Computer Vision Object Detection Algorithms for Collision Avoidance

By Posholi S. Nyamane, ECE 16

The Gold Team studied different object detection algorithms in relation to the collision avoidance system for its Autonomous Bridge Inspection Unmanned Aerial Vehicle(UAV) capstone project. The research focuses mainly on three categories of algorithms: feature-tracking, template-tracking, and machine learning. The applicability and feasibility of each algorithm are assessed based on its performance in terms of computation speed and the object features it uses to detect the object.

Introduction

Computer vision can be defined as an electronic perception of an image. It is comprised of image acquisition, processing, analysis and understanding. Understanding of an image in the context of computer vision involves extracting, from the image, information describing the physical world and transforming it into signals that can be used for control or decision-making. Object detection is one of the many sub-fields under the study of computer vision. It deals with detecting objects from still or moving images. It is widely applied in autonomous vehicles for collision avoidance, video surveillance, image retrieval, etc.

The collision avoidance system for the autonomous UAV that the Gold Team is designing will implement object detection to create obstacle maps. The goal is to have a fast and robust collision avoidance system. To build such a system, three categories of object detection algorithms (feature-tracking, template-tracking and machine learning) are considered. This tech note presents the general theory, applications and computational performance of the algorithms in each category.

Feature-tracking Algorithms

Feature based algorithms detect objects in images by finding salient points (also know as key points or corners) on the object and in its vicinity as identifiers. It is a three step process: key points detection, descriptors extraction and finally matching.

Key points are points surrounded by a greater variation in intensity of the pixels in the image (Szeliski, 2010a). They are also the points at which the direction of edges changes, or the point where two or more edges meet (hence the name *corners*). At the corner/key point the gradient (an increase or decrease in pixel intensity) around the point has high variation. This variation is used to detect the point and the higher the variation, the more detectable the point is. Some of the algorithms used to detect key points are Harris-Stephens Corner Detector (Harris and Stephens, 1988) and Shi-Tomasi Corner Detector (which is an improvement on the Harris Corner Detector). Both of these algorithms have

Figure 1. Key points (the blue dots) detected using the Harris Corner Detector

the advantage of being independent on variations in scale and illumination which can be useful in creating a robust collision avoidance system. Figure 1 shows the key points (the blue dots in the image) detected using Harris Detector.

After the key points have been detected, the next step in feature based object detection is descriptors extraction. Key points descriptors are needed to perform the object search in images. A descriptor of a key point is a histogram (represented by a vector) containing the gradient orientations of pixels surrounding the key point. It is a description of the key point. Just as there are different algorithms used to detect key points, there are different algorithms used to extract descriptors. One of the methods used is Scale Invariant Feature Transform (SIFT). A SIFT descriptor is computed by partitioning the region surrounding each detected key point into grids and computing the orientation histogram of the pixels in each sub-region. The final step is to find the object of interest in the image: the key point descriptors relating to the object of interest are matched with the descriptors extracted from the search image (the image being searched for the object).

Template-tracking Algorithms

Template-based detection is done by comparing every pixel in the image of the object to every pixel in the search image. This is called template matching (Bradski and Kaehler, 2008). The image of the object is called a template and it must be less or equal to the search image in area. When comparing the pixels, the template "slides" over the search image from the top left corner to the bottom right corner. There are different methods used for template matching. Some of them are square difference matching method, correlation matching method and correlation coefficient matching method. With the square difference matching method, the comparison is accomplished by taking the sum of the square difference of the intensity of the corresponding pixels in the template and the search image. The region in the search image which results in a minimum sum is more likely to be a match of the template. Figure 2 shows the result of a square difference matching method.

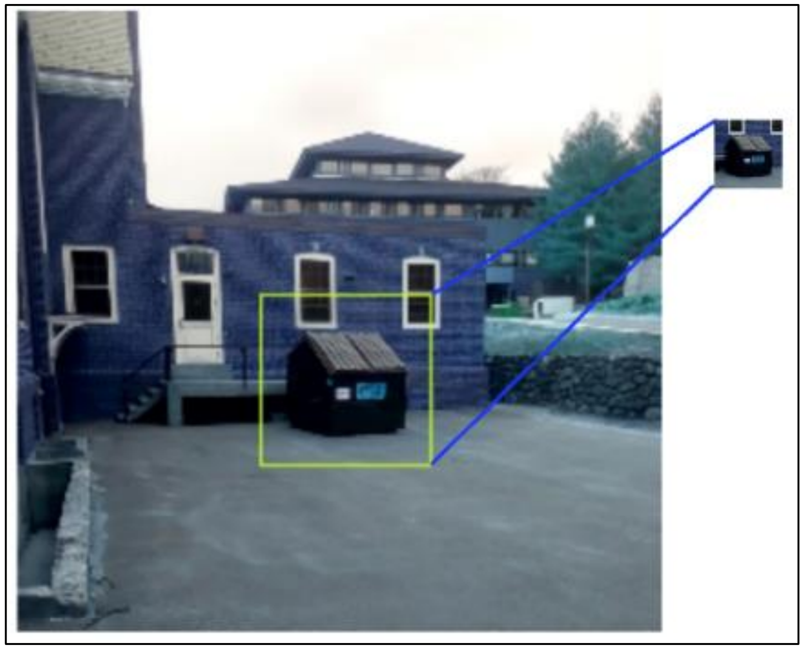

Figure 2. Template matching is used to find the template (on the right) in the image.

Machine Learning Algorithms

The machine learning approach to object detection can be done in many ways. One approach is to use cascade classifiers (classifiers are ways of categorizing the object based on shape, color, etc.). A classifier is first trained/computed with sample views of the object. The object can be a face, a building, a car, etc. The training first is done with a few hundred positive examples (images of the object) scaled to the same size. It is then done with negative examples (arbitrary images) with the same dimensions (Szeliski, 2010b). The training samples are usually kept in a database. A trained classifier can be applied to a search image for detecting the object of interest. The classifier can also be resized to detect the object at different scales. The term "cascade" means that the classifier is comprised of several simpler classifiers/stages that are applied sequentially to the search image until all the stages pass or until a rejection occurs at some stage. Besides cascade classifiers there are other methods such as the field histogram method (Linde and Lindeberg, 2004), which categorizes objects using object descriptors and histograms.

Performance and Applicability Comparison

As seen from the discussion presented above, the algorithms in the three categories have different approaches to object detection. Although they are all based on pixel intensities, the way the intensities are used varies from method to method. Because of this the algorithms depend on different features of the object. As a result some algorithms are more applicable to particular situations than are others. For example, a feature-based detector has higher success detecting objects with "corners" than objects with no corners. The algorithms in the three categories also have different computation speeds; as they need to check every pixel, template-based and feature-based detectors tend to be slower than cascade classifiers detectors. These computationally expensive algorithms can pose a significant problem when applied to videos, especially when the frame rate is higher than the algorithms' speed. The problem is that the next frame arrives before the algorithm completes the computation on the current frame. This leads to decreased precision or failure. For feature-based and template-based detectors, reducing the image size, thus reducing the number of pixels, can speed up the computation. With regards to obstacle detection system feature-based detectors have a huge advantage over template detectors and machine learning detectors which is that they can be designed to detect any obstacle in the field of view of the camera. This can be done by detecting any salient points/corners in the image. The motion of this points relative to the camera can then be used to estimate where the obstacles are and which way they are moving. On the other hand, in order to detect any obstacle, template matching detectors require templates for every possible obstacle, which is unfeasible. The same problem arises with machine learning algorithms in that the classifier will have to be trained to detect any obstacle. Table 1 summarizes the general properties of the algorithms from each category and their qualitative performance.

Conclusion

This Note presents three categories of object detection algorithms. The computation theory of each category shows that different image/object features are used to detect objects. Features used affect the speed, robustness and applicability of the algorithm. Based on the theory and the results summarized above, the Gold Team concludes that in order to detect obstacle in a video captured by a moving camera on a UAV, feature-tracing algorithms should be used as they can detect any object and their computation speed can be easily improved by scaling down the image.

Table 1. Performance summary for the three categories of algorithms

Algorithm Category	Object Features used	Qualitative Computation Speed	Advantages	Disadvantages
Feature-tracking Algorithms	Key points (pixel intensity variation)	Slow (can be sped up by downsizing the images)	Can be designed to detect any obstacle, accurate	Slow Resizing the image can blur it and reduce precision
Template-tracking Algorithms	Pixel intensity	Slow (can be sped up by downsizing the images)	Accurate detection of objects.	Slow, can be affected by illumination variation as it uses pixel intensity, Resizing the image can blur it and reduce precision, cannot be designed to detect any obstacle.
Machine Learning Algorithms	Object classifiers based on shape, color, etc.	Fast	Accurate, computationally inexpensive, fast.	Cannot be designed to detect any obstacle.

References

Bradski, G. & Kaehler, A. (2008). Histograms and Matching. In *Learning OpenCV* (pp. 214-218). Sebastopol, CA: O'Reilly Media.

Harris, C. & Stephens, M. J. (1988). A Combined Corner and Edge Detector. In *Proceedings of the Alvey Vision Conference*, 147–152. Retrieved from http://www.bmva.org/bmvc:1988

Linde, O, & Lindeberg, T. "Object recognition using composed receptive field histograms of higher dimensionality." *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 2. IEEE, 2004. doi: 10.1109/ICPR.2004.1333965

Szeliski, R. (2010a). Feature Detection and Matching. In *Computer Vision Algorithms and Applications* (pp. 207-259). London. New York: Springer.

Szeliski, R. (2010b). "Recognition". In *Computer Vision Algorithms and Applications* (pp. 668-709). London. New York: Springer.