BALLOONSAT: DESIGN, IMPLEMENTATION, AND APPLICATION OF A LOW-COST TETHERED WEATHER BALLOON REMOTE SENSING STATION

Dustin Blackwell, John Hupton, Robert Hursig, Jessica Kiefer,
Matt Schlutz, Scott Seims, Student Authors
Dr. John Saghri, Research Advisor

ABSTRACT

Design, implementation, and applications of a low-cost tethered helium-filled weather balloon remote sensing station called BalloonSat is discussed. BalloonSat remote sensing station has been designed, implemented, and continuously upgraded and improved by Cal Poly students in the remote sensing course taught by Dr. John Saghri over the past few years. BalloonSat provides the raw multispectral imagery data needed for a number of research projects. The onboard visible, infrared, and thermal sensors have wireless remote pan and zoom with real-time video monitoring capabilities. The collected remotely sensed data is transmitted to the ground station through a low-power wireless communication channel. Initial launches of the new and improved BalloonSat have demonstrated improved physical stability and wind-tolerance than previous designs. The collected multispectral data is exploited to provide useful information pertinent to the local terrain. The long term goal of this research is to develop digital image processing schemes to exploit the remotely sensed multispectral imagery collected by the BalloonSat in various applications useful for the campus and the surrounding community. The current application being investigated is the assessment of the status of the campus open parking lots. A robust image exploitation scheme has been successfully developed to identify and count vehicles in campus open parking lots. In the near future, with the recently added thermal imaging capability, more challenging and multi-sensor exploitation schemes will be developed for such applications as campus security, campus traffic monitoring, and campus tour.

INTRODUCTION

Remote sensing is a challenging multidisciplinary science with numerous commercial and military applications. The objective of this research is to develop a low-cost tethered helium-filled weather balloon remote sensing station capable of collecting and exploiting multispectral imaging data from the local terrain for various civilian applications. The BalloonSat remote sensing station has been designed, implemented, and continuously upgraded and improved by Cal Poly students in the remote sensing course taught by Dr. John Saghri over the past few years. The system is comprised of a tethered helium-filled weather balloon and an aerial multispectral imaging platform. The aerial multispectral imaging system, or "payload," consists of an upper box, which stores the electronics and batteries, a C-bracket which provided the stability for the panning motion, and a lower box which houses the two cameras, shortwave infrared lens and changing mechanism, and also a shutter release servo. The helium balloon is filled to about 8 feet in diameter and secured to the ground by four tethered lines. The two cameras and the interchangeable short wavelength infrared lenses provide an inexpensive multispectral imaging capability. The cameras have wireless remote pan and zoom with real-time video monitoring capabilities. The collected remotely sensed data is transmitted to the ground station through a low-power wireless communication channel. The multispectral data is then processed to provide useful information pertinent to the local terrain. Figure 1 shows the BalloonSat and samples of its wide-angle and telephoto visible imagery.

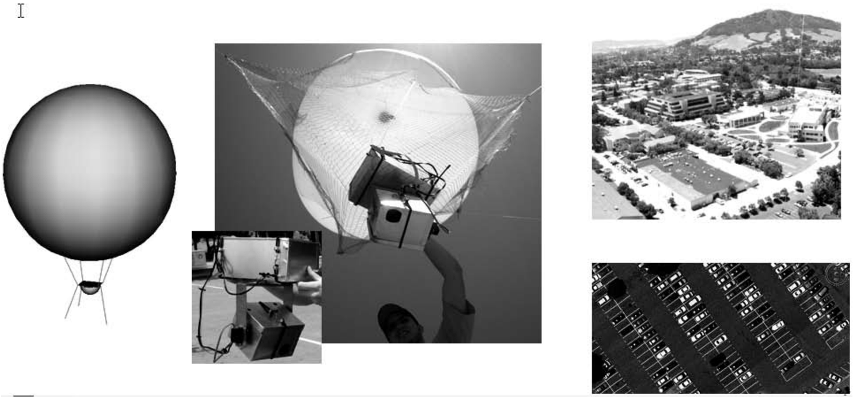

Figure 1. BalloonSat and samples of its wide-angle and telephoto visible imagery

The long term objective of this research is to develop digital image processing schemes to exploit the remotely sensed multispectral imagery collected by the BalloonSat in various applications useful for Cal Poly campus and the surrounding community. The current application being investigated is to provide a tool for the assessment of the status of the campus open parking lots. A robust image exploitation scheme to identify and count vehicles in a campus open parking lot has been successfully developed. New applications utilizing the recently added thermal imaging capability of BalloonSat will be developed in the near future.

HARDWARE SYSTEM DESIGN: PAYLOAD

The various functions can be broken down into ground station and BalloonSat platform areas. Figure 2 shows the electrical, communication, and control systems for the BalloonSat. The ground station houses the Radio Control (R/C) transmitter for control of the pan, tilt, take picture, and reset servo commands. The commands are sent to the R/C receiver on the balloon that is connected to the microcontroller which interprets the signals and activates the servos appropriately. The digital, visible light camera stores its images on the SD memory card when the servo physically operates the shutter. At the same time, the thermal camera in the camera box outputs a live video stream. The thermal camera's video stream is transmitted in real time through a video transmitter in the upper box down to a video receiver at the base station, where it is displayed on a laptop via a TV tuner card. Operators use the images from the video feedback to aim the camera while the payload is aloft.

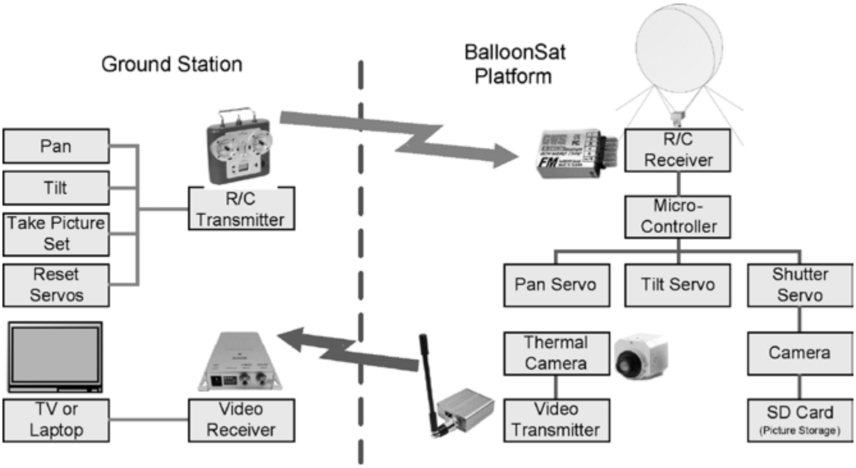

Figure 2. BalloonSat System Block Diagram

Chassis Design

The chassis is designed to house the visible light camera, infrared (IR) lens and servo, and shutter release servo. The cameras have pan and tilt servos that have been designed and implemented specifically for this application. The dimensions of the lower housing were changed to allow for the added thermal camera and the removal of the IR lens. The components are constructed from aluminum sheet stock.

Thermal Camera

The new addition to the BalloonSat platform is a commercial thermal camera from FLIR Systems in Goleta, California. The camera has 320 by 240 pixel resolution and a 19mm lens. This compact camera has a cross section of two inches by two inches and weighs 155 grams. It has an uncooled core and a spectral response range of 8-14 micron. The added spectral information gathered by this compact and light weight camera will provide useful information that will more easily and efficiently accomplish project goals. While able to be remotely controlled via an RS-232 connection, the camera can also operate autonomously, outputting an analog video signal which is optimized for current light conditions. The camera operates at supply voltages from 5 to 24 Volts and dissipates an average of 1.6 Watts. It is also shock tolerant, capable of withstanding a 70-g shock pulse and operates near room temperature without requiring costly cooling system [1]. The analog video output of the thermal imager is a 320 by 240 pixel, NTFS encoding at 30 frames per second.

The high-resolution thermal images captured by this camera improve the ability to differentiate vehicles from the background due to their unique heat signatures. Unlike visible images, thermal images will be invariant to the type, color, or position of vehicles within the parking lot. Moreover, the surrounding blacktop and vegetation will also have distinct heat signatures. Hence, thermal images provided by this camera will significantly increase accuracy in the image processing as compared to when the platform was limited to visible and near-IR wavelengths.

Stability of the Balloon Rig

Experience from previous launches raised several alarms about the stability and security of the weather balloon platform and its expensive payload. Previously, the payload was suspended from the balloon via netting over the balloon much like a hammock. This design did not provide adequate stability for the payload during wind gusts. While the netting was lightweight, it increased the pressure exerted on the balloon where the net met with it (especially on the side opposite of the incident wind). The net was also prone to tangling and was very difficult to make even slight adjustments to its orientations during the launch process. To overcome these difficulties, the net was replaced with a rain fly, lowering pressure exerted on the balloon and securely holding the weather balloon in place under harsh flying conditions. The rain fly was not prone to tangling, did not contain any sharp cable ties, and was much easier to reposition during the launch process. As the weight of the rain fly and new tether system, which will be discussed shortly, was much increased, the diameter of the weather balloon and the amount of helium needed to be increased. Hence, a 13'x15'x7', 6 lb rain fly was purchased to accommodate the increased balloon diameter. Instead of suspending the payload, heavy-duty nylon strapping was sewn to the rain fly's straps and to straps looped through newly created holes through the upper housing's four corners. The integrity of the sewn connections provided a tensile strength, which far exceeded the requirements of this project (two people pulling in opposite directions could not sever the connection). Refer to Figure 3 below for the implemented rain fly system. Notice, the additional rope inserted around the horizontal circumference of the balloon, used to better contain the balloon while aloft.

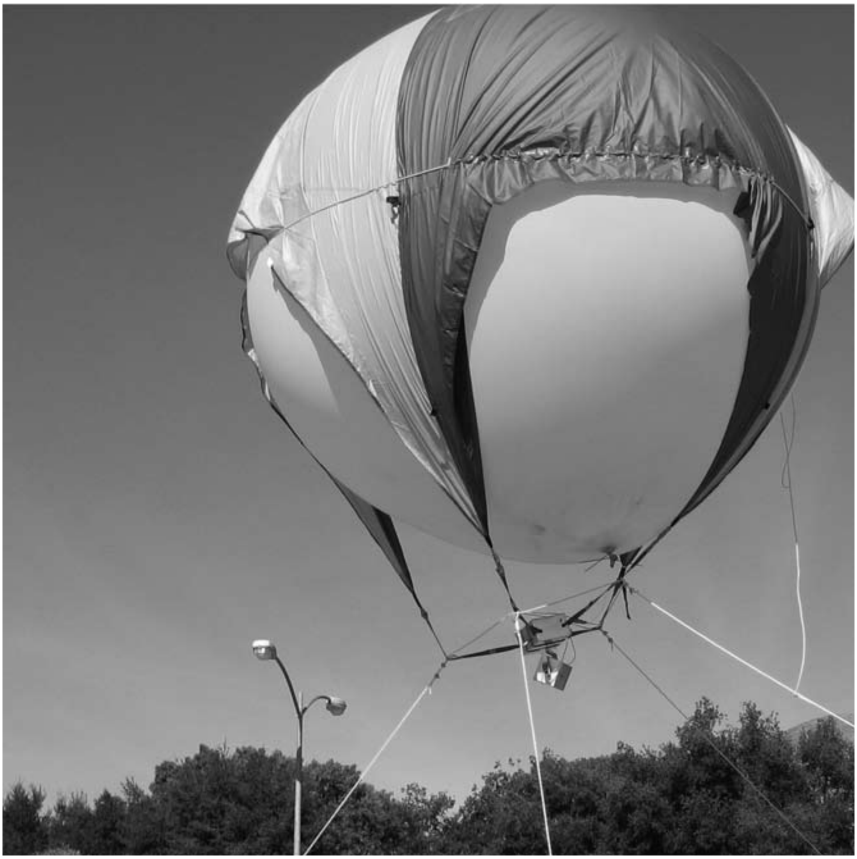

Figure 3. Modified Balloon Rig

Electronics

In order to organize the goals and capabilities of the electronics for the BalloonSat, a generalized block diagram for both the onboard and ground station electronics was created as seen in Figure 2. A controller board for the BalloonSat was designed and implemented. Figure 4 shows the controller board during construction. This board manages power and controls everything onboard the balloon. This circuit was designed to accomplish all the capabilities that were outlined in the system block diagrams. One important addition to the circuit was a 12 volt regulator that could be turned on and off from the microcontroller. This regulator supplies power to the video transmitter, which happens to be somewhat of a power hog. The ability to turn this transmitter on and off gives us the option to go into a low power mode to conserve limited onboard power. After the schematic was built and tested physically on a breadboard, it was exported to a board layout. Some of the goals for the board design were to keep its size under two inches square, minimize the connections on the top side of the board, and to include component labeling on the top side of the board.

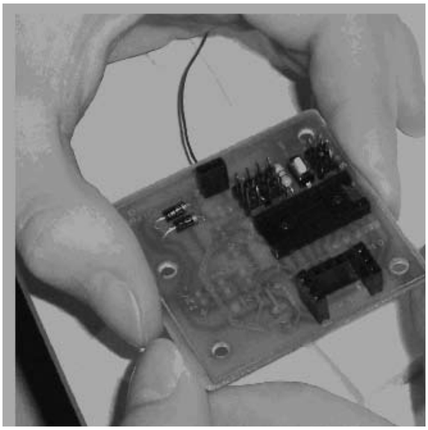

Figure 4. BalloonSat Controller Schematic

SOFTWARE SYSTEM DESIGN -- DATA EXPLOITATION

The goal of the software research portion of the BalloonSat project is to develop image processing algorithms and techniques to extract physical features and information from the imagery taken aboard our aerial platform. Specifically, the main image processing goal is to develop a unique algorithm for extracting the number of parked vehicles from an aerial photograph of a parking lot. The eventual application would be to be able to launch a balloon platform satellite and collect parking lot populations automatically from images taken onboard the satellite.

With the addition of the thermal camera, the imaging capability of the BalloonSat now extends from the visible light (0.3 to 0.7 micron wavelengths) to the thermal range (8 to 14 microns wavelength). However, experimentation and exploitation of the thermal imagery has just started and the final combined visible/thermal results will be presented in the future publications.

Automatic Vehicle Detection and Counting Problem

Various methods have been used for vehicle extraction in several different kinds of applications using both visible and thermal imagery [1-8]. Many techniques involve using several different kinds of gray-level, color-level, or binary operations to create new images and then using a quantifiable metric to count the number of vehicles in the new image or images.

A human being would easily be able to discern the number of vehicles present in a picture of a parking lot, but writing algorithms that duplicate the visual and mental processes of a human is not a simple task. Computer software works in series of simple sequential tasks and enumerated quantities, so discerning a qualitative feature in a parking lot image involves many complex sequences of these simple tasks.

Nevertheless, there are many applications for algorithms like this one, which seek an automated way of extracting a useful feature from an image or set of images. Figure 5 illustrates some of the challenging issues to be considered relevant to identifying of cars in aerial photos of parking lots.

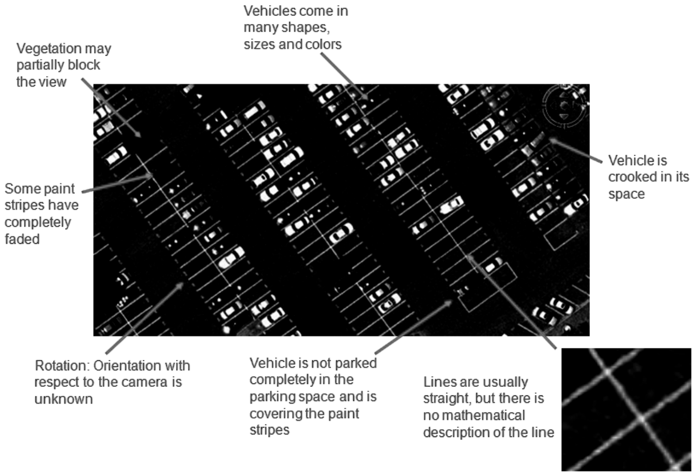
Vehicles come in many shapes, sizes and colors
Vegetation may partially block the view
Some paint stripes have completely faded
Vehicle is crooked in its space
Rotation: Orientation with respect to the camera is unknown
Vehicle is not parked completely in the parking space and is covering the paint stripes
Lines are usually straight, but there is no mathematical description of the line

Figure 5. Challenges for the automatic vehicle detection and counting

There are two common approaches for this task, Blobbing and Correlation Raster. The blobbing concept involves taking a grayscale or Red-Green-Blue (RGB) image, such as a visible light parking lot input image, and converting it into a binary image, consisting of only 1s and 0s. Ideally, all of the pixels that were 1s in the binary image would be pixels that were part of a vehicle, and all of the 0s would be cement, dirt, trees, or some other type of terrain. Thus, the blobs could simply be counted to yield a vehicle count in the image.

The process of counting blobs is algorithmically quite easy and fast, and has been implemented in many programming languages, platforms and in numerous different kinds of applications. The challenge in any blobbing method is in creating the binary image. There are two basic schemes, Base Image Subtraction and Manual Blobbing, which are commonly used for counting blobs.

Base Image Subtraction is a method that would be difficult to implement given BalloonSat's platform. The idea is to collect a set of baseline images in which there are no vehicles in the parking lot, then take pictures later when the vehicles are present. With both images converted from RGB to grayscale, the baseline images can be subtracted from the populated images and the pixels that have a grayscale value near 0 can be removed with thresholding (setting any pixel that is within some set distance of 0 to 0). The remaining pixels can therefore be "blobbed", the blobs can be counted and output as a vehicle count.

The difficulty in Base Image Subtraction is due to the fact that the scale, size and position of the baseline images must be nearly identical to the images taken later. If done with a mounted, near-permanent or predictable aerial platform this might be possible to achieve, but with the unpredictability of the BalloonSat platform, this method is most likely unattainable.

Manual Blobbing involves using a single input image (populated with vehicles) which is processed and converted to a binary image and then blobbed. This method actually involves a series of other embedded methods in order to convert the input image to a binary image. The difficulty in this method is due to all of the issues outlined in Figure 5, and is exacerbated by the parking lot lines. The parking lot lines, even when the cars park properly, often end up intruding into the binary image and blending multiple vehicle blobs into one another, yielding a very inaccurate vehicle count.

Other methods were attempted this year to filter out the parking lot lines for Manual Blobbing. The most successful involved using the Hough Transform to discover straight lines (parking lot lines) in the image and filtering out the pixels that contributed to these lines (or pixels that were very similar). The problem with this method is that a good amount of vehicle information was filtered out as well, including most white, gray, silver and light blue vehicles.

Correlation Raster is a vehicle count extraction method that uses a pre-extracted vehicle mask, a small image taken from another sample image which looked exactly like a vehicle. The mask is then run through the image in a Raster fashion (from the top-left to the bottom-right) and a discrete cross-correlation is calculated between the mask and the input image pixels it was placed upon. This correlation value is then plotted in a new image, which is filtered, based on a grayscale threshold, and the remaining pixel blobs are counted as vehicles. This method is quite slow, as it involved iterating through every single pixel in the input image and for each pixel, iterating again through all of the pixels located underneath the mask. This requires nearly n^3 (three nested loops) operations for an n by n image, as the size of the mask is approximately n in our test image set. The other major drawback of this method was that its results are not very accurate. For one, in our tests it only accounted for vehicles that matched the shape and color of the selected mask. Additionally masks could be used, but this would involve another Raster run, and given the number of different vehicle shapes and colors that need to be considered, the algorithm would very quickly become extremely slow to run, and still not necessarily be accurate. In addition to only partially accounting for all vehicle types, the method is not at all scalable. If the input images taken were not taken at *exactly* the same size and scale as the pre-extracted vehicle mask, then the method would not work at all, or a new mask would have to be manually created (which defeats the purpose of automation).

Developed Vehicle Identification and Counting Scheme

The new method was dubbed Percent-Fill Raster. In this approach, instead of using a Rastered vehicle mask, a parking lot space length and width are extracted from the input image and used to create an empty mask. This mask would be run through a binary form of the input image and if the number of pixels underneath the mask exceeded some percentage threshold of the size of the mask, a 1 would be plotted in an output image. The number of 1s in the output image then yielded a vehicle count. The difficulty, again, was in the accuracy of the creation of the binary image. However, because of the considerations of the Percent-Fill mask, it was no longer of great concern that the parking lot lines be successfully filtered out. In other words, the binary image did not have to have separate pixel blobs (as would be required by the blob counting methods outlined above). The *Percent-Fill Raster* algorithm steps are outlined below.

Step One: *Grayscale Transformation*

This step is a very basic image processing operation which consists of a color space transformation from RGB to Luma-Chrominance (Y'UV) color spaces. The input images had intensities describing the amount of light in the red, green and blue bands. These images are fed through a simple matrix transform into the Y'UV color space and the Y' component (intensity) is used as the output gray level intensity.

Step Two: *Cement/Asphalt Gray-Level Extraction*

The gray level of the cement/asphalt/ (empty parking lot space) is used as a threshold to distinguish the vehicle from the asphalt/cement background. Extracting the level is done in a very simple manner: using the statistical mode of *all* of the gray levels in the input grayscale image, e.g., the gray level histogram.

Step Three: *Filtration of Trees & Vegetation*

This step aims to set all pixels that correspond to trees and vegetation to the gray level of the cement (so they will be ignored later). This step yielded some improvement in the accuracy of the algorithm, as tree areas tended to be counted as vehicles beforehand.

Tree and vegetation filtration process involves the following procedure:

1. Computation of "green content" images:

   Two images are computed directly from the input RGB components, i.e., red, green, and blue, of the color image. The R, G, and B images are treated separately, and the two new "green content" images are computed by subtracting the B image from the G image and the R image from the G image.

2. Set all nonzero pixels in the "green content" images to 1 (forming a binary image by using a threshold of 0).

3. Compute Logical AND of the two "green content" images:

   If a pixel is a 1 in *both* "green content" images, then it is a 1 in the Logical AND image, otherwise it is a 0.

4. For all pixels, if it is a 1 in the Logical AND image, then set it to the cement gray level in the input grayscale image and use that as the output image with trees filtered out.

The basic concept behind the four filtering steps is to extract which pixels are dominantly green and have almost *no* red and blue content. If all pixels that were highly green were filtered out, then pixels in green cars or certain blue cars would be filtered out as well. But computing the two subtraction "green content" images and then computing their Logical AND ensures that the filtered pixels have only green information. Figure 6 shows a sample input image after the trees have been replaced with the cement gray level and the rest of the image has been converted to grayscale.

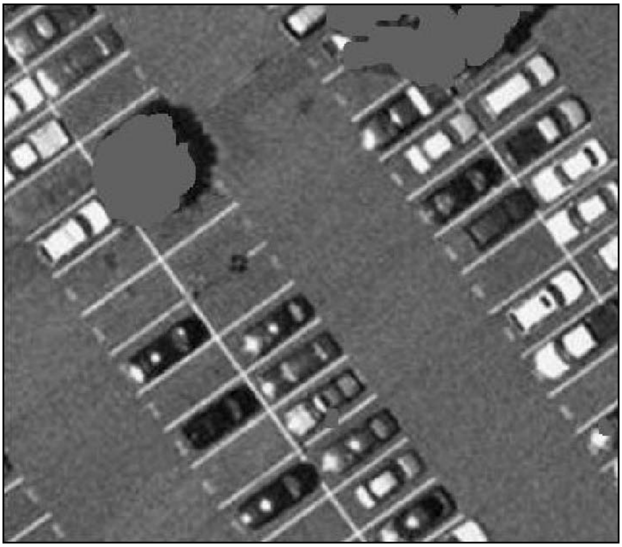

Figure 6. Grayscale Input After Filtration of Trees & Vegetation

Step Four: *Canny Edge Detection*

The goal of edge detection is to highlight pixels that are on the border or edge of a particular image region, such as the pixels on the edge of a vehicle next to parking lot cement. There are various gradient-based edge detectors including Prewitt, Sobel, and Laplacian, and Robert [9]. For this project, a non-mask Canny edge detector was used. Canny edge detection is a multi-stage edge detection algorithm based on Gaussian derivatives. The results of using Canny edge detection on the input grayscale images (after tree filtering) were quite successful compared to the results found with the classical Prewitt, Sobel, Laplacian and Roberts masks previously experimented with. These improved results also aided in the improved results of the next step, the Hough transform. A sample of an edge detected input image is shown in Figure 7. This image was taken after steps one through four were completed.

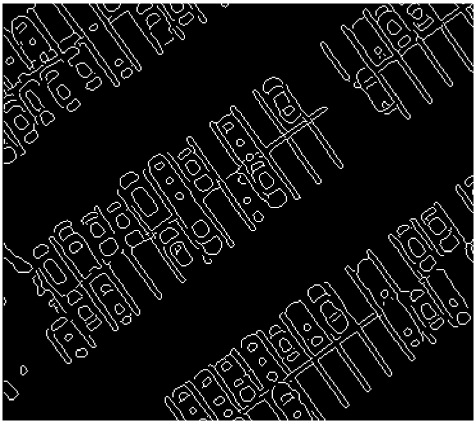

Figure 7. Sample Edge Detected Image

Step Five: *The Hough Transform*

The Hough Transform is used to find straight lines in images [10]. It is very powerful, especially in applications such as BalloonSat where it is known that straight lines will occur in the image, in this case the parking lot lines. The Hough Transform is a two-dimensional discrete transform (like the 2-D Discrete Fourier Transform or 2-D Discrete Cosine Transform) which transforms a binary image into what is called the Hough Space. The Hough Space is another two-dimensional matrix (or image) in which each pixel corresponds to a particular line that *could* be in the input image. If the level at some pixel in the Hough Space is high, or a maximum, then the line described by that pixel's location in the Hough Space probably exists in the input binary image as a series of 1s. A sample image was created in which the lines found in the Hough Space were overlaid on top of the input edge image in red to show which lines had been discovered. This image is shown in Figure 8.

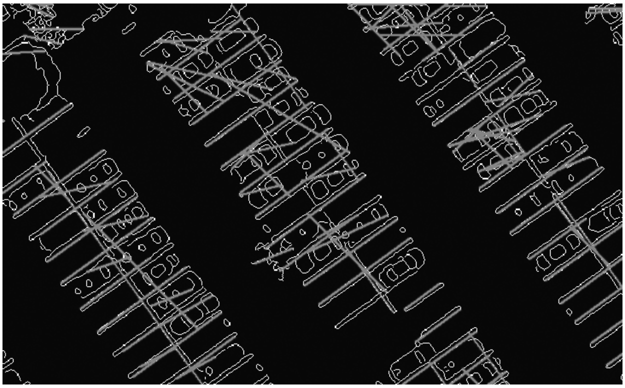

Figure 8. Sample Hough Transform Lines

Step Six: *Extraction of the Length & Width of the Mask, and the Angle of Rotation*
This step is unusual and important for the Percent-Fill Raster and image rotation steps. The goal is to take the returns from the Hough Transform and discern the length and width of a typical parking space, as well as the angle of rotation of the parking lot in the image. Since the results of the Hough Transform are obviously not perfect, this proves to be a difficult task.

To extract the angle of rotation, the statistical mode is again utilized on the Hough Space returns. The angle (theta) at which the most Hough Space maxima occur is the angle at which most of the lines are oriented. Since most of the lines returned by Hough are likely to be the parking lot lines, this angle will be the angle that the image needs to be rotated by to align it vertically.

This method produced very accurate results. The resulting lengths, widths, and angles of rotation were very consistently accurate for our test image set, and the length and width were only different from what a human could visually perceive (after zooming in) by 1 or 2 pixels at the most in each direction.

Step Seven: *Image Rotation*
The input grayscale image needs to be rotated so that the parking lot rows are aligned with the top and bottom of the image, and so that the Percent-Fill mask

can Raster through the image in a vertical alignment. Step six returned an angle that the image needed to be rotated by.

Step Eight: *Forming a Binary Image*

This step is actually a series of separate and unrelated simpler steps, some of which are commonly called morphological operations. If the method of vehicle counting were blobbing, then the separation of vehicle pixels from parking lot lines and other vehicles would be extremely crucial here. However, since blobbing is not the method used, this step has relaxed goals and characteristics, and occurs simply to separate out the parking lot cement (and effectively the trees) from the rest of the pixels. This leaves a binary image where all of the white pixels (1s) are either vehicles, parking lot lines, or some other unwanted disturbance. The resulting image is black and white, with the white pixels forming very large blobs made up of vehicles and parking lot lines. The next step in the algorithm will serve to differentiate which pixels contribute to vehicle space and which do not. A sample of the rotated binary image is shown in Figure 9.

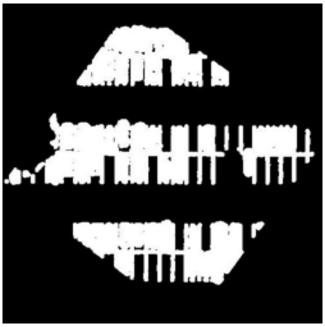

Figure 9. Sample Rotated Binary Image

Step Nine: *Raster Percent-Fill Mask*

Step six provided an accurate measure of the length and width of a typical parking space, and the idea behind step nine is to find how often in the binary image (from step eight) the white pixels fill up that mask in the way that a vehicle would. If the mask is placed at some point in the binary image, and it is populated with a large majority of white pixels, chances are the mask is sitting on top of a vehicle.

Drawing this concept out, the mask is simply run through the image in a Raster pattern to form a new binary image that displays white pixels, i.e., “1” pixels, corresponding to the center of the mask locations for which the percentage of pixels underneath the mask exceeds some threshold. The threshold was empirically selected to be 70%. This step effectively serves to filter out the consideration of the parking lot lines. As can be seen in Figure 9, the parking lot lines contribute to a good portion of the white pixels in the image, but since they will not be enough to meet a large percent-fill threshold, they will be omitted from the Percent-Fill output image, and thus will not be considered in the output vehicle count. This can be seen in the sample Percent-Fill output image, shown in Figure 10.

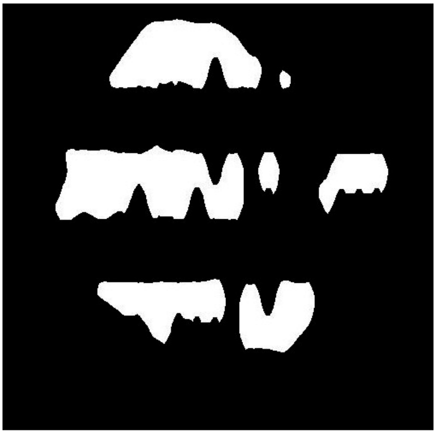

Figure 10. Sample Percent-Fill Output Image

Step Ten: *Counting the Vehicles*

The last step is extremely simple, being comprised of nothing more than a short calculation. All of the white pixels found in the Percent-fill output image are counted up (i.e. all of the pixels in the binary input from step eight that filled the mask to a percentage threshold). That count is then divided by the area of the mask (the extracted length multiplied by the width) and that value is printed out as the count of vehicles in the original input image. Essentially, it is assumed that the number of pixels that are white in the Percent-Fill image is relatively equal to the number of pixels that were part of a vehicle in the original image. Thus, if you divide the number of white pixels by the size of a vehicle (in pixels), you will get a count of the vehicles that are in the original input image. Theoretically, this same operation could be performed on the binary image obtained in step eight, but that image has a large number of pixels that correspond to parking lot lines. Step nine is therefore required to remove the consideration of these pixels.

EXPERIMENTAL RESULTS

The algorithm was tested on a sample set of 8 similar aerial parking lot photos taken from the BalloonSat. The algorithm did not do any pre-processing of the images, as the images were taken with a good camera and good lighting. Consequently, in order to process other image types, there might need to be additional front-end pre-processing, such as histogram equalization/specification or tone adjustment. The input test images were of varying size, scale and orientation. They were in RGB format and a sample is shown in Figure 11. The number of cars in each image was counted visually for comparison to the algorithm, and the results obtained from the algorithm's output are shown in Figure 12. The maximum percent difference between the actual count and the algorithm's output count is 12%, and the average percent difference for all eight images is 6.9%, which is considerably more accurate than previous BalloonSat results.

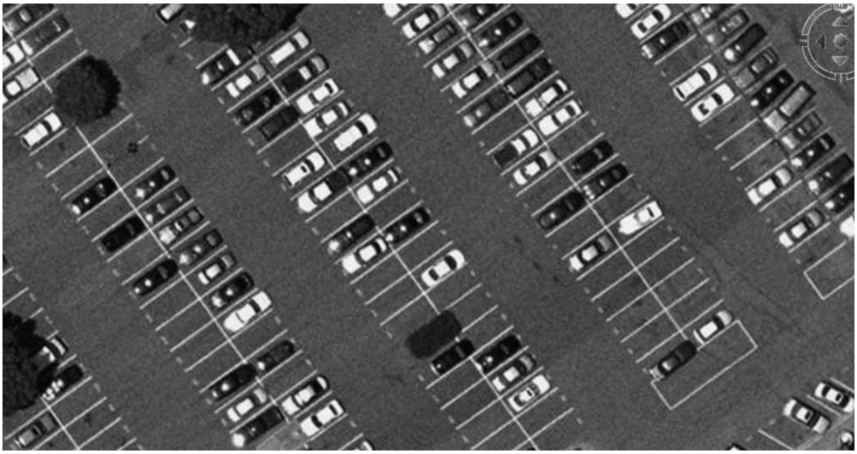

Figure 11. Input RGB Test Image

Test Image	Resolution	Vehicles in Image	Algorithm Output Count	% Difference
lot1.jpg	1035 x 524	80	80	0.00%
lot2.jpg	502 x 524	44	39	11.4%
lot3.jpg	332 x 524	25	23	8.00%
lot4.jpg	1035 x 352	60	58	3.33%
lot5.jpg	1035 x 206	20	20	0.00%
lot6.jpg	368 x 470	56	62	10.7%
lot7.jpg	524 x 332	25	22	12.0%
lot8.jpg	352 x 368	51	56	9.80%

Figure 12. Table of Image Processing Algorithm Results

CONCLUSION

Students were successfully able to design, implement, and test a low-cost tethered helium-filled weather balloon remote sensing station. The onboard visible, infrared, and thermal sensors have wireless remote pan and zoom with real-time video monitoring capabilities. The collected remotely sensed data was transmitted to a ground station through a low-power wireless communication channel. Initial launches of the new and improved BalloonSat indicated more physical stability and wind-tolerance than previous designs. BalloonSat was successfully launched on Cal Poly campus. The collected multispectral data was exploited to provide useful information pertinent to the local terrain including the assessment of the status of the campus open parking lots. A robust image exploitation scheme to identify and count vehicles in a campus open parking lot was successfully developed. In the near future, with the recently added thermal imaging capability, more challenging and multi-sensor exploitation schemes will be developed for such applications as campus security, campus traffic monitoring, and campus tours.

WORKS CITED

[1] Pyroelectric imaging, B.M. Kulwicki et al, Applications of Ferroelectrics, Proceedings of the Eighth IEEE International Symposium on, Proceedings of the Eighth IEEE International Symposium on, 1993

[2] Methods of analyzing traffic imagery collected from aerial platforms, A. Angel et al., IEEE Transactions on Intelligent Transportation Systems. Vol. 4, Issue 2, pp 99-107, 2003

[3] The Use of Photographic Methods for Traffic Data Collection, J. B. Garner, J et al. Uren, The Photogrammetric Record, vol. 7, Issue 41, pp 555-567, 2006

[4] Car detection in aerial thermal images by local and global evidence accumulation, S. Hinz et al, Pattern Recognition Letters, vol. 27, Issue 4, 2006

[5] Automatic Vehicle Detection in Satellite Images, J. Leitloff et al, International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences, vol. 36, Issue 3, 2006.

[6] Detection of Vehicles and Vehicle Queues for Road Monitoring Using High Resolution Aerial Images, S. Hinz, Proceedings of 9th World Multiconference on Systemics, Cybernetics and Informatics, July 2005

[7] Fast and Subpixel Precise Blob Detection and Attribution, S.Hinz, Proceedings of ICIP 05, Sept. 2005

[8] Estimation of vehicle movement in urban areas from thermal video sequences, E. Michaelsen et al, 2nd GRSS/ISPRS Joint Workshop on "Data Fusion and Remote Sensing over Urban Areas", 2003

[9] Edge detection techniques - an overview, D. Ziou et al, International Journal of Pattern Recognition and Image Analysis, vol. 8, pp 537-559, 1998

[10] A modified Hough transform for line detection and its performance, O. Chutatape et al, Pattern recognition, vol. 32, pp 181-192, 1999