

Master 1 Informatique

- **Rapport de projet Traitement Image**

Analyse d'une image contenant un Escalier

Bouzid Arezki
21911932

Année universitaire : 2019 – 2020

RÉSUMÉ

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles).

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du « domaine »).

L'objectif de notre travail consiste à développer un système qui puisse prédire le nombre marche d'une image contenant un escalier.

Pour cela on va utiliser la transformée de Hough pour détecter les lignes et extraire les rectangles de l'image séparément (composants connexes)

Et ensuite, on va utiliser la GFD (descripteur de Fourier générique) pour classifier les composants connexes (rectangle ou pas) les rectangles d'une marche.

Ce travail a nécessité une base de données qui contient un vecteur (ou plusieurs) de caractéristique d'un rectangle pour cela il nous faut une image (ou plusieurs) images d'un rectangle pour lui appliquer la GFD, et pour la phase de décision on a utilisé la distance minimale entre les vecteurs de caractéristiques (distance de Manhattan).

Dans ce document, nous étudions le descripteur de Fourier générique

Une seconde partie de ce document porte sur l'implémentation et l'évaluation de performances de cette solution.

TABLE DES MATIÈRES

| | |
|--|-----------|
| 1. INTRODUCTION..... | 04 |
| 1.1 INTRODUCTION..... | 04 |
| 2. TRANSFORMÉE DE HOUGH | 06 |
| 2.1 INTRODUCTION | 06 |
| 2.2 TRANSFORMÉE DE HOUGH | 06 |
| 2.2.1 APPROCHE THÉORIQUE | 06 |
| 3. LES DESCRIPTEURS DE FOURIER GÉNÉRIQUES | 08 |
| 3.1 INTRODUCTION | 08 |
| 3.2 Descripteur de Fourier générique | 08 |
| 3.2.1 <i>Descripteur de Fourier unidimensionnel</i> | 09 |
| 3.2.2 Transformation de Fourier polaire | 10 |
| 3.3 Dérivation de FD générique | 14 |
| 3.4 Mise en œuvre du GFD | 15 |
| 3.5 ALGORITHME DE GFD | 16 |
| 4. MODÉLISATION, IMPLÉMENTATION ET RÉSULTATS | 17 |
| 4.1 DEFINITIONS ET OBJECTIFS | 17 |
| 4.2 DESCRIPTION DE L'ALGORITHMES | 17 |
| 4.2.1 Le schéma globale | 18 |
| 4.2.1.1 Explication et justification plus profonde des étapes..... | 18 |
| 4.3 EVALUATION DE PERFORMANCES..... | 20 |
| 4.3.1 Table de confusion | 28 |
| 4.3.2 Environnement et modèle de simulation | 28 |
| 4.4 CONCLUSIONS | 28 |
| 5. BIBLIOGRAPHIE / LISTE DES REFERENCES | 29 |

INTRODUCTION

1. Introduction

1.1 Introduction

La forme est reconnue comme étant l'une des principales caractéristiques qui décrivent le contenu d'une image. La description et la discrimination des formes planes est l'un des problèmes de la vision par ordinateur et de la reconnaissance des formes (Dryden et al., 1998, Chen et al., 2009). Les principales qualités d'un descripteur est d'une part sa grande capacité de discrimination de sorte qu'il puisse grouper des formes similaires entre elles et séparer les formes dissimilaires dans des groupes différents (Kim et al., 2000) et d'autre part, le faible caractère redondant de ses composantes de sorte qu'un nombre minimal de celles-ci suffit à caractériser une forme efficacement. Des perturbations telles que le bruit, les distorsions et les occlusions doivent être prises en compte car ils peuvent changer la forme d'une manière complexe et subtile (Chen et al., 2008). Il est évident qu'un descripteur fiable doit être d'abord invariant par rapport aux transformations géométriques usuelles telle la translation, la rotation et le changement d'échelle car les paramètres régissant le processus de capture par caméra, tels que l'angle et la distance de prise de vue, peuvent être inconnus et peuvent aussi varier d'une manière significative. De plus, un système de vision humaine doit identifier un objet comme tel en dépit de sa position, son orientation ou sa taille. Les descripteurs de formes peuvent être scindés en deux catégories principales ; l'une est basée sur les contours et l'autre, sur les régions. Les descripteurs de la première catégorie ne peuvent pas détecter la structure interne de la forme puisqu'ils sont basés sur les contours uniquement. En outre, ces méthodes ne sont pas adaptées aux formes disjointes ou creuses, qui est le cas des symboles graphiques, car l'information

sur le contour n'est pas disponible. Par conséquent, elles sont limitées à un certain type d'applications (Tabbone et al., 2006). Une panoplie de descripteurs basés sur la région, i.e. tous les pixels contenus dans la forme, est proposée dans la littérature. Nous pouvons particulièrement citer les méthodes basées sur la théorie des moments (Belkasim et al. 1991, Prokop et al. 1992) incluant les moments géométriques de Legendre et Zernike (Mukundan et al. 1995). Une attention particulière est accordée dans le paragraphe suivant aux descripteurs de Fourier génériques qui sont proposés par (Zhang et al., 2002) car ce descripteur sera utilisé.

Le reste de ce rapport est organisé comme suit. Les chapitre 2 et 3 sont respectivement dédiées aux Transformée de Hough et la GFD. Les résultats expérimentaux seront présentés et commentés dans le chapitre 4. Enfin, la chapitre 5 sera consacrée aux conclusions

TRANSFORMÉE DE HOUGH

2. Transformée de Hough

2.1 Introduction

La transformée de Hough est une technique de reconnaissance de formes inventée en 1959 par Paul Hough, faisant l'objet d'un brevet, et utilisée dans le traitement d'images numériques.

Cette technique nous permet de détecter les lignes présentes dans une image, c'est une étape cruciale qui nous permet d'isoler les rectangles des marches en une seule composante connexe, ces étapes seront bien détaillées dans le chapitre 4.

2.2 Approche théorique

Le problème posé est celui de la recherche et de la détection de lignes qui seraient éventuellement présentes dans une image analysée malgré les imperfections de l'image : points manquants (la ligne pouvant être partiellement masquée par un objet), bruit. La transformée de Hough consiste à représenter chaque point de contour détecté dans un espace de paramètres à deux dimensions :

- Une droite est caractérisée par deux paramètres, elle est donc représentée par un point dans cet espace de paramètres ;
- Si l'on considère l'ensemble des droites passant par un point, l'image de cet ensemble est une courbe dans l'espace de paramètres ;

La transformée de Hough d'un point de l'image analysée est la courbe de l'espace des paramètres correspondant à l'ensemble des droites passant par ce point.

Si des points sont colinéaires, alors toutes les courbes de l'espace de paramètres se coupent au point représentant la droite en question.

Du fait des imperfections de l'image, les points détectés ne sont pas parfaitement alignés et donc les courbes ne sont pas parfaitement concourantes. La méthode consiste donc à discrétiser l'espace de paramètres, à le découper en petits rectangles, et à dénombrer pour

chaque rectangle le nombre de courbe y passant. On construit ainsi une matrice dite d'accumulation, les *maximas* locaux de cette matrice correspondant à des droites probables.

L'algorithme comporte donc trois étapes :

- Pour chaque point de contour détecté, détermination de la courbe correspondante dans l'espace des paramètres ;
- Construction de la matrice d'accumulation à partir de ces courbes ;
- Détection de pics dans la matrice d'accumulation

[1]

LES DESCRIPTEURS DE FOURIER GÉNÉRIQUES

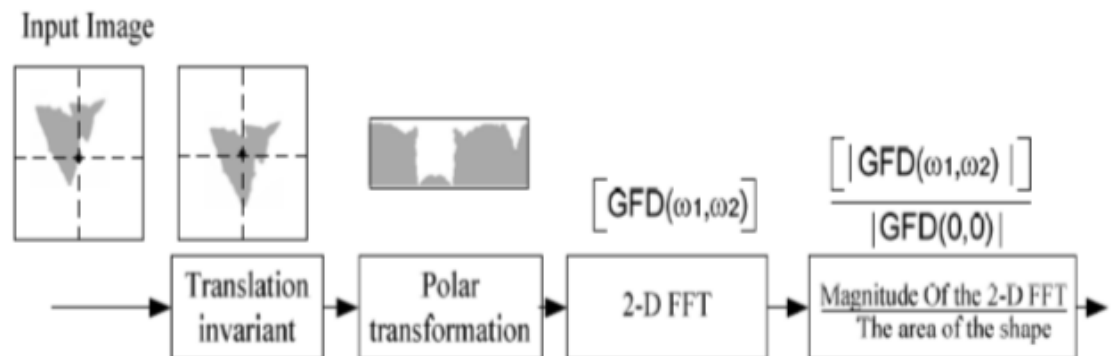
3. Les descripteurs de Fourier génériques

3.1 Introduction

Après avoir fait des prétraitements sur l'image du test et détecter ces lignes et avoir tous les composants connexes de cette image, on applique à chaque composant connexe la GFD Descripteur de Fourier générique.

3.2 Descripteur de Fourier générique

Avant de décrire Descripteur de Fourier générique (GFD) en détail, les principales étapes de l'extraction des descripteurs génériques de Fourier sont illustrées dans la figure suivante :



3.2.1 Descripteur de Fourier unidimensionnel [4]

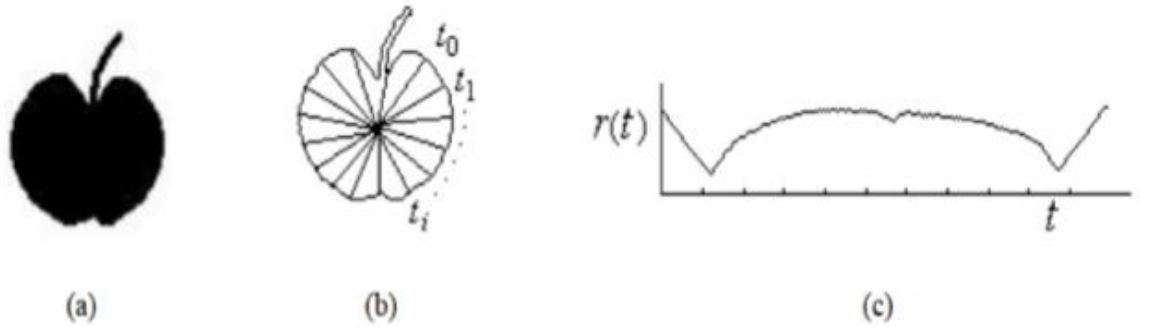
Un FD à une dimension a été appliqué avec succès à de nombreuses applications pour la représentation de forme, notamment à la reconnaissance des caractères. Il est invariant sous certaines transformations et robuste au bruit. Généralement, le Descripteur de Fourier unidimensionnel (FD 1-D) est obtenu par transformée de Fourier (FT) sur une fonction de signature de forme dérivée des coordonnées de la limite de forme $\{(x(t), y(t)), t = 0, 1, \dots, N-1\}$. Une forme typique de fonction signature est la fonction de distance centroïde qui est donnée par la distance du point limites du centre de la forme (x_c, y_c) .

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{\frac{1}{2}}, t = 0, 1, \dots, N-1$$

Où :

$$x_c = \frac{1}{N} \sum_{t=0}^{N-1} x(t) \quad y_c = \frac{1}{N} \sum_{t=0}^{N-1} y(t)$$

Un exemple de fonction de distance centroïde d'une forme de pomme est illustré dans la figure qui suit :



a) figure de pomme ; b) le contour de (a) ; c) fonction de distance centroïde de (a)

Un FT dimensionnel est ensuite appliqué sur $r(t)$ pour obtenir les coefficients transformés de Fourier.

$$a_n = \frac{1}{N} \sum_{t=0}^{N-1} r(t) \exp\left(\frac{-j2\pi nt}{N}\right), \quad n = 0, 1, \dots, N-1$$

Les grandeurs des coefficients a_n ($n = 0, 1, \dots, N-1$) normalisées par la grandeur du premier coefficient a_0 sont utilisées comme descripteurs de forme, appelés descripteurs de Fourier. Les FD acquises sont invariantes en translation, en rotation et en échelle. Il a été démontré que la représentation de forme utilisant un descripteur de Fourier (FD) prévient de

nombreuses autres formes de contour [[2], [3]]. Cependant, toutes ces méthodes supposent la connaissance des informations de limite de forme qui peuvent ne pas être disponibles dans des situations générales. Par exemple, il est difficile de dériver 1-D FD pour la forme de la figure (a) car le contour de la forme n'est pas disponible.

De plus, 1-D FD ne peut pas capturer le contenu intérieur de la forme ce qui est important pour la discrimination de forme. Par exemple, FD n'est pas capable de distinguer la forme de la figure (b) de celle de la figure (c). Les inconvénients limitent l'application du FD 1D.



(a) Une forme sans contour, (b) et (c) deux formes ayant le même contour mais un contenu intérieur différent [2].

3.2.2 Transformation de Fourier polaire

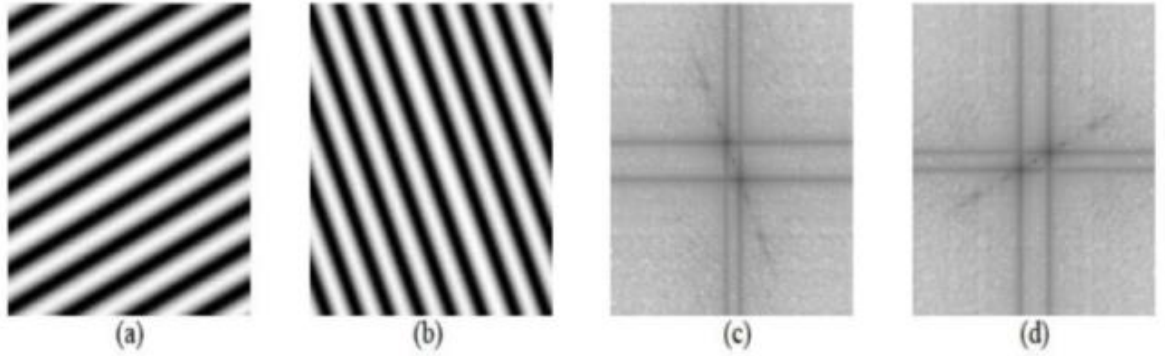
La transformée de Fourier a été largement utilisée pour le traitement et l'analyse d'images. L'avantage de l'analyse d'image dans le domaine spectral par rapport à l'analyse de la forme dans le domaine spatial est qu'il est facile de surmonter le problème de bruit qui est commun aux images numériques [5]. De plus, les caractéristiques spectrales d'une image sont généralement plus concises que celles extraites du domaine spatial. Un FT dimensionnel a été appliqué avec succès à la forme du contour (qui est généralement représentée par une signature de forme dérivée des coordonnées de la limite de forme) pour dériver FD. L'application de la forme unidimensionnelle sur la forme suppose la connaissance des informations sur les limites de la forme. Il n'y a pas de travail signalé sur la région FD. Dans cette section, nous introduisons des FD génériques dérivés de PFT 2-D [6]. Les transformées de Fourier 2D continues et discrètes d'une image de forme $f(x, y)$, ($0 \leq x < M$, $0 \leq y < N$) sont données par (eq1) et (eq2) respectivement.

$$F(u, v) = \int_x \int_y f(x, y) \exp[-j2\pi(ux + vy)] dx dy \quad eq1$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux/M + vy/N)] \quad eq2$$

Les valeurs u et v dans (eq2) sont les u_{th} et v_{th} fréquences spatiales dans les directions horizontale et verticale, respectivement. Le mode bidimensionnel FT peut être appliqué directement à n'importe quelle image de forme sans supposer la connaissance des informations de contour. Toutefois, l'application directe de FT bidimensionnel sur une image de forme dans l'espace cartésien pour dériver des FD n'est pas pratique, car les fonctions capturées par 2-D FT ne sont pas invariants en rotation. L'invariance en rotation d'une forme est importante car des formes similaires peuvent être orientées différemment. Par exemple, les deux modèles (formes) de la figure qui suit : (a) et (b) sont des modèles similaires (formes), cependant, leurs distributions de spectres de Fourier ((c) et (d)) sur le plan des fréquences sont différentes [7] [2].

La différence de distribution des caractéristiques rend difficile l'appariement des deux modèles, en particulier en ligne.



Par conséquent, nous considérons l'image de forme dans l'espace polaire et l'application de la transformée de Fourier polaire (PFT) sur l'image de forme. La PFT produit des données invariantes par rapport à la rotation, particulièrement bien adaptées à l'extraction précise des caractéristiques d'orientation. Dans ce qui suit, nous étudions et décrivons deux PFT. L'étude est nécessaire, car une méthode théoriquement valable peut ne pas être facilement applicable pour la mise en œuvre [8]. Pour dériver une PFT, les données $f(x, y)$ et les spectres $F(u, v)$ sont placés dans l'espace polaire, c'est-à-dire :

$$\begin{aligned}x &= r \cos \theta, & y &= r \sin \theta \\u &= p \cos \Psi & v &= p \sin \Psi\end{aligned}\quad eq3$$

(r, θ) est les coordonnées polaires dans le plan image

(p, Ψ) est les coordonnées polaires en plan de fréquence.

Les différentiels de x et y est :

$$\begin{aligned}dx &= \cos \theta dr - r \sin \theta d\theta \\dy &= \sin \theta dr + r \cos \theta d\theta\end{aligned}\quad eq4$$

Le jacobien de (eq4) est r en remplaçant (eq3) et (eq4) par (eq2), on obtient le Transformée de Fourier polaire (PFT1):

$$PF_1(\rho, \Psi) = \int_r \int_{\theta} r f(r, \theta) \exp[-j2\pi r \rho \sin(\theta + \Psi)] dr d\theta \quad eq5$$

La PFT1 discrète est alors obtenue sous la forme :

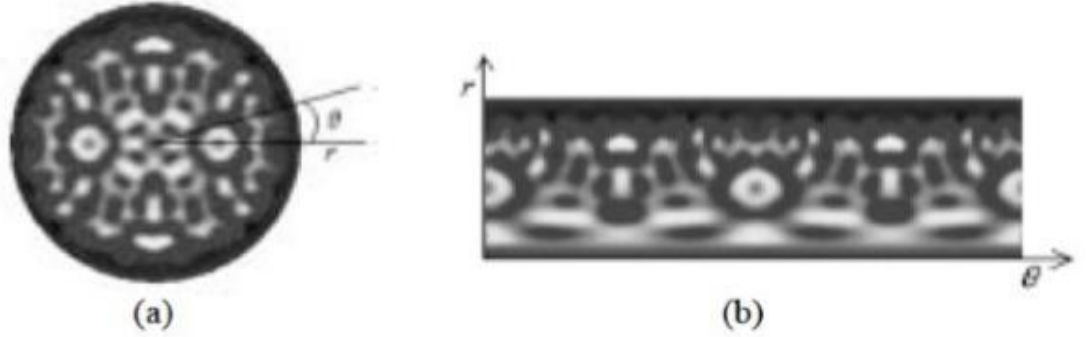
$$PF_1(\rho_l, \Psi_m) = \sum_p \sum_i f(r_p, \theta_i) r_p \exp[-j2\pi r_p \rho_l \sin(\theta_i + \Psi_m)] \quad eq6$$

Où : $r_p = p / R$, $\theta_i = i (2\pi / T)$ ($0 \leq i < T$); $\rho_l = l$ ($0 \leq l < R$) et $m = m\theta_i$.

R et T sont la résolution de fréquence radiale et de fréquence angulaire respectivement. Les coefficients de Fourier polaire acquis $F(\rho, \Psi)$ sont utilisés pour dériver FD normalisé pour la représentation de forme.

PFT1 est le résultat direct de la transformation polaire de (eq1). Cependant, en raison de la présence de m dans la fonction de péché $\sin(\theta_i + \Psi_m)$, la signification physique de m n'est pas la même fréquence angulaire. Les caractéristiques capturées par la PFT1 ont perdu leur signification physique dans une direction circulaire. Pour résoudre le problème.

Un FT polaire modifié (PFT2) est obtenu en traitant l'image polaire dans l'espace polaire comme une image rectangulaire bidimensionnelle normale dans un espace cartésien. Les figures qui suit illustre l'image polaire rectangulaire. La figure (a) est l'image de forme originale dans l'espace polaire, la figure (b) est l'image polaire rectangulaire tracée dans un espace cartésien.



(a) Image de forme originale dans l'espace polaire ; (b) image polaire de (a) tracée dans espace cartésien [2].

L'image polaire de la figure (b) est l'image rectangulaire normale. Par conséquent, si nous appliquons FT 2D sur cette image rectangulaire, le FT polaire a la même forme que le FT discret 2D normal de (eq6) dans l'espace cartésien. Par conséquent, le FT polaire modifié est obtenu comme :

$$PF_2(\rho, \varphi) = \sum_r \sum_i f(r, \theta_i) \exp[j2\pi(\frac{r}{R}\rho + \frac{2\pi i}{T}\varphi)] \quad eq7$$

Où : $0 \leq r < R$ et $\theta_i = i(2\pi / T)$ ($0 \leq i < T$); $0 \leq \rho < R$, $0 \leq \varphi < T$.

R et T sont les résolutions radiales et angulaires. PFT2 a une forme plus simple que PFT1. Il n'est pas nécessaire de contraindre la forme dans un cercle unitaire (la contrainte nécessite une normalisation d'échelle supplémentaire dans le domaine spatial). Et la signification physique de ρ et φ est similaire à u et v dans (eq2). Les ρ et φ sont simplement le nombre de fréquences radiales sélectionnées et le nombre de fréquences angulaires sélectionnées. La détermination de ρ et φ est réalisable physiquement, car les caractéristiques de forme sont généralement capturées par les quelques basses fréquences [2]. La figure 1: (a) (b) montre les images polaires des deux motifs de la figure précédente : (a) (b) et leurs spectres de Fourier polaire sont représentés sur les figures (c) et (d). On peut observer sur la figure 1 que la rotation d'un motif dans l'espace cartésien entraîne un décalage circulaire dans l'espace polaire. Le décalage circulaire ne modifie pas la distribution des spectres dans l'espace polaire. Ceci est démontré aux figures 1: (c) et (d). Les spectres de Fourier polaire sont plus concentrés autour de l'origine de l'espace polaire. Ceci est particulièrement bien adapté à la représentation de forme car, pour une représentation efficace de la forme,

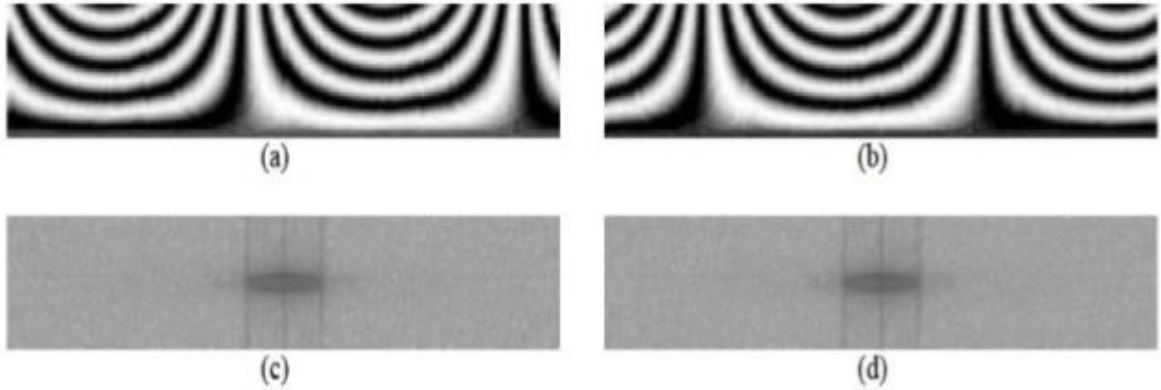


Figure 1 : (a) (b) images polaires des deux motifs de la figure 3 (a) et (b); (c) Fourier spectres de (a) ;(d) spectres de Fourier de (b)[2]

3.3 Dérivation de FD générique

Dans cette section, la dérivation de FD en utilisant la PFT est décrite en détail. Étant donné une image de forme $\{f(x, y); 0 \leq x < M, 0 \leq y < N\}$. Pour appliquer PFT, l'image de forme est convertie à partir de l'espace cartésien à l'espace polaire $l_p = \{f(r, \theta); 0 \leq r < R, 0 \leq \theta < 2\pi\}$, R est le rayon maximum de la forme. L'origine de l'espace polaire doit être le centre de la forme, de sorte que la forme est invariante de transformation [4]. Le centroïde (x_c, y_c) est donné par :

$$x_c = \frac{1}{M} \sum_{x=0}^{N-1} x, \quad y_c = \frac{1}{N} \sum_{y=0}^{M-1} y \quad eq8$$

Et:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \quad \theta = \arctang \frac{y - y_c}{x - x_c} \quad eq9$$

Les PFT sont appliqués sur l_p . Les coefficients acquis des trois transformées sont invariants en transformation en raison de l'utilisation du centroïde comme origine de l'espace polaire. L'invariance en rotation est obtenue en ignorant les informations de phase contenues dans les coefficients et en ne conservant que les grandeurs des coefficients. Pour obtenir l'invariance d'échelle, la première valeur de magnitude est normalisée par la surface du cercle dans laquelle se trouve l'image polaire ou par la masse de la forme. Toutes les autres valeurs de magnitude sont normalisées par la valeur du premier coefficient. Les coefficients PFT normalisés en transformation, en rotation et en échelle sont utilisés comme descripteurs de forme [9]. Pour résumer, le descripteur déforme dériver le FD dérivé de PFT1 et PFT2 sont respectivement FD1 et FD2. Ils sont représentés par suivant :

$$FD1 = \left\{ \frac{PF_1(0,0)}{area}, \frac{|PF_1(0,1)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(0,n)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(m,0)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(m,n)|}{|PF_1(0,0)|} \right\}$$

$$FD2 = \left\{ \frac{PF_2(0,0)}{area}, \frac{|PF_2(0,1)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(0,n)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(m,0)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(m,n)|}{|PF_2(0,0)|} \right\}$$

Où m est le nombre maximal de fréquences radiales sélectionnées et n le nombre maximal de fréquences angulaires sélectionnées. m et n peuvent être ajustés pour atteindre l'exigence de représentation hiérarchique grossière à fine. Pour une description efficace de la forme, seul un petit nombre des descripteurs acquis est sélectionné pour la représentation de la forme. Les descripteurs sélectionnés forment un vecteur de caractéristiques qui est utilisé pour indexer la forme. Pour deux formes représentées par leurs descripteurs de Fourier, la similarité entre les deux formes est mesurée par la distance Manhattan entre les deux vecteurs de caractéristiques des formes. Par conséquent, la mise en correspondance en ligne est efficace et simple [9].

3.4 Mise en œuvre du GFD

L'implémentation de GFD peut être résumée en 4 étapes: normalisation de la transformation, transformée de Fourier polaire, normalisation de la rotation et normalisation de l'échelle. L'algorithme de dérivation de GFD à l'aide de PFT2 est donné à la Figure II.

Les algorithmes de dérivation de GFD à l'aide de PFT1 sont similaires, la seule différence étant dans le calcul du pas de transformation de Fourier polaire et du pas de normalisation d'échelle[2].

3.5 Algorithme de dérivation de GFD

1. Données d'image de forme en entrée $f(x, y)$;
2. Obtenir le centre de la forme (x_c, y_c) ;
3. Définissez le centroïde comme origine; /* normalisation de la traduction */
4. Obtenez le rayon maximum de l'image de la forme (maxRad);
5. Transformation de Fourier polaire
 - Pour la fréquence radiale (rad) de zéro à la fréquence radiale maximale (m)
 - Pour la fréquence angulaire (ang) de zéro à la fréquence angulaire maximale (n)
 - Pour x de zéro à la largeur de l'image de forme
 - Pour y de zéro à la hauteur de l'image de la forme
 - {rayon = racine carrée $[(x - \text{maxRad})^2 + (y - \text{maxRad})^2]$;
 - thêta = arctan2 $[(y - \text{maxRad}) / (x - \text{maxRad})]$;
 - /* thêta se situe entre $[-\pi, +\pi]$ */
 - si (thêta < 0) thêta + $= 2\pi$;
 - /* étendre thêta à $[0, 2\pi]$ */
 - FR [rad] [ang] + $= f(x, y) \times \cos [2\pi \times \text{rad} \times (\text{rayon} / \text{maxRad}) + \text{ang} \times \text{thêta}]$;
 - /* partie réelle du spectre */
 - FI [rad] [ang] - $= f(x, y) \times \sin [2\pi \times \text{rad} \times (\text{rayon} / \text{maxRad}) + \text{ang} \times \text{thêta}]$;
 - /* partie imaginaire de spectres */
 - }
6. Calculer FD
 - Pour rad de zéro à m
 - Pour ang de zéro à n
 - {/* rotation et normalisation d'échelle */
 - Si (rad = 0 & ang = 0)
 - FD [0] = racine carrée $[(\text{FR2} [0] [0] + \text{FI2} [0] [0]) / (\pi \times \text{maxRad}^2)]$;
 - Autre
 - FD [rad \times n + ang] = racine carrée $[(\text{FR2} [\text{rad}] [\text{ang}] + \text{FI2} [\text{rad}] [\text{ang}]) / \text{FD} [0]]$;
 - }
7. Vecteur de caractéristique de sortie FD.

Figure II : Algorithme de GFD.

MODÉLISATION, IMPLÉMENTATION ET RÉSULTATS

4. Modélisation, Implémentation et Résultats

4.1 Définitions et Objectifs

Notre travail traite 5 tâches :

Prétraitement de l'image (appliquer des filtres, pour réduire le bruit).

Détecter les lignes de l'image et les dessiner sur cette dernier pour isoler les marches.

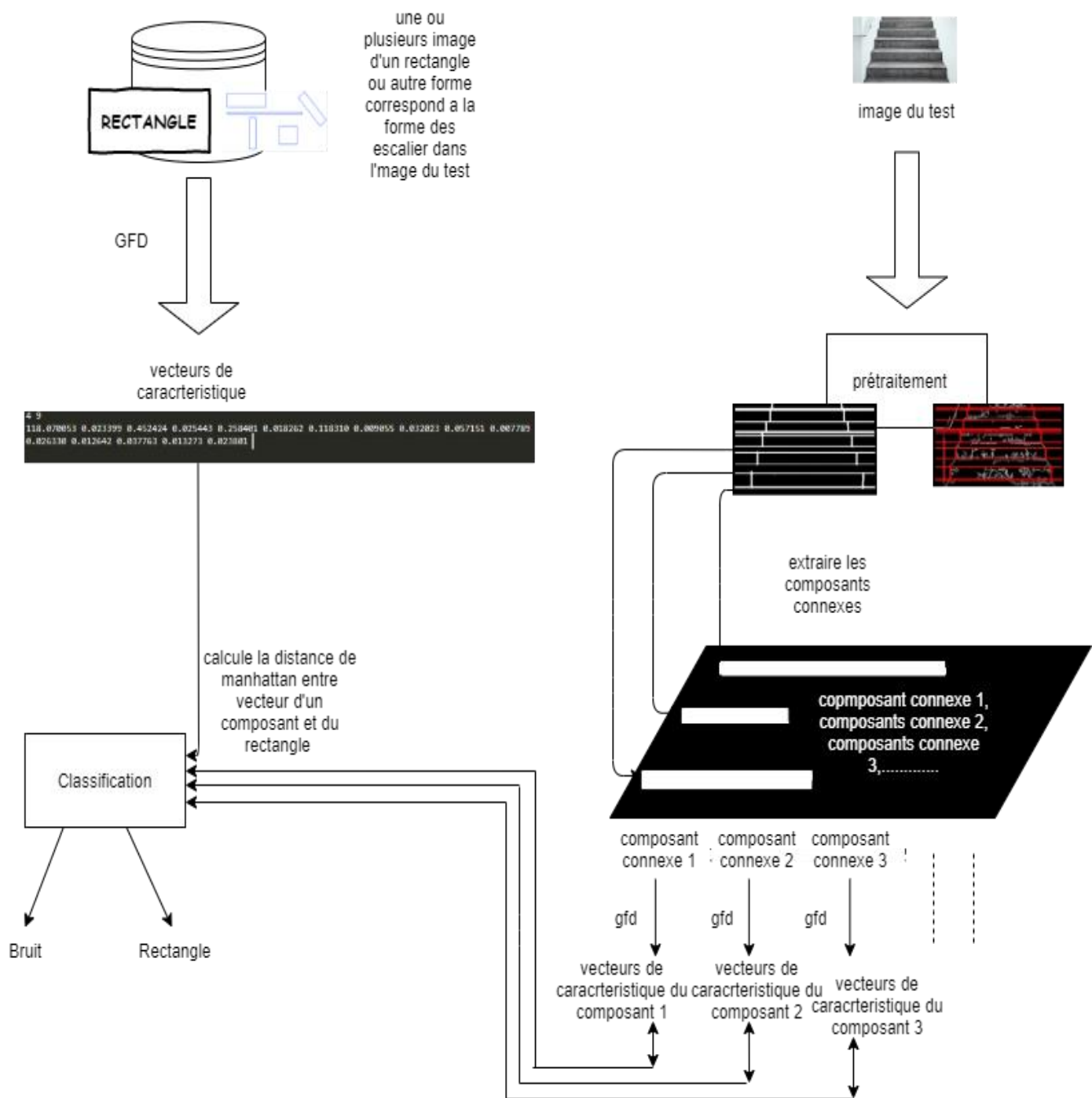
Décomposer l'image en extraira les composant connexe.

Calculer le vecteur de caractéristiques (GFD descripteur) de chaque composant.

Classifier les composants en calcule la distance de Manhattan entre ces vecteurs et le vecteur de caractéristique du rectangle.

4.2 Description de l'algorithme

4.2.1 Le schéma globale



Explication et justification plus profonde des étapes :

1) L'étape du **prétraitement** a pour but de sépare chaque rectangle d'une marche ou contre marche des escaliers dans l'image initiale a un composant connexe a part et pour faire ça, il est

nécessaire de bien détecter les lignes verticales et horizontales de l'image, nous avons procédé comme suit :

1.2.1 The Canny Edge detector :

1.2.1.1 filtrez tout bruit, un filtre gaussien est utilisé à cet effet.

1.2.1.2 trouvez le gradient d'intensité de l'image pour cela nous suivons une procédure analogue à Sobel.

1.2.1.2.1 appliquer un masque de convolution dans la direction x et y.

1.2.1.2.1 trouver la force du gradient et la direction.

1.2.1.3 une suppression est appliquée cela supprime les pixels qui ne sont pas considérés comme faisant partie d'un bord par conséquent seules les lignes fines resteront.

1.2.1.4 Canny utilise deux seuils

1.2.1.4.1 si le gradient de pixels est supérieur au seuil supérieur, le pixel est accepté comme bord.

1.2.1.4.2 si le gradient de pixels est inférieur au seuil inférieur, le pixel est rejeté.

1.2.1.4.3 si le gradient de pixels se situe entre les deux seuils, il ne sera accepté que s'il est connecté à un pixel supérieur au seuil supérieur.

1.2.2 appliquer la transformation de ligne de Hough standard

1.2.2.1 principes de la transformée de Hough :

Le but est de détecter les contours et les représenter dans un espace de paramètre à deux dimensions :

Détermination de la courbe correspondante dans l'espace des paramètres pour chaque point de contour détecté.

Stocker dans un vecteur à deux dimensions les valeurs des deux paramètres (angle et rayon)

Détection de pics dans la matrice d'accumulation.

II) la GFD :

A pour but d'analyse d'image dans le domaine spectral son avantage par rapport à l'analyse de la forme dans le domaine spatial est qu'il est facile de surmonter le problème de bruit qui est commun aux images numériques. De plus, les caractéristiques spectrales d'une image sont généralement plus concises que celles extraites du domaine spatial. Un FT dimensionnel a été appliqué avec succès à la forme du contour, c'est un bon choix pour éliminer le bruit aux milieux de la marche de l'escalier.

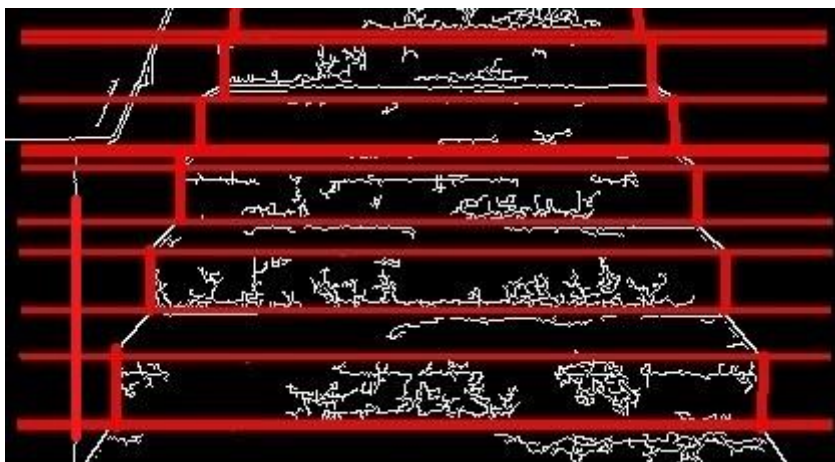
4.3 Evaluation de Performances

Exemple 1 :

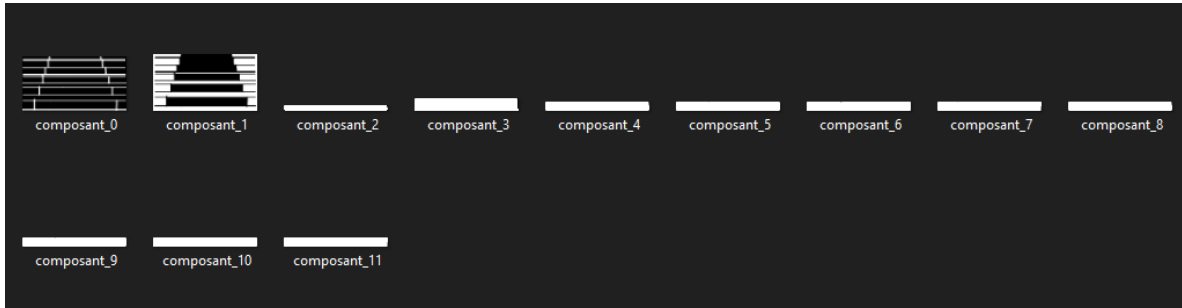
Image de base (5 marches)



Après l'application de Canny Edge detector et la transformation de Hough standard



Extraction des composants connexes après avoir transformé l'image en noir et blanc et faire une binarisation voici les composants connexes :



Classification : en calculant la distance de manhattan entre le vecteur de chaque composant et ce vecteur correspond à cette image du rectangle :



```
4 9
118.070053 0.023399 0.452424 0.025443 0.258401 0.018262
0.118310 0.009055 0.032023 0.057151 0.007789 0.160328 0.021811
0.163290 0.024444 0.119592 0.018962 0.071382 0.068112 0.016627
0.085748 0.015366 0.035537 0.004788 0.021556 0.008767 0.034701
0.056778 0.015730 0.030378 0.005392 0.026330 0.012642 0.037763
0.013273 0.023801
```

Et voici le résultat :

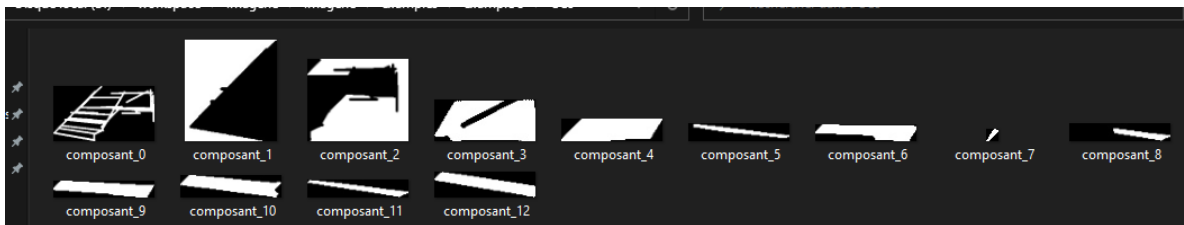
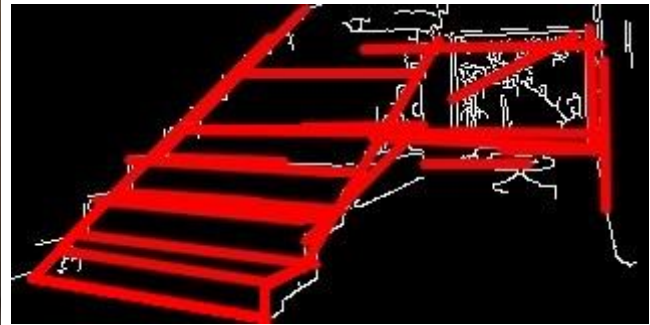
```
composant_1 =>Rectangle H 112.087 | => 112.087 => classe rejet
composant_2 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_3 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_4 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_5 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_6 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_7 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_8 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_9 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_10 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
composant_11 =>Rectangle H 28.8371 | => 28.8371 => Rectangle H
```

On a eu 10 composants classifiés comme rectangle horizontal,

Résultat $10/2=5$ marches (à cause des contres marches les contres marches)

Remarque : on ne prend pas en considération la composant « composant_0 » car elle correspond au fond de l'image.

Exemple 2 : (10 marches)



```

composant_1 =>Rectangle H 137.112 | => 137.112 => classe rejet
composant_2 =>Rectangle H 137.112 | => 137.112 => classe rejet
composant_3 =>Rectangle H 127.669 | => 127.669 => Rectangle H
composant_4 =>Rectangle H 85.645 | => 85.645 => Rectangle H
composant_5 =>Rectangle H 85.645 | => 85.645 => Rectangle H
composant_6 =>Rectangle H 45.8754 | => 45.8754 => Rectangle H
composant_7 =>Rectangle H 45.8754 | => 45.8754 => Rectangle H
composant_8 =>Rectangle H 45.8754 | => 45.8754 => Rectangle H
composant_9 =>Rectangle H 45.8754 | => 45.8754 => Rectangle H
composant_10 =>Rectangle H 45.8754 | => 45.8754 => Rectangle H
composant_11 =>Rectangle H 23.418 | => 23.418 => Rectangle H
composant_12 =>Rectangle H 23.418 | => 23.418 => Rectangle H

```

Résultat :10 marches.

Remarque : les seuils utiliser pour chaque image ne sont pas les mêmes (seuil pour filtres et/ou pour la transformation de Hough)

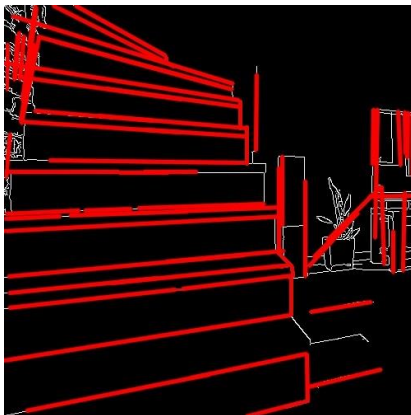
Exemple 3 : (15 matches)



```
composant_1 =>Rectangle H 202.137 => 202.137 => classe rejet
composant_2 =>Rectangle H 189.612 => 189.612 => classe rejet
composant_3 =>Rectangle H 189.612 => 189.612 => classe rejet
composant_4 =>Rectangle H 189.612 => 189.612 => classe rejet
composant_5 =>Rectangle H 189.612 => 189.612 => classe rejet
composant_6 =>Rectangle H 67.5777 => 67.5777 => classe rejet
composant_7 =>Rectangle H 67.5777 => 67.5777 => classe rejet
composant_8 =>Rectangle H 67.5777 => 67.5777 => classe rejet
composant_9 =>Rectangle H 33.3796 => 33.3796 => classe rejet
composant_10 =>Rectangle H 33.3796 => 33.3796 => classe rejet
composant_11 =>Rectangle H 33.3796 => 33.3796 => classe rejet
composant_12 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_13 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_14 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_15 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_16 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_17 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_18 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_19 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_20 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_21 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_22 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_23 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_24 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_25 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_26 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_27 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_28 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_29 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_30 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_31 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_32 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_33 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_34 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_35 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_36 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_37 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_38 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
composant_39 =>Rectangle H 15.3391 => 15.3391 => Rectangle H
```

$28/2 = 14$ marches alors que y'en a 15.

Exemple 4 : (8 marches) un exemple qui montre les limites de la méthode



Si vous avez remarqué y'a 3 rectangle pour une marche, et la GFD ne fait pas de différence entre un rectangle large et étroit.



```
composant_1 => 343.067 => classe rejet  
composant_2 => 33.0427 => Rectangle H  
composant_3 => 30.0425 => Rectangle H  
composant_4 => 33.0100 => Rectangle H  
composant_5 => 33.0420 => Rectangle H  
composant_6 => 15.0424 => Rectangle H  
composant_7 => 20.0938 => Rectangle H  
composant_8 => 19.0289 => Rectangle H  
composant_9 => 10.0425 => Rectangle H  
composant_10 => 33.0421 => Rectangle H  
composant_11 => 16.0475 => Rectangle H  
composant_12 => 33.0427 => Rectangle H  
composant_13 => 13.8099 => Rectangle H
```

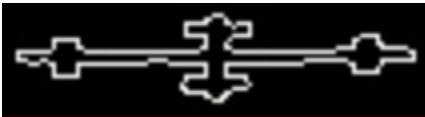
Résultat = 12 marches

Exemple 5 de la base de données commun (13 marches):



Si vous vous souvenez dans le schéma de l'algorithme j'ai mentionné que on peut mettre la forme de la marche pour calculer son vecteur de caractéristique et le comparer.

Dans cette image on n'a pas besoin de détecter les lignes car la forme elle représente un seul composant connexe, il suffit juste de remplacer le rectangle qui été dans la BDD par une photo de la croix qui apparait dans l'image comme suit.

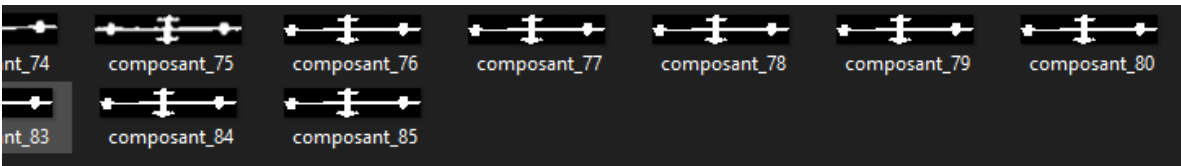


Résultat : 12 composants connexes classier comme une croix

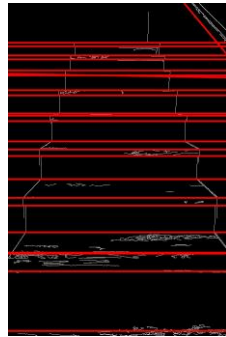
```
composant_71 =>Croix 22.156 | => 22.156 => classe rejet  
composant_72 =>Croix 22.156 | => 22.156 => classe rejet  
composant_73 =>Croix 22.156 | => 22.156 => classe rejet  
composant_74 =>Croix 22.0205 | => 22.0205 => Croix  
composant_75 =>Croix 22.0205 | => 22.0205 => Croix  
composant_76 =>Croix 22.0205 | => 22.0205 => Croix  
composant_77 =>Croix 22.0205 | => 22.0205 => Croix  
composant_78 =>Croix 22.0205 | => 22.0205 => Croix  
composant_79 =>Croix 22.0205 | => 22.0205 => Croix  
composant_80 =>Croix 22.0205 | => 22.0205 => Croix  
composant_81 =>Croix 22.0205 | => 22.0205 => Croix  
composant_82 =>Croix 22.0205 | => 22.0205 => Croix  
composant_83 =>Croix 22.0205 | => 22.0205 => Croix  
composant_84 =>Croix 21.9175 | => 21.9175 => Croix  
composant_85 =>Croix 21.9175 | => 21.9175 => Croix
```

Nombre de croix = nombre de marches =12

Alors qu'il y'a 13 car la dernière croix tout au fond n'apparait qu'à a moitié.



Exemple 6 de la base de données commun (9 marches):

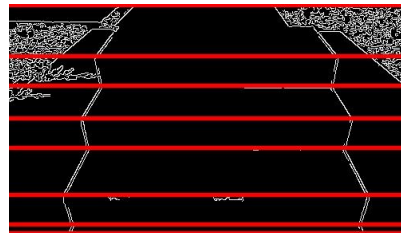


Ce cas-là a été déjà mentionner c'est la limite de la méthode.

Résultat =12 alors que y'a que 9 marches

(Détections de plusieurs rectangles pour la même marche).

Exemple 7 de la base de données commun (4 marches):



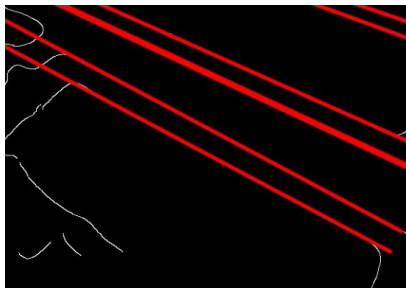
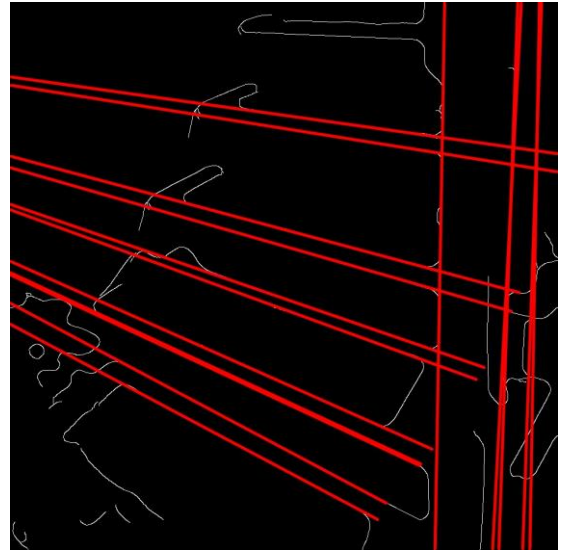
8 rectangles détecter /2 (pour retirer les contres marches) = 4 marches

```
composant_1 =>Rectangle H 56.5962 | => 56.5962 => Rectangle H  
composant_2 =>Rectangle H 32.3862 | => 32.3862 => Rectangle H  
composant_3 =>Rectangle H 32.3862 | => 32.3862 => Rectangle H  
composant_4 =>Rectangle H 32.3862 | => 32.3862 => Rectangle H  
composant_5 =>Rectangle H 32.3862 | => 32.3862 => Rectangle H  
composant_6 =>Rectangle H 32.3862 | => 32.3862 => Rectangle H  
composant_7 =>Rectangle H 8.62654 | => 8.62654 => Rectangle H  
composant_8 =>Rectangle H 3.78436 | => 3.78436 => Rectangle H
```

Exemple 8 de la base de données commun (5 marches) :

Les résultats sont juste pour cette image bien que les rectangles détecter sont pas toujours les rectangles de la marche, regarder l'image suivante :

Remarque : on a rajouté un filtre médian pour avoir ce résultat car il est Plus adapté au bruit impulsionnel.



La par exemple dans la première marche ce n'est pas le bon rectangle mais c'est assez suivant pour compter la marche.

4.3.1 Table de confusion

Dans cette table de confusion on a les résultats de détection de rectangle ie :

Même les rectangles (non pas les rectangles de la marche) sont détectés.

| Trouvé / Réel | Rectangle | Non Rectangle |
|---------------|-----------|---------------|
| Rectangle | 82 | 10 |
| Non Rectangle | 12 | |

Rappel = 0.87, précision = 0.89.

Le taux d'erreur de prédiction est calculé comme suit :

$$\text{pourcentage d'erreur} = \left| \frac{\text{valeur expérimentale} - \text{valeur théorique}}{\text{valeur théorique}} \right| \times 100\%$$

Est égale à 20% environ.

Environnement et modèle de simulation

Langage de programmation utiliser : c++.

Bibliothèque utiliser : OpenCv 4.3.0 dans la partie de prétraitement.

4.3.2 Conclusion

On a essayé de faire une méthode compatible avec n'importe quelle image contient des escaliers avec n'importe quelle forme de marche on n'a pas eu toujours le bon nombre de marche exacte mais ont été toujours proche avec 4 ou 3 de plus ou de moins du chiffre exact.

RÉFÉRENCES

5. Références

- [1] Transformée_de_Hough https://fr.wikipedia.org/wiki/Transformée_de_Hough.
- [2] D. S. Zhang and G. Lu. (August 22-25, 2001) "Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study". Dans Proc. Conférence internationale IEEE sur le multimédia et les expositions (ICME2001), Tokyo, Japon, pp.317-320.
- [3] Hannu Kauppinen, Tapio Seppanen and Matti Pietikainen. (1995) "An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification". IEEE Trans. PAMI-17(2):201-207.
- [4] D. S. Zhang and G. J. Lu. (June 2001) "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures". Dans Proc. Int. Conférence sur le multimédia et la formation à distance. Fargo, ND, États-Unis, pp1-9.
- [5] Eric Persoon and King-sun Fu. (1977) "Shape Discrimination Using Fourier Descriptors". IEEE Trans. On Systems, Man and Cybernetics, Vol.SMC-7(3):170-179.
- [6] Zhang, D., & Guojun, L. (s.d.). "Shape Based Image Retrieval Using Generic Fourier Descripteurs". Churchill, Victoria 3842 .
- [7] P. J. van Otterloo. (1991) "A contour-Oriented Approach to Shape Analysis". Prentice Hall International (UK) Ltd, pp90-108.
- [8] Palvides .T. (1996) "Algorithms for Graphic and image processing", Rockville, MD: Computer science press.
- [9] S. Derrode, M. Daoudi, and F. Ghorbel, (1999) "Invariant content-based image retrieval using a complete set of Fourier-Mellin descriptors," in IEEE International Conference on Multimedia Computing and Systems, pp.877-881.