KI 2013 Workshop on Visual and Spatial Cognition

Semantic Object Recognition with Segment Faces

Falk Schmidsberger and Frieder Stolzenburg

Harz University of Applied Sciences, Automation and Computer Sciences Department, Friedrichstr. 57-59, 38855 Wernigerode, Germany
{fschmidsberger,fstolzenburg}@hs-harz.de

Abstract. Recognizing objects from images becomes a more and more important research and application topic. There are diverse applications such as face recognition, analysis of aerial images from multicopters, object tracking, image-based web search, etc. Many existing approaches focus on shape retrieval from a single polygon of contour points, or they try to compare clouds of interesting points of an object. However, human object recognition concentrates on few points of the segments forming the object. Clearly, complex objects, strictly speaking the projections of their shape on the image plain, consist of several (polygonal) segments. Therefore, the procedure presented in this paper takes this into account, by composing objects hierarchically into a group of segments. We briefly introduce our procedure for semantic object recognition based on clusters of image segment contours and discuss the problem of recognizing objects from different perspectives.

1 Introduction

Already in [2], it has been stated, that the contour points of an object, for instance of a cat, are of particular importance for human semantic object recognition. By semantic, we mean in this context that we do not only want to recognize abstract geometric forms but real complex objects, given by (non-preprocessed) example images, which are not characterized by a single contour, but by a group thereof. The surface of a complex object can be considered as consisting of several polygons. When such an object is viewed from a certain viewpoint, its image may be perspectively distorted. Nevertheless, several features remain invariant, for instance segment neighborhood relations, among others. Therefore, we focus in our approach on segment contours and their adjacency relations. Our overall procedure of object recognition roughly works as follows (cf. [13]):

Each object in an image is decomposed into segments with different shapes and colors. In order to recognize an object, e.g. a house, it is necessary to find out which segments are typical for this object and in which neighborhood of other segments they occur. A group of typical and adjacent segments for a certain object defines the whole object in the image. Similar segments are clustered. A hierarchical composition of these 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 k-means clustering and decision trees with boosting [3, 5, 9], and from computer vision, e.g. image pyramid segmentation and contour signatures [4].

The rest of this paper is organized as follows: First, we introduce our procedure for semantic object recognition in some more detail (Sect. 2). After that, we consider the influence of perspective projection on object recognition (Sect. 3). Next, we provide a brief evaluation of the approach (Sect. 4) and discuss some other approaches on object recognition (Sect. 5). Finally, we summarize and conclude the paper (Sect. 6).

2 Semantic Object Recognition

In our procedure for semantic object recognition, at first, we train models with sample images for each object category. After training, the models are used to recognize objects in other images. This is done as sketched in Fig. 1. Most of the steps and data in this process (marked in black in the figure) are identical for the training and the recognition phase. The steps and data that are relevant only for the training phase are marked with blue boxes and arrows, whereas the data and steps that are relevant only for the recognition phase are marked in green.

1. **Image optimization and segmentation:**
   For each pixel in the image, similar neighboring pixels are colored with a 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 image segments [4].
2. **Segment feature vector extraction and normalization:**
   A feature vector is computed for each segment, using the data of four normalized distance histograms, computed from the segment contour. A distance histogram consists of a vector of distances computed with several related methods: polar distance, contour signature, and ray distance [1, 10, 15] (see Fig. 2).
3. **Compute/use cluster models over the feature vectors:**
   During *training*, a cluster model with all feature vectors from all images of one category is created. Each cluster represents a familiar segment of the actual object category. During *recognition*, the cluster model of a category is used to select the familiar segments in the actual image.
4. **Segment tree creation, using only familiar segments of a category:**
   A segment tree comprises typical adjacent segments and the hierarchical composition of segment clusters for a certain object. It defines the whole object in the image. This histogram method is invariant against translation, rotation, scaling, and partially also to perspective distortions.
5. **Compute/use the decision tree models:**
   During *training*, a decision tree model from the data of the segment tree for each object category is created. During *recognition*, the decision tree model of each category is used to recognize objects in the actual image.

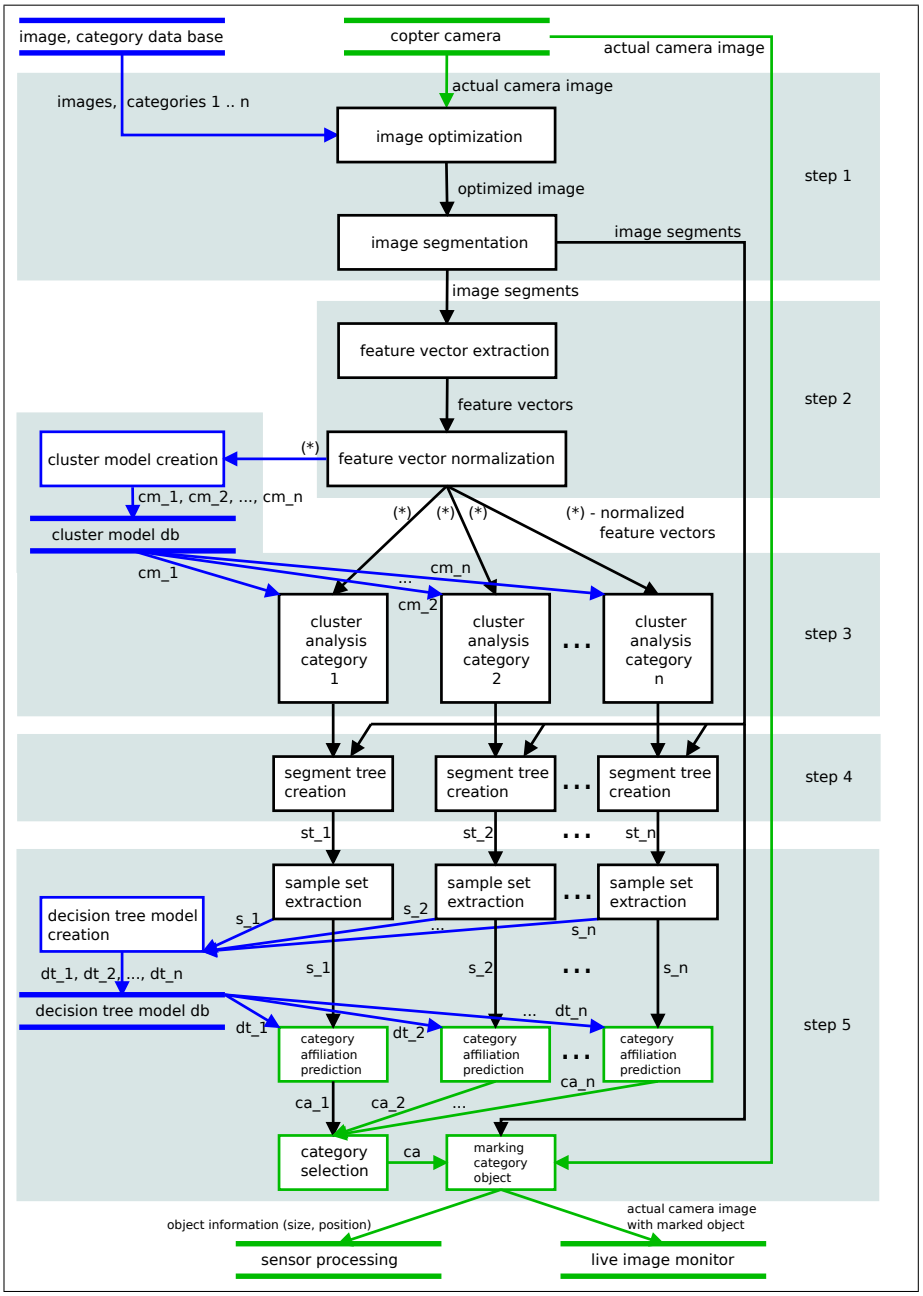

Fig. 1. Data flow diagram for the semantic object recognition procedure for training phase (blue) and recognition phase (green) with five steps: 1. image optimization and segmentation; 2. segment feature vector extraction; 3. compute/use cluster models over the feature vectors; 4. segment tree creation; 5. compute/use the decision tree models.

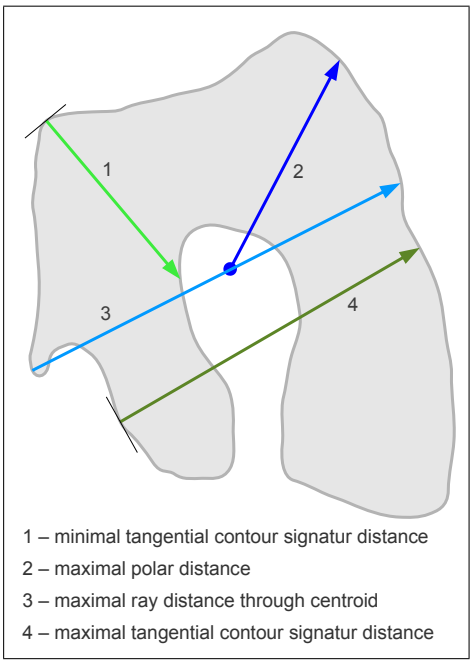

Fig. 2. Distance histogram methods: A distance histogram consists of a vector, where each element contains the distance between the centroid of the segment, strictly speaking the center of gravity, and a pixel in the segment contour (maximal polar distance) or the distance between two pixels in the segment contour computed with different methods (cf. [13, Sect. 3.1]).

3 Segment Contours and Perspective

Let us now consider the problem of perspective distortion in more detail, where objects are viewed from different viewpoints. In this context, the shape of an object is more or less defined by its surface. For the sake of simplicity, we assume that the surface is given as a polygon mesh. One question then is, what happens with the contour of a polygonal segment on the object surface, when it is viewed from different perspectives. Here, we restrict attention to the case where the segment is completely visible and not partially hidden. Let us consider the projective image of a cube as an example (Fig. 3). Clearly, by perspective projection, angles between lines and lengths of lines including their ratios may deviate significantly from their original values. In addition, parallelism of lines is not preserved, too.

Formally, the image of an object can be roughly described by central projection. Central projection (planar 3D projection) is determined in essence by the point of the observer, called central or focal point c and the position of the image plane, in particular its normal vector v, both consisting of 3D coordinates.

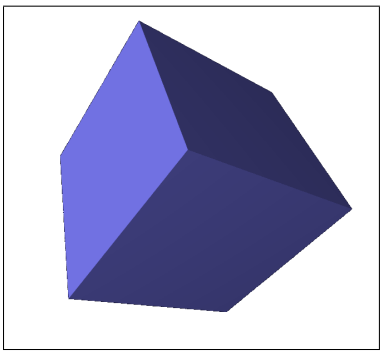

Fig. 3. Cube in perspective view.

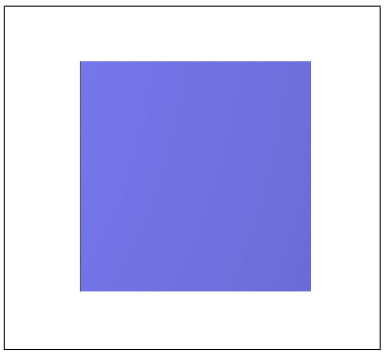

Fig. 4. Cube viewed from the front.

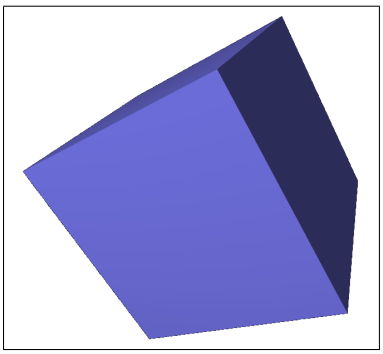

Fig. 5. Rotated cube.

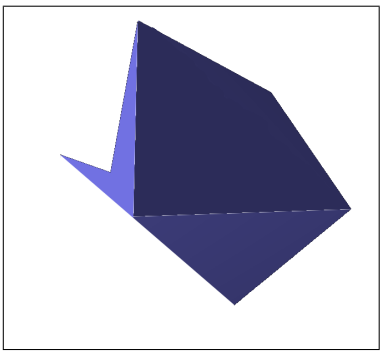

Fig. 6. Impossible projective view.

This gives us approximately 6 parameters. In contrast, a polygonal surface with n vertices is determined by n 2D coordinates, giving us $2n$ values. Since $2n > 6$ for $n \geq 4$, it follows that, although the shape of a polygon may vary widely by perspective distortion (cf. Fig. 4 and 5), for polygons with 4 or more vertices, there are clear limits for the distortion.

In fact, there are several invariants during perspective projection: First of all, straight lines remain straight lines. This implies that polygons remain polygons with the same number and order of vertices. Furthermore, left-right relations, which are important for localization, navigation and exploration [16], stay invariant, provided that the polygons are always viewed from the same side, which is usually the outside of the object. From this it follows in particular, that convex edges remain convex and never become concave by perspective projection, and vice versa. Therefore, the image in Fig. 6 definitely cannot be the projection of a cube, because the leftmost polygon in the image is concave, whereas a cube has only squares on its surface that are convex. Last but not least, adjacent segments remain adjacent. Hence, the image segment tree remains the same, although of course not all polygonal segment faces may be visible from all viewpoints.

In principle, full perspective projection can be taken into account: For this, a point x of a segment contour, given as 3D column vector in homogeneous coordinates, i.e. with an additional component (cf. [7, Sect. 5.6]), is projected onto the 2D plane, at position y, that is a 2D column vector (also in homogeneous coordinates). The corresponding mapping $M_0 : x \mapsto y$ consists first of a rotation R, that is 3×3 orthonormal matrix, and a translation T, a 3D column vector, and then the actual projection P from distance $-d$, as follows (cf. [7, Sect. 6.4]):

$$M_0 = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}}_{P} \cdot \begin{bmatrix} & R & & T \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let now y_1 and y_2 be the projections of a given object point x, whose position usually is not known, on two images. Therefore, we have $y_i = M_i \cdot x$ for $i = 1, 2$, where M_1 and M_2 are mappings as above, in general different. From this, we obtain (1) $\lambda\, y_2 = M \cdot y_1$ with $M = M_2 \cdot M_1^{\dagger}$, where $\dagger$ denotes the Moore-Penrose pseudoinverse operator and λ a scale factor, which is needed because the 3×3 matrix M, mapping the homogeneous 2D coordinates, is not affine in general, i.e., its last row may be different from $[0 \cdots 0\ 1]$. Eq. (1) can be expressed equivalently without reference to λ by the cross product $y_2 \times (M \cdot y_1) = 0$. This leads to three linear equations in the 9 components of the matrix M, of which only two are linearly independent however, for each projection point pair (y_1, y_2). Given n such pairs, we arrive at the matrix equation $A \cdot m = 0$, where A is a $(2n) \times 9$ matrix and m is the matrix M reshaped as column vector (cf. [17]).

Since A is overdetermined in general and it must be $m \neq 0$, the solution for $A \cdot m = 0$ can be found by minimizing the squared (L2) vector norm (2) $\varepsilon = |A \cdot m|^2$ with respect to the condition $|m| = 1$. Eq. (2) is equivalent to $\varepsilon = (A \cdot m)^{\top} \cdot (A \cdot m) = m^{\top} \cdot (A^{\top} \cdot A) \cdot m$, where $\top$ denotes transposition, thus $(A^{\top} \cdot A) \cdot m = \varepsilon\, m$ (after multiplication with m). Hence, the problem of determining whether two segment contours stem from the same object, taking perspective distortion into account, can be reduced to an eigenvalue problem. As distance measure in the clustering procedure, the smallest eigenvalue ε of the 9×9 matrix $(A^{\top} \cdot A)$ can be used. Nevertheless, a normalization with respect to the starting point of the segment contour has to be done.

4 Evaluation of the Approach

The object recognition method with distance histograms (as described in Sect. 2) has been implemented in *C++/OpenCV* [4] by the first author. For the full treatment of perspective distortion (Sect. 3), so far only an implementation in *Matlab/Octave* [8] by the second author is available. All in all, our object recognition procedure works in practice: The segment neighborhood relations in the image segment tree remain invariant, even after perspective distortion. Rotation and translation of polygons is treated by length normalization. Although projections may vary a lot, they often show still strong similarity to the original

image. The clustering of different segments takes this into account. The experiments with our *C++* implementation for object recognition with recognition rates usually between 50 and 100 % – far above chance – are encouraging in this direction.

To test and improve the first implemented algorithm in a controlled environment, it was used to classify images from the butterfly image dataset [11]. 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 would have a success rate of $1/5 = 20\,\%$ only. It takes about 0.7 seconds to classify a live image, which need not be pre-segmented into foreground and background.

5 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 [6] 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.

[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. This approach is very successful in practice. However, as already said in the introduction, here more or less only points of clouds and not groups of segment faces are considered, which appears to be more cognitively adequate.

[14] performs shape retrieval by considering qualitative relations. This means the qualitative relations of the line segments forming the contour of the polygon are considered during the object recognition phase. The approach is eventually based on the so-called double-cross calculus [18]. Each object is identified by exactly one contour built from more and more polygon vertices. The approach is successful, however it does not reflect the fact, that complex objects may consist of several segment contours.

6 Conclusions

With our approach, we can recognize objects in digital images, independent of scaling, translation, rotation, and perspective distortions of the object. To do this, we train and use models based on the segment shapes of the objects and the topological and spatial relations of these segments. The next step is to implement the approach as a real-time object recognition process on autonomous multicopters (cf. [13]).

References

1. Alegre, E., Alaiz-Rodríguez, R., Barreiro, J., Ruiz, J.: Use of contour signatures and classification methods to optimize the tool life in metal machining. Estonian Journal of Engineering **1** (2009) 3–12
2. Attneave, F.: Some informational aspects of visual perception. Psychological Review **61**(3) (1954) 183–193
3. Berry, M.J.A., Linoff, G.: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. 3rd edn. John Wiley & Sons Inc. (2011)
4. Bradski, G.R., Kaehler, A.: Learning OpenCV – computer vision with the OpenCV library: software that sees. O'Reilly (2008)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. The Wadsworth Statistics/Probability Series. Wadsworth Publishing (1983)
6. Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding **81**(2) (2001) 166–210
7. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics: Principles and Practice. 2nd edn. The Systems Programming Series. Addison-Wesley Publishing Company (1993)
8. Gilat, A.: MATLAB: An Introduction with Applications. 2nd edn. John Wiley & Sons (2004)
9. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd edn. Springer Series in Statistics. Springer (2009)
10. Jähne, B.: Digital Image Processing. 6th revised and extended edn. Springer (2005)
11. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: Proceedings of the British Machine Vision Conference. Volume 2. (2004) 959–968
12. Lowe, D.G.: Object recognition from local scale-invariant features. Computer Vision, IEEE International Conference on **2** (1999) 1150–1157
13. Schmidsberger, F., Stolzenburg, F.: Object recognition with multicopters. In Wölfl, S., ed.: Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI–2012), Saarbrücken (2012) 83–87 Obtained Best Poster and Demo Presentation Award.
14. Schuldt, A.: Shape retrieval with qualitative relations: The influence of part-order and approximation precision on retrieval performance and computational effort. In Bach, J., Edelkamp, S., eds.: KI 2011: Advances in Artificial Intelligence – Proceedings of the 34th Annual German Conference on Artificial Intelligence. LNAI 7006, Berlin, Springer (2011) 301–312
15. Shuang, F.: Shape representation and retrieval using distance histograms. Technical report, Dept. of Computing Science, University of Alberta (2001)
16. Stolzenburg, F.: Localization, exploration, and navigation based on qualitative angle information. Spatial Cognition and Computation: An Interdisciplinary Journal **10**(1) (2010) 28–52
17. Wikipedia: Projektionsmatrix – Wikipedia: Die freie Enzyklopädie (2013) [Online; Stand 14. August 2013].
18. Zimmermann, K., Freksa, C.: Qualitative spatial reasoning using orientation, distance, and path knowledge. Applied Intelligence **6** (1996) 49–58