Robust vehicle categorization from aerial images by 3D-template matching and multiple classifier system

Bertrand Le Saux and Martial Sanfourche
Onera - The French Aerospace Lab
F-91761 Palaiseau, France
bertrand.le_saux@onera.fr, martial.sanfourche@onera.fr

***Abstract*—We present a robust method for vehicle categorization in aerial images. This approach relies on a multiple-classifier system that merges the answers of classifiers applied at various camera angle incidences. The single classifiers are built by matching 3D-templates to the vehicle silhouettes with a local projection model that is compatible with the assumption of the little knowledge that we have of the viewing-condition parameters. We assess the validity of our approach on a challenging dataset of images captured in real-world conditions.**

I. Introduction

Object categorization in real-world scenes is a challenge that has received considerable attention from the computer vision and image analysis communities in recent years. After faces and people, vehicles have been targets of choice due to the wide variety of possible applications, from traffic monitoring and vehicle guidance to scene interpretation. Recent advances in the field of vehicle detection and categorization follow two main trends: geometric methods and learning-based methods, while the best performances are usually achieved by a blend of both approaches.

Learning-based approaches focus on the extraction of 2D features from several views of the vehicle, and gather them in bag of features or constellation models. In the context of aerial imagery, boosting has been used to learn how to detect vehicles seen from above in the nadir direction in urban areas [1]. A similar framework has also been used to model vehicle side-views [2], [3], but such discriminative models lack the precision needed for categorization, for which models that embed geometric relationships between the features are more satisfactory. Implicit Shape Models describe object patches by referencing to a visual codebook and estimate the distribution of the patch locations in the recognition framework [4], but different detectors are needed for different 2D aspects of a vehicle. A true 3D model that encodes the visual appearance under various 3D points of view and the 3D geometric relationships between the points of view have been proposed by Savarese and Fei-Fei [5]. The complexity of the model implies that this approach is particularly greedy for training data. The development of these methods has been made possible by the construction of several benchmark databases (UIUC Car [6], Caltech 101 [7]). However, images in such controlled datasets do not represent all the possible views of a vehicle that can happen in the actual conditions of an airborne sensor.

On the other hand, several approaches try to fit a geometric 3D-model to the image. Koller proposed several 12-parameter vehicle models (one per category) [8] that he matches to motion-based primitives extracted from video frames. This model can also be combined with the visual appearance of the vehicle [9] by extracting HoG features at various salient locations, thus allowing a real categorization in the vehicle class. The precision of this last approach is made possible by the extensive metada (geo-localization, plane attitude) recorded during image capture.

The typical scenario we want to address is vehicle categorization in a single image taken from a plane or UAV in real conditions. Illumination changes are usually not tested during flight, and this leads to poor image quality. This is a problem for using learning-based classifiers, since on one hand, the available training sets do not correspond to the real conditions we face, and on the other hand we do not have the possibility of building a new training set that encompass the whole variety of poses and situations. It also makes the extraction of reliable visual features in the images difficult.

Moreover, unlike the case of controlled image capture, there is scarce metadata about the viewing conditions. During the flight, all the information of localization or plane-attitude is usually lost, except the altitude. On the contrary, what is known before take-off is usually available, such as the camera focal length. This forbids us to use a precise and complete geometric model as in [9]. Instead we propose to fit a 3D template by using a simple local projection model, and rely on a multiple-view classifier system to achieve good performances.

In the rest of the paper, we describe the local projection model, the matching procedure and the classifier system in section II. We conclude with the experimental validation of the proposed approach on real-world images in section III.

II. Approach

A. 3D-template matching

a) Simple local projection model: The camera coordinate system is centered at the camera position (cf. Fig. 1), and defined such that the x-axis is the camera axis and $\mathbf{y}$ is perpendicular to x and the aeroplane direction. The local world coordinate system is also centered at the camera position, and defined such that $\mathbf{x}$ is the aeroplane longitudinal axis and $\mathbf{z}$ is the aeroplane vertical axis.

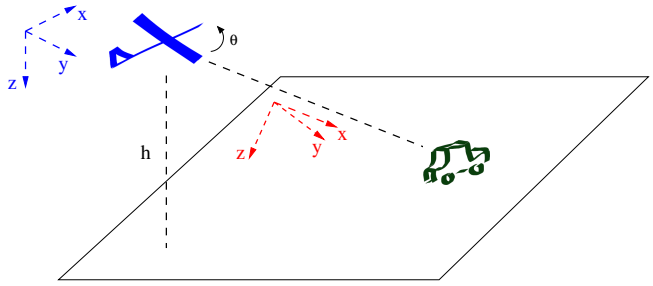


Fig. 1: We consider a simple local projection model where only the plane altitude h and the camera azimuth θ are known. The local world coordinate axes are represented in blue, the camera coordinate axes are in red.

The transform between pixel coordinates and a 3D point in the real world is given by:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K R_{ali} R_{Cl} X^{world} \quad (1)$$

where K, the camera intrinsic parameter matrix is defined by:

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

where the central point u_0, v_0 is assumed to be the center of the image, and the rotation matrices for the coordinate-system transform are:

$$R_{Cl} = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

$$R_{Ali} = R_X(\pi/2).R_Z(\pi/2)$$

We assume a locally plane world model, thus the z-coordinate of the target location is h and we can determine the factor s (the notation $[.]_3$ indicates the third coordinate of the vector within brackets):

$$s(h,u,v) = h / \left[R_{cl}^t R_{ali}^t K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right]_3$$

Finally we obtain the projection model of the target coordinates:

$$X^{world} = \begin{pmatrix} x \\ y \\ h \end{pmatrix} = s(h,u,v) R_{cl}^t R_{ali}^t K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (2)$$

b) Template matching algorithm: On one hand, the inputs of the algorithm 1 consist only of the 3 parameters needed to feed the projection model: h the plane altitude, θ the inverse of the elevation angle of the camera and the focal length α, and the image and a bounding rectangle delimiting roughly the position of the vehicle in the image. Such a bounding rectangle is a result of the detection algorithm [4] (the code of which is available on the Internet).

On the other hand, we dispose of the target models: basically they are 3D templates like the shoe-box model or the parametrized generic model used in [8]. The choice of the right template depends on the image resolution and the computation times we want to achieve.

In a nutshell, the method estimates the target position in the real world from the image, and back-project the template in the image to compare with the image.

Algorithm 1 3D-Template matching

Require: image I, bounding rectangle defined by (x, y, l, w), projection parameters K, θ, h

precise image position C_0 + vehicle shape $\leftarrow$ fit a 2D Gaussian distribution on $I(x : x+l, y : y+w)$.
for all model m **do**
  for $p = 0$ to 360 **do**
    template^{World} $\leftarrow$ init_world_template(model, C_0, K, θ, h)
    template^{Im} $\leftarrow$ project2image(template^{World})
    template silhouette $\leftarrow$ convex_hull(template^{Im})
    similarity $s_{m,p} \leftarrow$ match(template silhouette, vehicle shape)
  end for
end for
$m, p \leftarrow \arg\max(s_{m,p})$

First, a fine estimation of the vehicle position in the image is performed. The object shape is extracted by estimating the parameters of a 2-Gaussian distribution of the pixel intensities in the region delimited by the bounding rectangle. This allows a precise estimation of the vehicle center.

Then we use the projection model of Eq. 2 to estimate the real-world location of the object and try to fit various 3D models to the shape we extracted. Shape or silhouette similarity is typically measured by chamfer distance [10].

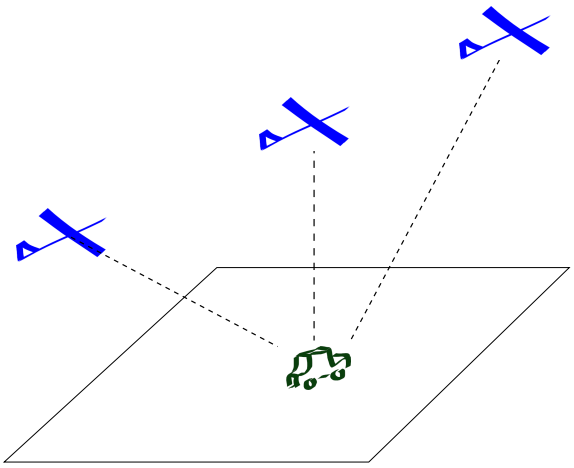
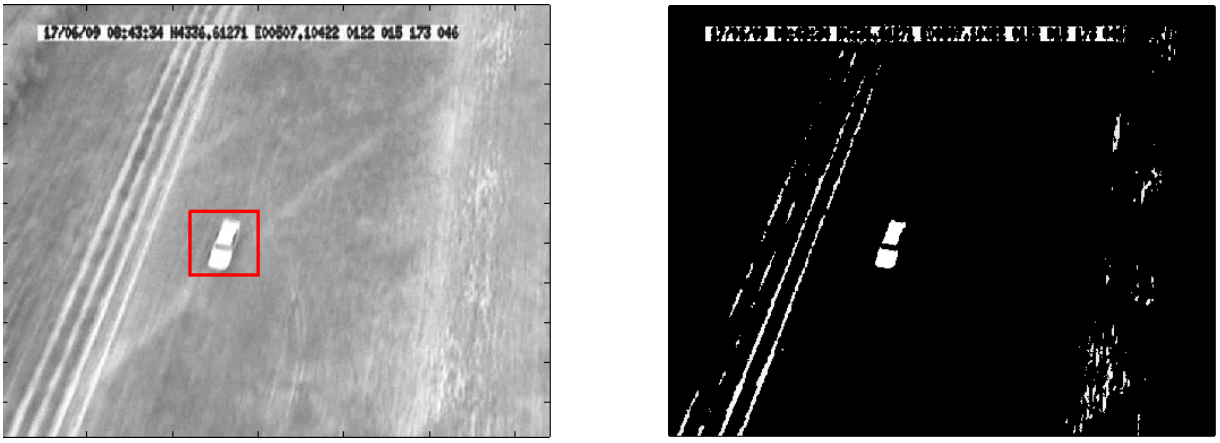


Fig. 2: A multiple view classifier system is used to combine multiple aspects of the target.



(a) (b)

Fig. 3: Target image example (a) with object of interest and its segmentation based on the object local intensity for silhouette extraction (b).

B. Multiple classifier system

Multiple Classifier Systems (MCS) have successfully been used to solve difficult classification problems [11]. In this paper we present a MCS based on the 3D-template-matching previously defined.

We use several views of the object of interest to gather more information on its geometry (cf. Fig 2). In order to achieve this, the target is imaged C times under different aspects during the passage of the plane (a tractable scenario yields 3 or 4 images in the time elapsed). Each view is then matched to the 3D-templates and classified according to the nearest-neighbor rule.

A similarity-weighted K-NN algorithm [12] (with K the number of possible templates) is used to combine the outputs of the individual classifiers into the final decision m_0 according to the rule:

$$m_0 = \arg\max_{m, m\in[1,K]}\left(\sum_{i=1}^{C} s^i_{m,p}\right) \quad (3)$$

III. Results and discussion

A. Dataset

The dataset consists of 68 images taken from a small plane flying at about 100m. 6 different vehicles (that span over 3 car segments) are parked on the target zone. The aeroplane circles the zone in order to get several snapshots of the vehicles. Along with the images, the following parameters are recorded: altitude, camera angle and camera focal length. An image example is shown in Fig. 3a.

B. Error rates and confusion matrices of the single classifiers

Table I gives the classification errors for the individual classifiers. While some distinct car-segments (e.g utility vehicles) are easily classified, the atypical elements of the other categories lead to errors in the classification process.

utility	compact	mid-size car
1	0	0
0.24	0.58	0.18
0.16	0.60	0.24

TABLE I: single-view vehicle category confusion matrix

utility	compact	mid-size car
1	0	0
0.05	0.73	0.22
0.02	0.61	0.38

TABLE II: multiple-view vehicle category confusion matrix

C. Error rates and confusion matrices of the multiple-classifier system

Table II shows the results for car-segment categorization using the multiple-view classifier. Compared with Table I, the number of images correctly classified in their actual class is improved.

Table III shows the results for car-model identification. Even if some classes are correctly retrieved, this clearly shows the limitations of the approach for distinguishing between similar-sized vehicles. For example the approach is confused by sedans and wagons in different car segments. Both a more detailed geometric model and some better data (with higher resolution and associated metadata) would be necessary.

C1	C2 wagon	C3	U	M1	M2 wagon
0.45	0.12	0.04	0.19	0.02	0.16
0.01	0.82	0.07	0	0.04	0.06
0.01	0.4	0.13	0.06	0.09	0.31
0	0	0	1	0	0
0	0.42	0.2	0	0.17	0.20
0.06	0.23	0.36	0.1	0.10	0.15

TABLE III: multiple-view vehicle category confusion matrix

D. Impact of the number of the views in the classifier

Fig. 4 shows the influence of the number of classifiers in the MCS on the percentage of true positives. At first, adding new classifiers is efficient in improving the error rates, since they correspond to new views of the object and thus bring new information to the classification process. However the discriminative power of the geometric information has its own limit, and the improvement then reaches a plateau.

Actually we are also constrained by a more practical barrier since the maximum number of classifiers is the number of views we can manage to get during one passage of the plane, and thus depends on the trajectory. The optimal number of classifiers (in terms of results and computational times) of the MCS is nevertheless compatible with this limit.

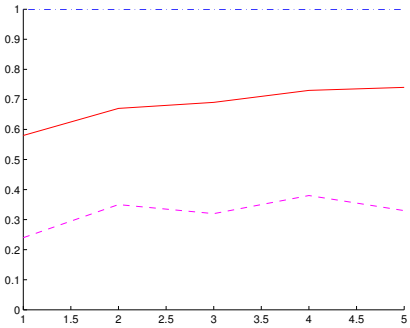


Fig. 4: Influence of the number of classifiers in the MCS, for various categories (dash-dotted blue: utility vehicles; solid red: compact cars; dashed magenta: mid-size cars).

E. Pose estimation

	utility	compact	mid-size car	overall
pose error	4.57°	4.74°	4.20°	**4.54°**

TABLE IV: RMS errors on pose estimation

The matching process of our approach allows us to precisely estimate the pose of the target, that we define as the target orientation on the ground with respect to the aeroplane plane heading. Table IV sums up the RMS errors for the various category models. It shows that the pose estimation is relatively precise (less than 5% overall error) for all target classes, independent of the categorization results.

F. Matching results

As a result of the approach, the precise location of the target is also determined both in the image and in the local aeroplane coordinate system (cf. Fig. 5 and Fig. 6). If coupled with a precise GPS, it allows us to georeference the target for further use.

Fig. 5: Image with the matched model silhouette.

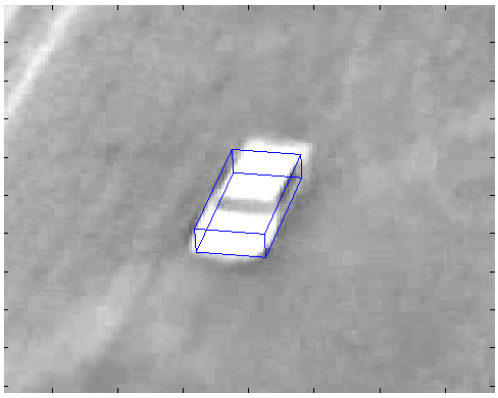


Fig. 6: Zoomed image with the matched model.

IV. Conclusion

In this paper, we presented a robust approach for vehicle categorization in airborne images. Our aim was to be able to process challenging real-world data, i.e. saturated, at low-resolution and with few associated metadata. This is achieved by defining a multiple-classifier system that merges the information from different single-view classifiers. To deal with the lack of viewing condition information, we defined a local projection model that allows us to match vehicle 3D-templates to the object silhouette in the image. The single-view classifiers are then based on a nearest-neighbor rule.

Though a more detailed geometric model could lead to the decrease of classification errors by a few percentage points, the real promising direction is to include local contrast or appearance information. In the future we aim to implicitly capture the statistically significant visual features of each category by training classifiers on such real-world datasets. The MCS framework we proposed will easily incorporate these new classifiers, in order to give a more accurate decision.

Acknowledgments

The authors gratefully acknowledge the Department of Theoretical and Applied Optics of Onera for providing us with the data captured in real-world conditions.

REFERENCES

[1] T. Nguyen, H. Grabner, B. Gruber, and H. Bischof, "On-line boosting for car detection from aerial images," in *International Conference on Research, Inovation and Vision for the Future (RIVF'07)*, 2007.

[2] W.-C. Chang and C.-W. Cho, "Online boosting for vehicle detection," *Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 40, no. 3, pp. 892–902, 2010.

[3] H. Grabner and H. Bischof, "On-line boosting and vision," in *Computer Vision and Pattern Recognition (CVPR), workshop on Generative-Model Based Vision*, 2006.

[4] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *European Conference on Computer Vision (ECCV), Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004.

[5] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, October 2007.

[6] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.

[7] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *Computer Vision and Pattern Recognition (CVPR), workshop on Generative-Model Based Vision*, 2004.

[8] D. Koller, K. Daniilidis, and H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *International Journal of Computer Vision*, vol. 10, no. 3, pp. 257–281, 1993.

[9] S. Khan, H. Cheng, D. Matthies, and H. Sawhney, "3d model based vehicle classification in aerial imagery," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[10] A. Rosenfeld and J. Pfaltz, "Distance functions in digital pictures," *Pattern Recognition*, vol. 1, no. 1, pp. 33–61, 1968.

[11] J. Kittler and F. Roli, Eds., *Multiple Classifier Systems, First International Workshop, MCS 2000*. Cagliari, Italy: Springer Verlag, june 2000.

[12] G. Toussaint, "Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining," *International Journal of Computational Geometry and Applications*, vol. 15, no. 2, pp. 101–150, 2005.