

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 histogramme 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 classifieurs 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 ??.

1.2 Descripteur HOG

Dans cette méthode, le descripteur (nommé HOG-D) est basé sur un ensemble d'histogrammes de gradient orientés. Le calcul de ces histogrammes est décrit ci-dessous:

1.2.1 Découpage de l'image

Le descripteur est calculé pour une image de 64x128 pixels, ces dimensions représentent les proportions choisis 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 un total de 105 blocs contenant chacun 4 cellules.

1.3 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 fig ???

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

$$\|S\| = \sqrt{S_x^2 + S_y^2}$$

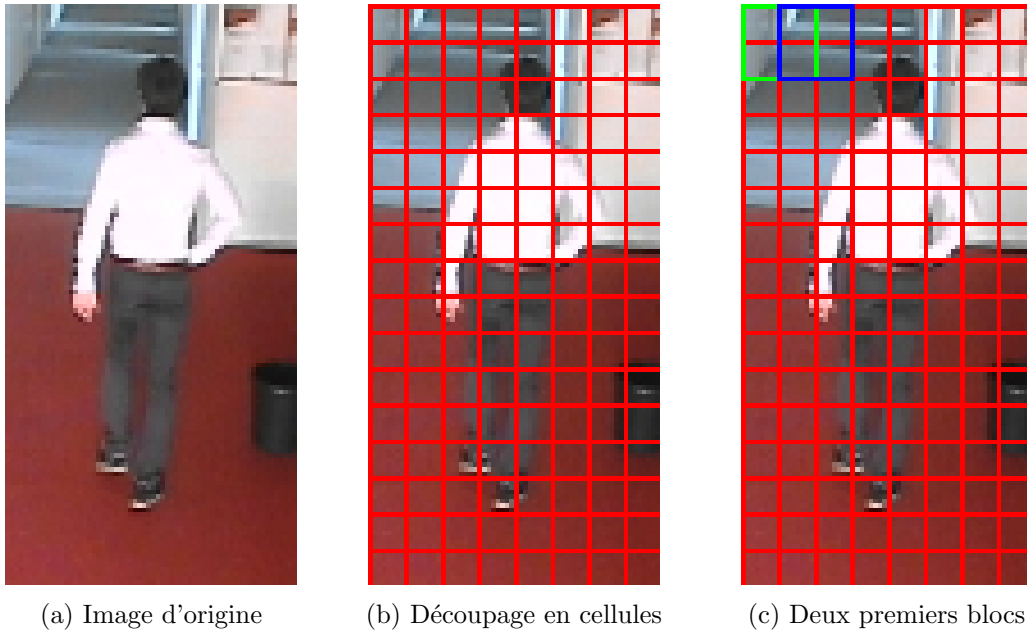


Figure 1: Découpage de l'image en blocs et en cellules

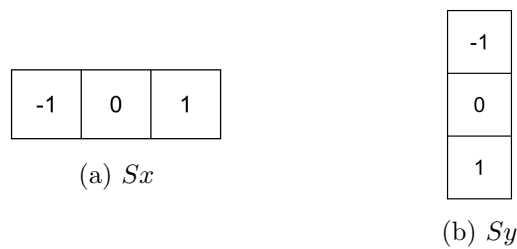


Figure 2: Noyaux des gradients horizontaux (a) et verticaux (b)

$$\Omega_S = \arctan\left(\frac{S_y}{S_x}\right)$$

1.4 Calcul des histogrammes

Pour chaque cellule, un histogramme est obtenue grâce à la somme pondérée des orientations des gradients

1.5 SVM

type de classifieur... machine learning toussa toussa

1.6 Paramètres et traitement des rectangles

-paramètres utilisés et pourquoi -gestion des rectangles multiples

1.7 Resultat obtenues

-10FPS -pratiquement aucun faux positifs (géré par le tracking sisi) -pratiquement complet niveau detection face/dos (géré aussi maggle) -lacune: detection de profil