

1 Détection de personne

Afin de mettre en place un suivi efficace des personnes, il faut commencer par être capable de détecter ces personnes. Il a donc fallu en premier lieu décider d'une méthode de détection de personnes adaptée à notre application. De nombreuses méthodes différentes existent dans ce domaine, mais l'environnement dans lequel nous effectuons cette détection fixe nos besoins:

-le suivi des personnes devant être fait en temps réel, il nous faut un algorithme suffisamment rapide pour ne pas avoir un impact trop important sur les performances du système complet;

-le fait que nous travaillons avec un caméra PTZ et que ce travail soit destiné à fonctionner dans des environnements variés élimine les méthodes de détection se basant sur un fond connu.

Mon choix s'est porté sur une détection basée sur les histogrammes de gradient orienté (HOG) pour son efficacité reconnue, sa rapidité et le fait que cet algorithme est suffisamment populaire pour pouvoir trouver des implémentations simples et des classificateurs entraînés sur des jeu de données correspondants à notre cas de figure.

L'implémentation utilisée est celle présente dans OpenCV sous le nom de "HOGDescriptor" associé à un classifieur de type machines à vecteurs de support (SVM). Le descripteur utilisé est celui par défaut d'OpenCV pour les personnes.

1.1 Principe de la détection HOG

Cette méthode a été présentée par Navneet Dalal et Bill Triggs en 2005 dans le papier "Histograms of Oriented Gradients for Human Detection". Elle consiste en une série d'étapes illustrées dans la figure 1 issue d'une slide par Sminchisescu.

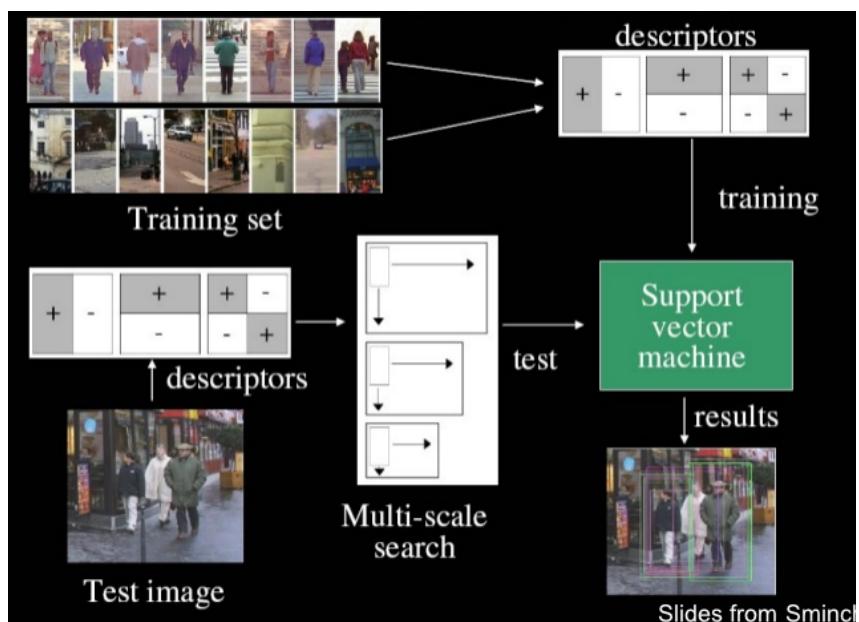


Figure 1: Illustration HOG+SVM

1.2 Descripteur HOG

Dans cette méthode, le descripteur (nommé HOG-D) est basé sur un ensemble d'histogrammes de gradient orientés. Les différentes étapes nécessaires au calcul de ces histogrammes sont décris ci-dessous:

1.2.1 Découpage de l'image

Le descripteur est calculé pour une image de 64x128 pixels, ces dimensions représentent les proportions choisi pour une personne. C'est également la résolution utilisée pour la base d'image "INRIA Person Dataset" réunie pour tester cette méthode.

Cette image est découpée en cellules de 8x8 pixels, ces cellules sont utilisées pour faire des blocs de 2x2 cellules (16x16 pixels). Les blocs se recoupent à 50%, ce qui donne un total de 7 blocs en largeur par 15 blocs en hauteur.

On a donc au final un total de 105 blocs contenant chacun 4 cellules de 8x8 pixels.

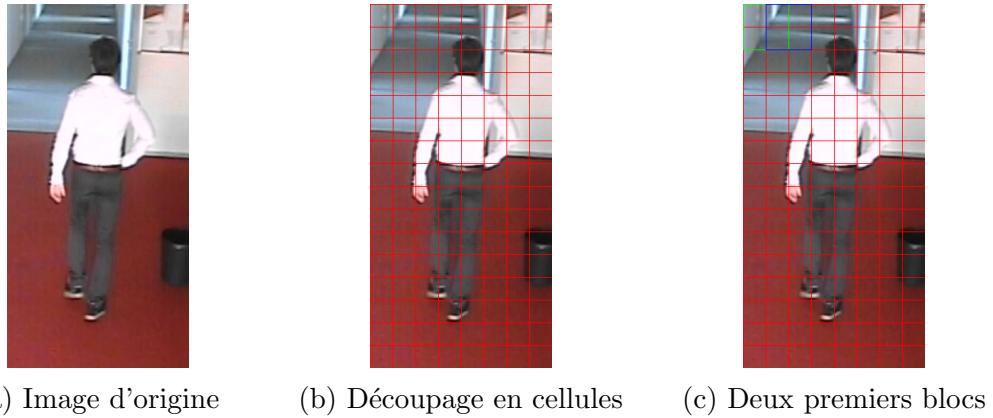


Figure 2: Découpage de l'image en blocs et en cellules

1.2.2 Calcul des gradients

Les gradients orientés sont calculés à l'aide de deux gradients pour chaque pixel: un gradient horizontal et un gradient vertical notés S_x et S_y tel que:

$$S_{x(i,j)} = I_{(i+1,j)} - I_{(i-1,j)}$$

$$S_{y(i,j)} = I_{(i,j+1)} - I_{(i,j-1)}$$

Les noyaux utilisés pour calculer ces deux valeurs sont illustrés dans la figure 3 Blabla le



Figure 3: Noyaux des gradients horizontaux (a) et verticaux (b)

gradient orienté est représenté par le vecteur: $\bar{S} = \begin{pmatrix} S_x \\ S_y \end{pmatrix}$. Les caractéristiques du gradient sont donc:

$$\bar{S} : \begin{cases} \|S\| = \sqrt{S_x^2 + S_y^2} \\ \Omega_S = \arctan\left(\frac{S_y}{S_x}\right) \end{cases}$$

1.2.3 Calcul des histogrammes

Pour chaque cellule, un histogramme est obtenue grâce à la somme pondérée des orientations des gradients. Cet histogramme est réalisé pour 9 valeurs d'orientation par pas de 20° ; les orientations vont donc de 0° à 180° . Les centres de ces intervalles sont donc

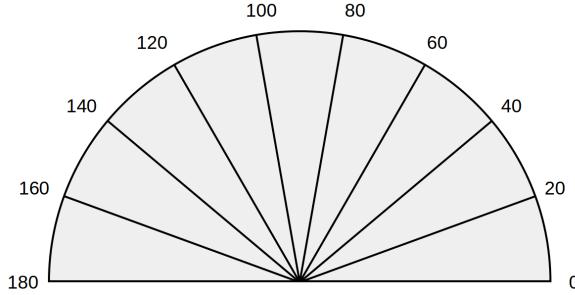


Figure 4: Découpage des orientations des gradients en 9 sections

$[10;30;50;70;90;110;130;150;170]^\circ$. Un gradient ayant pour orientation 50° verra donc son

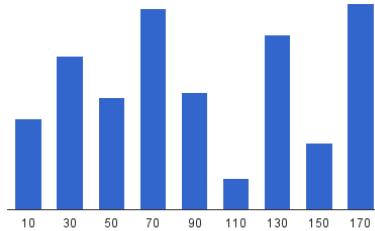


Figure 5: Histogramme obtenu avec centres des intervalles

poids (sa norme) ajouté à la barre $[40;60]^\circ$ de l'histogramme de sa cellule. Les gradients dont l'orientation se situe entre deux centres d'intervalles ont leur poids repartis entre les deux segments de l'histogramme par un principe d'interpolation trilinéaire.

Ainsi, si on prend un gradient \bar{X} , de norme $\|\bar{X}\|$ et d'orientation $\theta_X = 55^\circ$, son poids ($\|\bar{X}\|$) sera reparti entre les segments $[40;60]^\circ$ et $[60;80]^\circ$ de l'histogramme ayant pour centres 50° et 70° respectivement. Si on note les poids assignés aux deux segments P_{X50} et P_{X70} , on obtient:

$$\begin{cases} P_{50}(X) = \left(1 - \frac{|\theta_X - 50|}{20}\right) * \|\bar{X}\| = \frac{3}{4} * \|\bar{X}\| \\ P_{70}(X) = \left(1 - \frac{|\theta_X - 70|}{20}\right) * \|\bar{X}\| = \frac{1}{4} * \|\bar{X}\| \end{cases}$$

Cela permet d'éviter qu'un gradient avec une norme importante se situant à la limite entre deux sections de l'histogramme puisse faire varier le résultat de manière trop importante.

1.2.4 Concaténation et normalisation

Une fois les histogrammes des quatre cellules d'un bloc calculés, ils sont concaténés afin de former un vecteur de 36 valeurs (4×9).

Afin de diminuer l'impact des variations de contraste locales, les histogrammes des blocs sont tous normalisés en divisant leurs valeurs par la norme du vecteur obtenu. On a ainsi un descripteur invariant par rapport au contraste local (bloc de 32x32 pixels) ainsi que par rapport à la luminosité de l'image (caractéristique inhérente aux gradients). Une fois les histogrammes des blocs sont à leur tour concaténés pour obtenir l'histogramme final.

Le descripteur HOG-D nous donne donc vecteur de 3780 valeurs (105 blocs * 36 valeurs) pour représenter une image de 128x64 pixels.

1.3 Classifieur

Ce descripteur va être utilisé avec un classifieur afin d'effectuer notre détection de personne. Le classifieur utilisé ici est le même que celui utilisé par Dalal et Bill Triggs, une machine à vecteurs de support linéaire (*linear SVM*) entraînée à l'aide de *SVMLight*.

1.3.1 Classification

Le principe d'un classifieur linéaire est, à l'aide d'exemples positifs et négatifs, de déterminer un hyperplan permettant de séparer l'espace des positifs et celui des négatifs. Un exemple d'ensemble de points positifs et négatifs dans un espace en deux dimensions est présenté en figure 6. La particularité de la SVM est d'avoir pour but de trouver l'hyperplan le plus

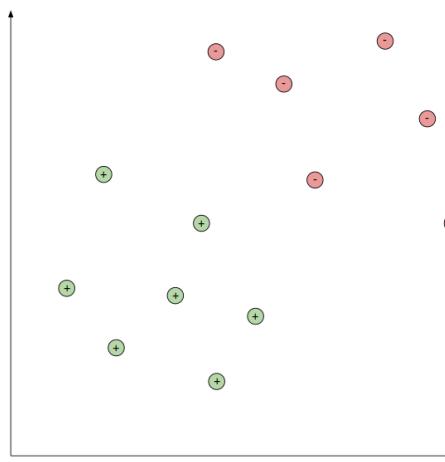


Figure 6: Exemples positifs et négatifs utilisés pour entraîner la SVM dans un espace 2D

distant des exemples négatifs et positifs. C'est le principe de la route la plus large, on cherche à trouver la route la plus large pouvant séparer les exemples, et une fois définie,

l'hyperplan choisi est au centre de la “route”. Ce principe est illustré dans les figures 7 et 8.

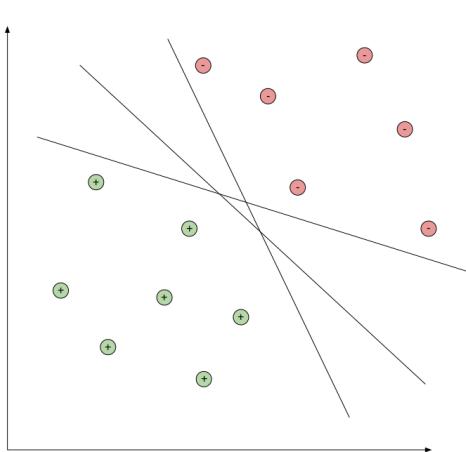


Figure 7: Différents hyperplans valides

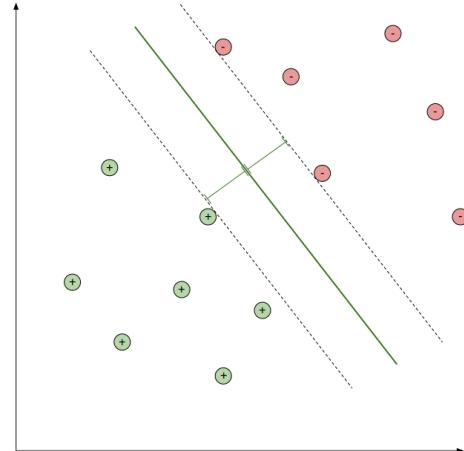


Figure 8: Hyperplan avec la marge la plus large

Une fois l'hyperplan défini lors de l'entraînement de la SVM avec des exemples positifs et négatifs, on peut s'en servir pour déterminer si un point donné dans cet espace est positif ou non selon sa position par rapport à l'hyperplan.

Pour cela, on utilise un vecteur unitaire normal à l'hyperplan sur lequel on projette le vecteur des paramètres du point expérimental, la valeur de cette projection est comparée avec celle déterminée par l'hyperplan.

Si on note la distance entre l'origine et l'hyperplan C , \bar{w} le vecteur unitaire normal à l'hyperplan, et \bar{X} le point expérimental à classifier, on a la relation suivante:

$$\alpha = \bar{X} \cdot \bar{u} \quad \begin{cases} \alpha \geq C \rightarrow \text{cas positif} \\ \alpha < C \rightarrow \text{cas négatif} \end{cases}$$

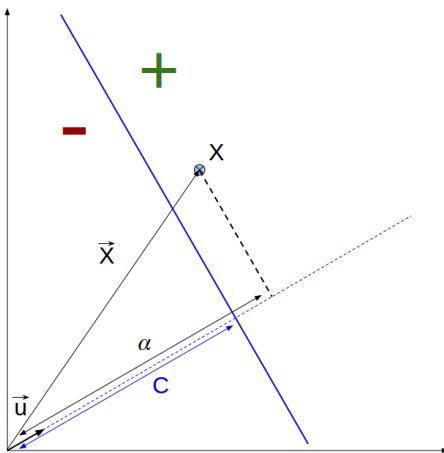


Figure 9: Utilisation de l'hyperplan pour classifier un exemple X

1.3.2 Fenêtre glissante multi-échelle et suppression des non-maxima

Une fois la SVM entraînée, on peut utiliser une fenêtre glissante le long d'une image afin de déterminer les parties de l'images classifiées comme des personnes. Pour pouvoir détecter les personnes de différentes tailles et à différentes distances on repasse la fenêtre à différentes échelles.

On a alors une liste des sous-rectangles de l'images contenant une personne, le problème étant qu'une même personne peut être détectée dans plusieurs rectangles de tailles et de positions légèrement différentes et on obtient une multitudes de rectangles positifs pour une seule personne.

Afin de résoudre ce problème, on utilise un algorithme de suppression des non-maxima sur l'ensemble des rectangles obtenus.

Le principe est de mesurer la similitude entre deux rectangle, notée ϵ , en fonction de leurs tailles et positions, au delà d'un certain seuil paramétrable, les deux rectangles similaires sont réunis.

On peut également choisir un seuil pour le nombre minimum de rectangles dans un cluster pour que la détection soit gardée, et enfin tout les rectangles étant inclus dans des rectangles plus grand avec plus d'occurrences sont également supprimés.

Le résultat cet algorithme une fois paramétré est illustré dans la figure 10.

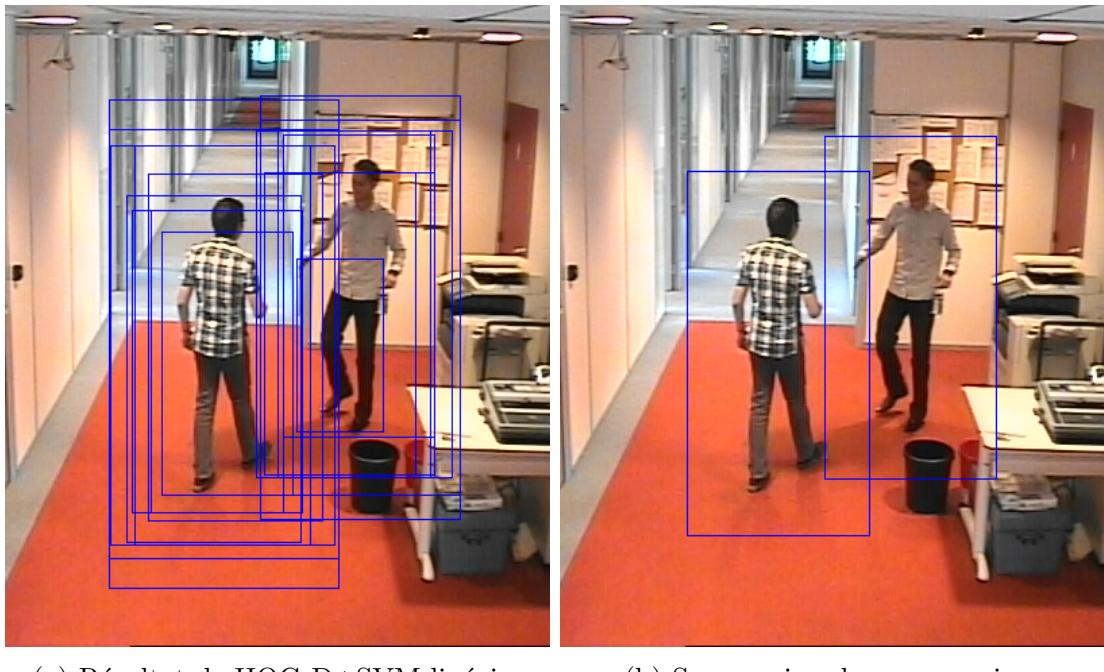


Figure 10: Algorithme de suppression des non-maxima

1.4 Paramétrage et résultats

Une fois la méthode de détection et son implémentation choisis, il reste à en régler les paramètres afin de correspondre à nos besoins.

Dans un cadre de suivi de personnes, le but est d'avoir un bon taux de détection mais

surtout de minimiser les faux positifs. En effet, ne pas détecter une personne est pénalisant, mais si on détecte un faux positif et qu'on zoom dessus, on perd énormément de temps, et ce temps perdu est plus pénalisant que de perdre une personne. Avec les contraintes physiques de la caméra et notamment les délais importants pour zoomer et dé-zoomer, la priorité est d'être certain d'avoir détecter une personne.

Notre "cahier des charges" de détection est donc le suivant:

- environnement temps réel: un minimum de 5 images par secondes;
- taux de faux positifs proche de 0;
- taux de détection le plus haut possible en respectant les deux premières contraintes.

Afin d'atteindre notre but, on peut jouer sur plusieurs paramètres dans l'algorithme de détection.

Le premier paramètre, qui influe le plus sur le temps de détection est la plus petite taille de fenêtre utilisée pour parcourir l'image et l'échelle de multiplication entre deux tailles de fenêtres.

La taille minimum de fenêtre est fixée sur une fenêtre de 8x16 pixels, l'algorithme n'étant pas fiable pour des images plus petites.

L'échelle est déterminée comme un compromis en rapidité et robustesse de détection, une échelle de 1.15 nous donne de bon résultats en détection (plusieurs fenêtres positives pour une personne), et nous permet d'obtenir une vitesse de calcul entre 60 et 70 millisecondes soit environ 15 images par seconde pour la seule détection. L'effet de l'échelle est illustrée dans la figure 11.



Figure 11: Influence de l'échelle de taille pour les fenêtres de détection

Ce temps de calcul nous laisse une marge confortable pour les différents calculs nécessaires

au suivi de personnes. La détection étant la partie nécessitant le plus de temps de calcul, on peut s'attendre un système final capable de fonctionner à plus de 10 images par secondes, ce qui est satisfaisant pour notre cas.

Le second paramètre est le seuil à partir duquel une fenêtre est considérée comme une détection positive, l'équation précédente devient en pratique $\alpha \geq C + \tau$ avec τ le seuil choisi. L'effet du seuil est illustrée dans la figure 12.

Ce seuil n'influence pas le temps de calcul de l'algorithme, c'est donc un compromis entre le taux de détection et le taux de faux positifs.

Notre priorité étant d'avoir très peu de faux positifs, le seuil a été mis à une valeur $\tau = 0.35$ permettant de supprimer pratiquement tout les faux positifs dans notre environnement de test.

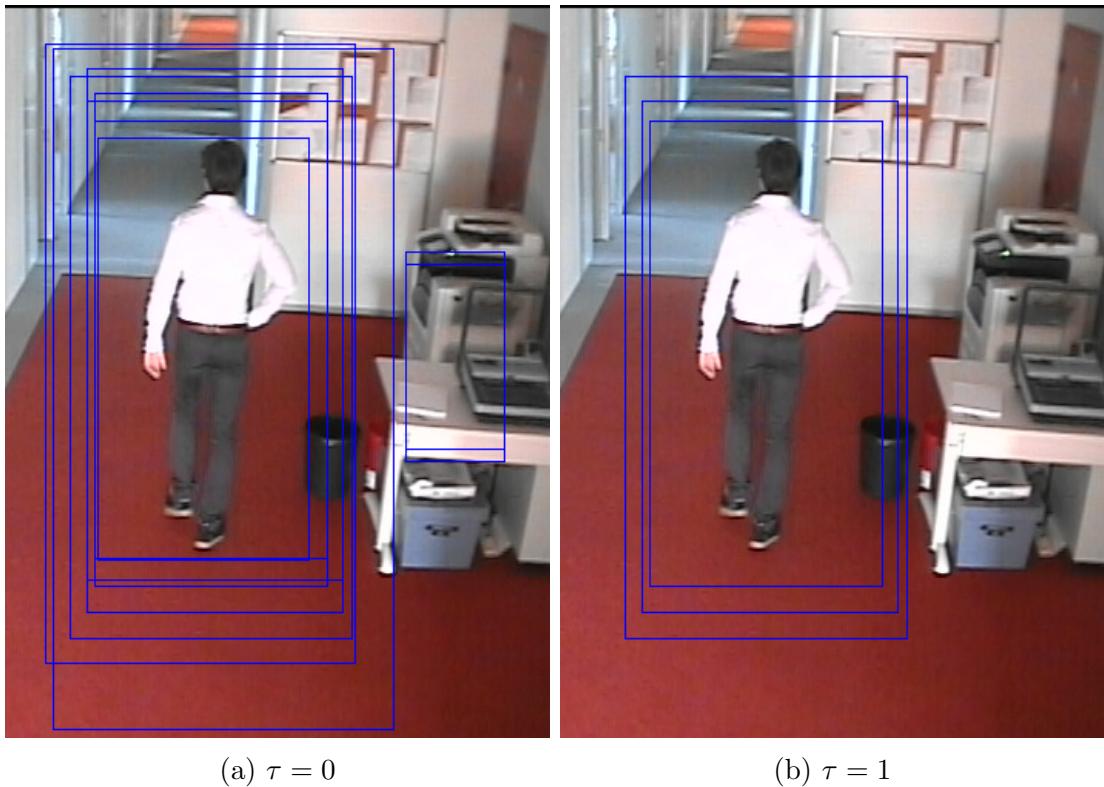


Figure 12: Influence du seuil de détection

1.5 Résultats

Après réglage des paramètres de l'algorithme, on obtient un détecteur nous permettant d'avoir un taux de faux positifs pratiquement nuls et un très bon taux de détection pour les personnes de face et de dos.

Pour des personnes représentées par plus de 50 pixels en hauteur, la détection est presque parfaite, la personne n'est que très rarement non-détectée, et ces pertes de détections seront gérées au niveau supérieur par le suivi.

Les personnes de profils ont un taux de détection légèrement plus variable, mais avoisinant les 80 à 90% lors des tests, ce qui est également suffisant pour le suivi.

Les faux positifs sont très rares, et éliminés au niveau supérieur également, toute détection intermittente étant filtrée par le suivi qui sélectionne les detections avec une certaine constance.

On a donc un système de détection robuste, ce qui nous permet d'avoir de bonnes bases pour construire notre système d'identification et de suivi des personnes.