Detecting Debris in UAV Imagery of Disasters

Matt Hegarty, Khuoung Nguyen, Robin Murphy
Department of Computer Science and Engineering
Texas A&M University
College Station, USA
hega58@tamu.edu, knn1989@tamu.edu, murphy@cse.tamu.edu

Abstract—In this paper, I present a method of identifying debris and man-made objects in UAV (Unmanned Aerial Vehicle) imagery that relies both on conventional corner and edge detection techniques, as well as improvements specific to the domain of wilderness search and rescue. This process enables first responders to more quickly sift through the massive amounts of data generated by UAV flights and identify potential locations of interest. Currently, UAV imagery is handled by having several responders view each picture and determine if the image contains features that might represent debris. Humans are not well suited to looking through hundreds or thousands of high resolution images for small features, and quickly become distracted and fatigued. This approach provides a way of reducing the number of images that human responders need to review by filtering out images that are less likely to contain debris, letting responders focus on the more important images. It has been tested on actual disaster images taken at the Wimberley Flood, and feedback from responders and search and rescue professionals has been positive.

Keywords—UAV; Computer Vision; Disasters; Search and Rescue; Detection

I. Introduction

In a disaster, sensing and processing data is a large and ongoing problem. Internet and cellular access are often either interrupted, impaired, or over operating over capacity. Even when communications are available, electricity, physical space, processing power, and manpower are all limited. In a disaster scenario, there is currently no automated method of processing UAV (Unmanned Aerial Vehicle) imagery available. Instead, a responder needs to sit at a monitor and manually look at every image taken. This process is time consuming, as a single short flight can yield hundreds of high resolution images, and many different groups may contribute imagery from multiple flights. This task costs manpower and space at a critical time in incident response, and any delay in handling the imagery also delays followup on any potential leads on missing persons present in the images. Additionally, most people are unable to search images for small and ambiguous features for a long period of time without becoming fatigued or beginning to skip through images quickly. An automated method of detecting interesting features, such as building debris and other clearly man-made objects, offers the possibility of freeing up a substantial amount of man-power for other aspects of a disaster response. Such a method also would allow responders to decrease the time required to respond to new information and follow up on leads present in imagery.

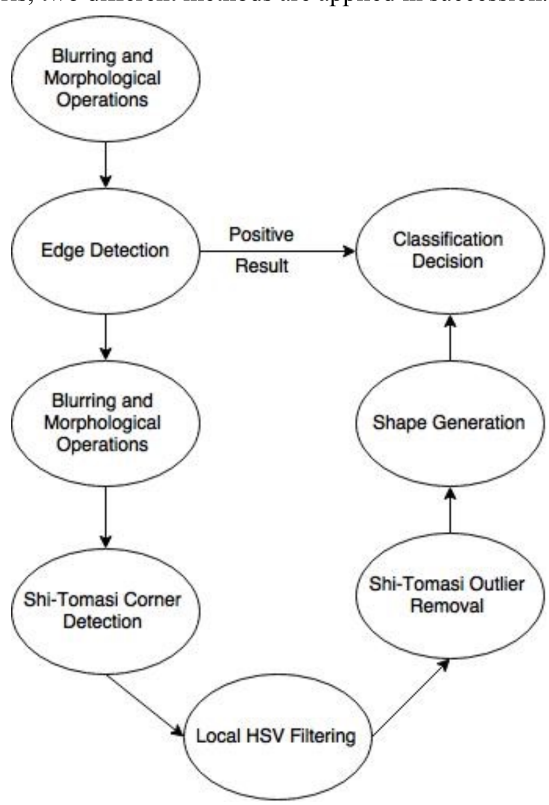
Because of the nature of disaster response, there are three additional constraints on such a method. In order for such a program to be conveniently integrated into a responder's existing workflow, it needs to be both easy to use and capable of running on standard consumer personal computers. In a disaster, specialized processing hardware and cloud computing capabilities probably will not be available. Most importantly, in order to be useful such a method needs to minimize false negatives at all costs. Any piece of debris that slips through the automated detection is potentially a vital piece of information that won't be seen until a full human review of the imagery is performed. Therefore, false negatives are substantially worse for performance than false positives, which represent only a loss of seconds of manpower. These requirements were determined after speaking with first responders and other search professionals.

II. Related Work

There is no existing method for automatically detecting debris in SAR (Search and Rescue) imagery. However, there are existing methods for human detection in SAR imagery [1]. Unfortunately, many of these methods are not practical for wilderness search and rescue. Many methods for human detection rely on either uncluttered environments, or unoccluded figures [2][4]. Unfortunately, SAR operations rarely take place in such convenient environments. Disaster zones almost always yield noisy imagery, and missing people may be trapped beneath logs, inside cars, or partially under the cover of vegetation. The existing work that is not solely focused on human detection is typically for military applications, and as such either focuses on methods for finding a specific target object, or focuses on identifying a general class of object, such as convoys of vehicles or multistory buildings [6]. Because debris can be nearly any color or shape, it is difficult to definitively identify with computer vision techniques. A method of detecting debris needs to detect abnormalities with certain characteristics, rather than match portions of an image with a target.

III. Approach

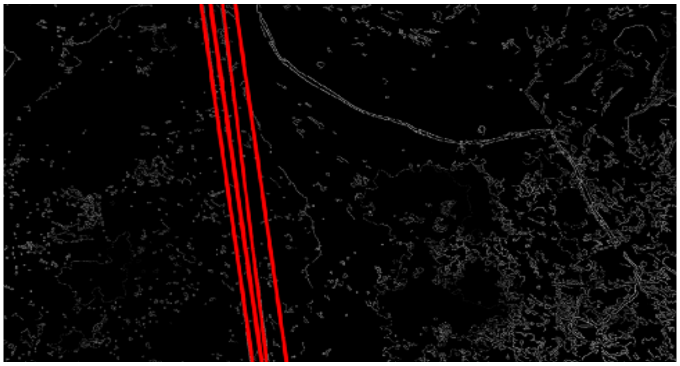
Here we propose a general method for identifying images containing debris and other features of interest. Here, debris is considered to be material present at a disaster site of human origin or manufacture. In order to detect different kinds of debris, two different methods are applied in succession.



1. Flowchart detailing the proposed method.

First, the image is preprocessed and then a canny edge detection algorithm is applied to generate an edge map. For the edge detection, preprocessing consists of first down-sampling the image to a constant size, then applying a sequence of filters. First, a small kernel Gaussian blur is applied to the image, then a small kernel dilation is applied in order to improve detection of thin bright lines like edge reflection and power lines. Finally, a bilateral blur is applied to the image in order to further reduce noise while conserving edges. A low sensitivity Hough transform is applied to the edge map, and if at least one edge spanning a substantial portion of the image is detected, the image is flagged as containing debris. For the purpose of this implementation, a substantial portion of the image was defined as two-thirds of the average of the dimensions of the resized image.

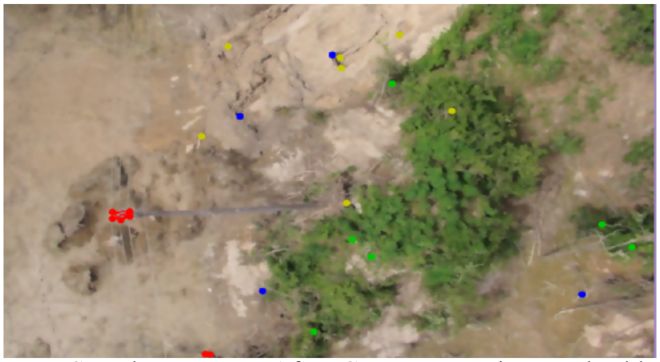
2. Sample Image A Prior to Edge Detection



3. Sample Image A After Edge Detection. Note that while the four power lines are detected, the long white ribbon is not because it lacks a straight stretch of sufficient length. With sufficient blurring and a high enough threshold, edge detection rarely produces false positives when run on actual images from disasters, which is why it is run first.

If the image does not contain any lines long and clear enough to trigger the edge detector, then the corner detector is run. This method is less specific, and uses the Shi-Tomasi corner detection algorithm to identify multiple gradient maxima on the image. Prior to the application of Shi-Tomasi, the image is again resized and preprocessed, but with some small differences. After applying a Gaussian blur, the image is eroded. After these steps, a stronger bilateral blur than was used for edge detection is applied. The difference in blurs and morphological operations reflects the different goals of the two detectors. The Shi-Tomasi algorithm returns a set of points, which can be seen in fig. 4 as the variously colored dots. This set of points is filtered based on the HSV values of their local pixels. Points returned by the algorithm that have only green or brown in all of their adjacent pixels are filtered out, as are points with both a high value and low saturation in all of their adjacent pixels. The remaining points are checked for outliers, and points that are too distant from all other points are also filtered out. After this pass, line segments are generated from the points that have not been filtered out based on their proximity. These segments are then connected into line segments or polygons as appropriate. If a shape contains a certain number of connected line segments, the image is also flagged as containing debris. Since the edge detector rarely produces false positives, it is run first. If it does not produce a positive result, the more general corner detection is run.

4. Sample Image B Prior to Corner Detection



5. Sample Image B After Corner Detection. The blue points were filtered out based on proximity, the green and yellow points based on local HSV values, and the red points were kept and used to judge whether or not to flag the image as containing debris. The above image is flagged based on the cluster surrounding the head of the power line.

IV. Implementation

This application was written in C++ using the OpenCV 3.0 and tinydir 2015 libraries for image processing and file operations respectively. It was implemented to run on all of the image formats supported by OpenCV's imread function, which includes tiff, jpg, png, and bmp. The program was written for windows machines, because windows is the most common operating system used by responders. The hue and value thresholds for HSV filtering in the corner detection module were determined based on the color ranges of Texas dirt and vegetation in normal daylight. These ranges were chosen because this implementation is specifically intended to be used by Texan agencies, and it is typical for both search operations and UAV flights to be performed primarily during the day. Values can be chosen by hand with an HSV picker and a few sample images of the local environment, though an automated method is a future goal. The target dimensions for resizing the image were chosen to make computation proceed at a reasonable pace on a laptop, and have no real bearing on the effectiveness of the implementation, except that values for target sizes and kernels should be scaled as appropriate. If computational resources are not an issue, the algorithm can be run on the original images, as long as the preprocessing kernel sizes are sufficient to handle image noise. It is recommended to preserve the aspect ratio of the original images and to avoid scaling any dimension under 800 pixels in order to preserve small features.

V. Results and Discussion

The classifier was run on a set of 145 images taken by a UAV flying over the Blanco River in Hays County, Texas. The classifier detected most pieces of debris, but registered many false positives, particularly from bright specular highlights caused by the pictures being taken directly over the river. In order to more accurately evaluate the effectiveness of this method, I plan to run the classifier on additional sets of images classified by domain experts. This method has a time complexity of *O(mn*log(mn))*, where m and n are the dimensions of an image in pixels after resizing, and can classify 13.2 images per minute on laptop with a Intel i7-4700MQ processor (2.40 Ghz). As implemented, the method does not depend on internet access for any functionality.

I. Results: Wimberley Flood Imagery

Test Set	Percentage Flagged
Debris	87.1
No Debris	55.6

Imagery contributed by Lone Star UAS

VI. Conclusions and Further Work

The classifier presented here is a useful utility that helps identify the most important images in a set, but no replacement for manual inspection. Ultimately, identifying and rescuing missing persons is not an area where computation can replace human interaction. Humans need both to view imagery to make a judgment call about whether or not a feature is worth a ground team followup, and a human team will likely effect the actual search and rescue at the site of interest. However, computation has a place in SAR as an extension of human capabilities, allowing for quicker and more effective management of UAV imagery in the fast paced environment of a disaster.

This work was intended to culminate in a tool that would be useful to first responders out of the box, without any work on their part. The current codebase is available at http://people.tamu.edu/~hega58/. Development and executable releases are planned to continue at least until the code reaches this point.

In order to improve this method, the two main goals are increased generality, especially with respect to different environments and image formats, and further improving detector specificity. Finding a method of filtering out the sort of intense specular highlights responsible for many false positives would be the logical next step. Looking further beyond that, adding color sensitive Probabilistic Hough Transform has the potential to improve detection accuracy, specifically of short linear objects like wooden beams. The tool could also be improved by adding an automatic method of determining color thresholds, such as using k-means and histograms to identify common hue ranges. Additionally, there are several possible methods of improving detector accuracy by considering the angles produced by a connected set of features. Another goal is to integrate the detector with a utility to superimpose a physically accurate grid over the image, in order to assist responders with determining the size of objects.

Acknowledgment

This work was made possible by NSF CNS 1263027. Additionally, thanks to Lone Star UAS for contributing their flooding imagery, and to the participants of the 2015 Summer Institute on Flooding for contributing their time and expertise, as well as for providing additional test imagery. Additionally, thanks to Jesus Suarez of Texas A&M University.

References

1. Blondel, P., et al. "Fast and Viewpoint Robust Human Detection for SAR Operations". Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on. Web.
2. ---. "Fast and Viewpoint Robust Human Detection in Uncluttered Environments". *Visual Communications and Image Processing Conference, 2014 IEEE.* Web.
3. Chen, Jie, et al. "The Comparison and Application of Corner Detection Algorithms." *JOURNAL OF MULTIMEDIA* 4.6 (2009)Print.
4. Ning Hui, et al. "Research on Human Body Detection Methods Based on the Head Features on the Disaster Scenes". *Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on.* Web.
5. Shuanghui Lei, et al. "Comparison of Feature Point Extraction Methods Based on UAV Remote Sensing Image". *Intelligent Control and Automation (WCICA), 2014 11th World Congress on.* Web.
6. Singh, S., and D. V. Rao. "Recognition and Identification of Target Images using Feature Based Retrieval in UAV Missions". *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on.* Web.