Creation

While our input data is an image file, the tactics we use for surface detection rely on existing polygons. To bridge the gap between pixels and polygons,a separate program is run on the pre-processed images. This program, firstOFF, was written by Professor Olaf Hall-Holt and effectively combines the pixels of an image into usable polygonal chunks, known as polygons. Although the polygons that firstOFF outputs are not large enough to detect most surfaces in the scene, it rarely creates polygons that overlap two surfaces. Because of the accuracy of this step, we can safely assume that these polygons are correct and do not attempt to modify them beyond merging two or more together.

[4.1] Average Color Test

When we began looking into whether two polygons were actually parts of the same surface, the most obvious connection was color. For example, imagine that there is a picture of a table with two polygons which constitute the surface of the table. In comparing all the pixels of two polygons inside of that table, they may not be equivalent. Perhaps there exists a pattern or design in one polygon that causes it to appear far different from the other. However, in examining the colors of the two polygons as a whole, it can be discovered that the average red, blue, and green channels, RGB channels, are largely equivalent. When two polygons have increasingly similar average RGB channels, we say there is a higher and higher probability that they should be merged. This idea of merging based on average RGB channels is a parallel to the idea of Similarity in the Gestalt Principles. Similarity in the Gestalt Principles is defined as the repetition of forms or colors that connect in a pleasing way [2]. Although our average color test does not account for consistent forms throughout a surface, we get a similar effect by comparing the average colors of the two polygons as the irregularities in color even each other out.

[4.2] Edge-Based Color Test

While comparing average color between two polygons is often enough to determine whether or not the two are a part of the same surface, using average color alone can often lead to many false-positives. For example, two small polygons may lie on either side of the division between a wall and a similarly-colored cabinet. While the separation is readily apparent along the shared edge, the average color values of each polygon would suggest that the two should be merged (Fig. 2). For this reason, comparing the color on either side of shared edges works well as a secondary test and safeguard after the average color test, as thin boundaries are preserved.

[4.3] Dominant Color Test

The dominant color test searches through the pixels of a polygon in order to locate the top six occurring colors. This test serves as a backup for average color, much like the edge-based color test, in cases where a strong outlier may throw the test. Dominant color picks up the slack by being able to ignore the outlier as long as the outlying color isn't in the group of predominant colors.

[4.4] Collinearity/Geometric Primitives

As a means of evaluating whether or not two polygons should be merged, color is only so useful. This is particularly the case in an environment where several unique objects in a given image will likely be very similar in color. With this in mind, we expanded our tests to include basic geometric primitive detection. The Gestalt Principle of Continuity states that the patterns and characteristics of one shape continue beyond the physical bounds of the shape itself. Along these same lines, the collinearity test works off of the idea that the long straight edges found in some of the small polygons are usually sections of the larger boundaries of planes in an image. To obtain these boundaries, we search through pairs of adjacent polygons for shared vertices, and at each one compared the unit vectors created by the up to 4 edges incident to the shared vertex. Any resulting dot product with an absolute value less than 0.001 signifies a pair of line segments that are near-parallel and, because they share an endpoint, collinear as well. Once we have these lines we then use them as uncrossable boundaries for the polygons. This method isn't without its flaws. There are certain places where there are many straight lines on a single plane, such as on a railing or striped wall, which will needlessly prevent polygon merging. Another issue with collinearity is merging around the lines. While we may prevent the merging of two polygons using the collinearity test, they may very likely be merged later in the iterative process via a chain of neighboring merges.

When we became aware of the shortcomings of the collinearity test, which is essentially a local geometric primitives test, we decided to create a global geometric primitives test, to be run before any of the other tests as it is used to find global object boundaries. This test attempts to merge polygon edges together to form large lines that follow the major contours of the image. From these lines, basic polygons can be formed that represent rough outlines of the surfaces in the image.

[5] Results

We found that testing based on average, dominant, and edge color, as well as testing for collinearity, seemed to work very well in making accurate merges of the basic polygons. Of these four tests we implemented, average color was by far the reliable one. For this reason, the other 3 tests were built around average color to catch common edge cases and intervene when average color was prematurely merging or separating two polygons needlessly. In places where there are small areas of color inconsistency, taking the average of the entire polygon reduces the problematic colors and allows for more confident and consistent color merging. Unfortunately average color isn't completely problem free. For example, aturally-occurring gradients, such as those created by shadows and lighting effects, would often cause average case to abandon two polygons that should be merged. These edge cases, along with many others, would be caught by on of the other 3 tests, in this case the edge color test.

Another useful color test was dominant color. It served as a backup for average color, where if there was a bad enough outlier that it threw off the average color test, dominant color would pick up its slack. Textures such as carpets became problem areas for average color because of the constantly varying colors in the floor. Dominant color on the other hand finds the the highest occurring color in the carpet and compares it to the dominant colors around it, making merges accordingly. Dominant color also helped ignore outliers. When one or two very different colored pixels in a polygon don't match with the rest of the pixels, dominant color completely ignores the outliers in favor of the colors that occur most frequently.

The final color test, the edge color test, was another test that served as a backup for average color. It guarded against similar colors being merged across two different planes, by testing for edge color differentiation. This proved very useful when dealing with things made of wood and other commonly used materials. The test would pick up on the shadows that form when two objects are next to each other and see that as a boundary that shouldn't be crossed. Edge color proved to be a good counter check to average color, but not a test that could stand up very well on its own.

The collinearity test was one of the few checks that didn't involve testing the color values in a given polygon, so in certain situations it had a leg up on the tests that required a color difference to work. When first implemented, collinearity helped reduce the number of false merges by a large amount, but it wasn't perfect. At the time of completion both the collinearity test and the global geometric primitive search were still unreliable in certain edge cases, with the latter still being in development.

[6] Future Work

As our exploration drew closer to an end we decided to try and find global lines in order to make global shapes that would serve as hard boundaries for merging. This thought was sparked through seeing the strengths and weaknesses of the collinearity test. This idea of finding shapes in the image and then using them as guidelines for the merges follows the Gestalt principle of Similarity. By finding shapes beforehand it gives the computer a shape to fit the polygons to.

This concept of taking smaller shapes and putting them together to make one cohesive whole is at the core of Gestalt theory, and we strongly believe that the continued development of the global geometric primitives test will significantly improve the merging results.

[7]Conclusion

As a whole, we found that applying the Gestalt principles of Similarity, Proximity, and Continuity to computer stereo vision yields positive results. The four tests we developed, when used in conjunction, produced merged polygon sets that represented the surfaces in the image with a reliable level of accuracy. We believe that, in time, we could further refine the existing tests as well as further our research into using geometry to find global lines in a given image.

Figures:

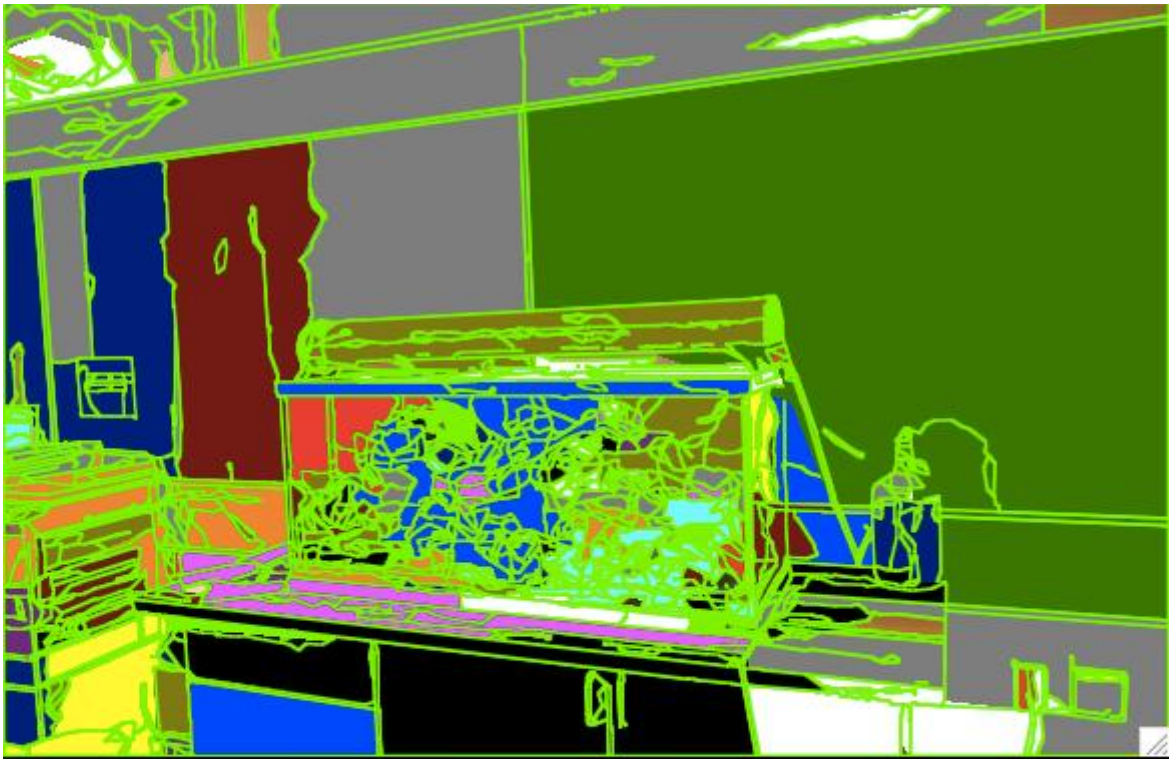

Figure 1: Typical Merged Output Generated by the Algorithm

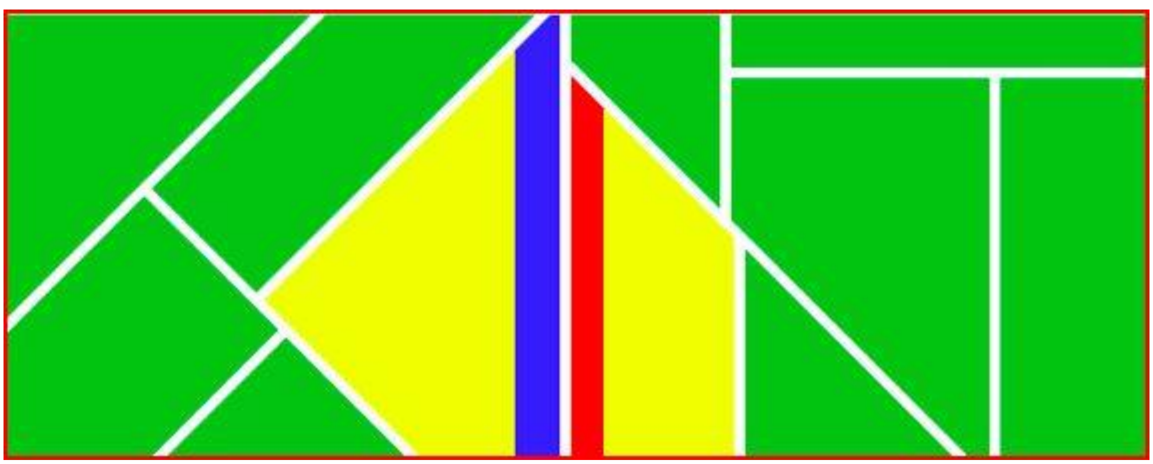

Figure 2: Edge-based Color Testing

References

Globfit
http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc.htm