2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Rapid Semantic Mapping: Learn Environment Classifiers On the Fly

Bertrand Le Saux and Martial Sanfourche*

Abstract— **We propose solutions to provide unmanned aerial vehicles (UAV) with features to understand the scene below and help the operational planning. First, using a visual mapping of the environnement, interactive learning of specific targets of interest is performed on the ground control station to build semantic maps useful for planning. Then, the learned target detectors are transformed to be applied to new images captured by the UAV. On the technical side, we present: (i) an online gradient boost algorithm to interactively design context-dependent detectors; (ii) a video-domain adaptation method to use object detectors on on-board-camera images. We verify our approach on challenging data captured in real-world conditions.**

I. INTRODUCTION

Extensive research work has been done to provide Unmanned Aerial Vehicles (UAVs) with a model of their environment ([1] for a recent example). Environment maps typically contain a 2D or 3D geometric representation combined with sensor-based information such as image textures when cameras are used. The underlying assumption is that it represents a step towards more autonomy, by allowing localization, path planning and object recognition. The step further consists in semantic maps [2] which in addition contain a mapping of the geometric features to higher-level semantic information like labels of known classes of objects.

In practical situations like security monitoring, search-and-rescue or other civil applications, UAVs are not fully autonomous yet, but at least partially remotely operated. At the ground countrol station (GCS), qualified professionals work in close collaboration with the UAV operator for designing intervention schemes. Environment maps are useful to give them the big picture of the situation and allow to spot targets of interest. For building semantic maps, we propose to interactively train the detectors on the GCS using the environment maps in order to overcome the problem of required training data and transform them for use on the UAV camera.

Among typical state-of-the-art approaches, [3] generates semantic labeling of places for a ground robot in an indoor environment by using supervised classification with the adaboost algorithm on laser and vision-based features. In [4], several known classes of objects are recognized by a bag-of-feature approach coupled with extraction of key-points in images captured by the robot. Both techniques require training samples with associated labels beforehand (in the latter case an internet connection was used to retrieve relevant images). For aerial vehicles, the aspect of objects varies depending on the altitude and this leads to different approaches. For indoor environment where sensors can capture images with enough details, [5] proposes to use scalable part-based models to detect people. Outdoor, at further distance with a more-or-less vertical point of view, a wide range of detectors for specific targets exists, such as cars [6], [7] or buildings and road artefacts [8]. All these detectors assume a pre-existing model. In [9], the authors propose to propagate and reinforce semantic labeling in new images captured while the robot evolves, but they still use prebuilt semantic classifiers to initialize the process.

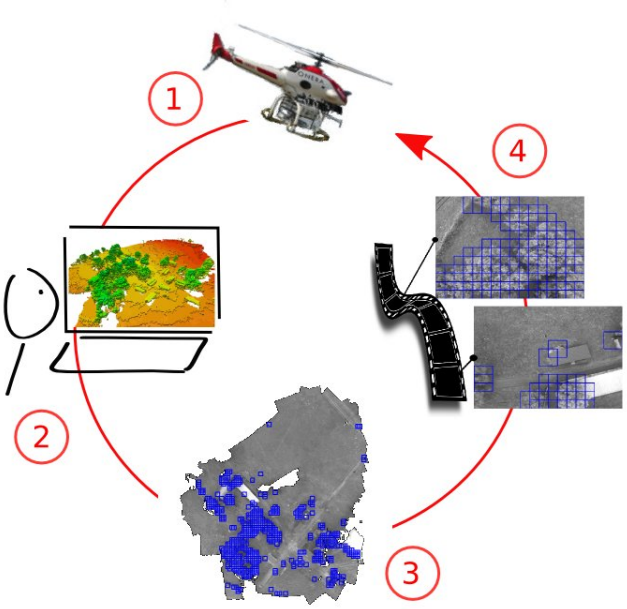


Fig. 1. **Graphical abstract**. Our approach for interactively designing ad hoc classifiers for search-and-rescue missions. (1) **Data collection** The UAV collects videos, 3D Lidar data and GPS positions. (2) **Geometric mapping** In the ground station, rapid 3D mapping of the environment is processed by bundle adjustement. (3) **Semantic mapping** An interpret designs classifiers for specific targets (trees, cars...) using online gradient boost. (4) **Video-domain adaptation of detectors** Classifiers are geometrically adapted to the onboard camera image domain and sent-back to the flying UAV.

In this paper, we introduce an alternative approach for designing semantic classifiers on site. Starting with an environment map that is build on the GCS from data (video images and laser measures) captured in flight by a UAV, we take benefit of the expert in the loop to define by online learning on the map what is a relevant target for the current operation. Concretely, (i) the expert selects target samples the appearance of which is used to interactively train an online gradient boost classifier; (ii) the resulting classifiers are adapted to execute on the camera images and sent back to the UAV. The outputs are detections of targets both geo-

*ONERA The French Aerospace Lab, F-91761 Palaiseau, bertrand.le_saux, martial.sanfourche at onera.fr

localized on the environment map for mission planning and in the video-stream for the UAV navigation (cf. Fig. 1).

The rest of the paper is organized as follows. In section II we describe the UAV setup and the process to build the global mapping of the environment. In section III we detail the interactive learning approach and in section IV the adaptation of the detectors to the video domain. Finally, the whole approach is assessed by results presented in section V, that we discuss in section VII.

II. ENVIRONMENT MAPPING

Though the construction of non-semantic mapping is beyond the scope of this paper, this section describes the system that was used to perform experiments and briefly explains the generation of orthomosaïcs.

A. System overview

The platform used for experiments is a Yamaha RMAX helicopter equipped with various sensors: a 1,3MP monochrome camera for video and a 4-line-scan laser measurement sensor (Sick LD-MRS) for range data. The localization of the helicopter is given by a decimeter-class GPS-RTK system. For security sake, the UAV is remotely controlled from the GCS which receives its attitude and speed information along with sensor data.

B. Environment maps

Given video and range data captured from the UAV, we build 3D environment maps in a 3-step workflow. First, the geometric constraint induced by points of interest tracked over the video-sequence is used to refine the trajectory, using a sparse bundle adjustment algorithm [10]. Second, given the new UAV positions, range data are aggregated to form a 3D-point cloud that is the basis for the environment map (cf. Fig 2). Finally, a Digital Elevation Model (DEM) generated from the point cloud can be textured by mapping the radiometry of each video-frame to the DEM pixels, which yields in easy-to-understand orthomosaïcs.

The trajectory refinement is required to obtain useful environment maps because in spite of the precise localization provided by a high grade GPS-RTK, a direct mapping of range data using these measures leads to artefacts (ghost buildings and misplaced landmarks) in the resulting DEM (cf. Fig 3)

III. INTERACTIVE LEARNING OF TARGETS OF INTEREST

Orthomosaïcs give an operator the global situation of the scene and allow him to enhance the UAV capability by learning objects of interest: for example trees for landing obstacle avoidance or cars for target detection. For this purpose, we adapted the method of [11] to such images. We show this is effective for learning patterns other than man-made structures, and specifically the ones that can be useful for UAV guidance.

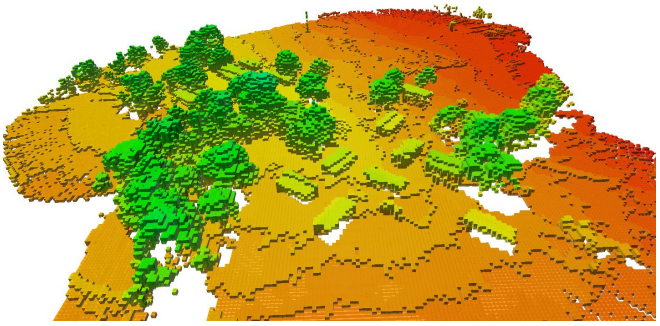
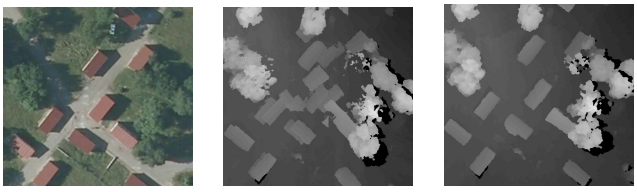


Fig. 2. 3D environment map built from the range data using the precise trajectory estimated by sparse bundle adjustment using the video-images.



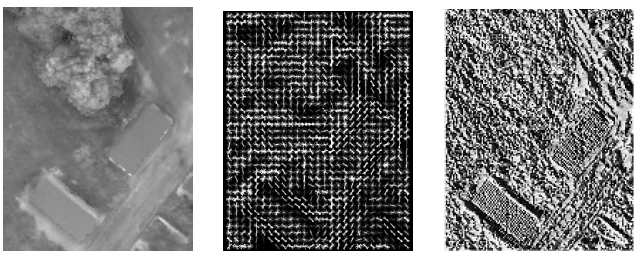
(a) (b) (c)

Fig. 3. Quality of the mosaïc: comparison of a ground-truth aerial photography (a) with 2 DEMs: (b) one built using only raw UAV parameters and which leads to reconstruction artifacts; (c) the output of our method which matchs the real buildings observable in the ground-truth.

A. Feature extraction

The operator is presented the generated orthomosaïc in an interface that allows to select areas of interest and irrelevant areas (cf. Fig. 5). Then the system extracts small patches from the selected zones and computes appearance descriptors for each of them (cf. Fig. 4). The resulting features constitute the training set (with both positive and negative samples) that is used to learn the target of interest. We use the combination of Histograms of Oriented Gradients (HOG [12]) and Local Binary Patterns (LBP [13]). These features were proven efficient in aerial images [11] as well as standard videos, since they were used as the low-level image descriptors in the approach that won the PASCAL visual object classes detection challenge in 2011 [14].

Orthomosaïc patches corrrespond to sets of locations denoted by $\{m_p\}_{1 \leq p \leq P}$. In our implementation, HOG quantifies the edge direction information with a 60-dimensional



(a) (b) (c)

Fig. 4. Detail of the original ortho-rectified image (a) and corresponding extracted features: (b) Histograms of Oriented Gradients (HOG) as edge direction descriptor and (c) Linear Binary Patterns (LBP) as texture descriptor.

smoothed histogram. LBP is a texture feature that computes a 10-interval histogram of rotation-invariant patterns at 2 different scales, resulting in a vector of dimension 20. We denote the concatenated vector (dimension $D = 80$) of HOG-LBP features by:

$$x_n = f_{\text{HOG-LBP}}(\{m_p\}_{1 \leq p \leq P}) \tag{1}$$

and the training set of the features with the label y_n of the area they come from by:

$$\mathcal{X} = \left\{(x_1, y_1), \cdots, (x_N, y_N), x_n \in \mathbb{R}^D, y_n \in \{+1, -1\}\right\}$$

B. Fast online learning

Learning is then performed by online gradient boost, the aim of which is to build a decision rule able to discriminate the image descriptors. An online procedure is preferred because it allows the operator to give relevance feedback in order to design an efficient detector. In a few words, boosting is a machine learning approach which builds a good (*strong*) meta-classifier F from a set of weak classifiers f_m (presently, the i^{th} component of the HOG-LBP descriptor):

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{2}$$

Several variants of the initial *adaboost* algorithm [15] have been proposed, including the *online boosting* used in [16]. The trick for obtaining an online version of boosting is to use cumulative errors over the set of samples for each weak classifier instead of computing an error that estimates the difficulty of each data x_n. Boosting can be considered as an approximate gradient descent in the weak-classifier space [17], and this result yields in a more generic family of boosting methods named *online gradient-boost* [18] that builds the strong classifier F by minimizing the empirical loss defined by:

$$\mathcal{L}(F) = \sum_{n=1}^{N} l(y_n F(x_n)) \tag{3}$$

where $l(.)$ is a loss function. Standard loss functions are:

$$\begin{aligned}
\text{exponential:} &\quad \exp(-x) \\
\text{logit:} &\quad \log(1 + \exp(-x)) \\
\text{doomII:} &\quad 1 - \tanh(x) \\
\text{savage:} &\quad ((1 + \exp(2x))^2)^{-1} \\
\text{hinge:} &\quad \max(0, 1 - x)
\end{aligned}$$

The area selection process often suffers from imprecise drawings of the user or wrong labelling. It implies more mislabelled data than in a carefully controlled training set. Now, it has recently been shown that boosting algorithms with a convex loss function (among which *adaboost*) are particularly sensitive to noise [19]. This leads us to implement Algorithm 1 with the non-convex *DoomII* loss function that is less sensitive to noisy data, as shown in [11].

Moreover, in the images we are dealing with, positive samples are often scarce while it is easier to find areas of non interest. It often implies the training set is unbalanced. To counterbalance this problem, we define a new set of loss functions that take into account the prior probabilities of the training sets:

$$l(x) \leftarrow \frac{l(x)}{p(y)} \tag{4}$$

where priors are estimated by counting the number of positive samples n_+ and negative ones n_-:

$$p(y = 1) = \frac{n_+}{n_+ + n_-} \tag{5}$$

$$p(y = -1) = \frac{n_-}{n_+ + n_-} \tag{6}$$

This results in Algorithm 1 which minimizes Eq. 3 in an online fashion (i.e. each stage of the optimization yields in a functionnal classifier and new samples can be added incrementally). All sample weights are divided by $p(y = y_n)$, such giving more importance to training samples of the class that is underrepresented.

Algorithm 1 Online Gradient-Boost with a priori information

Require: a training sample set $(x_n, y_n)_{1 \leq n \leq N}$, a differentiable loss function $l(.)$
Require: M selectors (pools of weak learners) of K weak learners (i.e. feature component) each
for all $x_n, n \in [1 : N]$ **do**
  Set $F_0() = 0$
  Set the weight $w_n = -l'(0)/p(y = y_n)$ associated with x_n
  for all selector $m \in [1 : M]$ **do**
    for all weak learner $k \in [1 : K]$ **do**
      update weak learner $f_m^k = 0.5 \log\left(\frac{p(x^{ik}|y=1)}{p(x^{ik}|y=-1)}\right)$ with (x_n, y_n)
      update cumulative error $e_m^k \leftarrow e_m^k + w_n \mathbf{1}(sign(f_m^k(x) \neq y)$
    end for
    Select best weak learner with the least total weighted error: $k_m = \arg\min_k(e_m^k)$
    Set $f_m(x_n) = f_m^{k_m}(x_n)$
    Set $F_m(x_n) = F_{m-1}(x_n) + f_m(x_n)$
    Set the weight $w_n = -l'(y_n * F_m(x_n))/p(y = y_n)$
  end for
  Output model at stage n: $F(x)$
end for
Output the final model: $F(x)$

IV. DETECTION

The classifiers trained online have two potential outputs: semantic maps of the local area, which can be superposed to the orthomosaïc, and detectors that produce a semantic labeling of what is seen by the UAV.

A. Semantic maps

Once the training has been done, the final classifier is expressed in a compact way by Formula 2. Weak learner values are given by:

$$f_m(x) = 0.5 \log \left(\frac{p(x^{i_m}|y=1)}{p(x^{i_m}|y=-1)} \right) \quad (7)$$

where i_m is the HOG-LBP coefficient that was selected as the best weak learner and $p(x^{i_m}|y=y_n)$ are estimated using online histograms computed over the training set.

Applying this classifier on the orthomosaïc produces detection maps of the defined object. For a UAV, the interest is two-fold. Target-detection maps (like cars or buildings) are useful for defining the target of the UAV flight and thus planning the path that leads to it. Obstacle-detection maps (such as trees or buildings) are useful for planning paths that avoid potential dangers, especially when approaching the target.

B. Detectors for on-board camera

The classifier parameters are then used in a detector that performs on frames of the video flow, and detect the objects of interest in it. The compactness of the model allows to upload it on a UAV even over a limited-bandwith channel. However, the orthomosaïc is obtained by image synthesis and has different viewing angle and resolution than the images captured by the UAV on-board camera, so domain adaptation has to be performed to use the classifiers in a different geometry.

In Eq. 2, x is the feature computed over a patch according to Eq. 1. In projective coordinates, a patch is composed of points $m_k = (u, v, 1)^\top$ in the orthomosaïc plane. The same real-world points are projected to points $m'_k = (u', v', 1)^\top$ in the video-frames. The intrinsic parameters of the UAV-camera are known and the 3D position of the UAV is given by the GPS system. It is therefore possible to compute the homography that relates both projections [20] according to:

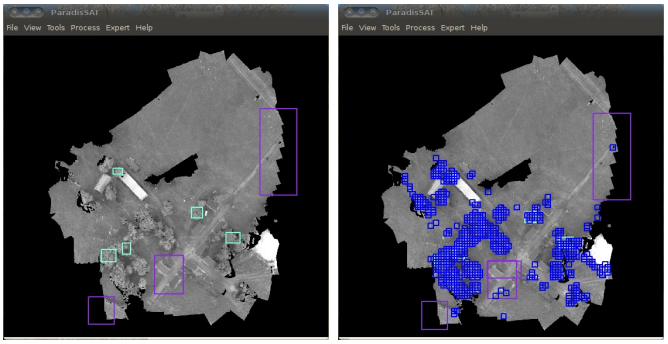
$$m_p = K_{\mathrm{OM}} \cdot H_{R,t} \cdot K_{\mathrm{Cam}}^{-1} \cdot m'_p \quad (8)$$

where K_{Cam} is the intrinsic-parameter matrix of the on-board camera, $H_{R,t} = R - \frac{tn^\top}{d}$ is the transform between the local coordinate systems of the camera and the world (R is the rotation matrix that depends on the UAV attitude; t the translation vector between the two origins; n the normal to the orthomosaïc plane and d the UAV altitude) and K_{OM} is the transform matrix that encodes the change of origin and resolution between the world and the orthomosaïc image. The procedure for detecting objects of interest in a given video-frame consists in Algorithm 2.

In practice, given that the camera viewpoint is vertical with respect to the UAV, rotations are only due to UAV motions and almost every area of the video-frames can be classified. The benefit is that the UAV is able to detect obstacles or targets that appear in its own field of view.

Algorithm 2 Detector adaptation to the video geometry

1: **for all** patch $\{m'_p\}_{1 \leq p \leq P} \in I'$: **do**
2: rectify patch in the orthomosaïc plane according to Eq. 8
3: interpolate a new patch following the orthomosaïc plane sampling grid
4: compute feature x according to Eq. 1
5: classify x according to Eq. 2
6: **end for**



(a) (b)

Fig. 5. (a) Initial selection of areas of interest (green) and non-interest (purple) for designing a tree classifier. (b) Result of the detector (blue areas) after 3 iterations of learning by online gradient boost.

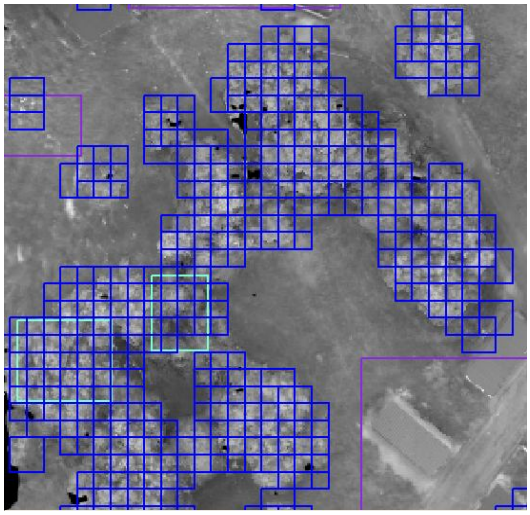
V. EXPERIMENTS AND RESULTS

Data were captured during successive UAV flights above a scene composed of buildings (individual housing), trees, meadows and cars (more-or-less 4000 640x480 frames, with resolutions from 5 to 10cm depending on the altitude). Data of the first flight was used to build the environment mapping (20cm-resolution mosaïc) to learn targets and create semantic maps. Data of subsequent flights was used to test the resulting detector on never-seen images.

A. Online Learning Experiments

In Fig. 5, we show an example of learning interactively a classifier. Initially the orthomosaïc is shown in an interface that allows to select positive and negative sample areas. The system answers by showing on the image the areas detected by the current classifier. After roughly 3 rounds of interactions, the user may consider the detection of objects of interest (here, trees for obstacle avoidance) is satisfactory with only a few remaining false alarms and stop the learning. Fig. 6 shows such semantic maps for two topics that exemplify objects of interest for the UAV: trees (obstacles) and cars (targets).

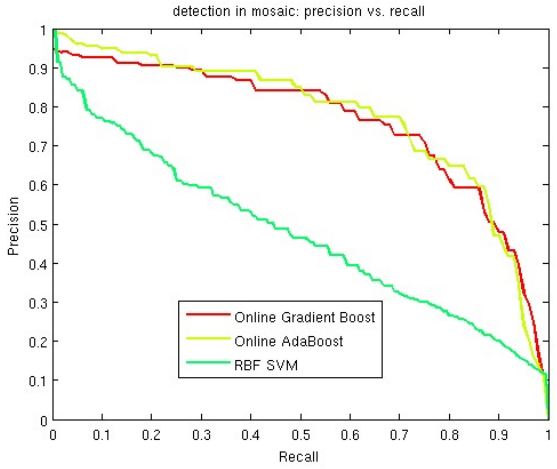
In Fig 7, we show precision-recall curves of tree and car detection in the orthomosaïc. Precision is the proportion of relevant detections in the retrieved results and is defined as $\mathrm{Prec} = \mathrm{TruePos}/(\mathrm{TruePos} + \mathrm{FalsePos})$ with respect to a ground-truth. Recall is the proportion of real targets that were retrieved and is defined by $\mathrm{Rec} = \mathrm{TruePos}/(\mathrm{TruePos} + \mathrm{FalseNeg})$. A trade-off has to be established between these



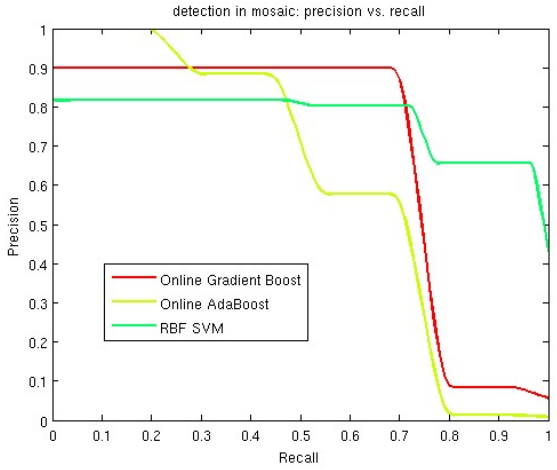
(a)

(b)

Fig. 6. Details of object-detection maps superimposed on the environment map: (a) tree detection for landing obstacle avoidance. (b) car detection for target localization.



(a)



(b)

Fig. 7. Precision-recall curves of detection in the orthomosaïc. (a) Tree detection: both online gradient boost and standard adaboost gives better detection results than the SVM. (b) Car detection: curves are less informative since there is only one car in the dataset, so true positives correspond to training data. However all three methods are able to learn what is not a car and to avoid too much false positives.

two values. Our online gradient boost with *a priori information loss function* was compared with two state-of-the-art learning methods: standard online adaboost, and Support Vector Machine (SVM). In this latter case, a parallel implementation of the SVM [21] helped to keep interaction times low. We used a Radial Basis Function (RBF) kernel and optimal parameters were chosen by cross-validation and grid-search. For each approach, we averaged detection results over 5 rounds of training and testing. Both boosting approaches outperform the SVM-based detector, which is flawed by too many false alarms when recall increases. Among the boosting variants, the gradient boost method we proposed appears to be more able to discriminate targets from clutter than the standard adaboost.

B. Target Detection in a Video Flow

Detectors for various objects (trees, buildings and cars) learned during the first flight are now applied on frames extracted from the second video. These detectors were geometrically adapted using the approach of section IV-B. Fig. 8 show that in spite of the change of viewpoint and scale, most objects of interest are retrieved in the new images. Car and building detections show that in spite of specific method to handle object rotation at learning stage, objects with various orientations can be detected.

Hardware requirements. We measured computing times on a UAV payload system with a core 2 duo processor (1.5 GHz clock rate). All detectors have a 1.6 Hz framerate which corresponds to a UAV advance of less than 1 m: for example landing-area obstacle detection remains tractable to deal with. The boosting models have a typical size of 13.5 kB and are on average 20 times smaller than SVM models, which make them more suitable for upload in restricted bandwidth conditions.

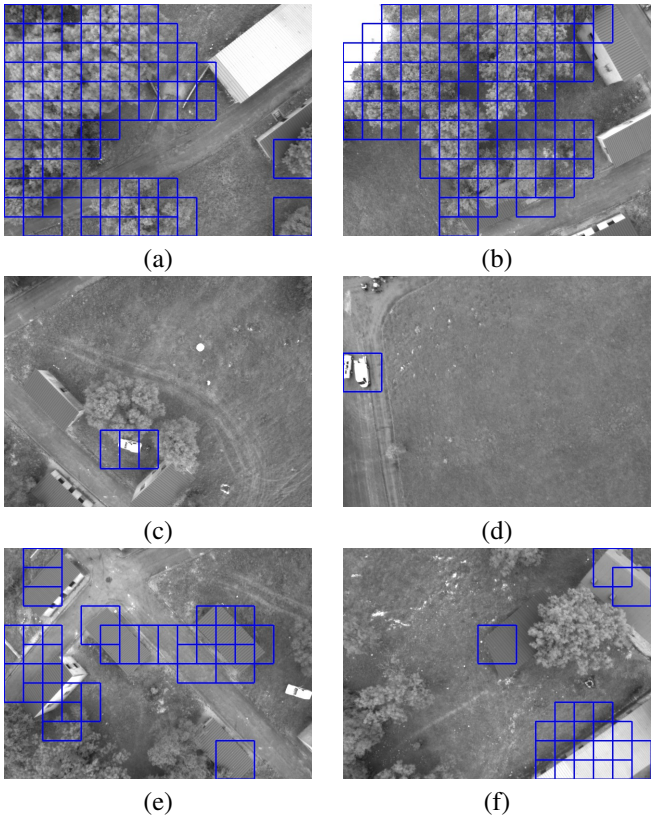


Fig. 8. Detection results (blue squares) in video frames after geometric transform. First row (a) & (b): tree detections; second row (c) & (d): car detections; third row (e) & (f): house detections.

VI. PERSPECTIVES

The proposed learning and domain adaptation approaches remain valid with different object models. Further work will investigate different kinds of models that may allow to discriminate a wider variety of objects. In aerial images with vertical point of view, rotation is handled by our method by relying on the variety of samples that are selected. Rotation-invariant features will be tested along with schemes to find objects at different orientations. Eventually, the real challenge is to be able to learn directly 3D-objects of interest by designing them in the 3D environment map, so the next step will be to build textured 3D models that allow to learn targets seen from any viewpoint.

VII. CONCLUSIONS

In this paper we presented a novel approach for rapid semantic mapping for a UAV. Key-outputs are semantic maps superimposable to environment maps of the explored area and object detectors directly usable on the UAV. The integrated workflow we presented goes beyond the standard mapping procedure for control, by using this mapping to generate *ad hoc* target classifiers in an interactive procedure based on online gradient boost. We think that real-world scenarios will require collaboration between a robot that is autonomous for what it is efficient at and an operator in the ground control station for defining objectives that require a high level of conceptualization.

REFERENCES

[1] F. Fraundorfer, L. Heng, D. Honegger, G.-H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, 2012.

[2] A. Nuechter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 915–926, 2008.

[3] O. Mozos, R. Triebel, P. Jensfelt, A. Rottman, and W. Burgard, "Supervides semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[4] P.-E. Forssen, D. Meger, K. Lai, S. Helmer, J. Little, and D. Lowe, "Informed visual search: combining attention and object recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, California, USA, 2008.

[5] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[6] K. Ali, F. Fleuret, D. Hasler, and P. Fua, "A real-time deformable detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 2, pp. 225–239, 2012.

[7] B. Le Saux and M. Sanfourche, "Robust vehicle categorization from aerial images by. 3d-template matching and multiple classifier system," in *IEEE International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, september 2011.

[8] J. Inglada, "Automatic recognition of man-made objects in high resolution optical remote sensing images by svm classification of geometric image features," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, 2007.

[9] R. de Nijs, S. Ramos, G. Roig, X. Boix, L. Van Gool, and K. Kühnlenz, "On-line semantic perception using uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, 2012.

[10] M. Lourakis and A. Argyros, "SBA: A Software Package for Generic Sparse Bundle Adjustment," *ACM Trans. Math. Software*, vol. 36, no. 1, pp. 1–30, 2009.

[11] N. Chauffert, J. Israël, and B. Le Saux, "Boosting for interactive man-made structure classification," in *IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, july 2012.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of Computer Vision and Pattern Recognition*, Washington DC, USA, 2005, pp. 886–893.

[13] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[14] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured hog-lbp for object localization," in *Porceedings of Computer Vision and Pattern Recognition*, Colorado Springs, USA, 2011.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, 1997.

[16] T. T. Nguyen, H. Grabner, B. Gruber, and H. Bischof, "On-line boosting for car detection from aerial images," in *IEEE International Conference on Research, Inovation and Vision for the Future (RIVF'07)*, 2007, pp. 87–95.

[17] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," *Advances in Neural Information Processing Systems*, vol. 12, pp. 512–518, 2000.

[18] C. Leistner, A. Saffari, P. Roth, and H. Bischof, "On robustness of on-line boosting: A competitive study," in *Proceedings of ICCV Workshop on On-line Learning for Computer Vision*, Kyoto, Japan, 2009.

[19] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Machine Learning*, vol. 78, no. 3, pp. 287–304, 2010.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*. Cambridge University Press, 2003.

[21] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast support vector machine training and classification on graphics processor," in *Proceedings of International Conference on Machine Learning*, Helsinki, Finland, 2008, pp. 104–111.