Washington University in St. Louis
Washington University Open Scholarship

All Theses and Dissertations (ETDs)

January 2009

Superpixel Segmentation of Outdoor Webcams to Infer Scene Structure

Rachel Tannenbaum
Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/etd

Recommended Citation
Tannenbaum, Rachel, "Superpixel Segmentation of Outdoor Webcams to Infer Scene Structure" (2009). *All Theses and Dissertations (ETDs)*. 498.
https://openscholarship.wustl.edu/etd/498

This Thesis is brought to you for free and open access by Washington University Open Scholarship. It has been accepted for inclusion in All Theses and Dissertations (ETDs) by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Computer Science and Engineering

Thesis Examination Committee:
Robert Pless, Chair
Ron Cytron
Tao Ju

SUPERPIXEL SEGMENTATION OF OUTDOOR WEBCAMS TO INFER SCENE STRUCTURE

by

Rachel Tannenbaum

A thesis presented to the School of Engineering
of Washington University in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

December 2009
Saint Louis, Missouri

ABSTRACT OF THE THESIS

Superpixel Segmentation of Outdoor Webcams to Infer Scene Structure

by

Rachel Tannenbaum

Master of Science in Computer Science

Washington University in St. Louis, 2009

Research Advisor: Professor Robert Pless

Understanding an outdoor scene's 3-D structure has applications in several fields, including surveillance and computer graphics. Scene elements' time-series brightness gives insight to their geometric orientation; and thus the 3-D structure of the overall scene. Previous works have studied the time-series brightness of individual pixels. However, there are limitations with this approach. Pixels are often quite noisy, and can require a lot of memory. This thesis explores the use of superpixels to address these issues. Superpixels, an approach to image segmentation, over-segment a scene but attempt to ensure that each segment lies on only one scene element. Applying superpixels to webcams reduces the effect of noise on pixels' time-series brightness, and conserves memory by reducing the number of pixel "entities". This thesis explores methods of solving for a superpixel's surface normal, and demonstrates that the time at which maximum brightness is achieved serves as a basic indicator of geographic orientation.

Acknowledgments

This thesis could not have been completed without the support of many individuals. First and foremost, I would like to express my sincere gratitude to my mentor, Dr. Robert Pless, for all of the guidance he has shown me throughout this project. I am extremely grateful for all of his help and enthusiasm. I would also like to thank Nathan Jacobs for his valuable advice and suggestions, as well as my Thesis Committee members, Dr. Tao Ju and Dr. Ron Cytron. Finally, I would like to thank my family for encouraging me to pursue a Master's degree, and for their continued love and support.

Rachel Tannenbaum

Washington University in Saint Louis
December 2009

Contents

List of Figures

Chapter 1

Introduction

Understanding an outdoor scene's 3-D structure has several useful applications. In our political environment with great concern over terrorism threats, surveillance is an important element of maintaining national security. An increasing number of surveillance cameras provides an overwhelming stream of images, requiring efficient techniques to review and process this data. In particular, it is important to detect people and specific objects, and determine whether they represent a threat. A scene's 3-D structure may give insight to the type of scene it is, the scale of particular objects, and where potentially important objects, such as cars or people, could possibly appear. Another application lies in computer graphics: a real-world scene's geometric structure may be useful for rendering the scene in 3-D.

Much can be learned about a scene's structure by analyzing images of the scene throughout a period of time. Conveniently, image sequences can be easily obtained from thousands of webcams (if not more) that are publicly available on the internet. This thesis uses several scenes from AMOS, an archive of images taken from over one thousand outdoor webcams [6].

Webcam videos give strong cues to understand a scene's 3-D, geographic structure. A scene's structure manifests itself in how a pixel varies in brightness throughout the day. In outdoor scenes, for example, eastward facing walls are bright in the morning. In general, a pixel's brightness over time tells us the geographic orientation of the surface on which it lies. Previous works have looked at attributes of a pixel's brightness over time, such as its *brightness extrema* (a function of a pixel's peaks and troughs in brightness) [9] or components in shadow or sunlight [20]. This thesis

Figure 1.1: A sample webcam image

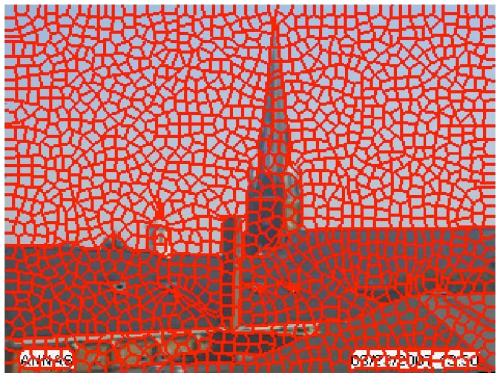


Figure 1.2: Superpixel segmentation

demonstrates that the time at which maximum brightness is achieved serves as a basic indicator of geographic orientation.

3-D scene structure has been heavily studied on the pixel level; in other words, the orientation of the object on which a particular pixel lies [9, 3]. However, there are limitations with this approach. First, considering individual pixels requires a lot of memory. Webcam images have hundreds of thousands of pixels, sometimes over one million. This requires a lot of memory when doing computations on several images of a scene, especially when explicitly looking at the time-series values of pixels throughout the video. Second, individual pixels are often quite noisy, especially with jpeg compression artifacts.

This thesis explores the use of superpixels to address these issues. Superpixels, introduced by Ren and Malik as a method for image segmentation, over-segment a scene, but attempt to ensure that each segment lies on on only one object, surface, or scene element [19]. An example superpixel segmentation is shown in Figure 1.

Superpixels are a robust measure of brightness. Individual pixels may be noisy, but taking the median over all of a superpixel's interior pixels is more reliable. In addition, the number of pixels in an image is high even at low resolutions, but using superpixels greatly reduces the number of pixel "entities" to store. For example, a scene containing one million pixels can be segmented into one thousand superpixels. This thesis explores methods of solving for a superpixel's surface normal, and introduces a software system in which an input scene is segmented into superpixels and surface normals are calculated.

Chapter 2

Previous Work

Superpixels have been used in several applications, including 3-D scene structure. We review some uses for superpixels, as well as methods for their calculation. In addition, time-lapse videos have also been used for determining 3-D structure, among other applications.

2.1 Superpixels

Superpixels were introduced by Ren and Malik as a preprocessing step in their segmentation algorithm [19]. They have since been widely used for image segmentation and labeling [1, 7, 21, 2, 10].

Ren and Malik use normalized cuts in their original superpixel segmentation algorithm. Subsequent superpixel algorithms include "TurboPixels", which offer a significant speedup over normalized cuts in higher resolution images [11], and "Superpixel Lattices" which constrain superpixels to conform to a grid [13]. This thesis uses Mori et al.'s superpixel code, which is based on normalized cuts, and incorporates Gestalt-cue based image boundary detectors [14, 12].

2.2 Superpixels for 3-D Scene Structure

Hoiem et al. employ superpixels to segment scenes into three structural components: the ground, sky, and anything that stands on the ground (buildings, trees, cars,

etc). As an interesting application, they use this segmentation technique to create automatic "photo pop-ups" from a scene. [5, 4].

2.3 Time-Lapse Videos for 3-D Scene Structure

Time-lapse video data has been used in many applications, including computing intrinsic images [22] and geolocating webcams [6]. Kim et al. use time-lapse data to compute a camera's response function and exposure values [8]. Jacobs et al. also use time-lapse data to compute camera calibration and 3-D scene depth, using cloud shadows [16]. Narasimhan et al. built a large database (WILD) containing images of a scene taken every hour for a year. They take advantage of the varying weather conditions captured in the images, and compute the camera's calibration, and the depth of scenes [15, 17].

In "Factored Time Lapse Video", Sunkavalli et al. compute a matrix factorization of pixels into shadow, illumination, and reflectance components [20]. Their factorization allows the user to select pixels with a particular surface normal. In "Clustering Appearance for Scene Analysis", Koppal and Narasimhan cluster pixels by surface normal [9]. They gather time-lapse data by randomly moving a light source around indoor scenes, and additionally include some outdoor scenes. They analyze a pixel's *appearance profile*, or its brightness over time. Specifically, they show that pixels with similar *brightness extrema*, a function based on the peaks and troughs in brightness, have similar surface normals. This clusters pixels by surface normal, but does not report what the surface normal is at a particular location.

This thesis also looks at appearance profiles, though of a superpixel rather than an individual pixel. We define the *appearance profile of a superpixel* as the median appearance profile of its interior pixels. In addition, rather than looking at a superpixel's brightness extrema, which is a function of all peaks and troughs in its brightness, this thesis shows that the global maximum alone is a good indicator of surface normal. It also attempts to improve upon Sunkavalli's work for scenes with noisier data.

Chapter 3

Applying Superpixels to Webcams

Two problems can arise when segmenting a single scene into superpixels. First, noise in an image can affect the segmentation. Second, an arbitrary snapshot may contain objects that are not "true", consistent components of a scene. For example, a large shadow or car can appear in a snapshot, and can influence the segmentation to include a boundary around it, potentially at the expense of bounding true objects in the scene.

Figure 3.1 (a) shows a single snapshot where cars and large shadows are present. The generated boundary probability image has high probabilities of boundaries around the cars, and the shadow that is cast over the red brick building on the left side of the image. Thus, in the superpixel segmentation, there are borders around the cars, and the boundary between the building and the pavement is misplaced because of the shadow (shown highlighted in the bottom image).

Taking the median image from one day is an improvement over a single snapshot. Figure 3.1 (b) shows a median image of snapshots taken throughout one day. This eliminates elements that appear in only a few images, such as cars and cast shadows. However, the cars in the parking lot remain in the image, since they are present in all images from that day. The segmentation is a slight improvement over the first one. It correctly bounds the bottom of the right red brick building, but it poorly segments the white building behind it.

In Figure 3.1 (c), the median is taken over all images from five days. Now, elements that are present throughout one day but absent in others, such as the cars in the parking lot, are eliminated. However, the median image that results is somewhat dark, and the resulting boundary probability image does not recognize some of the

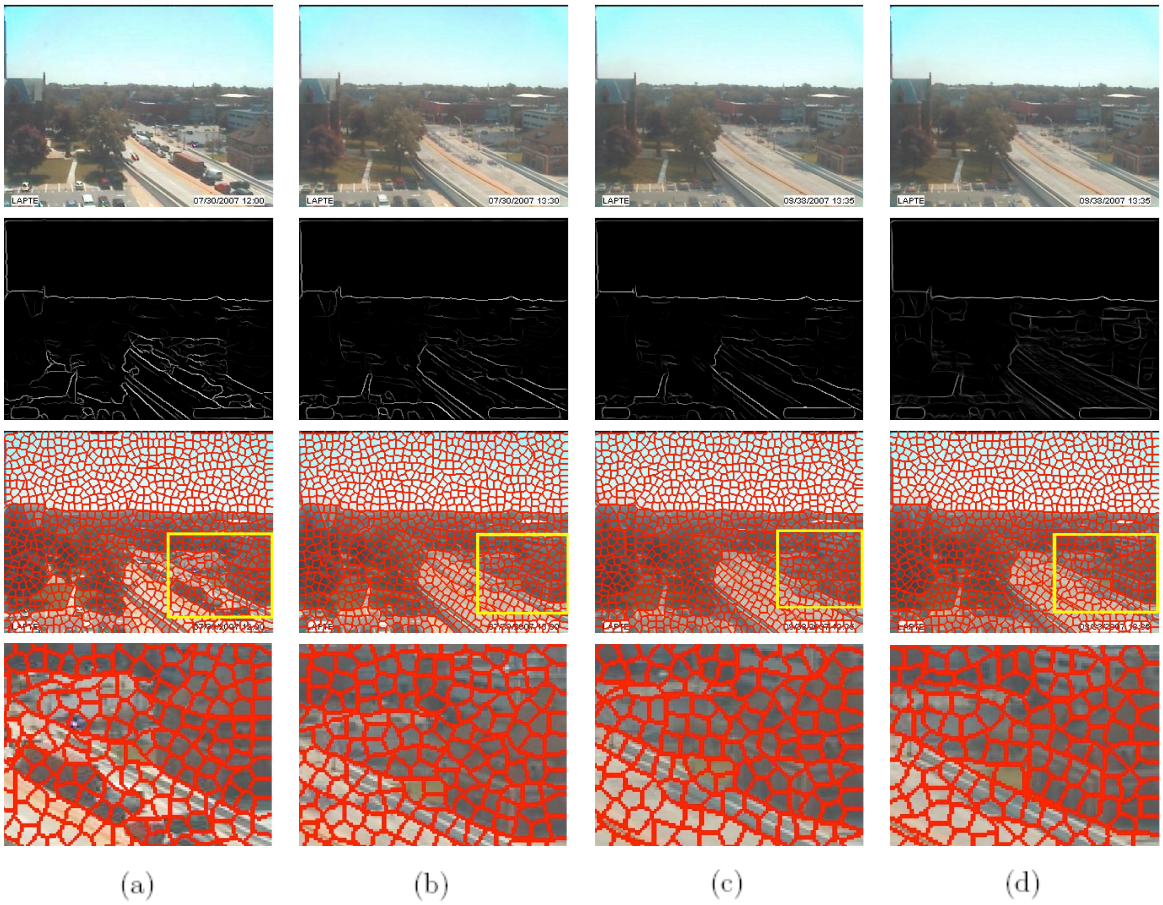


Figure 3.1: Segmenting webcam images into superpixels

intra-object boundaries. For example, the part of the probability boundary image corresponding to the right brick building is quite dark. As a result, the walls of this building are poorly segmented.

Finally, in Figure 3.1 (d), the same median image is shown, but the probability boundary image is the mean over the probability boundary images of all snapshots throughout the five days. Now, since boundary probability images of brighter snapshots are incorporated, the intra-object boundaries have higher probabilities. The resulting segmentation is accurate, as the superpixels do not cross boundaries. In particular, both the right obrick building and the white building behind it are correctly segmented. Thus, superpixel segmentations are often more robust on webcam

videos than on single images, particularly in scenes with temporary elements, such as cars, people, or cast shadows.

Figure 3.2 shows three sample segmentations. In each of them, the median image and mean probability boundary image were used in calculating the segmentations. Note that superpixels almost never cross object boundaries.

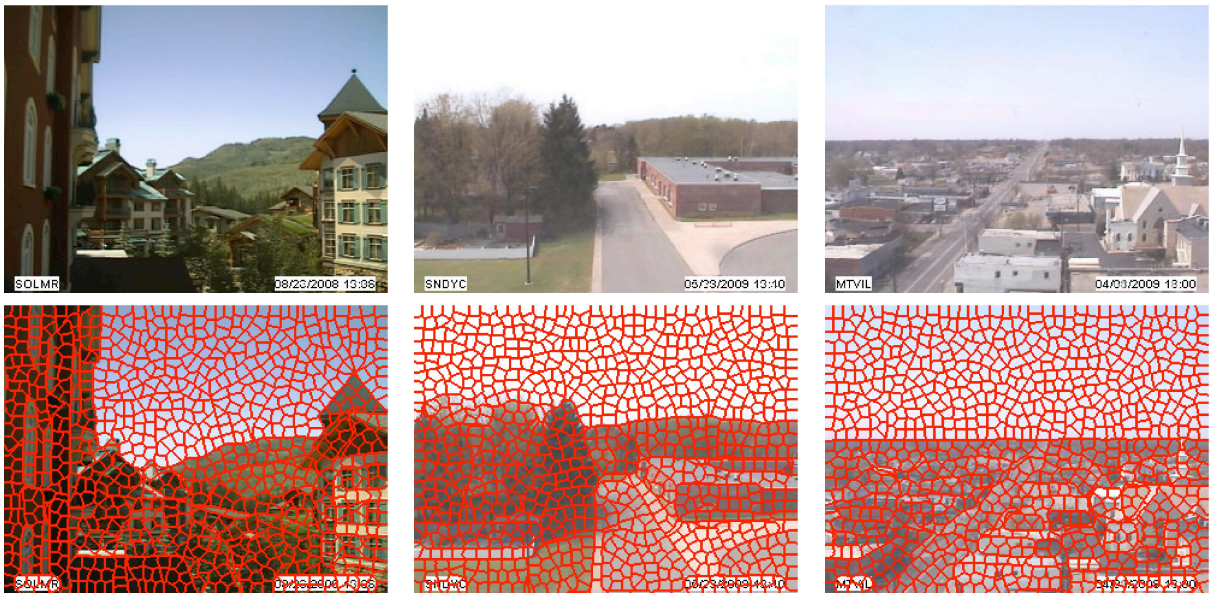


Figure 3.2: Sample segmentations

Chapter 4

Using Superpixels to Infer Scene Structure

Inferring scene properties using superpixels is more robust than using individual pixels. This thesis examines superpixels' appearance profiles, which we define as the median of all interior pixels' appearance profiles. A superpixel's appearance profile is usually much smoother than that of its interior pixels. In addition, the profile is often a more representative sample of the underlying surface's brightness over time, as there may be high variance between its interior pixels. We demonstrate that the time at which a superpixel plot reaches its global maximum allows us to calculate the north and east components of the superpixel's surface normal, which we call a *partial normal*. In addition, using the entire appearance profile, we can calculate the superpixel's complete surface normal, i.e. its north, east, and upward components.

4.1 Superpixel Appearance Profiles

Figure 4.1 shows four superpixels and their appearance profiles. The top images show a scene with a superpixel highlighted in red. The plots immediately below show the appearance profile of a sample pixel within the highlighted superpixel. The next plot shows the appearance profiles of all pixels within the superpixel. Finally, the bottom plot shows the appearance profile of the superpixel, which is the median of all interior pixel appearance profiles.

Note first that individual pixel plots can be bumpy. Since surface normals are eventually calculated from the time at which global maximum in brightness is reached, bumpiness in individual curves could affect this measure and result in an innacurate surface normal calculation. The variance in individual pixel plots may also cause inaccurate max times. Using the superpixel appearance profile rather than individual pixel appearance profiles reduces the effect of both issues. Observe in Figure 4.1, the superpixel appearance profiles are smooth, and are a robust indicator of the individual pixel plots.

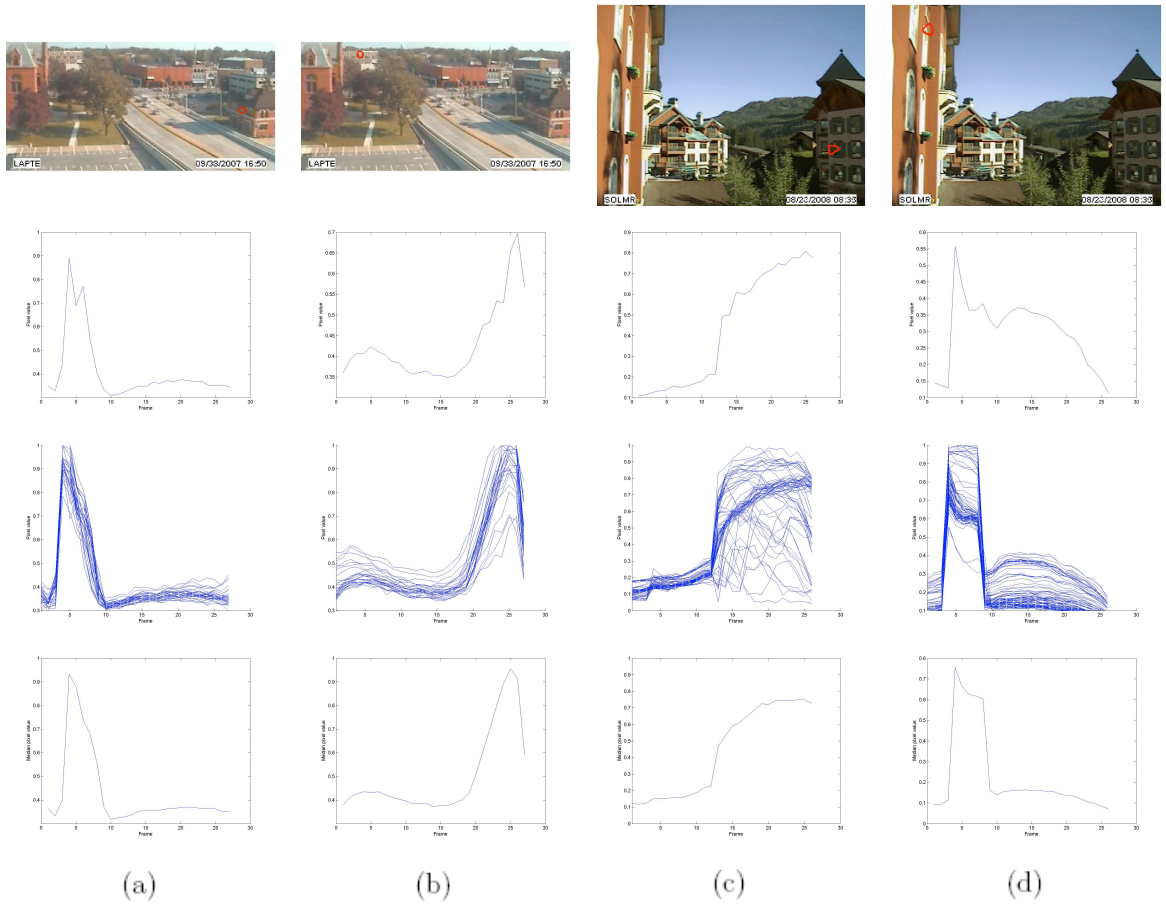


Figure 4.1: Appearance profiles of superpixels versus interior pixels. Each column shows a scene with a superpixel highlighted in red. The second row shows the appearance profile of a sample pixel within the highlighted superpixel. The third row shows the appearance profiles of all individual pixels within the superpixel. The fourth row shows the superpixel's appearance profile.

4.2 Expected Brightness of a Surface

Given a normal $\vec{N}$ for a surface, we can predict the surface's expected brightness throughout a day. According to the Lambertian lighting model, the intensity I of surface's brightness is given by

$$I = ba + b\vec{N} \cdot S \tag{4.1}$$

where b is the albedo of the surface, a is the ambient light, and S is the direction of sunlight. Assuming that ambient light, and the intensity of sunlight $|S|$ remain constant throughout the day, the surface is brightest when $\vec{N} \cdot S$ is largest; i.e., when the sunlight is most in the direction of the surface's normal.

We consider only sunny days in this thesis, since clouds and rain may block the path of sunlight. We look at images taken about every half hour between sunrise and sunset. This amounts to about 25 frames per scene; which is a large enough sampling rate to yield smooth brightness curves, while not so large as to take up too much memory. We exclude nighttime images, since the sun is not directed toward the scene so there is no additional information about surface normals to be gained. Also, at night, there is often artificial illumination from street lamps, etc., that could cause artifial peaks in appearance profiles.

Note Figure 4.1 (a) and (d) have brightness plots that reach their peak in the morning. Thus, we would expect the highlighted superpixels to have surface normals with large east components. Geolocating the webcams that generated the images verifies that this is in fact true [6], and as we will show in our results, our algorithm does calculate surface normals with high east components for those particular superpixels.

4.3 Calculating Partial Normals

Based only on the time at which a superpixel reaches its global maximum brightness, we can calculate the superpixel's *partial normal*, or north and east components. Given the time alone, we know only *when* a superpixel is brightest, but not *how* bright it is at that time. Thus, two superpixels could be brightest at the same time, but one

could be brighter than the other. For example, consider a superpixel on an east-facing building wall, and another superpixel on a slanted roof attached to that wall. They both are brightest at the same time, namely when sunlight is facing east. However, the wall will be brighter at that time, since its normal is facing directly east, while the roof's normal contains a larger upward component. Thus, if we assume vertical walls (i.e. assume the up component of normals is 0), we can calculate the east and north components.

Given a surface normal and the scene's geographic location, we can calculate its expected appearance profile. We first obtain the latitude, longitude, and offset from GMT [6]. Given these, and the local times during which the images were captured, we can obtain the azimuth and zenith angles of the sun throughout the day [18]. Geographic information is available for scenes from AMOS. We then calculate the direction of sunlight $S = (x, y, z)$, given the azimuth angle (ϕ) and zenith angle (θ), at a given time as follows.

$$x = sin(\theta)cos(\phi) \tag{4.2}$$

$$y = sin(\theta)sin(\phi) \tag{4.3}$$

$$z = 0 \tag{4.4}$$

Recall that for partial normals, we set z to 0. We test potential surface normals every two degrees on the NE-plane for matching global max times; a total of 180 potential normals. We test appearance profiles of $N \cdot S$ for N every two degrees on the xy-plane. Note that $N \cdot S$ may be negative if the sunlight direction is away from a building; we set negative brightness values to 0. We return the normal with the closest global max time to that of the the superpixel as the superpixel's surface normal.

This has the advantage of a small speed-up over calculating the complete normals, since we test every two degrees in only two dimensions (north, east) where calculating complete normals tests every two degrees in all three dimensions. Depending on the application, partial normals may be sufficient.

4.4 Calculating Complete Normals

Given the entire appearance profile, we may calculate the complete surface normal, i.e. all three components, of a surface. We no longer constrain z to be 0, and we test every two degrees in the entire 3-D space. Here, given zenith angle θ of the sun, the z component of the direction of sunlight is

$$z = cos(\theta) \tag{4.5}$$

Rather than comparing just the time of maximum brightness, we compute an error score between the entire appearance profiles of the superpixel and potential surface normal. In particular, we use the sum of absolute value differences as our measure, because they do not penalize heavily for large errors. It is important not to penalize heavily for large errors in a few special cases, such as shadows. If a shadow is cast over an object, that object will remain dark for the duration of the shadow. We discuss shadows further in our results.

We return the surface normal that yields the smallest error score. Calculating the complete normal takes slightly longer, but produces all three components of the surface normal, which may be necessary depending on the application.

Chapter 5

Visualization Tool

We present a software tool for easy visualization of surface normals. The tool has two overall functions: viewing information for individual selected superpixels, and querying for superpixels with normals within a specified range.

5.1 Viewing Information for Selected Superpixels

The user may choose either partial or complete normals to be calculated, in the "Method" box. The user selects a superpixel to highlight in the top-left frame. When the user selects a superpixel by clicking on it, a plot of the superpixel's appearance profile appears in the top-right frame. The user may also click on the appearance profile plot to show the scene at the time defined by the x-coordinate of their click location. This is useful if the user wants to investigate what is happening to a superpixel at a particular time; for example, when there is a peak in its appearance profile.

Upon selecting a superpixel, a plot of the best-matched surface normal's expected appearance profile appears in the bottom-right frame. Recall that when calculating a superpixel's partial normal, the best-matched normal is the one whose expected appearance profile reaches its global maximum at the same time. When calculating complete normals, the best-matched surface normal is the one for which the sum of absolute value differences is smallest. Figure 5.1 shows a screenshot in which three superpixels are selected.

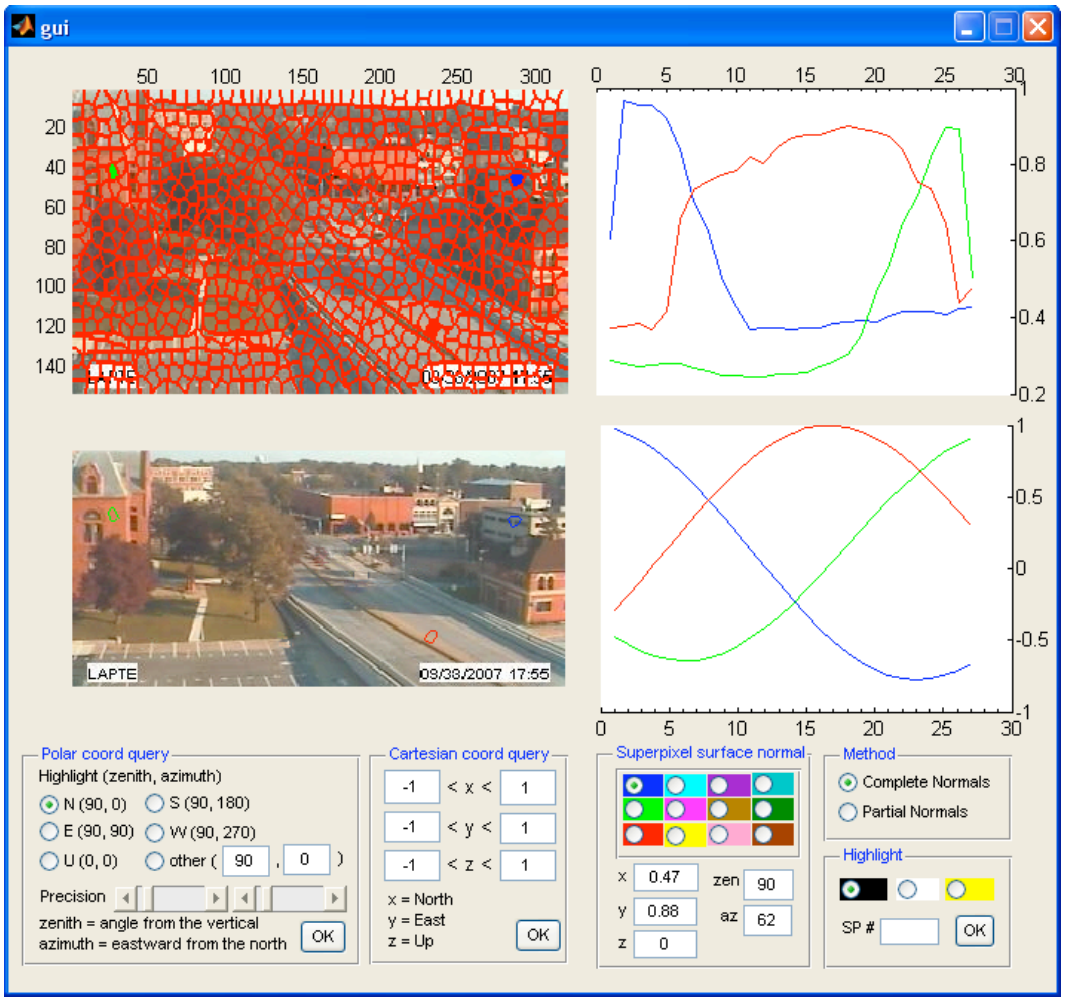


Figure 5.1: Viewing appearance profiles for selected superpixels. Here, three superpixels are selected, shown highlighted in blue, green, and red.

The user may click a radio button in the "Superpixel surface normal" box to view the corresponding-colored superpixel's calculated normal. In Figure 5.1, the blue radio button is selected, which corresponds to a superpixel on the white wall in the right half of the image. The selected superpixel is also shown highlighted in blue in the bottom-left frame. The normal is displayed in both Cartesian coordinates (x, y, z), where x is North, y is East, and z is Up; and polar coordinates $(zenith, azimuth)$, where the zenith angle is the angle from the vertical and the azimuth angle is the angle eastward from the north. The blue-colored superpixel's surface normal is $(.47, .88, 0)$ in Cartesian coordinates, or $(90°, 62°)$ in polar coordinates.

5.2 Surface Normal Queries

The user can also submit queries for superpixels with surface normals within a specified range. Depending on the user's preference, the query can be entered in Cartesian or polar coordinates. When querying by polar coordinates, the user may control the precision with which surface normals match the query. In Cartesian coordinate queries, the user explicitly sets a range for x, y, and z coordinates. The superpixels matching the query are shown highlighted in the bottom-left frame. A sample query is shown in Figure 5.2

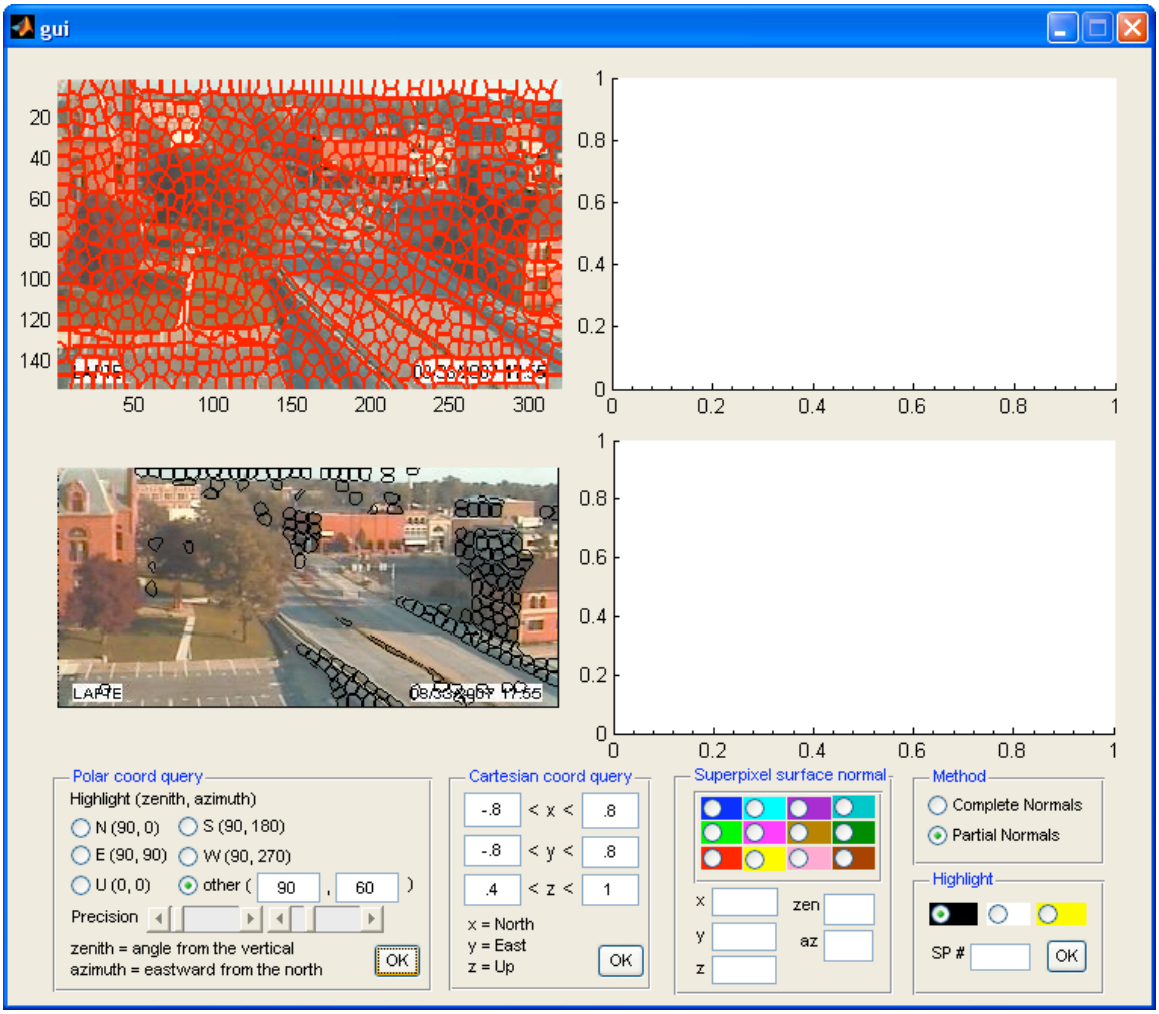


Figure 5.2: A surface normal query in polar coordinates.

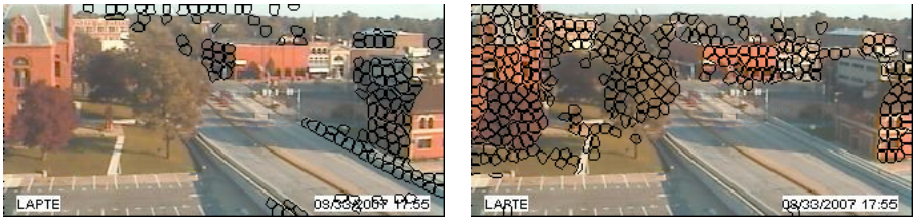
Chapter 6

Results

In general, both the partial and complete normal calculations are reasonably accurate. However, the calculations are thrown off in certain cases, which we discuss below. Here, we present a comprehensive analysis of one particular scene. See Appendix A for surface normal queries on other scenes. Finally, we also compare our superpixel results to results using individual pixels.

6.1 Partial Normals

Recall that partial normals operate under the "vertical wall hypothesis", and thus hold the z component equal to 0; or in polar coordinates, hold the zenith angle to 90°. The scene shown in Figure 6.1 is geographically oriented such that the camera is facing approximately SE; so walls facing the camera face NW, and perpendicular walls facing the street face 90 degrees clockwise, or approximately NE.



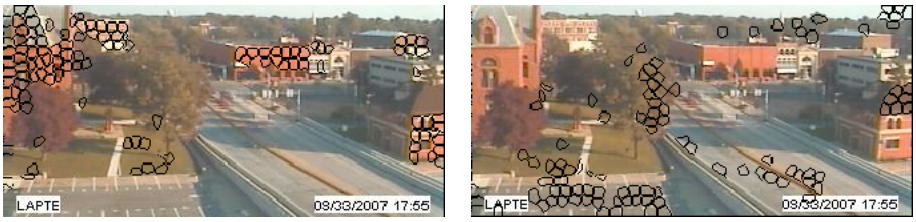
(a) (b)

Figure 6.1: Partial normal queries

Figure 6.1 (a) shows a cropped screenshot of superpixels with normals close to $(90°, 70°)$, or approximately NE. Note that the query returns the walls we would expect, namely the ones facing the street. Also, under the vertical wall hypothesis, the superpixels on the roof of the red building on the left side of the image are assumed to be facing directly outwards, i.e. have a zenith angle of 90°, so they too are returned as part of the result. Similarly, Figure 6.1 (b) shows superpixels that are facing approximately NW. Again, superpixels on the roof are returned as part of the result. This query captures all of the superpixels we would expect, but it also captures some superpixels that we might not expect to see – superpixels that lie on trees. This is because trees are difficult to segment into meaningful superpixels; they contain many leaves that may be small, have different surface normals, have space between them, and move frequently due to wind or other weather conditions. Thus, their appearance profiles may classify them as facing NW, but this is not a meaningful result.

6.2 Complete Normals

In calculating complete normals, we test every two degrees in *both* zenith and azimuth angle, i.e. we test every two degrees in the entire 3-D space. Figure 6.2 shows polar surface normal queries for $(90°, 300°)$ and $(35°, 300°)$. Because we are querying complete normals, we can distinguish between 90° and 35° in the zenith angles. For $(90°, 300°)$, we would expect to see vertical walls facing NW; and for $(35°, 300°)$, we would expect to see roofs or other *non*-vertical elements that also face NW.



(a) (b)

Figure 6.2: Complete normal queries

For the red building in the bottom right and the *vertical* wall of the red building on the left, this holds. The vertical walls are highlighted for the $(90°, 300°)$ but the roof of the right building is not; as we would expect. The same roof is highlighted for the $(35°, 300°)$ query, whereas the vertical walls are not; as we would expect. However, note that the left red building's NW-facing roof is incorrectly classified.

In Figure 6.3, two superpixels are highlighted; one from the incorrectly classified NW-facing roof (highlighted in blue), and one from the correctly classified NW-facing roof (highlighted in green). The appearance profile plot has just been clicked on, and the cursor indicates that the bottom-left frame is showing the scene at frame $x = 6$. Notice that around frame $x = 5$, the blue appearance profile begins to decrease, but the green appearance profile increases. We see from the 6th frame that this is because the blue superpixel is under a shadow. Thus, the blue superpixel's appearance profile does not begin to increase until later than that of the green superpixel, so it is classified incorrectly.

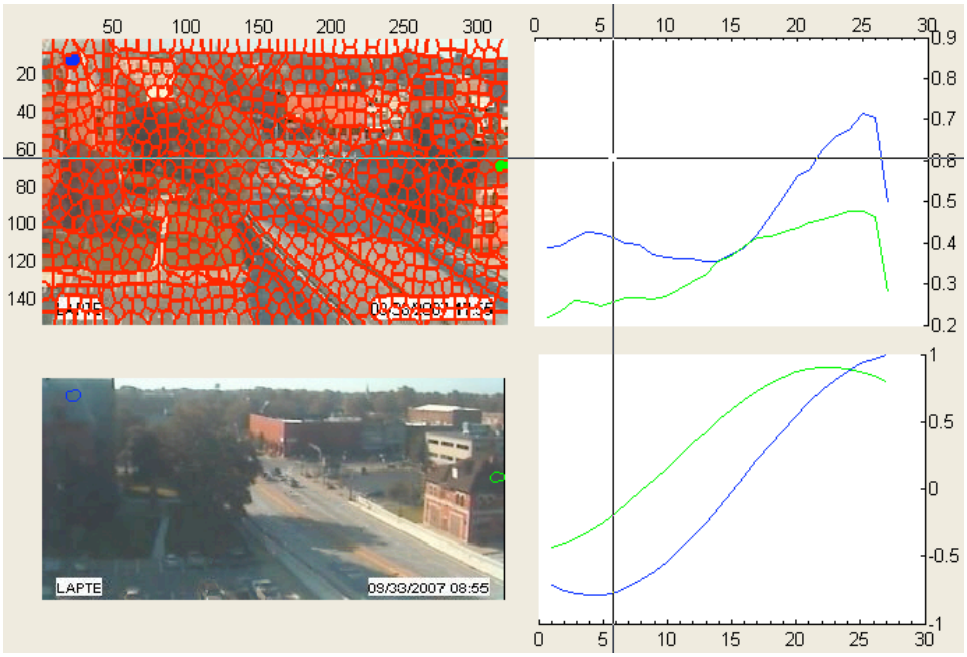


Figure 6.3: Misclassification of surface normal due to a shadow.

Shadows may also cause the misclassification of ground superpixels. Consider the grass superpixels in this scene, one of which is highlighted in Figure 6.4. We see in the appearance profiles that the blue-highlighted grass superpixel comes under a

shadow from about frame 9 to frame 22. This causes its appearance profile to be matched to the blue one shown, which is shifted to the right from the green, more accurate, match.

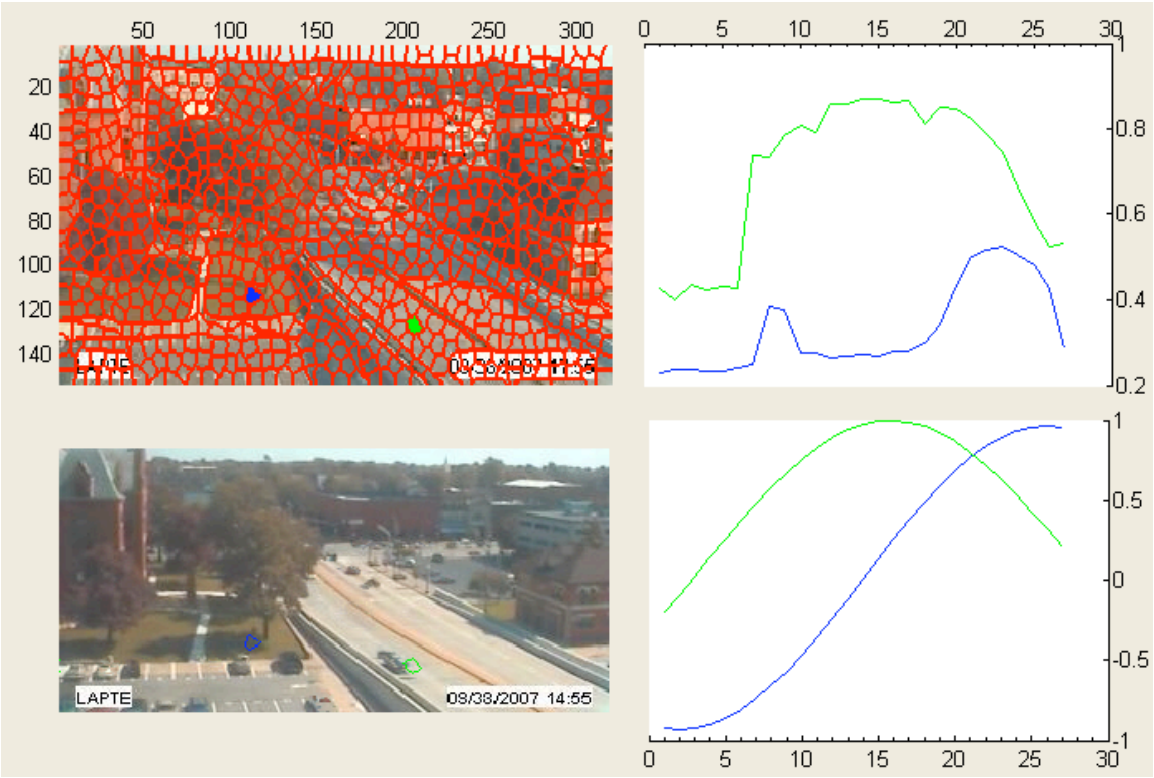


Figure 6.4: A ground superpixel misclassified because of shadows

It is also worth noting circumstances that do *not* lead to incorrect surface normal calculations. Figure 6.5 shows a street superpixel whose appearance profile contains a sharp trough where a car passes through, as is seen in the frame on the bottom-left. Even though that particular image is the median of several images taken at the same time (hence the "ghost car" artifacts seen in the parking lot at the bottom left), some cars are still predominant because of consistent traffic at that time and location over multiple days; perhaps rush hour. However, the surface normal that is returned is fairly accurate: the z (up) component returned is 0.85. Because we take the sum of absolute value differences to match appearance profile curves, large errors such as the ones at the sharp peaks where cars are passing, are not penalized heavily. A different error measure, such as sum of squared differences (SSD) would penalize more for large errors. Thus, short-duration shadows may not throw off the algorithm too badly, but

as we saw in the case of the grass superpixels, long-duration shadows often cause an incorrect appearance profile classification with this error measure.

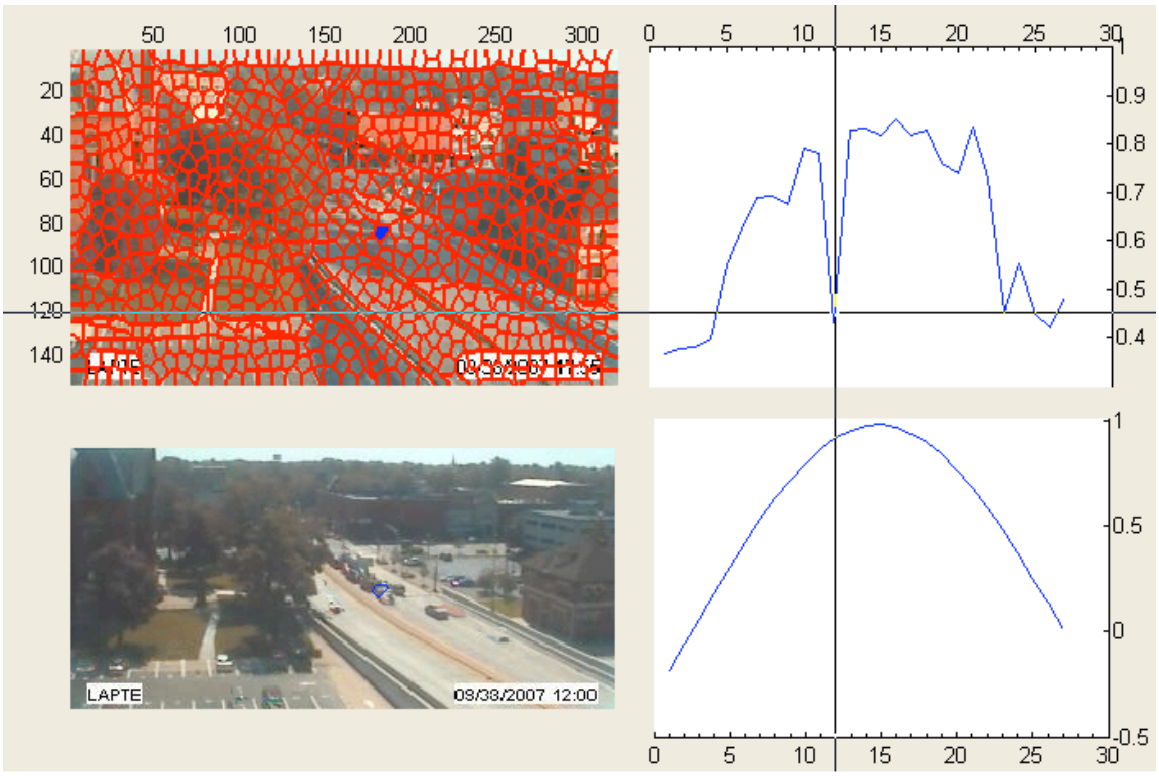


Figure 6.5: Short-duration deviations, such as the trough in this profile, do not have a significant effect on the surface normal result.

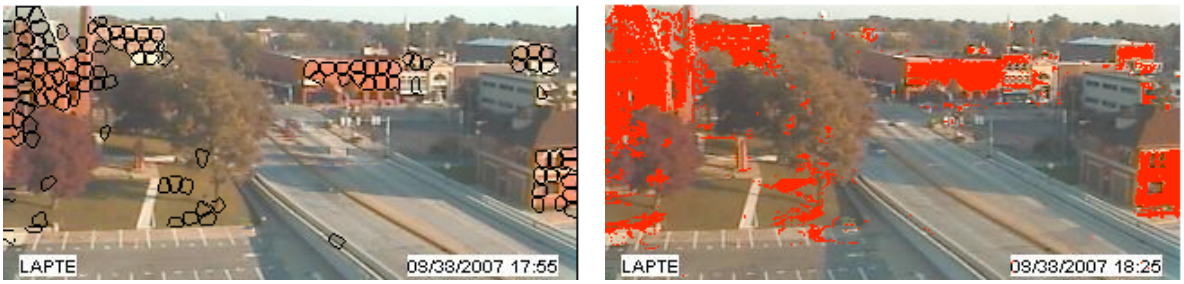
6.3 Comparison to Individual Pixel Normals

We apply the same algorithms for calculating partial and complete normals to individual pixels. Below, we show comparisons of pixel and superpixel normals for a few example queries. The pixel surface normal results are often noisier than the superpixel results; more so in some scenes than others.

Though in some cases, the pixel results may be as good, or almost as good, as the superpixel results, superpixel normals are always faster to calculate. Recall that there are about 1000 times as many pixels as there are superpixels, so superpixel calculations are significantly faster.

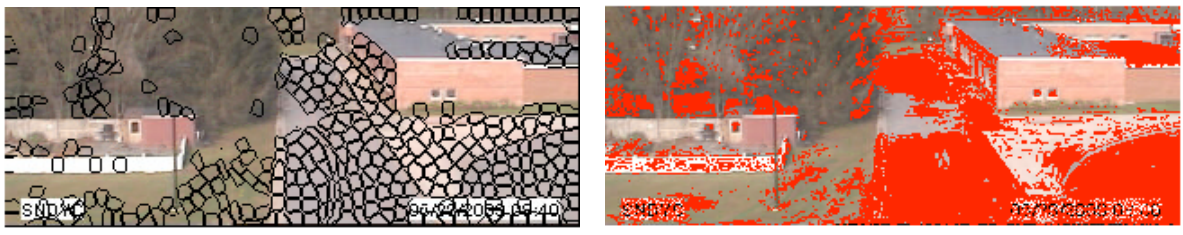
(a) (b)

Figure 6.6: Scene 1: Partial normal query (North-West)



(a) (b)

Figure 6.7: Scene 1: Complete normal query (North-West)



(a) (b)

Figure 6.8: Scene 2: Partial normal query (South)

Figure 6.9: Scene 3: Complete normal query (East)

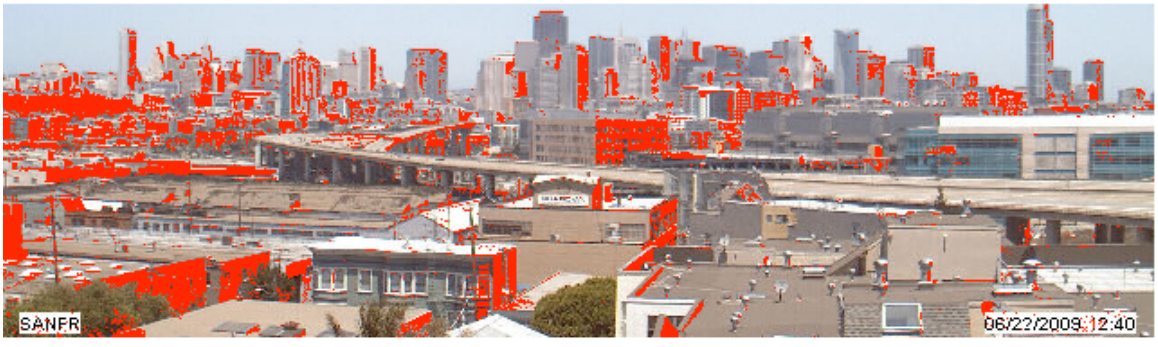


Figure 6.10: Scene 3: Complete normal query (East)

Figure 6.11: Scene 3: Complete normal query (North)

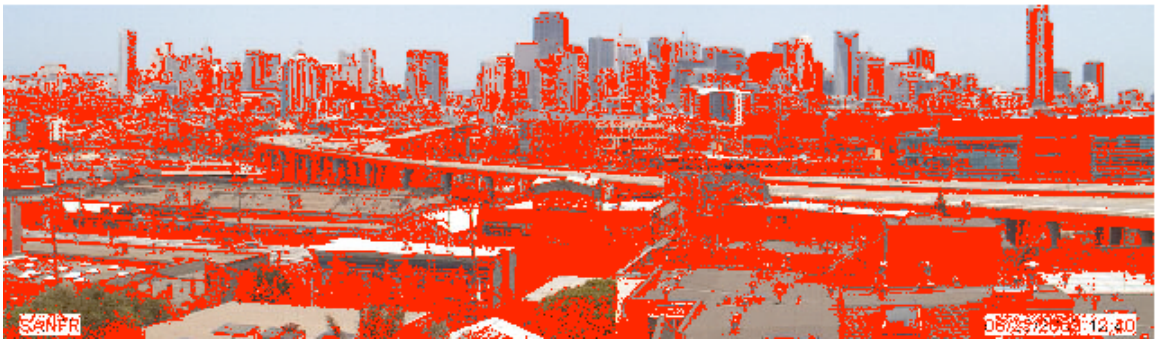


Figure 6.12: Scene 3: Complete normal query (North)

Chapter 7

Conclusion and Future Work

In general, the partial normal algorithm yields good results assuming vertical walls, and the complete normal algorithm gives good results for complete, 3-D normals. The algorithm may fail in a few situations; most notably, long-duration shadows. One area for future work may be to improve the error measure in appearance profile classification to reduce the effect of shadows. Also, combining our results with those of Hoiem et al. in "Automatic Photo Pop-up" [4] may produce more accurate results.

7.1 Reducing the Effect of Shadows

As shown in our results, long-lasting shadows may cause a misclassification of a superpixel's appearance profile. If we could reconstruct our images without shadows, or detect where they are and simply ignore them, we might get better results than with our current method which leaves them as they are, but tries not to penalize heavily for them in our error measure. We could also look into taking the median over several time-lapses of various days throughout the year, rather than a few time-lapses that are all around the same day. At various points throughout the year, the varying location of the sun could cast shadows in different directions, and hence the shadows may cancel each other out.

7.2 Combining with "Automatic Photo Pop-up"

Shadows have the greatest effect on ground pixels, since they are often surrounded by many buildings. Recall "Automatic Photo Pop-up" segments a scene into three components: sky, ground, and anything else (such as buildings); and creates a 3-D photo popup. Their algorithm could be run as a preprocessing step to segment out the ground and sky, and the remaining superpixels could be classified geographically using our algorithm. Alternatively, we could generate a 3-D photo popup using their algorithm, and define its geographic coordinate axis directions with our algorithm.

Appendix A

Surface Normal Queries

Figure A.1: Scene 1: Partial normal query (North-East)

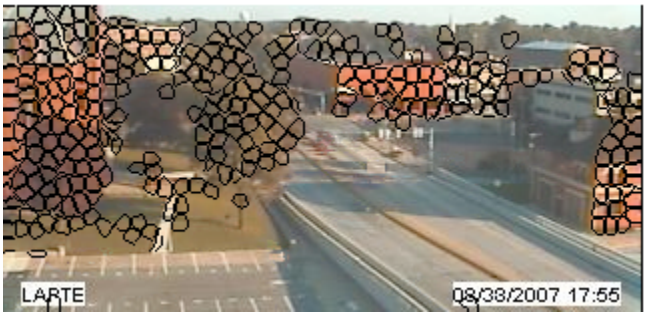
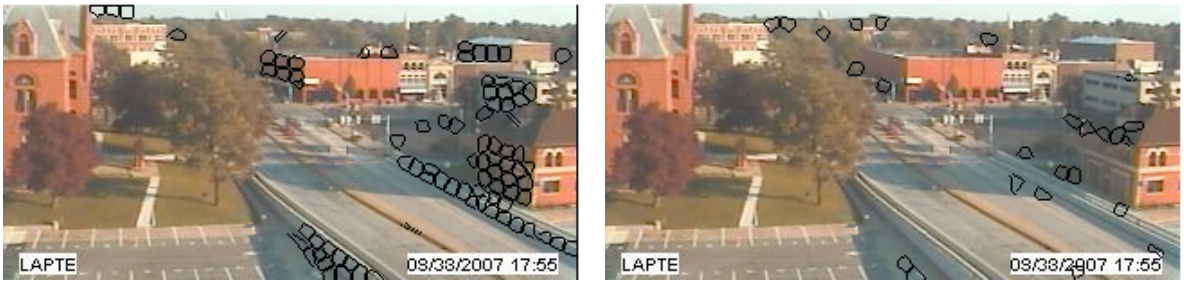
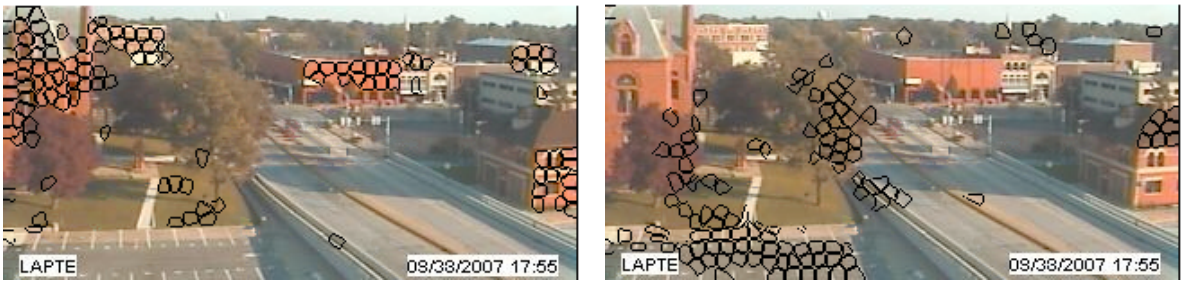


Figure A.2: Scene 1: Partial normal query (North-West)



(a) (b)

Figure A.3: Scene 1: Complete normal query (North-East). (a) Zenith = 90°. (b) Zenith = 45°



(a) (b)

Figure A.4: Scene 1: Complete normal query (North-West). (a) Zenith = 90°. (b) Zenith = 45°

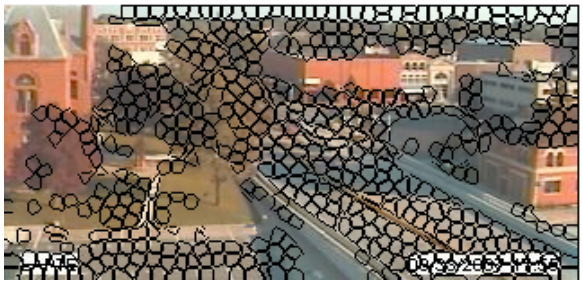


Figure A.5: Scene 1: Complete normal query (Up)

Figure A.6: Scene 2: Partial normal query (South). Note that partial normals cannot distinguish the ground from walls, since it holds the upward component of surface normals to be 0. However, it distinguishes between the directions that vertical walls face. Here, of all of the *vertical walls*, it only returns the left wall of the large building, which is the only wall that faces south.

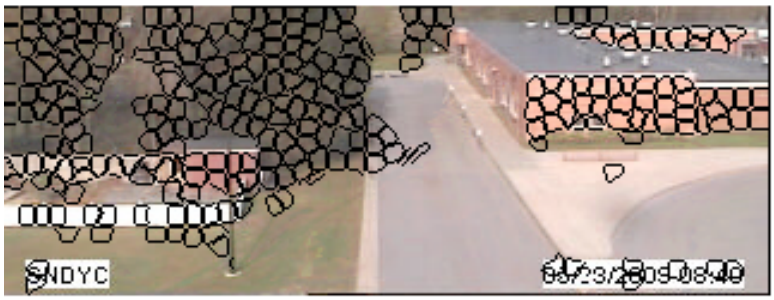


Figure A.7: Scene 2: Partial normal query (East)

Figure A.8: Scene 3: Complete normal query (East)

Figure A.9: Scene 3: Complete normal query (North)

Figure A.10: Scene 3: Complete normal query (Up)

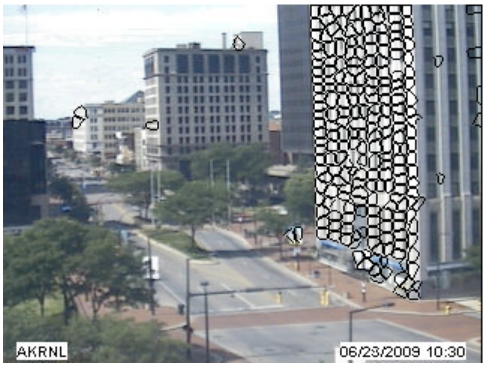


Figure A.11: Scene 4: Complete normal query (East)

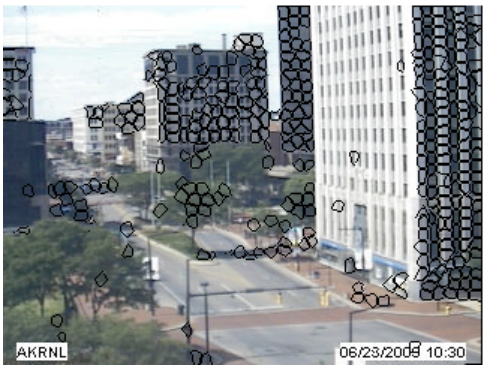


Figure A.12: Scene 4: Complete normal query (North)

Figure A.13: Scene 4: Complete normal query (Up)

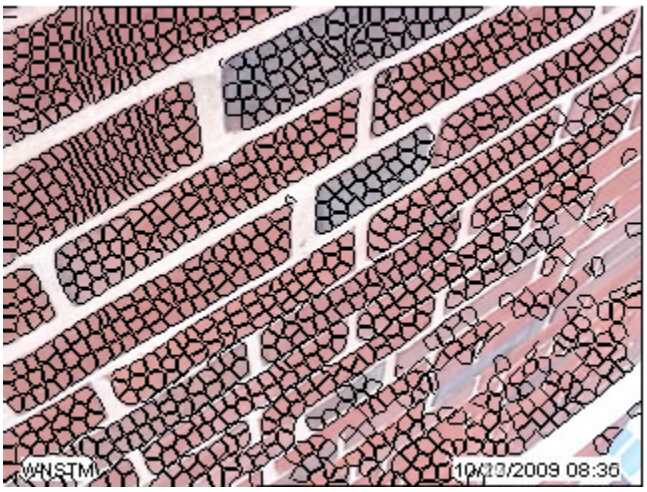


Figure A.14: Scene 5: Complete normal query $(90°, 5°)$

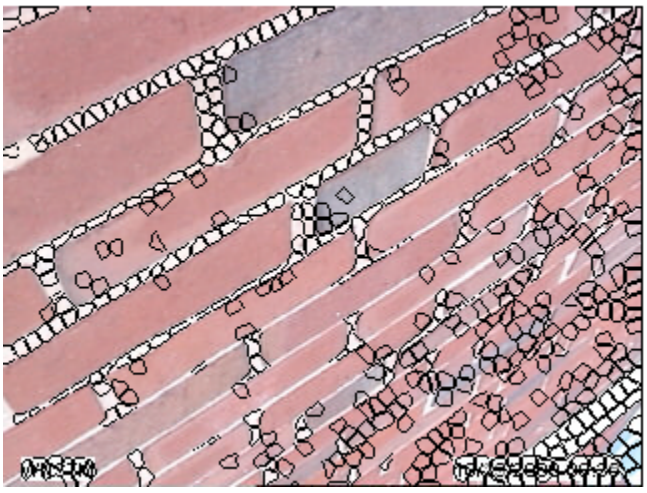


Figure A.15: Scene 5: Complete normal query $(90°, 12°)$. This example shows how precise surface normal calculations can be – here, these two queries' azimuth angles are only 7° apart.

References

[1] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *Int. J. Comput. Vision*, 80(3):300–316, 2008.

[2] Xuming He, Richard S. Zemel, and Debajyoti Ray. *Learning and Incorporating Top-Down Cues in Image Segmentation*. 2006.

[3] Aaron Hertzmann. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1254–1264, 2005. Member-Seitz, Steven M.

[4] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 577–584, New York, NY, USA, 2005. ACM.

[5] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *Int. J. Comput. Vision*, 75(1):151–172, 2007.

[6] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. pages 1–6, 2007.

[7] J. Kaufhold, R. Collins, A. Hoogs, and P. Rondot. Recognition and segmentation of scene content using region-based classification. pages I: 755–760, 2006.

[8] S.J. Kim, J.M. Frahm, and M. Pollefeys. Radiometric calibration with illumination change for outdoor scene analysis. pages 1–8, 2008.

[9] Sanjeev J. Koppal and Srinivasa G. Narasimhan. Clustering appearance for scene analysis. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1323–1330, 2006.

[10] Bogdan Kwolek. Object segmentation in video via graph cut built on superpixels. *Fundam. Inf.*, 90(4):379–393, 2009.

[11] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

[12] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[13] Alastair P. Moore, Simon J. D. Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[14] G. Mori, Xiaofeng Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–326–II–333 Vol.2, 2004.

[15] Srinivasa G. Narasimhan, Chi Wang, and Shree K Nayar. All the images of an outdoor scene. In *ECCV 2002, LNCS 2352*, pages 148–162, 2002.

[16] Robert Pless Nathan Jacobs, Brian Bies. Using cloud shadows to infer scene structure and camera calibration. *CVPR 2010 (under review)*, 2010.

[17] S.K. Nayar and S.G. Narasimhan. Vision in Bad Weather. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 820–827, 1999.

[18] Ibrahim Reda and Afshin Andreas. Solar position algorithm for solar radiation applications. Technical Report NREL/TP-560-34302, National Renewable Energy Laboratory, January 2008.

[19] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 10, Washington, DC, USA, 2003. IEEE Computer Society.

[20] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 101, New York, NY, USA, 2007. ACM.

[21] Xiaofeng Wang, Xiao-Ping Zhang, Ian Clarke, and Yury Yakubovich. A new gaussian mixture conditional random field model for indoor image labeling. In *IMCE '09: Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pages 51–56, New York, NY, USA, 2009. ACM.

[22] Y. Weiss. Deriving intrinsic images from image sequences. pages II: 68–75, 2001.

Vita

Rachel Tannenbaum

Date of Birth	August 31, 1987
Place of Birth	Chicago, IL
Degrees	B.S. Computer Science, December 2009 M.S. Computer Science, December 2009

December 2009

Segmenting Videos for 3D Structure, Tannenbaum, M.S. 2009