



**HAL**  
open science

# Un générateur de boîtes englobantes parcimonieux pour la détection d'objets dans des vidéos

Adrien Chan-Hon-Tong, Stephane Herbin, Alexandre Boulch

## ► To cite this version:

Adrien Chan-Hon-Tong, Stephane Herbin, Alexandre Boulch. Un générateur de boîtes englobantes parcimonieux pour la détection d'objets dans des vidéos. XXV colloque Gretsi 2015, Sep 2015, Lyon, France. hal-01175556

**HAL Id: hal-01175556**

**<https://hal.science/hal-01175556v1>**

Submitted on 10 Jul 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un générateur de boîtes englobantes parcimonieux pour la détection d'objets dans des vidéos

Adrien CHAN-HON-TONG, Stephane HERBIN, Alexandre BOULCH

ONERA - The French Aerospace Lab  
F-91761 Palaiseau, France  
prenom.nom@onera.fr

**Résumé** – La capacité à détecter automatiquement des objets dans des flux vidéos en respectant des contraintes de temps de calcul semble encore aujourd'hui hors d'atteinte. Cependant, beaucoup d'applications nécessitent seulement de détecter chaque objet dans au moins une image, et non pas, de détecter chaque objet dans chaque image. En tenant compte de cette spécificité, nous proposons un algorithme de proposition de boîtes englobantes dont l'objectif est de recouvrir au moins une fois chaque objet en ne proposant qu'un très faible nombre de boîtes par image. Notre algorithme obtient des résultats intéressants sur le jeu de données *VIRAT aerial* tout en respectant largement les contraintes de temps réel.

**Abstract** – The ability to automatically detect object in video stream while dealing with computation time constraint is still not reached by the state of the art algorithms. However, some applications need only the detection of each object at least one time, but not necessarily, the detection of each object in each frame. This context allows to soften algorithmic constraints. Particularly, in this paper, we offer a bounding box proposal algorithm designed to recover each target at least one time in the video with a very small set of boxes per frame. This algorithm achieves interesting result on the *VIRAT aerial* dataset while handling more than 25 frames per second.

## 1 Introduction

Les performances des algorithmes de classification et de détection d'objets dans des images ont récemment connu une très forte progression sur ILSVRC et Pascal challenge, notamment depuis les publications [6, 3] représentatives d'apprentissage profond.

Cependant, la classification [6] (au coeur de [3]) est trop lent pour fonctionner avec le paradigme de la fenêtre glissante en détection d'objets (même pour une taille d'objet donnée) puisque dans ce paradigme, l'algorithme devrait classer plus de 100000 boîtes par image alors qu'il n'est capable que d'en traiter 800 par seconde (même à l'aide d'un matériel spécifique [5]). Ainsi, [3] n'aurait pas été possible sans l'utilisation du paradigme de la proposition de boîtes ([12] dans [3]) qui consiste à proposer des boîtes englobantes là où l'on pense trouver un objet d'intérêt avant même de chercher à classer la boîte. Mais, les algorithmes de proposition de boîtes sont usuellement réglés pour produire 2000 boîtes par image [4], ordre de grandeur est incompatible avec le temps réel.

La seule alternative générique est d'utiliser des algorithmes de détection rapide [1, 10] moins performants que [3].

Cependant, des solutions rapides et performantes sont possibles pour certaines applications. Par exemple, dans de nombreuses applications, les objets ne peuvent apparaître que par des zones d'entrée sortie connues, ce qui permet d'y focaliser l'utilisation des algorithmes de détection. Dans d'autres applications, la caméra est immobile, ce qui impose que les objets soient mobiles (au moins pour entrer dans le champs de vue de

la caméra) ce qui peut permettre de les détecter par soustraction de fond.

Dans cet article, nous proposons une alternative qui ne fait aucune hypothèse sur le contenu de la vidéo (pas de zones d'entrée sortie, caméra mobile, objets immobiles) mais qui fait une hypothèse sur *l'objectif du traitement* : l'objectif du traitement n'est plus de détecter chaque objet dans chaque image mais de détecter chaque objet au moins une fois dans la vidéo. Un exemple de telle application est la recherche de blessés par un drone/avion : l'objectif est de localiser des personnes au cours d'un vol au dessus d'une zone comme une forêt. La caméra est alors mobile et les personnes immobiles (puisque blessés), et pour tenir compte des occultations (ici, dues aux arbres), il est nécessaire de supposer que la personne peut apparaître n'importe où dans l'image. Enfin, dans cette application, il est évidemment suffisant de détecter chaque personne une fois.

Après avoir positionné le problème vis à vis de la littérature existante, un algorithme tirant parti des spécificités de cet objectif est présenté. Cet algorithme est ensuite évalué sur le jeu de données *VIRAT aerial* [8].

## 2 Positionnement du problème

**Vérité terrain** : La spécificité du problème présenté nécessite une métrique spécifique entre celles du suivi et celles des algorithmes de proposition de boîtes. Ici, la vérité terrain est composé d'un ensemble de pistes comme en suivi d'objets. Mais, l'objectif n'est pas de produire des pistes proches

de celles de la vérité terrain. L'objectif est de produire un ensemble de boîtes ( $M$  par image) de sorte que chaque objet (au sens de la vérité terrain) soit recouvert par une des boîtes proposées, au moins une fois dans la vidéo.

On note  $b^*(o, t)$  la boîte englobante de l'objet  $o$  dans l'image  $t$ . Alors, comme en proposition de boîtes, une boîte  $b$  recouvre un objet  $o$  dans une image  $t$  si le ratio de Jacard IoU (intersection sur union) entre  $b$  et  $b^*(o, t)$  est supérieur à un seuil  $\alpha$  (si l'objet n'est pas présent le ratio sera  $-\infty$  par convention).

**Métrique :** Formellement, on note  $T$  la longueur de la vidéo,  $O$  l'ensemble des objets et  $B$  l'ensemble des boîtes proposées avec  $B_t$  celles extraites dans l'image  $t$  (donc  $|B_t| \leq M$  avec  $|\cdot|$  le cardinal). Le recouvrement des objets  $O$  par  $B$  est :

$$|\{o \in O / \exists t \in [1, T], b \in B_t / \text{IoU}(b, b^*(o, t)) \geq \alpha\}| / |O|$$

(On omet le dénominateur  $|O|$  dans la suite).

Au vu de cette métrique, l'application d'un algorithme de proposition de boîtes sans mémoire (utilisant chaque image indépendamment) avec  $M$  faible conduit *volontairement* à un faible recouvrement. Il est désiré que, d'une image à l'autre dans une vidéo, un tel algorithme produise des boîtes proches, puisque les images sont proches. Ainsi, l'algorithme sans mémoire devrait *volontairement* recouvrir densément les objets les plus saillants alors que l'objectif est ici de recouvrir tous les objets même de façon éparsée.

C'est pourquoi, dans la section suivante, nous proposons un algorithme utilisant à la fois des techniques de proposition de boîtes et des techniques de suivi pour obtenir un fort recouvrement.

**Contexte académique :** La littérature qui se rapproche de cet article est donc principalement la littérature sur la proposition de boîtes [2, 12, 13] (comparée dans [4]).

En terme d'objectif, cet article est proche des détecteurs rapides [1, 10] et de la littérature de détection d'objets mobiles dans des vidéos aériennes (dont [7] est un exemple représentatif).

En terme de coeur algorithme, cet article pourrait sembler proche de [9, 11]. Mais, des différences importantes avec ces travaux seront soulignées dans la section suivante.

## 3 Boîtes englobantes éparsées

### 3.1 Le suivi et le suivi faible

L'idée clé de notre algorithme est de proposer des boîtes mais de se souvenir des boîtes précédemment proposées pour ne pas proposer des boîtes qui correspondraient à des objets déjà proposées. La seconde idée principale est d'utiliser des méthodes de suivi pour, étant donnée une nouvelle boîte, décider si elle semble correspondre à un objet déjà recouvert.

Il peut cependant sembler contradictoire d'utiliser des outils de suivi pour proposer des boîtes alors que le problème du suivi d'objets est (sans doute) plus difficile que celui de la proposition de boîtes. Mais, le point important est que l'algorithme n'a besoin que d'un *suivi faible*. Par exemple, c'est un suivi dans

lequel, il n'y a que deux identités *connu* et *inconnu*. Ainsi, il n'y a aucun problème pour notre algorithme à laisser deux objets (au sens de la vérité terrain) échanger leur pistes (c'est à dire leur identité dans un suivi classique) puisque ici l'identité correspond à être pisté. De même, il n'est pas grave que les trajectoires soient très segmentées (plusieurs identifiants par objet réel) du moment que chaque objet réel est à un moment recouvert par une boîte.

C'est la différence majeure entre cet article et [9, 11]. L'utilisation de techniques de suivi et de proposition de boîtes pour extraire des volumes spatio temporel saillants dans des vidéos est commune. Mais, dans cet article, l'aspect temporel ne sert qu'à réduire le nombre de boîtes proposées alors qu'il doit porter une information dans [9, 11] (typiquement si l'objectif est d'utiliser le volume spatio temporel pour déterminer l'action de la personne observée). Ainsi, ces deux articles ont besoin d'un suivi classique qui est un problème difficile, là où notre algorithme n'a besoin que d'un suivi faible.

### 3.2 De la métrique à l'algorithme

Nous montrons dans cette sous section que les idées clés de l'algorithme proposé sont naturelles étant donnée la métrique utilisée. Soit  $S(B_{[1, \tau]}, O)$  le recouvrement (abrégé en  $S_\tau$ ) de  $O$  par  $B_{[1, \tau]}$  (boîtes extraites entre 1 et  $\tau$ ). Par définition, on a :

$$S_\tau = |\{o \in O / \exists t \in [1, \tau], b \in B_t / \text{IoU}(b, b^*(o, t)) \geq \alpha\}|$$

Ce recouvrement peut se décomposer entre les choix passés et futurs, ce qui est pertinent pour un algorithme *en ligne* :

$$S_{\tau+1} = S_\tau + |\{o \in O / S(B_{[\tau, \tau+1]}, \{o\}) = 1 \wedge S(B_{[1, \tau]}, \{o\}) = 0\}|$$

Si  $B_{[\tau, \tau+1]}$  est le singleton  $\{b_c\}$ , ce dernier terme devient

$$|\{o \in O / \text{IoU}(b_c, b^*(o, \tau+1)) \geq \alpha \wedge S(B_{[1, \tau]}, \{o\}) = 0\}|$$

Lors de l'utilisation de l'algorithme,  $O$  est inconnu mais il est possible d'estimer certaines probabilités sur  $O$ . Cela amène à considérer l'équation précédente en espérance :

$$E[S(B_{[1, \tau+1]}, O)] = E[S(B_{[1, \tau]}, O)] + P_{\tau+1}$$

où  $P_\tau$  est :

$$P(\exists o \in O / \text{IoU}(b_c, b^*(o, \tau)) \geq \alpha \wedge S(B_{[1, \tau-1]}, \{o\}) = 0)$$

Ainsi, la boîte  $b_c$  de l'image  $\tau$  qui maximise l'espérance de recouvrement compte tenu de la probabilité estimée  $P$  est celle qui maximise  $P_\tau$ . Or  $P_\tau$  est décomposable ainsi :

$$P(\exists o \in O / \text{IoU}(b_c, b^*(o, \tau)) \geq \alpha) \times P(S(B_{[1, \tau-1]}, \{o\}) = 0 | \exists o \in O / \text{IoU}(b_c, b^*(o, \tau)) \geq \alpha)$$

Cette dernière décomposition est intéressante car chacun des deux facteurs peut être rapidement (bien que grossièrement) estimés :  $P(\exists o \in O / \text{IoU}(b_c, b^*(o, \tau)) \geq \alpha)$  la probabilité de contenir un objet peut être grossièrement estimée par des techniques de proposition de boîtes classiques (notons  $P_{\text{objet}}$  l'estimation), et, la probabilité

$$P(S(B_{[1, \tau-1]}, \{o\}) = 0 | \exists o \in O / \text{IoU}(b_c, b^*(o, \tau)) \geq \alpha)$$

de ne pas avoir déjà recouvert l'objet peut être grossièrement estimée par des techniques de suivi (notons  $P_{\text{inconnu}}$  l'estimation). L'algorithme proposé consiste à utiliser cette dernière équation : à chaque image,  $P_{\text{objet}}$  et  $P_{\text{inconnu}}$  sont calculées, le gain maximal de recouvrement (en espérance) apportée par une nouvelle boîte est alors  $\arg \max_{b_c} (P_{\text{objet}}(b_c) \times P_{\text{inconnu}}(b_c))$

Pour extraire non pas 1 mais  $M$  boîtes, on propose une méthode gloutonne consistant à répéter  $M$  fois : extraction de la boîte maximisant le gain de recouvrement et actualisation de  $P_{\text{inconnu}}$ .

### 3.3 Détails d'implémentation

L'algorithme proposé n'est utile que s'il est suffisamment rapide. Aussi,  $P_{\text{objet}}$  et  $P_{\text{inconnu}}$  doivent être calculées rapidement même si cela impose des estimations grossières.

[2] semble être le seul algorithme de proposition de boîtes de l'état de l'art pertinent ([4]) pour estimer  $P_{\text{objet}}$  dans ce contexte. Cette méthode consiste à utiliser un algorithme de classification rapide avec une fenêtre glissante pour donner aux boîtes un score de présence d'un objet. Les boîtes sont d'abord transformées en un patch  $8 \times 8$  encodant une information de gradient puis classées par un SVM. Cependant, sur notre architecture (incompatible SSE), nous n'avons pas réussi à obtenir une vitesse suffisante à partir du code [2] (5FPS observée contre 300FPS attendue). Aussi, nous avons utilisé une version plus simple (mais suffisamment rapide) consistant simplement à appliquer une convolution à l'image de gradient pour obtenir une carte de score de présence d'objet (le masque de la convolution correspond à une ellipse lissée). Bien entendu, cet algorithme est moins performant que BING sur des jeux de données image comme le Challenge Pascal mais cet écart de performance semble inexistant dans notre contexte de vidéos faibles résolutions (voir table 1).

Le même problème se pose pour l'algorithme de suivi qui doit gérer typiquement 300 cibles par images, ce qui amène à utiliser un simple appariement local.

Enfin, toujours pour répondre aux problématiques de vitesse, l'estimation  $P_{\text{inconnu}}$  est binaire : 0 pour les boîtes chevauchant une piste existante, 1 pour les autres boîtes. Avec cette estimation pour  $P_{\text{inconnu}}$ , il est suffisant de savoir trier les boîtes vis à vis de  $P_{\text{objet}}$  mais la valeur intrinsèque n'est plus utile. Les boîtes sont donc simplement triées par leur score de convolution.

## 4 Évaluation

### 4.1 Jeu de données

L'évaluation de l'algorithme repose sur la détection de personnes dans le jeu de données *VIRAT aerial* [8].

Ce jeu de données est difficile car il contient des vidéos de faibles résolutions acquises avec une caméra mobile contenant des zones très texturées (char, voiture, bâtiments), des change-



FIGURE 1 – Exemple de vidéos annotées de VIRAT.

ments de plan, des changements en zoom, et même des changements de modalités (infrarouge/couleur).

Comme, il n'y a pas d'annotations associées à ce jeu de données, nous en avons annoté un sous-ensemble. Nous avons annoté une trentaine de vidéos couleur de 400 images recouvrant la diversité du jeu de données (figure 4.1). L'annotation est typique d'un scénario de suivi, et consiste d'une part à détecter les personnes et d'autre part de leur associer un label temporellement consistant (mais qui change quand la personne est temporairement occultée à l'inverse de la réidentification). Étant donné la faible résolution de la vidéo, l'annotation manuelle n'est elle-même pas évidente. Il est difficile, d'une part, de détourner une personne en l'absence de mouvement, et d'autre part, d'associer un label consistant quand de nombreuses personnes similaires se croisent. Deux annotations ont été effectuées pour former une annotation moyennée. Par contre, la taille des personnes est suffisamment constante dans chaque vidéo pour être considérée comme une méta donnée de la vidéo.

### 4.2 Proposition de boîtes classiques

Il n'existe pas (à notre connaissance) d'algorithme comparable à celui présenté vis à vis de l'objectif. Aussi, nous proposons de comparer notre algorithme avec des algorithmes de proposition de boîtes n'exploitant pas ou naïvement les spécificités du problème traité par cet article.

**proposition de boîtes sans mémoire :** Dans chaque image, les  $M$  boîtes non chevauchantes pour  $P_{\text{objet}}$  sont extraites (voir section 3.3 pour le calcul de  $P_{\text{objet}}$ ). Chaque image est traitée indépendamment de la précédente.

**proposition de boîtes parfaite mais sans mémoire :** Dans chaque image, les  $M$  premières boîtes de la vérité terrain sont sélectionnées. Ainsi, s'il y a moins de  $M$  personnes, le recouvrement est de 100% mais en présence  $K > M$  personnes, le recouvrement peut être aussi bas que  $\frac{K}{M}$ .

**proposition de boîtes aléatoire :**  $M$  boîtes sont proposées uniformément dans chaque image.

**proposition de boîtes aléatoire et suivi :**  $M$  boîtes sont proposées par tirage uniforme avec rejet des boîtes suivies.

**non chevauchement spatio temporel :** Dans chaque image,

méthode	recouvrement*	fréquence**
parfait sans mémoire	81%	528
sans mémoire	34%	132
uniforme	27%	528
uniforme suivi	39%	280
non chevauchement	75%	91
notre méthode	87%	68

\* : recouvrement défini en section 2 pour  $M = 4$  boîtes par image moyenné sur les vidéos extraites de VIRAT

\*\* : nombre d’images traitées par seconde avec 1 cpu 2.8Ghz et 4Go de RAM (implémentation c++ efficace mais sans optimisation fine, ni instruction SSE) prenant en compte la lecture des images.

TABLE 1 – Résultat sur le sous ensemble de [8]

les  $M$  boîtes (non chevauchantes) les plus saillantes ne chevauchant pas celles des quelques images précédentes sont sélectionnées. Formellement, si  $b$  est un boîte extraite à l’image  $t$  alors une boîte  $b'$  de l’image  $t'$  ne peut pas être extraite si  $\text{IoU}(b', b) \geq \alpha$  et  $t' - t \leq \delta_t$ .  $\delta_t$  est fixé a posteriori pour optimiser les performances de cette méthode.

### 4.3 Résultats

Le recouvrement des différentes méthodes dépend fortement de  $M$ . Cependant,  $M$  n’est pas un paramètre, c’est un élément dimensionnant du système : typiquement étant donné une architecture matériel et un algorithme de classification,  $M$  devrait être l’inverse du produit de la cadence vidéo par le temps nécessaire pour traiter appliquer l’algorithme de classification. Par exemple, avec une carte Quadro 2000 GPU entièrement dédié à la classification ([6] par [5]), on ne peut même pas avoir plus de 3 boîtes par image. Ainsi, il est pertinent de regarder les résultats pour de faibles valeurs de  $M$ .

Typiquement, pour  $M \leq 4$ , le système proposé obtient un recouvrement nettement supérieur à celui des autres algorithmes y compris le pseudo algorithme *parfait sans mémoire* utilisant la vérité terrain (qui devient meilleur pour  $M = 5$ ). Les résultats pour  $M = 4$  sont présentés en table 1.

Les résultats montrent que la métrique proposée en section 2 nécessite d’utiliser à la fois l’image et l’enchaînement temporel : l’enchaînement temporel seul (*uniforme suivi*) ou l’utilisation (même parfaite) de l’image seule (*parfait sans mémoire*) obtiennent de mauvais résultat vis à vis de *non chevauchement*.

De plus, la méthode proposée moins naïve que *non chevauchement* obtient un recouvrement meilleur de 10%. Cela montre à la fois la non trivialité du problème traité et la pertinence de la solution proposée pour ce type d’application.

## 5 Conclusion

Cet article présente un algorithme parcimonieux de proposition de boîtes pour la détection d’objets dans des vidéos. Cet

algorithme est pertinent pour les applications sans a priori sur le contenu de l’image et où l’objectif est de détecter chaque objet au moins une fois (mais pas forcément dans chaque image). Cet algorithme est évalué sur le jeu de données *VIRAT aerial* où ses performances sont largement meilleures que celles des algorithmes de référence. Cet algorithme combiné avec des techniques d’apprentissage profond pourraient former un détecteur à la fois temps réel et très performant. Cette perspective guide naturellement nos futurs travaux.

## Références

- [1] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012.
- [2] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing : Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [4] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really ? In *BMVC*, 2014.
- [5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. *arXiv*, 2014.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [7] Tahir Nawaz, Andrea Cavallaro, and Bernhard Rinner. Trajectory clustering for motion pattern extraction in aerial videos. In *ICIP*, 2014.
- [8] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.
- [9] Dan Oneata, Jerome Revaud, Jakob Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014.
- [10] Mohammad Amin Sadeghi and David Forsyth. 30hz object detection with dpm v5. In *ECCV*, 2014.
- [11] Gilad Sharir and Tinne Tuytelaars. Video object proposals. In *CVPR Workshops*, 2012.
- [12] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.
- [13] C Lawrence Zitnick and Piotr Dollár. Edge boxes : Locating object proposals from edges. In *ECCV*, 2014.