Object Recognition with Multicopters*

Falk Schmidsberger and Frieder Stolzenburg

Hochschule Harz
Automation and Computer Sciences Department
Friedrichstr. 57-59
D-38855 Wernigerode
Germany
<fschmidsberger,fstolzenburg>@hs-harz.de

Abstract. Data acquisition with semi-autonomous flying robots, e.g. multicopters, has several advantages over conventional inspections or aerial photographs. However, in order to facilitate the handling of the flying robot for the pilot, it seems to be appropriate to employ semantic object recognition, making the robot more autonomous. In this paper, we therefore report ongoing work on applying semantic object recognition, where the image recognition procedure works as follows: Each object in an image is composed of segments with different shapes and colors. In order to recognize an object, e.g. a plane, it is necessary to find out which segments are typical for this object and in which neighborhood of other segments they occur. Typical adjacent segments for a certain object define the whole object in the image. A hierarchical composition of segment clusters enables model building, taking into account the spatial relations of the segments in the image. The procedure employs methods from machine learning, namely clustering and decision trees, and from computer vision, e.g. image pyramid segmentation and contour signatures.

Keywords: Multicopters, Semantic Object Recognition, Machine Learning, Computer Vision, Applications

1 Introduction

Mobile data acquisition with unmanned autonomous-flying systems (UAS) is an inexpensive alternative compared with conventional aerial photography. Since these systems often are equipped with multiple sensors, it seems to be a good idea to improve the autonomy of such vehicles, because otherwise the personnel may not be able to control the whole robot system. In addition, it is not always possible to obtain radio contact during the flight. This means, it might be important to detect known interesting objects automatically during flight, and then to sense the environment around these objects or making photographs, whatever is appropriate. In the sequel, we will introduce the procedure for semantic object recognition in more detail.

* This research has been partially supported by the grant *ZIM KF2488207HM1* from the German ministry of economics in the *Airmeter* project. Preliminary work has been reported in [14].

S. Wölfl (Ed.): Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI-2012), pp. 83-87, 2012.

2 Related Works

The problem of recognizing and locating objects is very important in applications such as robotics and navigation. Therefore, there are numerous related works. The survey [5] reviews literature on both the 3D model building process and techniques used to match and identify free-form objects from imagery, including recognition from 2D silhouettes. [7] describes shape surfaces by curves and patches, represented by linear primitives, such as points, lines, and planes. Results are presented for data obtained from a laser range finder. Hence, these results cannot be transferred directly to the analysis of video camera images, as done here. [12] presents an object recognition system that uses local image features, which are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. This proposed model shares properties with the object recognition in primate vision. A nearest-neighbor indexing method is employed that identifies candidate object matches. [13] describes a model-based recognition system of planar objects based on a projectively invariant presentation of shape, using projective transformations. Index functions are used to select models from a model base, exploiting object libraries. However, for general semantic object recognition as considered here, fixed object libraries are certainly not sufficient. In [11], another approach for modeling visual context is introduced. The authors consider the leaves of a hierarchical segmentation tree as elementary units. This method has similarities to the one presented here. The approach here, however, exploits normalized contour feature vectors in the semantic object recognition method, which we will explain now.

3 Object Recognition

Each object in a digital image is composed of a number of segments with different shapes and colors. In order to recognize an object, it is necessary to find out which segments are typical for which object and in which segment neighborhood they occur. If such a segment in a characteristic neighborhood is found, it is considered as part of the object. Typical adjacent segments for a certain object constitute the whole object in the image and allow its identification. The data mining methods clustering, decision trees and boosting are used to implement this approach [2, 4, 9].

We extract the image segments by their colors in two steps. For each pixel in the image the similar neighboring pixels are colored with an uniform color by a flood fill algorithm. With an image pyramid segmentation algorithm the shapes of the resulting blobs of uniform color are extracted as the image segments [3].

3.1 Segment Feature Vectors

To process the segments of an image, a normalized feature vector is computed for each segment. The normalized feature vector of a segment pixel set comprises the data of four normalized distance histograms and is computed from the segment

contour. A distance histogram consists of a vector, where each element contains the distance between the centroid of the segment, i.e. the center of gravity, and a pixel in the segment contour or the distance between two pixels in the segment contour. These distance histograms are computed with the following three related methods: polar distance, contour signature and ray distance [1, 10, 15].

For the polar distance, fixed angle steps are used to select individual pixels from the segment contour with the maximum distance to the centroid of the segment. For non-convex segments, if there is no pixel with the actual angle, the pixel with the angle $+ \pi$ and the minimum distance to the contour is chosen. All selected pixels p are stored in the pixel set B and the distance of each pixel to the centroid is stored in the vector $\boldsymbol{MPD}$ (maximum polar distance) with a constant number of elements for each segment. In the contour signature histogram vectors, $\boldsymbol{MCD}$ (maximum contour distance) and $\boldsymbol{MinCD}$ (minimum contour distance), the distance of each pixel in B to the corresponding opposite segment contour pixel is stored. In this case, the straight line between the two pixels has to have a 90° angle to the tangent through the actual pixel in B. The corresponding opposite pixel is the pixel with the greatest distance to p for $\boldsymbol{MCD}$ (minimum distance for $\boldsymbol{MinCD}$). $\boldsymbol{MCD}$ and $\boldsymbol{MinCD}$ have the same cardinality as $\boldsymbol{MPD}$. In the ray distance histogram, the distance of each Pixel in B to the corresponding segment contour pixel is stored. Here, the centroid is on the straight line between the two pixels and the result is the vector $\boldsymbol{MCCD}$ (maximum center contour distance) with the same cardinality as $\boldsymbol{MPD}$.

3.2 Feature Vector Normalization

In most cases, the distance histograms have different values even for the same segment, when this is rotated or resized. To get a normalized segment feature vector, each distance histogram has to be normalized by shifting the distance values of the vector, so that the angle with the maximum value and the maximum angle difference to the next angle with the maximum value is the first element in the feature vector. In a second step, the distance values itself are normalized to $[0.0, 1.0]$, with the respective maximum distance value. After the normalization, all four distance vectors are joined to the new feature vector V which is invariant against translation, rotation and resizing.

3.3 Clustering and Decision Trees

In order to reduce the number of feature vectors, clustering algorithms (k-means and agglomerative) are used to build a cluster model [2, 8]. Each resulting cluster represents a set of similar feature vectors, identified by its respective centroid. The cluster model is used to decide the cluster affiliation for a new given segment feature vector.

For all segments in one image, the cluster numbers for each segment are computed and stored in a segment cluster tree (cf. Fig. 2). The root node of the tree represents the image itself. A child node represents a segment which is immediate part of the segment of the upper level. Nodes on the same level

are marked as neighbors by dotted lines, if the corresponding segments in the image are connected. Different colors in the cluster assignment visualization mean different levels in the segment hierarchy (cf. Fig. 1).

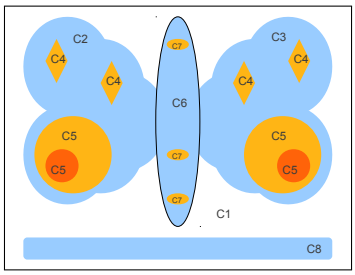

Fig. 1: Segment Cluster Assignments.

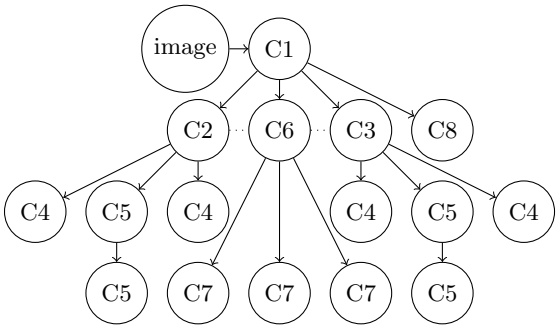

Fig. 2: Segment Cluster Tree.

We can now extract five different types of feature vectors from the segment cluster tree: The first one contains all paths from a leaf node to the root, the second one all child nodes of a node one level above, the third one all nodes marked together as neighbors, and the fourth one all child nodes of all nodes in the tree. The last feature vector contains the numbers of each cluster, found in the image. These five different feature vector types are used to train 10 decision tree/boost models [4, 9], which are combined to predict the right object category of unknown images.

3.4 Results

To test and improve the first implemented algorithms in a controlled environment, they were used to classify images from the butterfly image dataset [16]. For all seven categories, the right category of an image is predicted with a success rate of 99.5 % if the image is from the training set and 27.14 % if the image is from the test set. A random guess would give us only a success rate of $1/7 = 14.28\,\%$. On images made by the first author the success rates were 100.00 % and 46.00 % (5 categories). Here, a random guess has a success rate of $1/5 = 20\,\%$. Next tests will be made on the images from "The Pascal Visual Object Classes (VOC) Challenge" [6]. It takes about 0.7 seconds to classify a live image with the computational power of our actual multicoper hardware.

4 Conclusions

Thus, our first results are encouraging, but in the future, the implementation of our approach will be improved further to become faster with an increased object recognition success rate. For this, a distributed implementation seems to

be promising. Using more spatial relations of the segments including different perspective views for a more accurate decision tree/boost model is also desirable. The final goal is to implement the approach as a real-time object recognition working on autonomous multicopters.

References

1. Enrique Alegre, Rocío Alaiz-Rodríguez, Joaquín Barreiro, and Jonatan Ruiz. Use of contour signatures and classification methods to optimize the tool life in metal machining. *Estonian Journal of Engineering*, 1:3–12, 2009.
2. Michael J. A. Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley & Sons Inc., 3rd edition, 2011.
3. Gary R. Bradski and Adrian Kaehler. *Learning OpenCV - computer vision with the OpenCV library: software that sees*. O'Reilly, 2008.
4. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees (The Wadsworth Statistics/Probability Series)*. Wadsworth Publishing, 1983.
5. Richard J. Campbell and Patrick J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166 – 210, 2001.
6. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.
7. O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.
8. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2nd edition, 2006.
9. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. Springer, 2nd ed. 2009. Corr. 3rd printing 5th Printing. (20. April 2011).
10. Bernd Jähne. *Digital Image Processing*. Springer, 6th revised and extended edition, 2005.
11. Joseph J. Lim, Pablo Arbelaez, Chunhui Gu, and Jitendra Malik. Context by region ancestry. In *ICCV*, pages 1978–1985. IEEE, 2009.
12. David G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150, 1999.
13. C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16:57–99, 1995.
14. Falk Schmidsberger and Frieder Stolzenburg. Semantic object recognition using clustering and decision trees. In Joaquim Filipe and Ana Fred, editors, *Proceedings of 3rd International Conference on Agents and Artificial Intelligence*, volume 1, pages 670–673, Rome, Italy, 2011.
15. Fan Shuang. Shape representation and retrieval using distance histograms. Technical report, Dept. of Computing Science, University of Alberta, 2001.
16. Cordelia Schmid Svetlana Lazebnik and Jean Ponce. Semi-local affine parts for object recognition. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 959–968, 2004.