DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

Optimizing color information processing inside an SVM network

J. Pasquet [*#1], ***G. Subsol*** [#2], ***M. Derras*** [*3] ***and M. Chaumont*** [†#4]

[*] ***Berger-Levrault, Labège, France***
[1] pasquet@lirmm.fr
[3] mustapha.derras@berger-levrault.com

[†] ***Univ. Nimes, France***
[4] chaumont@lirmm.fr

[#] ***LIRMM, Univ. Montpellier / CNRS, France***
[2] subsol@lirmm.fr

Abstract

Today, with the higher computing power of CPUs and GPUs, many different neural network architectures have been proposed for object detection in images. However, these networks are often not optimized to process color information. In this paper, we propose a new method based on an SVM network, that efficiently extracts this color information. We describe different network architectures and compare them with several color models (CIELAB, HSV, RGB...).

The results obtained on real data show that our network is more efficient and robust than a single SVM network, with an average precision gain ranging from 1.5% to 6% with respect to the complexity of the test image database. We have optimized the network architecture in order to gain information from color data, thus increasing the average precision by up to 10%.

Introduction

Object detection in aerial images is a complex task which is usually dealt with by a machine learning approach where classification is based on a single or cascade of SVM [1, 2]. However, it is hard to concatenate different features in a single vector without any normalization problem [3].

A way to overcome this difficulty is to distribute the different features in different inputs of a neural network and to use a classifier as output node. Many network architectures have recently been proposed to create deep convolutional neural networks [4, 5]. However, this kind of approach usually requires millions of parameters and thus a huge database to determine them. To solve this problem, we can replace each neuron of the network by a linear SVM which is known to converge faster [6]. Similar to a neuronal network, each SVM uses an activation function.

Currently most image databases are in color. But, surprisingly, most object detection algorithms use only a single color channel, which can be obtained by different methods [8], and then the resulting gray-level image is processing by the network. Color information from the image is then only very partially used.

In this paper, we first show experimentally that using all the color information may lead to a significant improvement in accuracy[1]. Then we discuss how to choose the network architecture in order to use color information in an optimal way. For this, we give an overview of our SVM network system and describe the experimental database in section 2. In section 3, we analyze the performance by using entire color information and studying the influence of the color space. In section 4, we describe several architectures for our network and then find and analyze the optimal one. Finally, we conclude and present future work.

SVM network and data

Our SVM network

Our object detection procedure classifies images based on the Histogram of Oriented Gradients descriptor [9]. First we normalize the bounding box of each object image of the training database to an image of constant size, as in [2]. Then we extract the HOG features in a sliding window on the normalized image of the object. Thus, for an object, we obtain a vector composed of many HOG features. To get a multi-resolution descriptor, we repeat the extraction step with different sizes of sliding windows, as in [10].

The classical method [2, 10] is to build a single large vector by concatenating vectors corresponding to the different sizes of sliding windows and then use a SVM classifier. But, in our case, we decided to use the different vectors as inputs of an SVM network [11]. To get better multi-resolution robustness, each input neuron learns on a single size. In addition, to introduce some invariance in our SVM network, we also create a set of input neurons which are randomly connected to part of the vector HOG in different resolutions [12].

Then a hiddin layer is randomly connected to input neurons to get a better high-level abstraction. To break the linearity an asymmetric sigmoid function (see equation 1) is apply as an activation function after each SVM.

$$f(x) = \frac{1}{(1+e^{-\alpha x+\lambda})} \tag{1}$$

Where α and λ are two parameters computed with a validation database according to [7].

[1]We thank L. Deruelle and F. Bibonne (Berger Levrault, Labège, France) for providing access to image data and for fruitful discussions.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

The output neuron is fully-connected to the last hidden layer. During the classification step, this neuron will return the probability for the input window to contain a target object.

Our experimental database

We are interested in the detection of urban objects in high-definition aerial images. More specifically, we focus on detecting man-hole covers [13] and tombs [14] for geo-localization purposes. In particular, tomb detection appears to be a very challenging problem as tombs vary substantially in appearance, color and size in aerial images. Moreover, vegetation, shadows created by the numerous buildings, people walking or utility vehicles may create many distortions and occlusions in the images. We thus selected this application for our tests.

For the training database, we used 19 industrial aerial images of cemeteries with a resolution of 2.5 cm/pixel, which contained about 4,500 tombs. For the validation database, we used 3 images containing about 750 tombs. To evaluate our algorithms, we used 2 cemetery images (about 700 tombs) and manually delineated the rectangular bounding boxes of tombs which are then considered as ground truth.

During the learning and test step, we use the High Performance Computing resources of HPC-LR[2] in order to have a reasonable computational time.

During the evaluation step, detection results are given as a list of rectangular bounding boxes. To be considered as a correct detection, the area of overlap, noted A (see equation 2), between the detected boxes B_d and the ground truth B_g must exceed 58%.

$$A = \frac{Area(B_d \cap B_g)}{Area(B_d \cup B_g)} \tag{2}$$

Using all the color information

How to integrate information in the network?

Color information is defined in a color space **S** and is given as a set of S channels c (in general **S**=3). For example, $c \in \mathbf{S} = \{\mathbf{R},\mathbf{G},\mathbf{B}\}$ for the *RGB* color space. In the following, f_r^c denotes the HOG vector computed for a given window for channel c at a resolution of $r \in \{0,1,...,N\}$, with N being the number of extracted windows of different sizes. Let p_r^c denotes the output of a neuron which takes vector f_r^c as input.

The simplest idea consists of directly sending the f_r^c vectors inside the network. This means that each channel is processed separately and the number of input neurons is N times c. The hidden layer allows the network to combine the different channels and manage the color information. For this purpose, we need to increase the number of hidden neurons and thus the overall complexity. Figure 1 shows this kind of architecture. We will call this architecture the *separate architecture* in the case of a color space with $S = 2$ channels

Choice of color space

We test the separate configuration network through five standard color spaces. The *RGB* space is the addition of three fundamental colors, i.e. red, green and blue to form all the colors. The luminance is a definition of the intensity level ($S = 1$),

[2]HPC@LR: High Performance Computing from Languedoc-Roussillon, https://www.hpc-lr.univ-montp2.fr

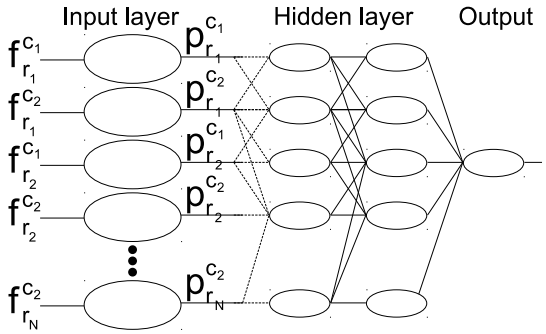

Figure 1. *The separate architecture.*

$Y = 0.21R + 0.71G + 0.07B$ according to [15]. $CIE-LAB$ and $CIE-LUV$ are two color spaces based on human perception. These models give optimal results in some image processing applications as in [16]. The *HSV* (Hue, Saturation and Value) color space is a cylindrical coordinate representation of the *RGB* model. It is very effective for image segmentation, as shown in [17].

Optimal Parameters for SVM Network

The best number of hidden SVN neurons and *random neurons* in the SVM network is an important parameter. If the number of *random neuron* is large enough the precision converges to a constant asymptote[12]. We thus took 400 random neurons in all the experiments. In Figure 2 we use a validation database which allows us to determine the number of hidden neurons. As expected, this number is directly linked to the size of first layer (input neurons) and thus to the color space used. The best number of hidden neurons is 300 and 600 for a color space of size $S = 1$ and $S = 3$ respectively.

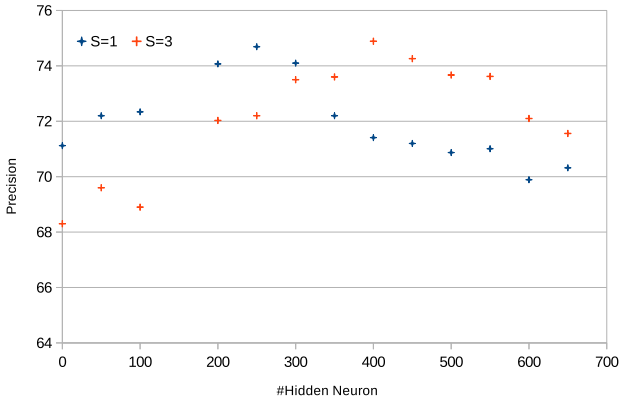

Figure 2. *Average of the precision compute with the validation database for a recall between 45% and 80% as function of the number of hidden*

Experimental results

First, we compare our network based on the *separate architecture* with the classical approach where we use an unique SVM which receives an input vector composed of concatenations of f_r^c.

In Figure 3, the experiments show that for any color space, the SVM network outperforms a single SVM. We even obtain a

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

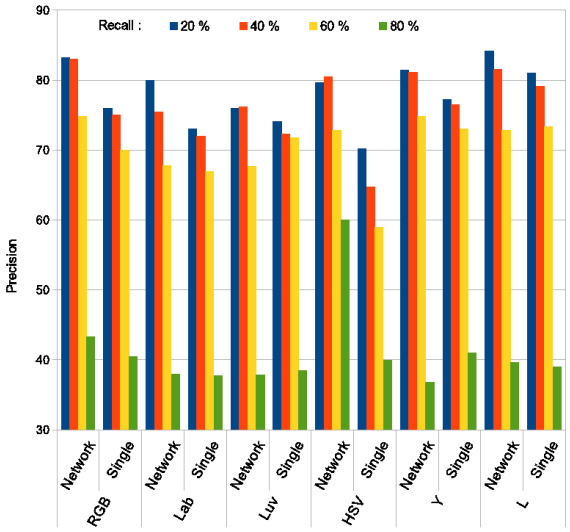

Figure 3. *Comparison of the performance of our SVM network and a single SVM approach. For each experiment we present 4 results in blue, orange, yellow and green bars which respectively correspond to a recall value of 20%, 40%, 60% and 80%.*

small average gain in precision of about 1.5% if we only use the lightness L. In fact, if we look at all the results for the single SVM, we can see that using a 3-channel space such as *RGB* or *Lab* does not increase the precision compared to the 1-channel space. This shows that a single SVM does not efficiently process the full color information.

In the case of SVM networks, we can see that the L channel gives slightly better results than the Y channel, but the coefficients used to compute the luminance are maybe not optimal for our image application. However, unlike the single SVM, there is a clear advantage to using all color information. Indeed, the average precision increases from 69.5% for the L channel to 71.1% for the *RGB* space. In a similar way, the *HSV* space gives interesting good results, and it is very efficient for a recall value of over 80%, with precision increased by threefold compared to the L channel. Color spaces based on human perception like *Lab*, *Luv* decrease the average precision by 5%.

In this section, we show that our SVM network is slightly better than a single SVM when we use only one channel. It becomes much more efficient when we use all the color information, with an average precision increase of about 3%. Note that if we select a sub-part of the test database where the objects to detect are considered complex, i.e. when the tombs are randomly oriented or very close to each other, the SVM network appears to be much more robust than the single SVM, with a marked increment in precision of about 6%, as we can see in Figure 4.

Nevertheless, the overall performance gain of the SVM network remains quite low (about 1.5%). In the next section, we present other network architectures which improve efficiency.

A Network Architecture to Optimize Color Information Processing

As shown in the previous section, information from the *RGB* and *HSV* model increases the performance. Nevertheless, compared to the results obtained with one grey level (L or Y), results with color are closed. The *separate architecture* is probably not well enough designed to integrate the color information. This can be explained because the *separate architecture* does not directly create relationships between neurons with the same resolution. In this section, to deal with this problem we introduce and test five other architectures to process 3-channel color information.

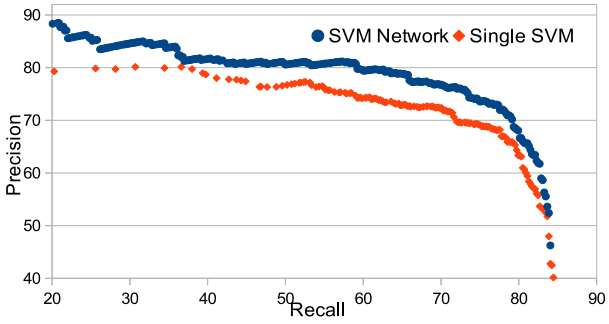

Figure 4. *Comparison of the performance of our SVM network and a single SVM approach, using only the L channel, for the most complex test database.*

The Fusion architecture

In this approach, we propose to concatenate the different f_r^c for a number of windows N and with the entire c value of the channel space $\mathbf{S}$.

The advantage of this method is that each input neuron simultaneously treats the entire color space, so during the learning step input neurons will be able to adapt to the database. Moreover, the number of input neurons does not increase, so we do not increase the size of the hidden layer and the computational time. We call this the *fusion architecture*.

The Maximum and Product architecture

This method was inspired by the following paper [18]. Indeed, using the orientation and magnitude of the gradient image, we do not want to focus on the low variations. Moreover, we propose to only select signals with the highest variation. The first step requires building the input neurons as in the *separate architecture*. The second step is to connect all input neurons with the same resolution r to a specific *max neuron*. We define a *max neuron* as a neuron which relays the highest signal without any activation function (see equation 3).

$$p_r^{max} = \max_{c \in \mathbf{S}} p_r^c \tag{3}$$

The second layer contains all the *max neurons* and it also has two advantages. Firstly, it selects only the highest channel intensity independently of r and secondly, only the *max neuron* outputs are used by the hidden layer. Schema 5 is an example of this architecture with $S = 2$. We call this the *maximum architecture*.

Similarly, we can define the product architecture where each *max neuron* is replace by a *product neuron*. By contrast with *max neurons*, the *product neurons* perform a linear quantification as defined in equation 4. This transformation is effective because the *product neurons* returns different scores if few intensity channels

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

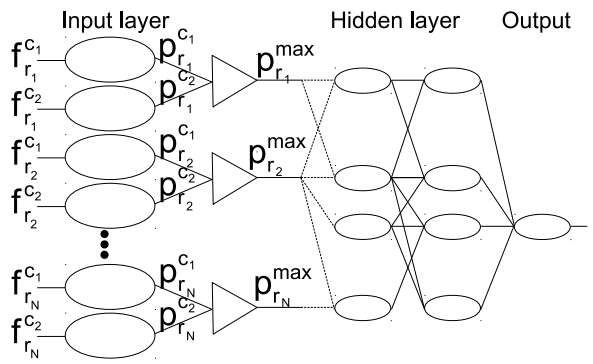

Figure 5. *The maximum (and product by replacing p^{max} by p^{prod}) architecture.*

have stray variations. We call this the *product architecture*.

$$p_r^{pro} = \prod_{c \in \mathbf{S}} p_r^c \tag{4}$$

The PCA architecture

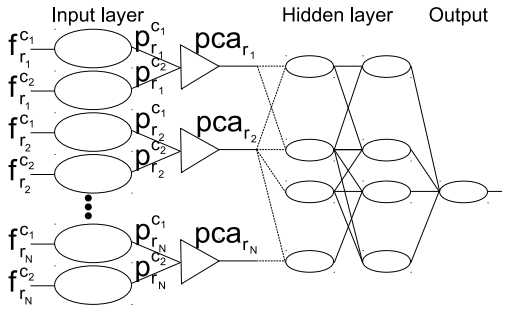

Figure 6. *The PCA architecture*

To choose the best color subset, the PCA is one classic alternative [8]. We assume that the best color subset is not always the same for each feature resolution. We thus have to fit a PCA projection noted pca_r for each resolution r, Schema 6.

Like the *maximum architecture*, this architecture has two design steps. The first one consists in the building input neurons similarly to the separate architecture. The second step consists to connect all input neurons with the same resolution r to a specific *PCA neuron*. We then calculate the eigenvalues after the learning of the input neurons. Each *PCA neuron* returns only the first principal component which is a simple scalar product. To break the linearity, we add an asymmetric sigmoid activation function as for SVM neuron. This architecture is called *PCA architecture*.

The Stack architecture

In the *fusion architecture*, each input neuron treats all vectors from $\mathbf{S}$. More generally, we may want to use a set of feature vectors which do not belong to the same space as for example SURF and HOG vectors. This requires first to scale each feature in order than the network can process all of them in a consistent way.

For this, the *fusion architecture* is refined by integrating a sub-layer which takes the features f_r^c as separate inputs and computes p_r^c. These values allow to compute a new feature vector f_r^{weight} which is a weighted combination of f_r^c for a given resolution r by:

$$f_{r_1}^{weight} = p_{r_1}^{c_1} f_{r_1}^{c_1}, p_{r_1}^{c_2} f_{r_1}^{c_2}, ..., p_{r_1}^{c_S} f_{r_1}^{c_S} \tag{5}$$

The low level input vectors will be used for the higher neurons [19] with a rescaling in this new architecture called *stack architecture*, as in Figure 7.

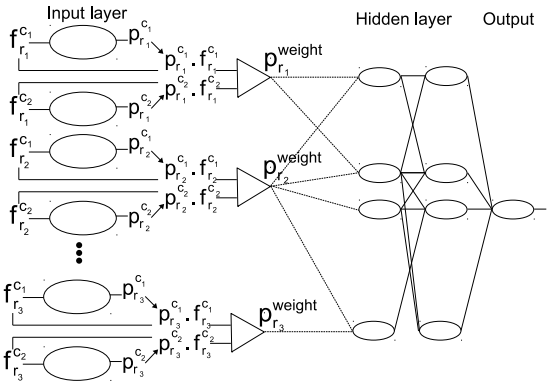

Figure 7. *The stack architecture.*

Experimental Results

In Figure 8, the *maximum architecture* does not enhance the results. Indeed, in natural images the intensity in the three channel *RGB* often fluctuates at the same time. The addition of color information is then weak compared to luminance. However, the *product architecture* is more sensitive to any fluctuations in different color components and thus has about 7% better precision for a recall range of 55% to 82%. In the same idea, the precision of the *PCA architecture* is similar or less than the precision of the *product architecture*.

The *fusion architecture* also increases the performance for a recall range of 50% to 82%. But this architecture can only be effective for features of the same type, unlike other proposed architectures.

The *stack architecture* increases the precision by up to 10% compared to the *separate architecture*. In fact, this architecture combines all advantages : it has a minimal number of hidden neurons like the *maximum architecture* and all of the information from channels is kept like the *separate architecture*. Moreover, there is no requirement to give an explicit formula to transform the different channels to only one as in the *maximum* or *product* architecture. In addition, the *stack architecture* can be used to combine descriptors from different feature spaces.

However, the disadvantage of this method lies in the computational cost during the learning step and during the testing step. To deal with the calculation time during the evaluation step the activation path could be used which reduced the requiring time by 80% [12].

Conlusion and Future Work

In this paper, we first compare a single SVM-based approach to an SVM network approach. We show that an SVM network

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

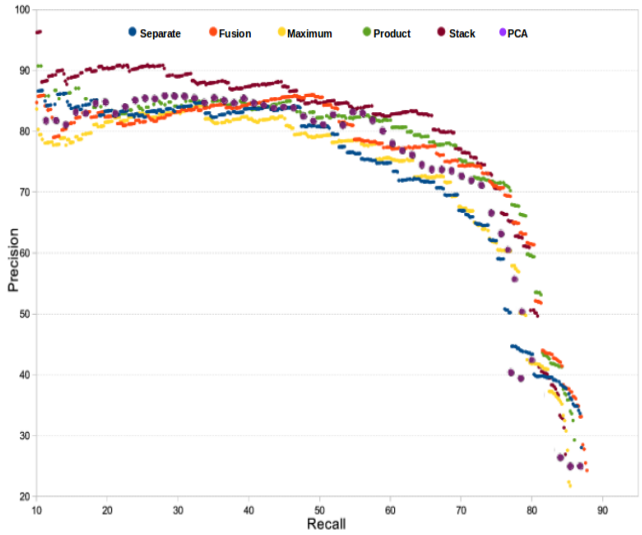

Figure 8. *Results obtained with the RGB model and different architectures are represented with a ROC curve (precision as a function of the recall) where the probability of being a tomb varies.*

outperforms the single SVM by an average precision gain ranging from 1.5% to 6%. In addition, we notice that using the three color components does not significantly increase the single SVM results. On the contrary, we can see that an SVM network is able to extract more information from the different color components and thus increases the final performance.

We then introduce different network architectures in order to effectively combine the features of each color component. The best proposed takes into account all the different features and normalizes them using a smart weight. We show that the so-called *stack architecture* outperforms other architectures by an average gain in precision ranging from 4% to 10%.

Future works will deal with the addition of features such as color components, SURF, SIFT or pixel intensity where the stack architecture could be used. But more tests are required to validate the real effectiveness of this design.

Another extension of our work would be to extend the stack architecture to a deep neural network. Thus the weighted combination of f_r^c will be learned by back propagation.

Acknowledgement

The authors would like to thank the Berger-Levrault group for supporting this research. Berger-Levrault is a French public regulation expert that addresses healthcare and local public administrations. With almost 60 thousands customers and more than 1,000 employees, the Berger-Levrault group is a key enabler for the development of e.Government in France. Bringing people/citizens/patients and their public administrations closer is a key focus of the Berger-Levrault group.

References

[1] S. Sahli, Y. Ouyang, Y. Sheng, and D. A. Lavigne, "Robust vehicle detection in low-resolution aerial imagery," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Apr. 2010, vol. 7668.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–511–I–518 vol.1.

[3] H. Meng, D. R. Hardoon, J. Shawe-Taylor, and S. Szedmak, "Generic object recognition by combining distinct features in machine learning," in *the International Society for Optical Engineering (SPIE), 9th Applications of neural networks and machine learning in image processing IX*, Aug 2005.

[4] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, Eds., pp. 2553–2561. Curran Associates, Inc., 2013.

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 2155–2162.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[7] Jing Gao and Pang ning Tan, "Converting output scores from outlier detection algorithms into probability estimates," in *Proc. of the Sixth IEEE Int. Conf. on Data Mining (ICDM06)*, 2006, pp. 212–221.

[8] A. Krizhevsky, Ilya S., and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, June 2005, vol. 1, pp. 886–893 vol. 1.

[10] Qiang Zhu, M.-C. Yeh, Kwang-Ting Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 1491–1498.

[11] Sangwook Kim, Swathi Kavuri, and Minho Lee, "Deep network with support vector machines.," in *ICONIP (1)*, Minho Lee, Akira Hirose, Zeng-Guang Hou, and Rhee Man Kil, Eds. 2013, vol. 8226 of *Lecture Notes in Computer Science*, pp. 458–465, Springer.

[12] Pasquet J., Chaumont M., Subsol G., and Derras M., "An efficient multi-resolution SVM network approach for object detection in aerial images," in *Machine Learning for signal processing, 2015*. IEEE, 2015.

[13] J. Pasquet, T. Desert, O. Bartoli, M. Chaumont, C. Delenne, G. Subsol, M. Derras, and N. Chahinian, "Detection of manhole covers in high-resolution aerial images of urban areas by combining two methods," in *Joint Urban Remote Sensing Event (JURSE)*, 2015.

[14] M. Chaumont, L. Tribouillard, G. Subsol, F. Courtade, J. Pasquet, and M. Derras, "Automatic localization of tombs in aerial imagery: Application to the digital archiving of cemetery heritage," in *Digital Heritage International Congress (DigitalHeritage), 2013*, Oct 2013, vol. 1, pp. 657–660.

[15] M. Anderson and S. an Stokes M. Motta, R. an Chandrasekar, "Proposal for a standard default color space for the internet - sRGB," in *Proc. of IS&T and SIDś 4th Color Imaging Conference: Color Science, Systems and Applications*, 1996, pp. pages 238–246.

[16] P. Ganesan, V. Rajini, and R.I. Rajkumar, "Segmentation and edge

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-243

detection of color images using CIELAB color space and edge detectors," in *Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on*, Dec 2010, pp. 393–397.

[17] Gwanggil Jeon, "Measuring and comparison of edge detectors in color spaces," *International Journal of Control and Automation*, vol. 6, pp. 21–29, Oct. 2013.

[18] D. Aldavert, A. Ramisa, R. López de Mántaras, and R. Toledo, "Real-Time Object Segmentation using a Bag of Features Approach," in *13th International Conference of the ACIA*, L'Espluga de Francolí, Catalonia, Spain, 2010, IOS Press, IOS Press.

[19] Kai Ming Ting and Ian H. Witten, "Stacked generalization: when does it work," in *Procs. International Joint Conference on Artificial Intelligence*. 1997, pp. 866–871, Morgan Kaufmann.

Author Biography

Jérôme Pasquet received a master degree in computer science from the university of Montpellier II, France, in 2013. He is currently working toward the PhD degree in LIRMM (Montpellier Laboratory of Informatics, Robotics and Microelectronics). His research interests are urban objects detection and segmentation in aerial photography.

Gérard Subsol obtained a Ph.D. Thesis in Computer Science in 1995. He was successively Ph.D. student, then Expert-Engineer at INRIA Sophia Antipolis, Research Engineer at University of Avignon and R&D Engineer with the start-up company Intrasense. Since 2006, he has been a CNRS Researcher at LIRMM located in the South of France. He is currently working on several applications of 2D and 3D image processing.

Mustapha Derras obtained a Ph.D. Thesis in Computer Science in 1993. He is currently Director of Technology, Innovation and Research of Berger-Levrault. Previously he held positions with the aim of creating innovative new products as Business Unit Director Catia, Dassault Systèmes, Marketing and R&D at TIMEG, Director of Business Unit CAT at Cadence Design Systems, software development architect at General Electric Medical Systems and project Manager at CLAAS where he completed his mobile robotics and image processing studies.

Marc CHAUMONT received his Engineer Diploma in Computer Sciences at the INSA (National Institute of Applied Sciences) of Rennes, France in 1999, his Ph.D. at the IRISA Rennes in 2003, and his HDR ("Habilitation à Diriger des Recherches") at the University of Montpellier in 2013. Since September 2005, he is an Assistant Professor in the LIRMM laboratory of Montpellier and the University of Nmes. His research areas are multimedia security (steganography, watermarking, digital forensics, video & image compression) and segmentation & tracking in images and videos. He is member of the TC of IEEE SPS - Information Forensics and Security for the period 2015-2017. He was program chair of ACM IH&MMSec'2013. He is reviewer for more than 20 journals (IEEE TIFS, IS&T JEI, ...) and for more than 10 conferences (EI MWSF, IEEE WIFS, ACM IH&MMSec, IEEE ICIP, ...).