Comparaison de la segmentation pixel et segmentation objet pour la détection d'objets multiples et variables dans des images

J. Pasquet[1,3], M. Chaumont [1,2], G. Subsol [1]

[1]LIRMM, CNRS/Université Montpellier 2, France
[2]Université de Nîmes, France
[3]Berger Levrault, 31676 Labège, France

Résumé

Cet article étudie et compare deux méthodes de segmentation. La première est la segmentation par objet [VJ01] où l'on cherche à détecter des fenêtres à partir d'un modèle. La seconde est la segmentation pixel [ARTLdM10, FVS08], où l'on cherche à déterminer à quelle classe appartient chaque pixel. De plus, nous proposons une extension au modèle classique de détection par cascade de HOG [ZYCA06] en utilisant les sacs de mots visuels. Des expérimentations sur des jeux de données réelles permettront la comparaison et mettront en avant un gain de performance non négligeable de notre méthode.
In this paper we compare two classical methods of segmentation. First one, the object segmentation, where we try to find an object in a sliding window from a generated model. The second one, the objet level pixel segmentation, where we try to classify all the pixels in an image. Moreover, we propose an extension to the object segmentation, this extension uses bags of features. Finally, we apply some experementations on these two methods using real data set in order to compare the performance.

Mots clé : Segmentation d'images, détection d'objets, sac de mots visuels

1. Introduction

Dans ce papier, nous comparons deux méthodes classiques de segmentation à savoir la segmentation au niveau objet et la segmentation au niveau pixel.

La première approche consiste à construire par apprentissage un modèle des objets recherchés. Pour cela, une base d'apprentissage constituée de multiples imagettes de chaque objet est utilisée. Pour chaque imagette, nous pouvons extraire un vecteur caractéristique décrivant l'objet et servant à l'apprentissage d'un modèle [VJ01, RBK95]. Ensuite, afin d'évaluer notre modèle et de l'utiliser nous cherchons la présence des objets sur une image test. Pour cela, il est nécessaire de lancer un test de présence via une fenêtre glissante de taille variable en tous points de l'image. Le nombre de fenêtres étant très important, pour obtenir un résultat dans des temps raisonnables, nous devons réaliser plusieurs optimisations. De nombreuses solutions existent afin de lutter contre ce problème comme l'utilisation de cascades [VJ01] réduisant fortement la complexité de la phase de tests, ou une présélection des zones d'intérêts dans l'image et donc une diminution des zones à tester [LBH08].

Cependant, une autre solution, la segmentation pixel, consiste à changer l'approche et à classifier directement chaque pixel de l'image [FVS08, RnAK*11]. Ainsi, lors de la face d'apprentissage, un vecteur caractéristique est extrait de chaque pixel et associé à un label. L'ensemble des vecteurs caractéristiques servent à l'apprentissage d'algorithmes. Lors de la face d'évaluation, chaque pixel de l'image test est classifié et associé à une probabilité d'appartenir à un objet. Ainsi, le nombre de vecteurs à tester est faible en comparaison à l'approche objet. Toutefois, la phase d'évaluation retourne une carte de probabilité qu'il est nécessaire de post-traiter afin d'obtenir des régions.

Afin, de comparer les deux méthodes nous utilisons une base de données constituée d'images aériennes couleur haute résolution de cimetière. Les objets recherchés étant les tombes, il a été montré dans [CTS*13] que ce problème est difficile. En effet, les tombes sont des objets à forte variabilité, multiples et très proches.

Dans une première section 2, nous présenterons la segmentation pixel ainsi que les descripteurs utilisés. Puis, dans la section 3 nous présenterons en détail l'approche par

© CORESA 2014, COmpression et REprésentation des Signaux Audiovisuels

construction d'un modèle. Nous proposons également une extension du modèle objet dans la partie 3.2. Dans notre approche, nous proposons d'introduire les sacs de mots visuels multi-fenêtre pour décrire les objets. Ensuite, nous présenterons notre base d'apprentissage ainsi que les résultats dans la section 4. Finalement, nous conclurons et proposerons des perspectives.

2. Approche pixel

L'approche pixel consiste à classifier directement chaque pixel de l'image [FVS08, RnAK*11]. Le schéma classique de fonctionnement de cette approche se divise en quatre étapes : l'extraction d'un vecteur caractéristique par pixel, la quantification des vecteurs en mots visuels, la statistique des mots visuels présents dans le voisinage du pixel étudié, c'est à dire la création d'histogrammes représentant la fréquence d'apparition des mots visuels, et la classification des histogrammes de mots visuels.

2.1. Extraction des caractéristiques

De nombreuses caractéristiques peuvent définir l'information un pixel. Dans son article [ARTLdM10], Aldavert et coll. montrent que les histogrammes de gradient orienté (HOG) sont bien adaptés. De plus, le temps de calcul de cette caractéristique extraite, de façon dense, peut être considérablement amoindri grâce à l'utilisation d'image intégrale [ZYCA06]. Les vecteurs HOG permettent de décrire la forme locale d'une zone à partir d'une distribution d'orientation et d'amplitude de gradient. Pour extraire le vecteur HOG d'une région (ou bloc) de dimension D_H nous divisons cette région en cellules de dimension réduite D_C. Dans chacune des cellules, nous formons un histogramme des N orientations du gradient (équation 1). Pour $N = 8$, chaque pixel vote pour son orientation principale, c'est à dire 0°, 45°, ... ou 325°.

$$\theta = floor\left(\frac{arctan(\frac{G_y}{G_x}) + \pi}{2\pi}.N\right).\frac{2\pi}{N} \quad (1)$$

Avec G_x et G_y les gradients selon l'axe x et y et $floor$ la fonction qui arrondit à l'entier inférieur.

Le vote de chaque pixel est pondéré par la norme du gradient (équation 2) afin de différencier les fortes et faibles discontinuités. Les histogrammes de chaque cellule sont ensuite concaténés puis normalisés avec la norme L2. Cet histogramme normalisé forme le vecteur HOG. La figure 1 récapitule le processus de formation du vecteur HOG.

$$G = \sqrt{G_x^2 + G_y^2} \quad (2)$$

Avec G_x et G_y les gradients selon l'axe x et y.

2.2. Création du dictionnaire

Chaque pixel est caractérisé par un vecteur HOG. Afin d'utiliser l'approche par sac de mots visuels, nous devons trouver les meilleurs représentants des vecteurs HOG. Pour obtenir les K meilleurs représentants nous utilisons une méthode de classification non supervisée tel que le

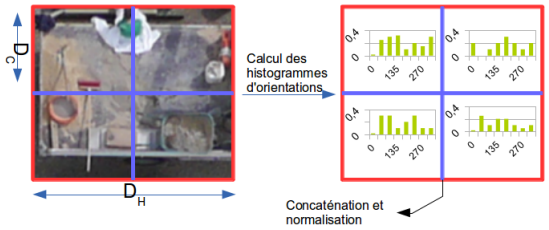

Figure 1: *Les contours rouges représentent la zone où le HOG est extrait et les contours bleus les délimitations dans le cas où nous souhaitons 4 cellules.*

classique Kmeans. Cependant, la complexité du Kmeans étant importante pour un nombre élevé de clusters (représentants) [MM09], nous pouvons utiliser des méthodes de Kmeans hiérarchiques [NS06] ou des forêts d'arbres aléatoires [MNJ08] afin d'obtenir avec des coûts raisonnables un grand nombre de mots visuels. Dans notre cas d'étude nous utiliserons des ERF.

2.3. Création du vecteur caractéristique et apprentissage

Une fois le dictionnaire créé, chaque vecteur HOG de chaque pixel est quantifié en un mot visuel. Afin d'effectuer l'apprentissage, il est nécessaire d'obtenir une liste de vecteurs caractéristiques pour chaque objet. Pour ce faire, nous passons une fenêtre glissante de largeur L_f sur chaque pixel de l'image. Nous créons dans la fenêtre, un histogramme des fréquences d'apparition des mots visuels. Cet histogramme est le vecteur caractéristique du pixel central à l'image. Lors de la création de la base d'apprentissage, connaissant la vérité terrain, nous associons au vecteur caractéristique le label du pixel central. Une fois la base d'apprentissage construite nous effectuons l'apprentissage avec un classifieur linéaire. Le schéma 2 récapitule tout le processus d'extraction de vecteurs caractéristiques dans le cas d'une approche pixel.

2.4. Phase de test

Lors de la phase d'évaluation, à l'aide du dictionnaire de la phase d'apprentissage nous quantifions l'image de test en mots visuels. Une fenêtre glissante sert ensuite à évaluer chaque pixel de l'image en créant un histogramme de fréquence. Cet histogramme de fréquence constitue le vecteur caractéristique qui est classifié par le classifieur. Notons que Aldavert et coll. [ARTLdM10] proposent une version optimisée de la phase d'évaluation. En effet, grâce à l'utilisation d'un classifieur linéaire à chaque mot visuel peut correspondre un score constant. En remplaçant chaque mot visuel par son score nous pouvons construire une image intégrale [VJ01] de l'image de scores. L'évaluation d'une région autour de pixel correspond à la somme des valeurs des scores des mots visuels de la région. Elle ne prend alors que quatre calculs et s'effectue en temps réel. De la phase de tests, il

© CORESA 2014.

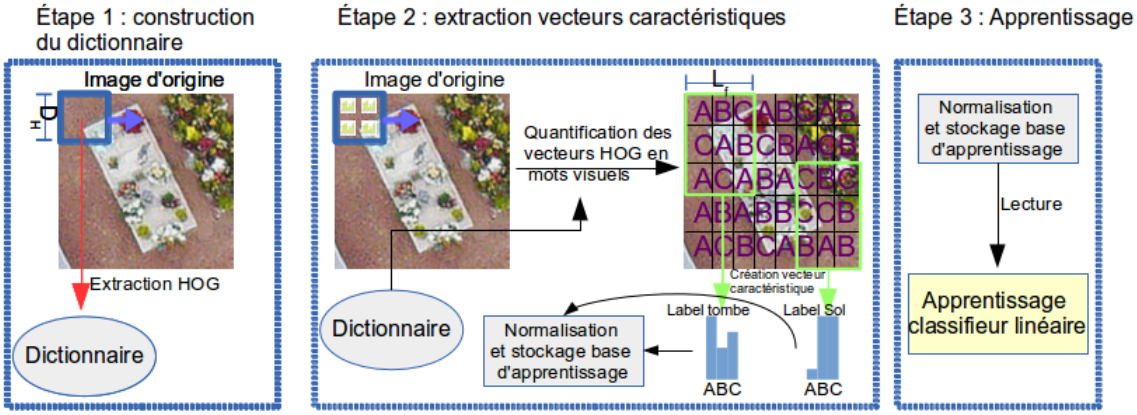

Figure 2: *Récapitulatif du processus d'apprentissage afin d'effectuer une segmentation par pixel*

résulte une carte de probabilité, il faudra effectuer un post-traitement afin de détecter et localiser les objets.

3. Segmentation objet

Dans l'approche objet ou modèle, le modèle de l'objet est construit. Contrairement à l'approche pixel, où chaque pixel de l'objet est caractérisé par un vecteur, nous allons former un seul vecteur caractéristique pour chaque fenêtre englobant un objet recherché. Notre base d'apprentissage sera constituée d'imagettes des objets recherchés ainsi que d'imagettes n'étant pas pas des objets recherchés. Dans la section 3.1, nous présentons le protocole classique d'extraction de vecteurs caractéristiques par objet. Dans la section 3.2, nous proposons une amélioration de cette méthode en y rajoutant une approche par sac de mots.

3.1. Extraction des caractéristiques

Sur une image, un objet peut avoir des tailles différentes de part sa nature où à cause d'effets de zoom. Afin d'être robuste à ce changement d'échelle, les objets sont redimensionnés à une taille constante. Ensuite, comme pour l'approche pixel, il est nécessaire d'extraire des caractéristiques comme les histogrammes de gradients orientés pour décrire nos objets. Afin d'obtenir une description plus fine de chaque objet, nous utiliserons la méthode de Qiang Zhu et coll. [ZYCA06] pour extraire de multiples HOG. Dans leur papier [ZYCA06], Qiang Zhu et coll. extraient des HOG de tailles différentes d_i avec $i \in 0..T$ et T le nombre de tailles possibles dans toutes les positions de la fenêtre. Une fois extraits tous les HOG sont concaténés en un vecteur de très grande dimension. La figure 3 résume ce processus. Afin de limiter le temps nécessaire lors de la phase de test, Qiang Zhu utilise un système de cascades de classifieurs [VJ01]. Ce processus d'extraction de caractéristiques est identique à celui mis en place par Viola et Jones [VJ01] avec un descripteur plus HOG robuste. Ceci créé un modèle multi-fenêtre robuste de nos objets.

Une fois extraits, les vecteurs caractéristiques ne sont pas quantifiés en mots visuels et servent à l'apprentissage d'un classifieur linéaire.

3.2. Extension de la segmentation objet

Nous proposons une extension du modèle de Qiang Zhu et coll. [ZYCA06] en y rajoutant la statistique et distribution de mots visuels. Notre idée est de décrire l'objet à l'aide de mots visuels de différentes résolutions. Pour chaque taille d_i de vecteur HOG extrait nous construisons en amont un dictionnaire de N mots visuels. Lors de la phase d'extraction de vecteurs caractéristiques, nous remplaçons chaque vecteur HOG par sa quantification en mots visuels. Pour chacun des T dictionnaires (pour chaque résolution) nous construisons un histogramme des fréquences d'apparition des mots visuels. Tous les histogrammes sont ensuite concaténés pour former le vecteur caractéristique de dimension $N * T$ (voir schéma 8).

Le fait d'utiliser comme vecteur caractéristique une statistique, provoque une perte de l'information sur la localité des vecteurs HOG. Plusieurs solutions existent pour lutter contre ce phénomène []. Dans notre cas, nous avons choisi de représenter les mots visuels dans une pyramide [LSP06] ce qui agrandit considérablement la dimension des vecteurs caractéristiques mais permet de garder partiellement la localité des vecteurs (voir schéma 5).

4. Implémentation et résultats

Dans cette section, nous allons décrire la base de données que nous avons utilisée ainsi que la façon dont nous avons évalué les différentes approches. Ensuite, nous donnerons et discuterons de nos résultats.

4.1. La base de données et protocole d'évaluation

Dans notre cas d'étude, nous allons chercher à détecter les tombes dans des images de cimetière de dimension 4000^2 pixels. Un exemple de la base de donnée est montré en 6. La détection de tombes, comme montré dans [CTS*13], est une

© CORESA 2014.

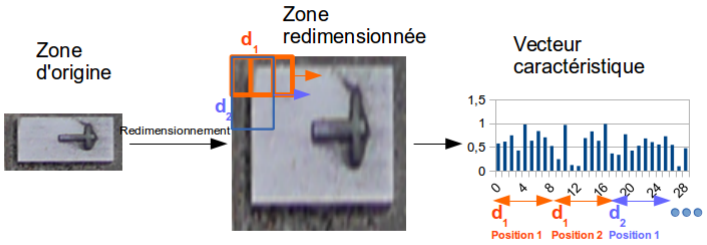

Figure 3: *Récapitulatif du processus d'extraction de multiples vecteurs HOG dans une région.*

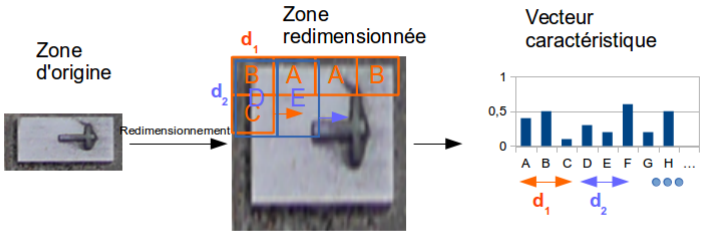

Figure 4: *Processus d'extraction de multiples vecteurs HOG et leur quantification en mots visuels par un dictionnaire dépendant de leur taille.*

Figure 6: *Image provenant de [CTS*13], montrant le cimetière de Lecey.*

tâche difficile. En effet, les tombes sont de tailles différentes (hauteur allant de 80-140 pixels et largeur de 50-60 pixels), avec des ombres portées et souvent des contours abîmés par le temps ou recouverts de terre les rendant quasiment identiques au sol. De plus, l'espace entre les tombes varie de 2 à 20 pixels nécessitant une détection précise. Pour cela, nous utilisons des méthodes d'apprentissage automatique. La société Berger-Levrault a fourni des images aériennes haute définition (2.5cm/pixel) en couleurs (rouge, vert, bleu) de 21 cimetières du département de Haute-Marne. Le nombre de tombes dans la totalité des images avoisine les 1300. Afin de ne pas biaiser nos résultats, les bases d'apprentissage et de test sont distinctes. Ainsi, 19 images serviront à l'apprentissage et 2 images (soit environ 120 tombes) serviront aux tests.

4.2. Paramètres utilisés

Dans cette section, nous décrivons les paramètres utilisés pour chacune des approches. Dans chacun des trois cas, à savoir l'approche pixel, objet et notre proposition d'amélioration, le classifieur utilisé est un SVM linéaire de *liblinear* [FCH*08].

4.2.1. Approche pixel

Dans l'approche pixel, la taille des blocs d'extraction des HOG est fixée à 32 pixels, avec pour chaque HOG une concaténation de quatre cellules. Chaque HOG est ensuite quantifié avec un dictionnaire de type forêt ERF [MNJ08] de profondeur maximale seize avec dix arbres. Le dictionnaire est donc formé de 655360 mots visuels. Le vecteur caractéristique est construit en représentant la fréquence d'apparition de ces mots visuels dans une fenêtre de dimension 40.

4.2.2. Approche objet

Chaque tombe est redimensionnée en fenêtre de dimension 70x70 pixels. Dans cette fenêtre, nous faisons varier des

© CORESA 2014.

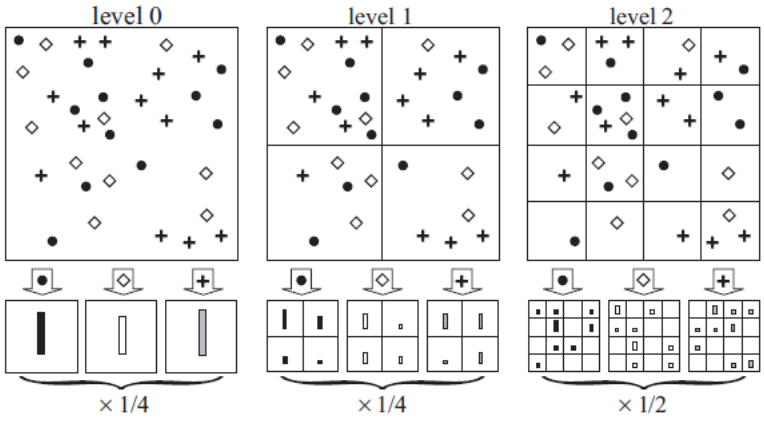

Figure 5: *Schéma provenant de [LSP06], montrant la formation d'un histogramme de mots visuels et leurs poids à partir d'une pyramide de profondeur 3.*

tailles de dimension minimale (5,5) jusqu'à (80,80) afin de tester 1260 tailles différentes. Chaque HOG constitué d'une seule cellule est extrait avec une surface de recouvrement de 50%. Au final, nous obtenons un vecteur caractéristique de dimension 430000.

4.2.3. Proposition d'extension de l'approche objet

Les paramètres utilisés sont les mêmes que ceux de l'approche par modèle de la section 4.2.2 avec un dictionnaire. Le dictionnaire choisi est un Kmeans Hierarchique [NS06]. Celui-ci permet d'obtenir un grand nombre de mots visuels avec un faible coût en complexité. Nos dictionnaires de profondeur 3, ont à chaque niveau 7 centroïdes, soit un total de 343 mots visuels par dictionnaire. La taille du vecteur caractéristique finale est donc de l'ordre de 430000.

4.3. Résultats

Nous allons comparer les différentes approches. En premier, nous comparerons l'approche pixel et l'approche objet. En second, nous comparerons l'approche objet à l'amélioration que nous proposons.

4.3.1. Comparaison approche pixel et objet

Afin de comparer l'approche pixel et l'approche objet il est nécessaire de segmenter la carte de probabilité de l'approche pixel pour obtenir des régions. Pour cela, en utilisant l'image d'origine et la transformée de Hough nous avons cherché à détecter des rectangles dans l'image. A l'aide de la carte de probabilité, nous associons à chaque rectangle un score et gardons uniquement les meilleurs protagonistes. Un exemple des résultats obtenus par les deux approches est donné dans la figure 7. Une fois la carte de probabilité segmentée, nous comparons l'approche de Qiang Zhu et coll. et l'approche pixel dans le tableau ci-dessous.

	Segmentation pixel	Segmentation objet
Précision (%)	0.23	0.3
Rappel (%)	0.41	0.41

Pour un rappel fixé, nous constatons que la segmentation objet est plus précise. Une explication possible est causée par la proximité des tombes. En effet, l'espace entre deux tombes voisines est compris entre 2 et 20 pixels. Aussi, l'approche pixel ne distingue pas les frontières entre les objets laissant cette tâche à la post segmentation. L'approche objet, de par son fonctionnement intrinsèque, connaît le modèle d'une tombe et donc les détecte directement. Dans le cas d'une détection d'objets multiples très proches, l'approche pixel n'est, à priori, pas très adaptée et demande un post traitement fort.

4.3.2. Comparaison approche multiple HOG avec et sans sac de mots

Lors d'une approche objet, le classifieur peut à chaque zone donner une probabilité d'être une tombe ou autre chose. La probabilité seuil minimal est requise pour considérer qu'une zone est une tombe. Dans la figure 8, nous faisons varier ce paramètre de seuil et traçons les courbes ROC de l'approche de Qiang Zhu et coll. (en bleu) et de notre approche décrite en 3.2 (en rouge). Pour un rappel fixé notre approche est en moyenne 9% plus précise que l'approche de Qiang Zhu et coll. Ce résultat est significatif et représente un gain relatif en précision moyen de 17%.

Sur la figure 9, on constate que la taille des dictionnaires influence directement sur les performances de notre méthode. Aussi, un faible nombre de mots visuels provoque une chûte de performance. Ce résultat est classique, en effet, une faible taille de dictionnaire ne permet pas de décrire correctement un système. Cependant dans le cas d'un dictionnaire trop grand, les performances diminuent également. Une explication possible serait que le vecteur caractéristique devienne trop grand et que l'information ne serait pas assez dense. Cependant mieux étudier ce phénomène semble être intéressant pour l'avenir.

5. Conclusion

Dans cet article, nous avons comparé l'approche pixel et l'approche modèle. Dans notre cas d'étude, il résulte

© CORESA 2014.

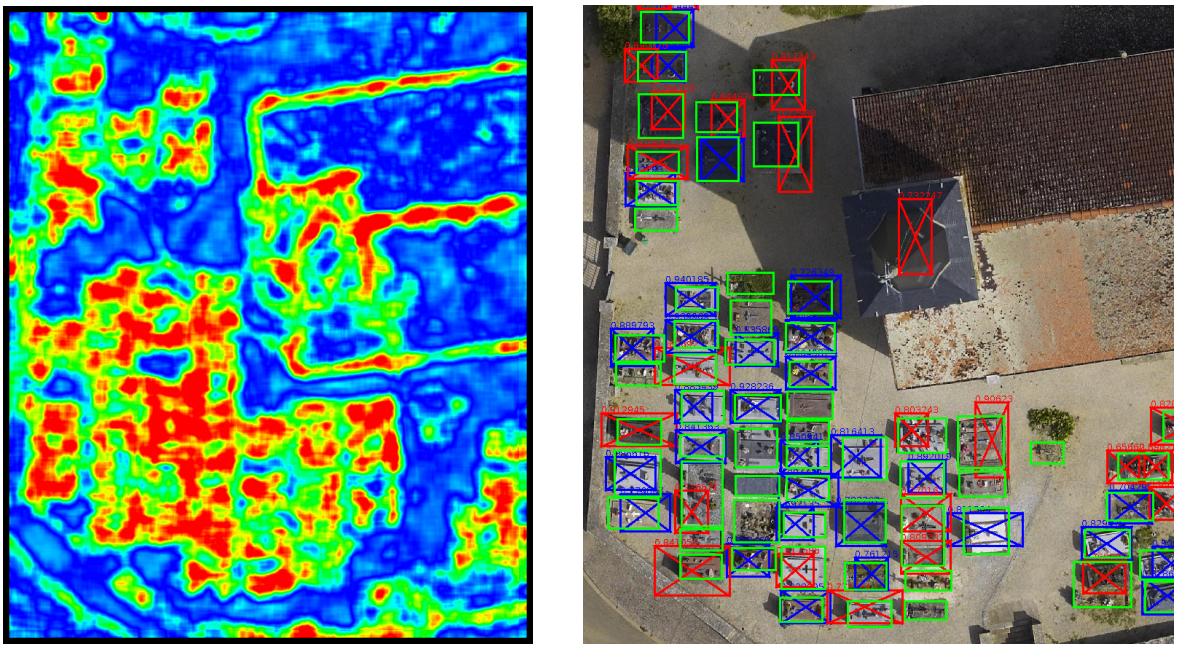

Figure 7: *Les deux figures illustrent l'approche pixel à gauche et l'approche objet à droite. L'approche pixel génère une carte de probabilité où les pixels rouges sont ceux avec une forte probabilité d'être une tombe et inversement les pixels bleus ont une forte probabilité de ne pas être une tombe. L'approche objet, génère directement des régions, les rectangles bleus sont les tombes détectées, les rectangles rouges sont les faux positifs et en vert nous affichons la vérité terrain.*

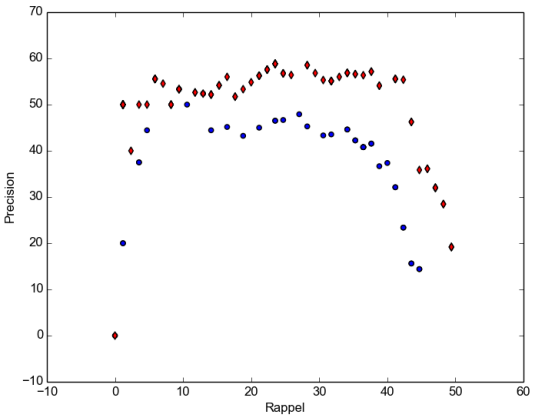

Figure 8: *La courbe ROC de l'approche décrite en 3 est représentée par les points ronds bleus, la courbe ROC de notre approche décrite en 3.2 est représentée par les diamants en rouges.*

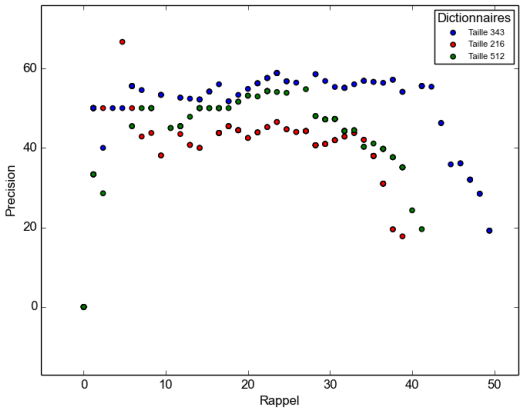

Figure 9: *Représente les courbes ROC de notre proposition avec des dictionnaires de type Kmeans Hiérarchique de différentes tailles.*

que pour le cas d'objets très proches et multiples, l'approche pixel est moins adaptée et précise de 7 %. Nous nous sommes donc focalisés sur la segmentation par modèle et proposons une extension du modèle classique de cascades de HOG. Dans la partie 4.3.2, nous montrons que l'utilisation de mots visuels augmente de façon significative, soit un gain relatif en précision de 17%. Afin de mieux comprendre ce gain, il serait intéressant, d'étudier la transformée inverse des représentants(du dictionnaire) des vecteurs HOG en image [VKMT13]. Afin d'améliorer notre extension du modèle de cascades de HOG, plusieurs pistes sont possibles. La première consiste à changer la méthode de construction du dictionnaire ainsi que sa taille ; laquelle influence grandement, comme nous l'avons montré dans la partie 4.3.2. De plus, la taille du dictionnaire peut varier en fonction de la taille des fenêtres glissantes.

© CORESA 2014.

6. Remerciements

Nous souhaitons remercier tout particulièrement la société Berger Levrault pour son soutient dans nos travaux. De plus, nous remercions également Laurent Deruelle, Francois Bibonne et Pol Kennel pour leurs conseils.

Références

[ARTLdM10] ALDAVERT D., RAMISA A., TOLEDO R., LÓPEZ DE MÁNTARAS R. : Fast and Robust Object Segmentation with the Integral Linear Classifier. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 1046–1053.

[CTS*13] CHAUMONT M., TRIBOUILLARD L., SUBSOL G., COURTADE F., PASQUET J., DERRAS M. : 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.

[FCH*08] FAN R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R., LIN C.-J. : LIBLINEAR : A library for large linear classification. *Journal of Machine Learning Research. Vol. 9* (2008), 1871–1874.

[FVS08] FULKERSON B., VEDALDI A., SOATTO S. : Localizing objects with smart dictionaries. In *Proceedings of the 10th European Conference on Computer Vision : Part I* (Berlin, Heidelberg, 2008), ECCV '08, Springer-Verlag, pp. 179–192.

[LBH08] LAMPERT C. H., BLASCHKO M., HOFMANN T. : Beyond sliding windows : Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–8.

[LSP06] LAZEBNIK S., SCHMID C., PONCE J. : Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, pp. 2169–2178.

[MM09] MEENA MAHAJAN PRAJAKTA NIMBHORKAR K. V. : The planar k-means problem is np-hard. *Lecture Notes in Computer Science* (2009).

[MNJ08] MOOSMANN F., NOWAK E., JURIE F. : Randomized clustering forests for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.. Vol. 30*, Num. 9 (septembre 2008), 1632–1646.

[NS06] NISTER D., STEWENIUS H. : Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (Washington, DC, USA, 2006), CVPR '06, IEEE Computer Society, pp. 2161–2168.

[RBK95] ROWLEY H. A., BALUJA S., KANADE T. : Human face detection in visual scenes. In *NIPS* (1995), Touretzky D. S., Mozer M., Hasselmo M. E., (Eds.), MIT Press, pp. 875–881.

[RnAK*11] RUSIÑOL M., ALDAVERT D., KARATZAS D., TOLEDO R., LLADÓS J. : Interactive Trademark Image Retrieval by Fusing Semantic and Visual Content. In *Advances in Information Retrieval*, Clough P., Foley C., Gurrin C., Jones G., Kraaij W., Lee H., Mudoch V., (Eds.), vol. 6611 de *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 314–325.

[VJ01] VIOLA P., JONES M. : 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.

[VKMT13] VONDRICK C., KHOSLA A., MALISIEWICZ T., TORRALBA A. : HOGgles : Visualizing Object Detection Features. *ICCV* (2013).

[ZYCA06] ZHU Q., YEH M.-C., CHENG K.-T., AVIDAN S. : Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (Washington, DC, USA, 2006), CVPR '06, IEEE Computer Society, pp. 1491–1498.

© CORESA 2014.