DÉTECTION DE VISAGES EN DOMAINES COMPRESSÉS

par

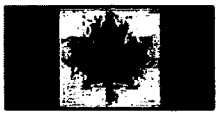
Guido Manfredi

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 7 juin 2011



Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-83649-1

Our file Notre référence
ISBN: 978-0-494-83649-1

Le 31 mai 2011

le jury a accepté le mémoire de Monsieur Guido Manfredi
dans sa version finale.

Membres du jury

Professeur Djemel Ziou
Directeur de recherche
Département d'informatique

Professeure Marie-Flavie Auclair-Fortier
Codirectrice de recherche
Département d'informatique

Professeur Nizar Boughila
Évaluateur externe
Université Concordia
Concordia Institute for Information Systems Engineering

Professeur Richard Egli
Président rapporteur
Département d'informatique

Sommaire

Ce mémoire aborde le problème de la détection de visages à partir d'une image compressée. Il touche également à un problème connexe qui est la qualité des standards de compression et l'estimation de celle-ci. Ce mémoire est organisé sous la forme d'une introduction générale sur la détection de visages et de deux articles soumis à des conférences internationales. Le premier article propose une amélioration de la méthode classique pour comparer la qualité de deux standards. Le deuxième propose une méthode de décompression spécialisée pour faire fonctionner le détecteur de visages de Viola-Jones dans le domaine compressé.

Mots-clés: DCT ; Viola-Jones ; Qualité ; Compression ; Détection de visages ; Domaine compressé.

Remerciements

Je tiens à remercier en premier lieu le Pr. Ziou pour les connaissances qu'il m'a transmises mais également pour son aide et son soutient. Egalement, la Pr. Auclair-Fortier pour ses conseils avisés et son humour. Ainsi que tous les membres du MOIVRE pour les discussions enrichissantes.

Abréviations

LBP *Local binary Pattern* (Motif binaire local).

DCT *Discrete cosine transform* (Transformée en cosinus discrets).

IDCT *Inverse Discrete cosine transform* (Transformée en cosinus discrets inverse).

DCL Domaine des cosinus locaux.

SSIM *Structural SIMilarity.*

PSNR *Peak Signal to Noise Ratio.*

MIT *Massachusetts Institute of Technology*

CMU *Carnegie Mellon University*

IR Infrarouge.

DCM *Digital Microscope Camera* (Microscope digital).

Table des matières

Liste des figures

Liste des tableaux

Introduction

La définition de ce qu'est un visage est un problème qui se pose depuis longtemps et qui n'a toujours pas été résolu. Il n'existe pas de critère qui permette de certifier que quelque chose est un visage ou ne l'est pas. Le choix a toujours été subjectif. Voilà pourquoi la détection automatique de visages est un problème difficile. Des solutions ont été proposées pour le résoudre pour des images non compressées et les taux de détection atteint sont à présent élevés. Néanmoins, dans certaines situations, l'image devra être traitée dans le domaine compressé. Dans ces cas-là, les algorithmes de détection de visages sont rares et peu satisfaisants. L'objectif de ce travail est de trouver quel domaine est le plus favorable pour la détection de visage et comment la mener à bien dans le dit domaine.
Dans le chapitre 1 de ce mémoire, on s'intéressera au fonctionnement de la détection chez l'homme, comme chez la machine. Nous visiterons l'état de l'art dans le domaine, ce qui nous amènera à nous concentrer sur un détecteur particulier de l'état de l'art : le détecteur de Viola-Jones. On verra dans le chapitre 2 que la détection est liée à une notion de qualité de l'image et que cette qualité dépend du format de compression utilisé sur l'image. De plus nous verrons une méthode qui permet d'estimer de manière précise les qualités relatives offertes par différents standards de compression. Cette étude nous mène à étudier, dans le chapitre 3, la détection dans une image compressée au format JPEG. On montrera qu'un outil mathématique, l'image intégrale, classiquement obtenue à partir des pixels d'une image, peut être obtenue à partir de coefficients d'une image compressée. Ainsi, le détecteur de visages de Viola-Jones pourra être utilisé directement sur une image compressée plutôt que sur une image décompressée.

Chapitre 1

Détection automatique de visages.

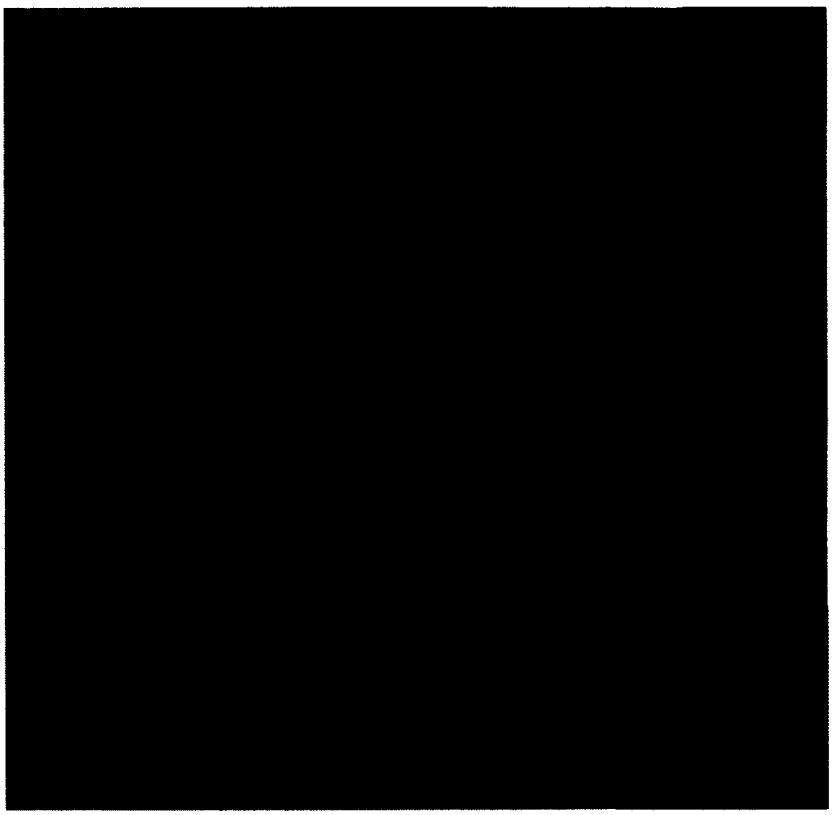


figure 1.1 – Le visage de Lena est détecté et repéré par le grand carré blanc.

1.1 Pourquoi la détection de visages

La détection de visages est le fait de trouver les coordonnées spatiales délimitant un visage dans une image ou une vidéo. En termes simples, cela revient à trouver les carrés qui délimitent le mieux les visages visibles dans une image (figure 1.1). Pour ce faire, les algorithmes doivent utiliser une définition du visage explicite ou implicite. Celle-ci est le plus souvent construite par apprentissage. Dans le cas explicite, la définition du visage est accessible une fois l'apprentissage terminé. C'est le cas par exemple des cascades de classifieurs [18] : on peut retrouver les caractéristiques qui ont permis de classifier correctement le jeu d'entraînement. Dans le cas d'un apprentissage implicite, par exemple pour un réseau de neurones [14], les caractéristiques sélectionnées au terme de l'entraînement sont inconnues.

figure 1.2 – Le programme détecte la position des visages puis reconnaît l'identité des deux personnes.

La détection de visages est le premier maillon de toute chaîne de traitement de visages. En effet, la majorité des techniques de traitement de visages nécessitent une image normalisée et bien cadrée du visage pour fonctionner. C'est le rôle de la détection de fournir cette image.

En reconnaissance de visages on utilise une base contenant des images de plusieurs personnes. Chaque personne est représentée par plusieurs images. Quand l'algorithme reçoit une image d'un visage inconnu, il doit décider s'il s'agit d'une des personnes connues ou non (figure 1.2). Cependant, la reconnaissance exige que l'image inconnue donnée en entrée soit sous la même forme que les visages présents dans la base d'images. Cela exige un cadrage contrôlé du visage, qui est assuré par la détection du visage, puis éventuellement d'autres points de repère tels que les yeux, afin de centrer le visage. La détection a récemment fait parler d'elle car elle est incluse dans les nouveaux logiciels de classement d'images, tels que Picasa, et permet de construire un album pour une personne donnée en retrouvant les photos contenant son visage dans une collection.

Une autre application est la reconnaissance d'expressions du visage (figure 1.3). La reconnaissance d'expression est cruciale dans l'interaction homme machine. En ef-

figure 1.3 – Le programme détecte la position du visage puis reconnaît l'expression de la personne.

fet, les mimiques faciales forment une grande part de la communication humaine. Une machine telle qu'un robot par exemple, qui doit communiquer avec des êtres humains, devra être capable de reconnaître les expressions de ses interlocuteurs pour comprendre pleinement leur message. Mais avant de pouvoir reconnaître une expression, il faut être capable de détecter le visage de l'interlocuteur.

Enfin, la détection de visage peut tout simplement être utilisée pour suivre le déplacement d'une personne et ainsi donner une base solide aux algorithmes de suivi. La première étape avant de commencer un suivi est d'avoir les coordonnées initiales de l'objet. C'est donc grâce à ce point de départ que les positions futures pourront être estimées. D'autres détections permettront de rendre le suivi plus robuste. Là encore, la détection est non seulement la première étape, mais aussi une étape cruciale au bon fonctionnement du système. Dans la suite nous nous concentrons uniquement sur la détection de visages.

La détection de visages est pour tout traitement du visage, ce que sont les fondations pour une maison. De sa robustesse vont dépendre les performances de tous les autres

éléments de la chaîne. Dans le cas idéal, la détection doit être rapide et économique. De plus, elle doit capturer tous les visages présents dans une image et ne doit pas confondre une région de l'arrière-plan avec un visage. C'est là que réside la difficulté : comment faire la différence entre une région d'arrière-plan et un visage ? Dans l'espoir de répondre à cette question nous allons, dans la prochaine section, analyser la façon dont procède le cerveau humain.

1.2 La détection de visages chez l'homme

D'après certaines études [4], des troubles de la reconnaissance de visage, tels que certains cas de prosopagnosie (mauvaise mémoire des visages), sont dus à une mauvaise détection des visages. Ces troubles ont permis de mettre en évidence les mécanismes de détection de visages chez l'homme. Cette activité s'avère être essentielle à la vie humaine, la preuve la plus probante est donnée par la nature : la détection de visages mobilise la majorité des zones du cerveau [6]. Cette tâche fait donc appel à des fonctions complexes et nombreuses : notre perception lui accorde une grande importance, mais ces mécanismes restent mal compris. Peu d'études ont été faites sur le sujet [9] et la rareté des troubles de la détection de visages prive ces études de nombreuses informations.

Pour comprendre le fonctionnement d'une détection de visage, il faut d'abord comprendre ce qu'est un visage. Intuitivement, nous aurions tendance à dire que c'est une forme ronde assortie de deux yeux, d'un nez et d'une bouche. Néanmoins, l'absence de ces organes ne nous empêche pas de reconnaître la présence d'un visage (figure 1.4).

Un visage peut également être considéré comme une alternance de zones sombres et moins sombres. Les différents creux et bosses du visage forment des contrastes qui nous aident lors de la détection d'un visage (figure 1.5). Néanmoins, même sans ces contrastes, l'être humain est capable de détecter un visage.

Nous avons donné deux exemples de définitions, mais elles sont loin d'être exhaustives : beaucoup d'autres existent sans pour autant couvrir toute la diversité des visages. Trouver une définition générale qui s'applique à toutes les situations reste donc un problème ouvert. Pourtant, des études ont permis de tirer certaines observa-

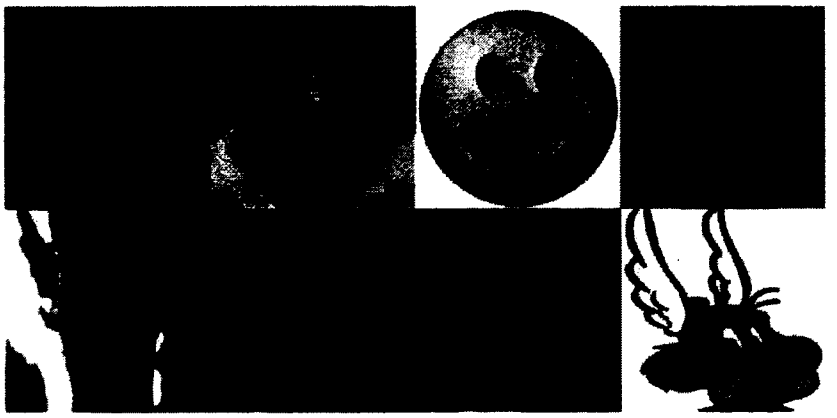


figure 1.4 – Un visage est quelque chose de très variable, il n'existe pas de définition qui prend en compte tous les cas possibles.

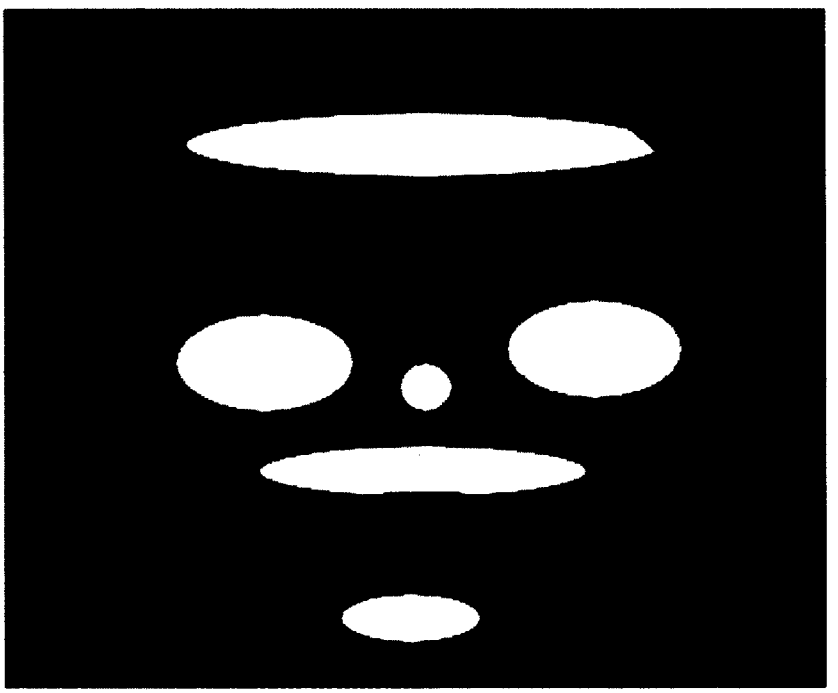
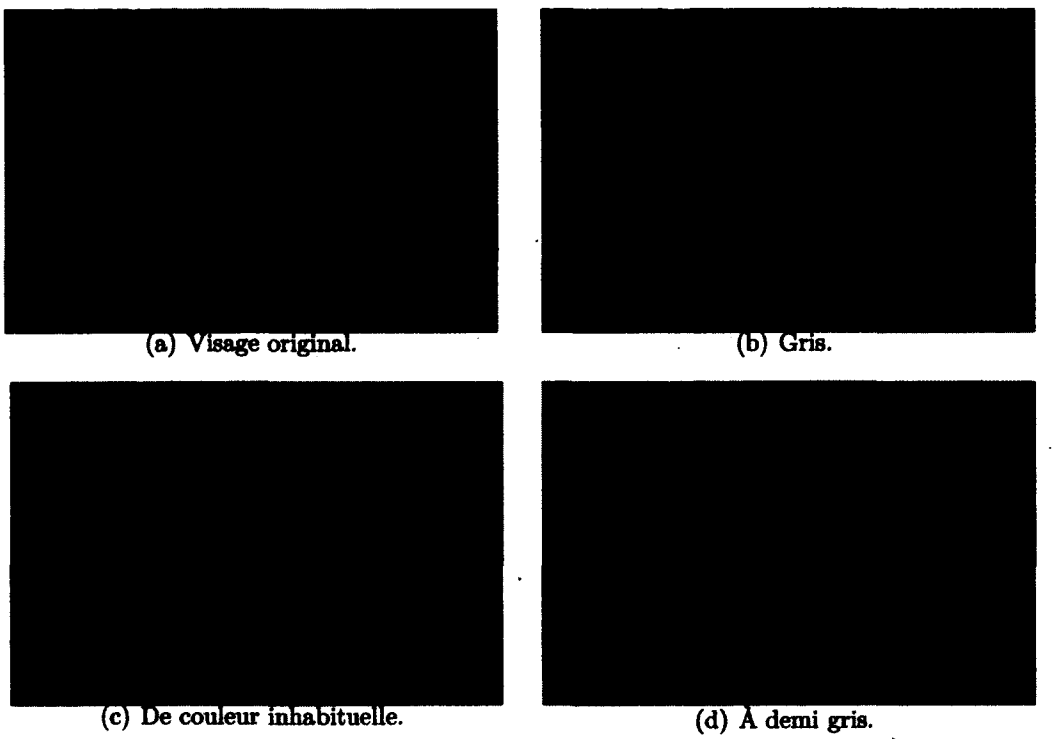


figure 1.5 – Un visage peut être vu comme un ensemble de zones sombres et moins sombres. En bas, la bouche (sombre) entourée du menton et de la lèvre supérieure (moins sombre). Au dessus, la pointe du nez (moins sombre) et de chaque coté les pommettes (moins sombres). Finalement, en haut les yeux (sombres) et l'arcade (moins sombre).

tions que nous allons rapporter.

La première question à laquelle se sont attaqués les chercheurs est de savoir si oui ou non la couleur est une information nécessaire à la détection. Dans [1], les auteurs ont mis en évidence le fait que des visages en couleurs sont plus rapides à détecter que des visages en niveaux de gris, que ces derniers soient placés dans une image noire et blanche ou en couleur (figure 1.6(b)). Changer la couleur d'un visage pour une couleur non naturelle, par exemple une peau bleue, ralentit la vitesse de détection (figure 1.6(c)). Par ailleurs, la détection d'un visage qui n'est qu'à moitié couleur peau est aussi lente que pour un visage gris (figure 1.6(d)).



(a) Visage original. (b) Gris.

(c) De couleur inhabituelle. (d) À demi gris.

figure 1.6 – Différentes expériences sur le rôle de la couleur dans la détection.

Bien que la détection soit plus rapide pour des visages de couleur peau, elle à également lieu pour des visages de couleur non naturelle. Cela suggère que la sélection d'une région d'intérêt ne se fait pas seulement par segmentation de peau mais aussi par l'intervention d'autres informations, comme la forme de l'objet. Une autre information

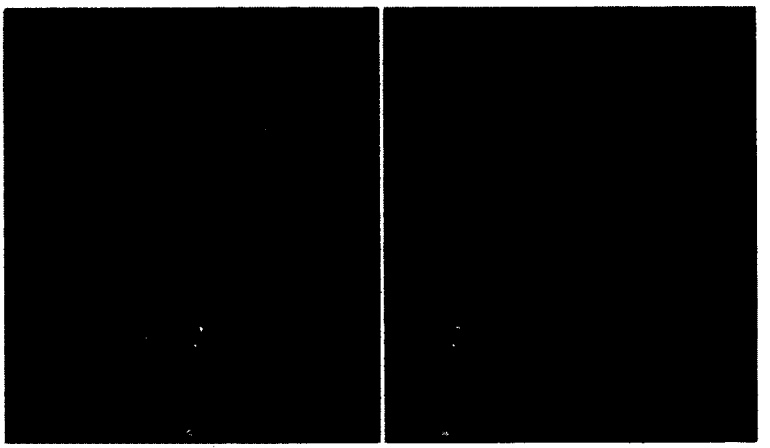


figure 1.7 – Les informations contextuelles nous aident à retrouver un visage. Dans ce cas, la présence d'un corps nous dit qu'au dessus des épaules il y aura un visage.

importante s'avère être le contexte : Lewis et Edmond [9] montrent que le contexte influence la vitesse de détection de visages. Le contenu de la scène oriente et accélère la détection. Par exemple, une tête sur deux épaules sera plus facilement détectée qu'une tête placée en plein milieu de nulle part (figure 1.7). Cela implique qu'il y a une recherche de régions d'intérêt préalable à la détection elle-même. Cette recherche discrimine entre ce qui a des chances d'être un visage et ce qui forme le reste de l'image.

De plus, la détection peut être ralentie par différents types de « bruits ». Dans [9] et [4], les auteurs comparent les vitesses de détection pour des visages normaux, à luminance inversée, à couleurs inversées, dont un organe a été masqué, à contraste diminué, rendus flous, spatialement inversés (figure 1.8). D'après les résultats de cette étude, les yeux sont des éléments très importants pour la détection. Leur absence n'empêche pas la détection mais la ralentie considérablement [9]. On peut donc supposer qu'il existe un système de détection rapide, qui utilise les yeux et un second, plus lent, qui utilise d'autres caractéristiques présentes dans le visage. La réduction du contraste et le flou s'accumulent pour ralentir la détection de manière plus qu'additive. En effet, la somme des ralentissements dus au contraste seul et au flou seul est inférieure au ralentissement provoqué par la présence des deux [9]. L'inversion, pour

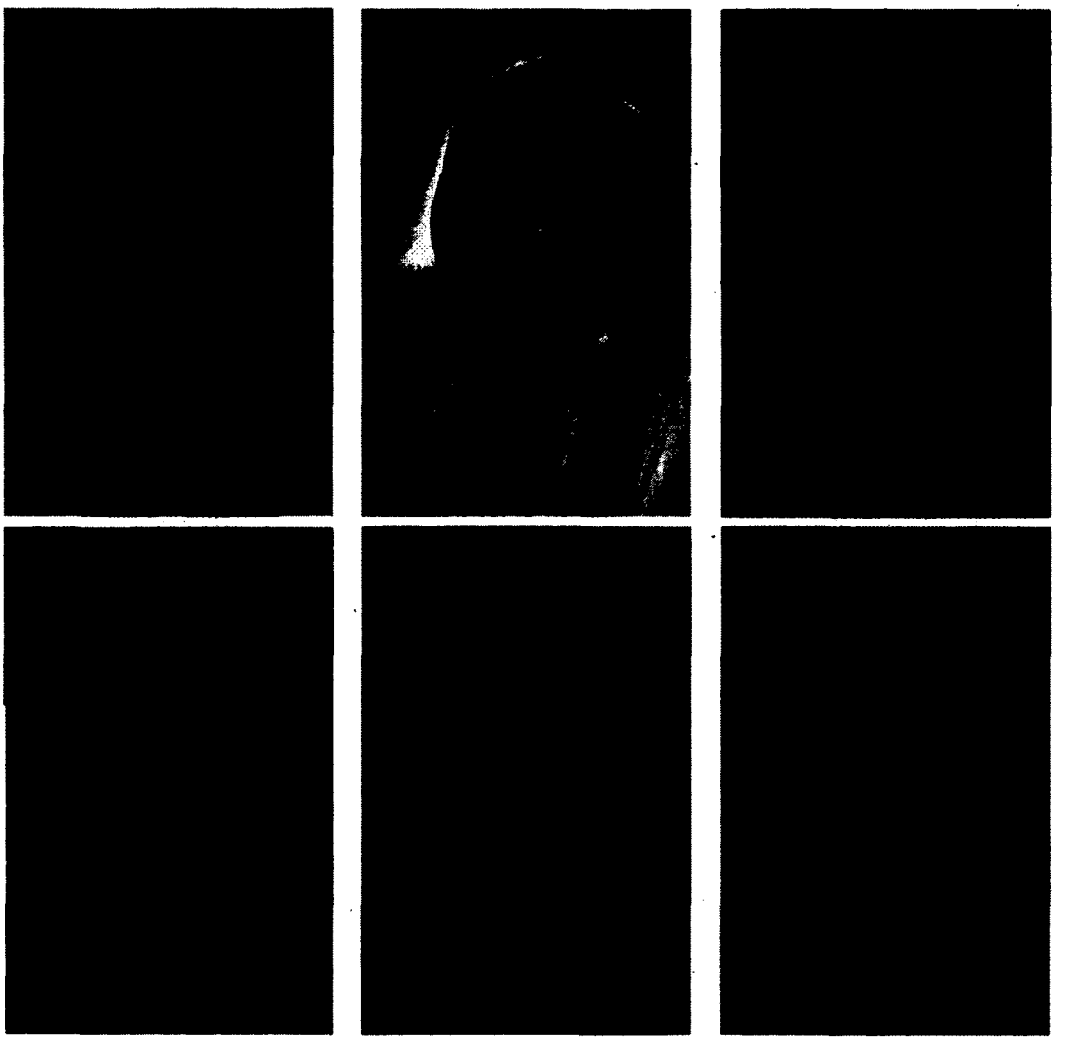


figure 1.8 – Différents bruits appliqués aux images.

sa part, ralentirait de moitié le classement d'une image entre visage et non visage. Cet effet est encore plus fort lorsqu'il s'agit de décider s'il s'agit d'un visage ou d'une partie de visage [4].

Une conclusion, à laquelle aboutissent fréquemment les auteurs [1, 4, 9], est l'existence de plusieurs patrons du visage (figure 1.9) testés successivement. Par exemple, si on regarde un visage à l'envers, le cerveau va d'abord tenter de le détecter en utilisant un patron pour les visages à l'endroit et de face, puis un autre pour les visages à l'endroit et de profil. Enfin il essayera un patron pour les visages à l'envers et de face. C'est seulement ce dernier qui lui permettra de détecter le visage. Cependant une question majeure subsiste : comment est faite l'invariance à l'échelle ? Il ne semble pas y avoir de réponse à cette question du côté neuropsychologie. On va voir par la suite que les traitements automatiques ont déjà une longueur d'avance sur ces études. Néanmoins, on pourra constater, dans la section suivante, une similitude entre le fonctionnement humain et celui des algorithmes automatiques.

1.3 État de l'art

La grande majorité des algorithmes de détection automatique de visages s'inspirent du fonctionnement humain. Ils procèdent en deux étapes. La première consiste à supprimer les régions qui ne contiennent pas de visages. Ce faisant, ils conservent uniquement quelques zones d'intérêt sur l'ensemble de l'image. Ce traitement est une sorte de détection de « non visages ». Cette première étape est très importante puisqu'en réduisant le champ de recherche, elle réduit le nombre de traitements nécessaires pour la suite, ce qui diminue grandement le coût de la détection. Une technique souvent utilisée pour cette tâche est l'identification de la peau [3, 5, 7]. D'autres méthodes existent telles que la recherche spatiale multi-résolution [2, 15, 18] ou la recherche de points saillants [17]. La deuxième étape est la confirmation que les régions candidates sont bien des visages. Si la première étape permet de localiser des candidats, la seconde permet d'éliminer ceux qui ne sont pas susceptibles d'être des visages. À la fin de ces deux traitements, seuls restent les visages (figure 1.10). Les méthodes les plus utilisées sont la comparaison à un patron connu et l'extraction de caractéristiques.

Avant de détecter un visage il faut le modéliser. Deux approches sont généralement

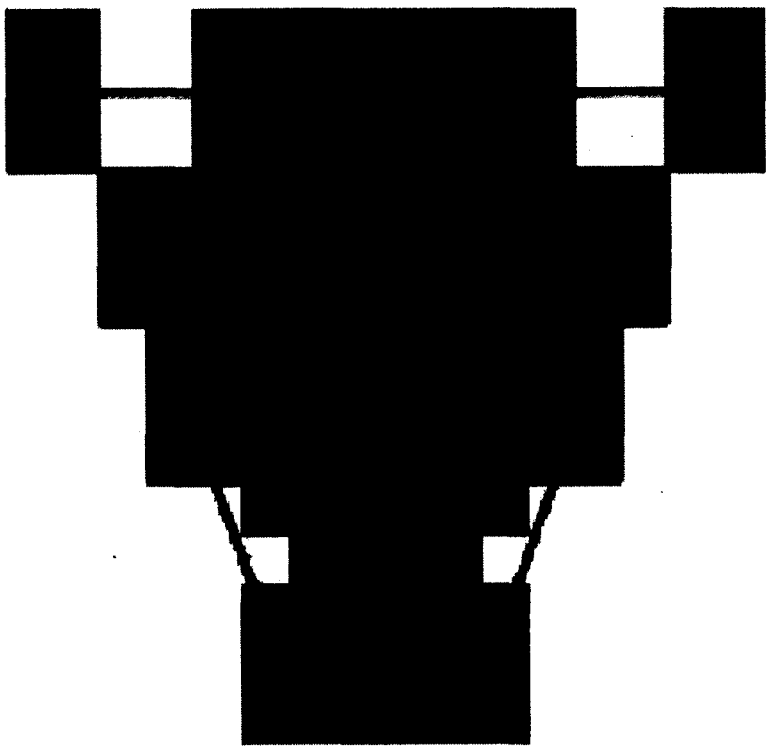


figure 1.9 – Un exemple de patron de visage. Les flèches indiquent des relations entre les différentes parties du visage.

figure 1.10 – La détection se déroule en deux étapes principales .

utilisées. La première suppose un modèle *a priori* et teste si une région candidate respecte les règles de ce modèle. On appelle cela un patron, ou *template* en anglais. Si la zone correspond au patron, alors c'est un visage [3, 12]. Ces méthodes ont tendance à être spécialisées pour des situations particulières, par exemple pour les images ne contenant qu'un seul visage.

La deuxième approche consiste à extraire de manière automatique un patron d'un ensemble d'images de visages. Cette solution demande un apprentissage sur une base de visages. Dans ce type de détecteur, l'algorithme extrait un certain nombre de caractéristiques de la zone candidate. Les invariances de ces caractéristiques aux changements d'échelle et à la position sont le minimum requis pour pouvoir détecter un visage à différentes échelles et à différents endroits [14]. D'autres invariances sont souhaitables pour augmenter la robustesse du détecteur, telles que l'invariance à la luminosité [10] ou aux rotations dans le plan [7].

La capacité descriptive des caractéristiques est également essentielle à une bonne détection. En effet, les caractéristiques doivent discriminer facilement les visages des autres zones. Elles forment un dictionnaire qui va permettre de décrire des visages et des « non visages ». Leur choix est donc crucial. Les plus utilisées sont les histogrammes [15], les caractéristiques de Haar [18], les motifs binaires locaux (LBP) [13] ou encore les coefficients de diverses transformées telles que les ondelettes [11] ou la transformée en cosinus discrète (DCT) [8]. Trouver la base, ou le mélange de bases, optimale pour la détection de visages reste un problème ouvert. Un autre élément important est la vitesse de calcul des éléments de ces dictionnaires, ainsi que leur nombre. Plus les éléments sont nombreux, meilleure sera la description. Malheureusement, plus le nombre d'éléments sera élevé, plus l'entraînement et la détection risquent d'être longs.

Nous avons résumé dans les tableaux 1.1 et 1.2 les travaux importants dans le domaine de la détection de visages. Ils sont organisés en trois groupes selon l'information utilisée pour le calcul des caractéristiques ; ces trois groupes sont décrits par la suite. Dans le tableau 1.1, nous avons relevé les caractéristiques et l'espace mathématique utilisés, ainsi que les invariances propres à chaque caractéristique. Le tableau 1.2 résume les performances en termes de taux de détection (TD), de fausses alarmes (FA) et de vitesse de détection en fonction de la puissance de l'ordinateur et de la

tableau 1.1 – Caractéristiques utilisées par les principaux travaux dans le domaine de la détection de visages et les invariances qu'elles offrent.

Travail	Caractéristique(s)	Domaine(s)	Invariances
[2]	Contours	Spatial	Taille, position, rotation dans plan +/- 20°.
[14]	Pixels	Spatial	Taille, position.
[19]	Pixels	Spatial	Taille, position.
[18]	Haar, LBP	Spatial	Taille, position.
[15]	Histogrammes	Ondelettes	Taille, position, rotation dans le plan.
[16]	Filtre de fréquences	Fourier	Taille, position, rotation dans le plan.
[17]	Filtre Gabor/ saillance	Fourier et spatial	Taille, position, rotations hors plan faibles.
[3]	Coefficients	DCT locale	Taille, position; luminosité, rotations hors plan.
[8]	Coefficients	DCT locale	Taille, position, rotations hors plan.

base de test utilisée.

Du côté des classifieurs, les réseaux de neurones [14, 17, 19] ont l'avantage d'être précis. Néanmoins, on ne sait pas quels éléments sont utilisés pour décrire les visages et ces classifieurs sont lents. Leur manque de rapidité et de transparence les rendent donc peu pratiques. L'utilisation de cascades de classifieurs en conjugaison avec l'algorithme Adaboost à résolu ces deux dernier problèmes [8, 15, 18].

Pour ce qui est des caractéristiques, les premières utilisées ont été des histogrammes, dans le domaine spatial. L'utilisation de caractéristiques plus simples, telles que les caractéristiques de Haar ou les LBP, conjuguées à l'utilisation de l'image intégrale a permis d'accélérer les calculs. De plus, les calculs dans le domaine spatial doivent, en partie, leur vitesse au fait qu'ils utilisent des informations directement accessibles. Ce premier groupe d'approches utilise donc les informations disponibles directement dans le domaine spatial. Un autre type d'approche tire parti des transfor-

mées mathématiques telles que la transformée en ondelettes. Celles-ci fournissent des caractéristiques capables de décrire, de manière riche, des visages à des résolutions différentes. De plus, elles offrent une sensibilité réduite aux changements de luminosité. Malheureusement, le calcul des caractéristiques est relativement lent dans ce cas. Ces techniques utilisent des informations venant de deux sources : l'image dans le domaine spatial et l'image dans le domaine transformé.

Les avantages d'une bonne description et d'un accès rapide à l'information peuvent être combinés en exploitant les coefficients utilisés pour compresser l'image. En effet, ceux-ci sont directement accessibles, sans même avoir besoin de décompresser l'image, d'où un gain de temps. Comme la majorité des images et des vidéos sont compressées avec la DCT locale, certains travaux [3, 8] se sont concentrés sur les coefficients de la DCT afin de détecter des visages. Contrairement aux précédentes, cette approche n'utilise que l'information donnée par l'image dans le domaine transformé et ne dispose pas de la version spatiale de l'image. L'un des principaux obstacles d'une telle approche, est la difficulté à faire une analyse multi-résolution dans le domaine de la DCT locale. Dans [8], l'auteur résout le problème en changeant le nombre de coefficients utilisés selon la résolution considérée. Malheureusement, cela revient à n'utiliser que les coefficients DC de la transformée pour des visages dont la taille est supérieure à 56×56 pixels. De plus, si la technique fonctionne pour des données synthétiques, les résultats présentés sont faibles dans le cas d'images naturelles. Dans [3] la méthode décrite n'utilise pas d'approche multi-résolution mais plutôt un patron de proportions qui permet d'identifier les visages. Mais là encore, les résultats ne sont concluants que dans certaines situations précises, par exemple si le visage est isolé de toute autre partie couleur peau et si le visage est suffisamment grand pour pouvoir distinguer les diverses parties qui permettent la mesure de proportions. Il n'existe pas à ce jour d'algorithme dans le domaine compressé qui surpasse, à tout point de vue, les algorithmes dans le domaine spatial.

Le domaine compressé se résume principalement à deux transformées : la transformée en cosinus discret, avec JPEG et MPEG, et la transformée en ondelettes, avec JPEG2000. Avant de s'atteler à résoudre le problème de la détection dans le domaine compressé, il faut déterminer quelle transformée est la plus adaptée. En effet, pour atteindre un bon taux de détection, les images doivent avoir une bonne qualité visuelle.

Néanmoins, plus une image est compressée, plus sa qualité visuelle est dégradée. Pour détecter des visages, on cherche donc la transformée qui donne la meilleure qualité visuelle pour un taux de compression donné. Dans le chapitre suivant nous allons évaluer les performances de JPEG et JPEG2000 afin de comparer les descriptions offertes par les deux transformées et de choisir le domaine qui donnera les meilleur résultats pour la détection de visages.

tableau 1.2 – Performances des principaux travaux dans le domaine de la détection de visages.

Travail	TD	FA	Vitesse	Machine	Jeu de test (nombre d'images)	Faiblesses
[2]	92%	112	0,48s	1Ghz	custom (77)	Cascade faite à la main, caractéristiques choisies à la main.
[14]	89,9%	422	2 à 4s	200MHz	MIT + CMU (507)	Caractéristiques utilisées inconnues, lent.
[19]	94,8%	78	N/A	N/A	MIT + CMU (125)	Caractéristiques utilisées inconnues, lent.
[18]	93,7%	422	0,067s	700Mhz	MIT + CMU (125)	Pas encore assez rapide ?
[15]	94,4%	65	40,2s	N/A	Custom (208)	Lent.
[16]	98,5%	1,5%	27 images/s	N/A	ORL (400)	Sensible aux rotations, multi-résolution lente.
[17]	65%	7	N/A	N/A	Custom (100)	Très sensible à la saillance.
[3]	N/A	N/A	N/A	N/A	videos custom(2)	Le visage doit être isolé d'autre parties de peau.
[8]	87%	N/A	N/A	N/A	Données synthétiques	Equivalent à [18] sur les coefficients DC pour les hautes résolutions (visage >56x56 px)

Chapitre 2

Comparaisons de qualité d'images compressées

Afin de détecter au mieux des visages, il faut utiliser des images de la meilleure qualité possible. Dans le cas de la détection en domaine compressé se rajoute une contrainte : la qualité des images est diminuée par la compression. Selon le standard utilisé la qualité sera plus ou moins réduite. On cherche donc quel standard offre le meilleur compromis taux de compression/qualité visuelle.
Deux standards largement utilisés ont été considérés : JPEG et JPEG2000, en plus d'un modèle qui nous est propre et que nous nommerons WJPEG2000. Les performances des trois compresseurs ont été comparées sur une base d'images afin de déterminer le standard le plus performant, en termes de rapport taux de compression/qualité d'image, pour ce problème. L'ajout d'un paramètre supplémentaire, le type d'image compressé, a permis d'obtenir plus de détails sur les performances des algorithmes. En effet, en divisant la base d'images en plusieurs familles nous avons déterminé, pour quatre familles d'images, les compresseurs les plus efficaces. Cet article apporte un nouveau modèle de compression nommé WJPEG2000, qui combine des techniques utilisées dans JPEG et dans JPEG2000. De plus, il introduit une nouvelle méthode de comparaison de standards, par l'utilisation d'une base de données fractionnée, pour obtenir des résultats plus détaillés lors de la comparaison de standards de compression.
Dans cette étude, la modélisation de WJPEG2000 est venue de D. Ziou. La supervision a été effectuée par D. Ziou et M.F. Auclair-Fortier. Enfin, la réalisation des expériences et l'utilisation d'une base d'image fractionnée sont dues à moi-même. Cet article a été soumis à la conférence Advanced Concepts for Intelligent Vision Systems 2011 (ACIVS) et est en cours d'évaluation.

Content Makes the Difference in Compression Standard Quality Assessment

Guido Manfredi, Djemel Ziou, and Marie-Flavie Auclair-Fortier

Centre MOIVRE, Universite de Sherbrooke, Sherbrooke(QC), Canada.
djemel.ziou@usherbrooke.ca

Abstract. In traditional compression standard quality assessment, compressor parameters and performance measures are the main experimental variables. In this paper, we show that the image content is an equally crucial variable which still remains unused. We compare JPEG, JPEG2000 and a proprietary JPEG2000 on four visually different datasets. We base our comparison on PSNR, SSIM, time and compression rate measures. This approach reveals that the JPEG2000 vs. JPEG comparison strongly depends on compressed images visual content.

1 Introduction

Still image number is increasing with the growing use of mobile phones, personal computers and digital cameras. They are stored, accessed, shared and streamed. However storage space and bandwidth are limited, that is why these images need to be compressed. Therefore several compression standards are widely used, such as JPEG, JPEG2000 and JPEG-XR (also know as HD-Photo format). However compression comes with side effects, like quality degradation and additional processing time. Given a specific application, it is hard to decide which standard should be used to minimize these drawbacks. To make a clear decision, users must rely on trial and error process or conclusions drawn from subjective quality assessments. The automatic version of such procedure is difficult to implement unless the subjective criteria used by human to assess image quality are replaced by objective criteria. Such assessment is carried out by using quality metrics, which calculation is fast and cost-effective, but less trustworthy [10].

A typical compression standard quality assessment experiment depends on an image dataset, compressor parameters and performance measures. The images in the dataset are characterized by the classes to which they belong. Compressor parameters can be algorithms such as JPEG, AVC/H.264 or options such as lossy or not, progressive or not. Finally, the performance measures are the criteria used to measure the influence of parameter sets on compressors [3, 4, 6]; they are for example quality measures, compressor complexity or functionalities (e.g. region of interest, coding or multiscaling).

Three central works [5, 8, 9], carried out between 2002 and 2008, in the field of compression standards objective quality assessment have influenced our work. In these three studies, the authors extensively tuned the compressors parameters

Table 1. Experimental variables of three existing works along with our's.

Work	Ebrahimi et al. [5]	De Simone et al. [9]	Santa Cruz et al. [8]	Our
Dataset name	Live database	Microsoft test set	JPEG2000 test set	Various
Dataset size	29	10	7	396
Classes	N/A	N/A	N/A	4
Compression algorithm compared	JPEG, JPEG2000	JPEG2000, H.264, JPEG-XR	JPEG, JPEG-LS, MPEG4-VTC, SPITH, PNG, JPEG2000	JPEG, JPEG2000, proprietary JPEG2000
Compression parameters	Compression rate, quality required, implementation	Compression rate	Compression rate, lossless/lossy, progressive or not	Compression rate, quality required
Performance measures	Blockiness, blur, MOS prediction, PSNR, compression rate	PSNR, SSIM	PSNR, error resilience, compression rate, speed	PSNR, SSIM, speed

and tried various criteria. They covered some of the most recent compression algorithms. They observed the effects of wavelets and cosine transform, of various coding algorithms like EBCOT, SPITH and Huffman and considered lossiness. They used various quality metrics and image artifacts as performance measures. Finally, they not only assessed compressors' speed but also error resilience (i.e. robustness to transmission error) and if it supports ROI coding and multiscaling. According to these studies, JPEG2000 and AVC/H.264 perform better than JPEG-XR. Moreover, JPEG2000 outperform JPEG in terms of image quality for medium and strong compression. Unfortunately, it lacks of speed when compared to JPEG. JPEG is the best algorithm for weak compression in terms of quality and speed. For lossless compression, JPEG-LS stands as the best choice regardless of the criteria. Table 1 summarizes experimental information about these works.

Those studies greatly improved our knowledge about the most popular compression algorithms. However, the datasets used in these experimentations do not exceed 30 images which cannot show the influence of image content on the compression process. Indeed, tests are made on a whole dataset and conclusions are drawn from the means of some performance measures.

In this work, we address these issues by adding the image content as a new criterion to the standard comparison framework. By varying the image classes, and using a greater number of images, we show that new results can be obtained. We give an example of using this framework with a well known and extensively studied comparison, the JPEG-JPEG2000 comparison. Only two standards are compared in order to keep the example simple, still it allows understanding the benefits of our approach. Moreover we propose to go beyond the standards spec-

ifications by adding the wavelets transform window size as a new parameter to JPEG2000. Varying images visual characteristics, in regard of the new criterion, reveals interesting results for medical image compression.
The next section will describe the experimental protocols used in our standards comparison along with three experiments realized following these protocols. The third section shows the results of the comparisons. Then, the fourth section offers a discussion about our methodology. We conclude by proposing an add-on to standards comparison frameworks.

2 Experimental Protocols

For quality assessment experiments, we propose to highlight the effects of image content according to compressor parameters and some performance measures. Because the measures will be the same for all experiments, let us first define them once and for all. Let us choose objective quality metrics. In order to draw conclusions coherent with the related works, we use the Structural SIMilarity (SSIM) [11] and the Peak Signal to Noise Ratio (PSNR). For a decompressed image X and a ground truth image Y, those metrics are defined as follow:

$$PSNR(X,Y) = 10\log_{10}(\frac{255^2}{MSE(X,Y)}),$$

where $MSE(X,Y)$ is the Mean Squared Error between X and Y.
The $SSIM(X,Y)$ is defined by

$$SSIM(X,Y) = \sum_{a\in X, b\in y} (\frac{(2\mu_{x(ab)}\mu_{y(ab)} + C_1)(2\sigma_{xy(ab)} + C_2)}{(\mu^2_{x(ab)} + \mu^2_{y(ab)} + C_1)(\sigma^2_{x(ab)} + \sigma^2_{y(ab)} + C_2)}),$$

where $C_1 = (0.01 \times 255)^2 = 6.5025$ and $C_2 = (0.03 \times 255)^2 = 58.5225$; $\mu_{x(ab)}$, $\sigma_{x(ab)}$ (resp. $\mu_{y(ab)}$, $\sigma_{y(ab)}$) and $\sigma_{xy(ab)}$ are respectively the mean, variance and covariance calculated over 8×8 pixels windows centered on (a,b) in image X (resp. Y). Although it has been shown that PSNR and SSIM are correlated [7], we need both of them for interpretation purposes. Indeed, when SSIM values are close to zero or one, we must use the PSNR to discriminate and vice versa. In addition to these metrics, we define the compression rate as,

$$CR(X) = \frac{\text{(size in bits of compressed image } X)}{\text{(size in bits of image } X)}.$$

We choose this definition of compression rate, instead of bpp (bit per pixel), because it is the one used in Jasper's implementation of JPEG2000. The lower the compression rate, the more compressed the image and vice versa. Finally, the computational time is used as performance measure. Tests were carried out on a 2.66GHz Pentium 4 with 1GB of RAM. We used the lossy compressors JASPER 1.900.1 [1] for JPEG2000, and libjpeg7 [2] for JPEG.

We claim that image content influences compressed image quality. The content of images can be defined by features such as contrast, textures, and shapes.

Table 2. Four visually different datasets used in this experiment.

Class	Sensor	Number of images	Resolutions (in pixels)	Frequency and spatial content
aerial	IR	38	512 × 512, 1024 × 1024, 2250 × 2250	High frequencies with little spatial extent.
outdoor	RGB camera	129	800 × 530	Highly textured and low frequencies with large spatial extent.
texture	Digital microscope camera	64	512 × 512, 1024 × 1024	Pure high frequencies.
medical	X-Rays	28	256 × 256, 768 × 575	Low frequencies with almost no high frequencies.

Table 3. Correspondance table between JPEG qualiy and JPEG2000 CR.

Class/ JPEG quality	6	25	50	75	100
aerial	0,009585	0,02603	0,042403	0,067592	0,357631
outdoor	0,012287	0,031723	0,049161	0,073795	0,334806
texture	0,046196	0,11628	0,177208	0,261664	0,919663
medical	0,009238	0,016283	0,027441	0,033954	0,094335

Different image types will have different quality loss during compression. Indeed, as some high frequencies are lost during the compression process, an image composed mainly of high frequencies will suffer a heavy quality loss. In order to explore this assumption, we gather four datasets. In our case, spatial and frequency content distinguish datasets' visual content. The whole dataset is formed by 259 uncompressed 24 bits depth color images, except for texture and medical images which are grayscale 8 bits depth. All color images are in the sRGB color space. Table 2 summarizes and describes the four classes used in this work.

The evaluation framework can be summarized as follows. We first compute the three measures at fixed Compression Rates (CR) for both JPEG and JPEG2000: PSNR, SSIM and computation time. For a given class we compute the mean PSNR obtained with JPEG2000 and JPEG over all images. Then we compute the difference between mean PSNR(JPEG2000) and mean PSNR(JPEG). If the difference is positive JPEG2000 performs better than JPEG, if the value is negative JPEG outperform JPEG2000.

We want to compare JPEG and JPEG2000 at the same compression rate. Note that JPEG compressor is not parameterized by a given compression rate but by the quality expressed in percentage. We choose five quality levels that are 6, 25, 50, 75 and 100. In order to use the same compression rates for JPEG and JPEG2000, from each of these quality values the compression rate is estimated and used for JPEG2000. However, as we have no direct control on compression rate, values are different for each class (Tab. 3). That is why, in Fig. 2, 3, 4 and 5, classes span different compression rate intervals. For the second experiment we compare JPEG2000 to a proprietary JPEG2000, named WJPEG2000

for Windowed JPEG2000. In order to obtain a greater spatial resolution, in WJPEG2000, the wavelet transform is windowed. We apply the wavelet transform on non-overlapping fixed size square windows (see Fig. 1). Unlike tiling procedure of JPEG2000, only the DWT step is ran separately for each window. The quantization and coding are performed on the whole resulting wavelet coefficients. The modification leads to a compromise between spatial and frequency resolutions. Indeed, windowing the transform increases its spatial resolution. On the other side, the transform blocks are smaller so they will have a decreased frequency resolution compared to a holistic transform.

The transform window size is a new parameter of the compressor. In order to highlight the impact of this parameter on compression, we compress our dataset with various window sizes from 2 to 64 pixels width.

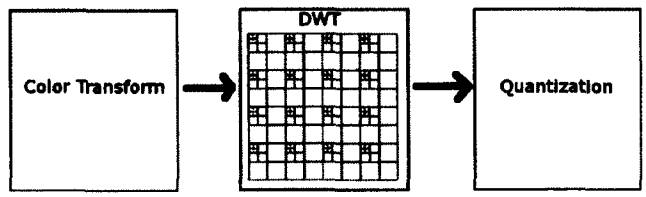


Fig. 1. The windowed DWT.

3 Experimental Results

According to Fig. 2, JPEG2000 reaches the best SSIM for high compression rate (> 0.2). Let us zoom on lower compression rates (Fig. 3). We see that the difference is hardly measurable for three out of four classes, their values are between 0.01 and -0.01. Even for aerial images, the difference is not far from 0.01. However, at these compression rates such differences are difficult to interpret. That is why we must use the PSNR for further analyze. For the rightmost points of the curves (Fig. 2), JPEG2000 outperforms JPEG, except for some points of the medical image class. At compression rates higher than 0.4, for texture class, JPEG outperform JPEG2000. Now let's focus on the compression rate interval 0 to 0.1 (Fig. 3) because the majority of image's compression rate are within such interval. As we can see, the maximum JPEG2000/JPEG difference in PSNR is of 8db. Only one point is over a difference of 6db, regardless of compression rate. Above a compression rate of 0.03, the maximum difference lowers to 3db. Table 4 sums up the results of this comparison. For compression rate between 0 and 0.03, JPEG2000 shows better quality than JPEG, except for highly textured image from texture class. At compression rates between 0.03 and 0.4, JPEG2000 outperforms JPEG, except for medical images. At compression rate superior than 0.4 and for textured images, JPEG has better quality than JPEG2000. Table

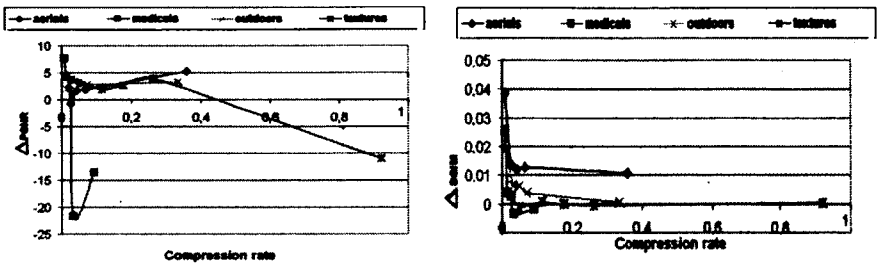


Fig. 2. JPEG2000/JPEG difference in PSNR and SSIM.

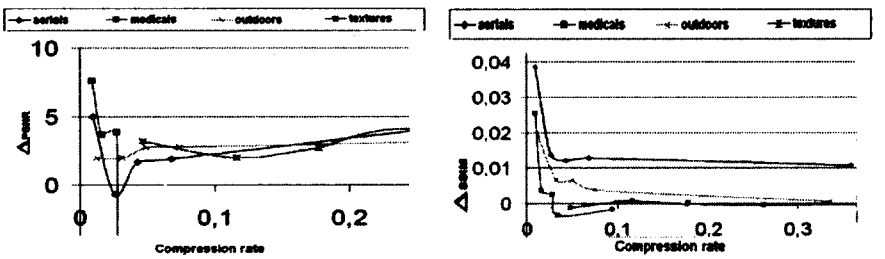


Fig. 3. Zoom of JPEG2000/JPEG difference in PSNR and SSIM in Fig. 2.

Table 4. Best compressor depending on compression rate and image content.

CR	0 - 0.03	0.03 - 0.4	0.4+
aerial	JPEG2000	JPEG2000	N/A
outdoor	JPEG2000	JPEG2000	N/A
texture	JPEG	JPEG2000	JPEG
medical	JPEG2000	JPEG	N/A

4 put forward the difference in the comparison depending on the chosen image class. However, these results show that each standard use a transform which will be effective for some type of data and less effective for others data.

Now let us compare WJPEG2000 with JPEG2000. Once more we observe differences in terms of PSNR and SSIM in order to see if the wavelet transform windowing affects the quality. The first experiment is carried out at a compression rate of 0.03. The PSNR difference is over 6db whatever the window size (Fig. 4).

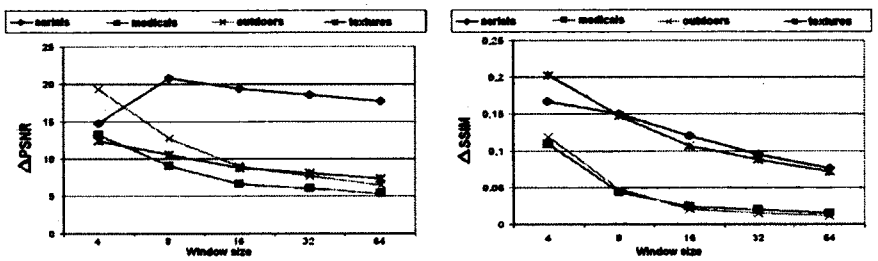


Fig. 4. JPEG2000/WJPEG2000 difference in PSNR and SSIM for CR = 0.03.

The SSIM is coherent with the PSNR analysis and shows that JPEG2000 gives better quality than WJPEG2000. Now, it becomes more interesting for higher compression rate. The next scores are obtained with a CR of 0.3. In Fig. 5 the difference in SSIM terms is too small to help us (< 0.01). Therefore we look at the PSNR. For medical images WJPEG2000 outperforms JPEG2000. The difference in terms of PSNR is mostly over 15db and reaches 18db for the window of 17 pixels width. For other classes, JPEG2000 outperform WJPEG2000 of more than 5db. Therefore JPEG2000 has better quality than WJPEG2000 for three classes out of four. However, for medical images, WJPEG2000 shows a huge improvement over JPEG2000. In fact, windowing the wavelets transform

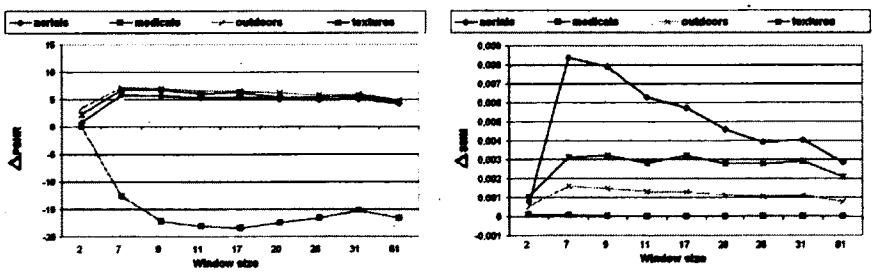


Fig. 5. JPEG2000 - WJPEG2000 difference in PSNR and SSIM for CR = 0.3.

greatly improves the compression of medical images. This provides an interesting direction to explore in further work. Indeed, the main feature of medical images is their smoothness.

We conclude this section pointing out the fact that the separation of our dataset in classes allowed us to see this result which would have gone unnoticed in a non classified dataset. Regarding the mean computational time, Fig. 6 shows

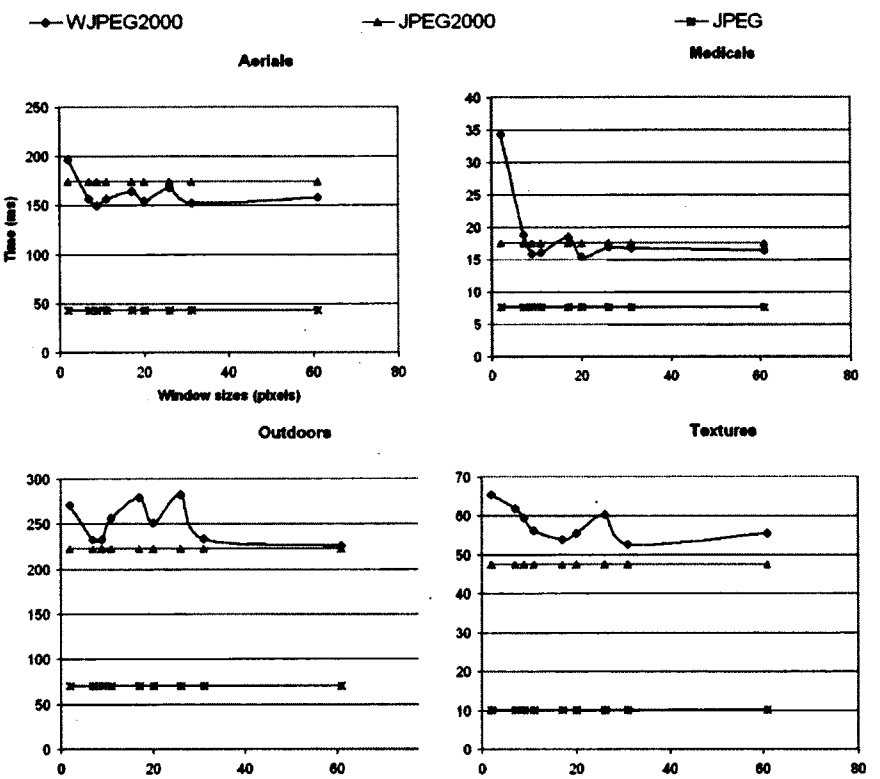


Fig. 6. Compression time for JPEG, JPEG2000 and WJPEG2000 for each class using CR = 0.3.

that for CR= 0.3, JPEG is the most efficient algorithm with a speed between two (medicals) and five times (textures) greater than those of JPEG2000 and WJPEG2000. WJPEG2000 is up to 1.14 times slower than JPEG2000.

For CR=0.03 (Fig. 7), results are similar. JPEG is two to five times faster than the others. WJPEG2000 is up to 1.36 times slower than JPEG2000. Note that the computational time is image content free.

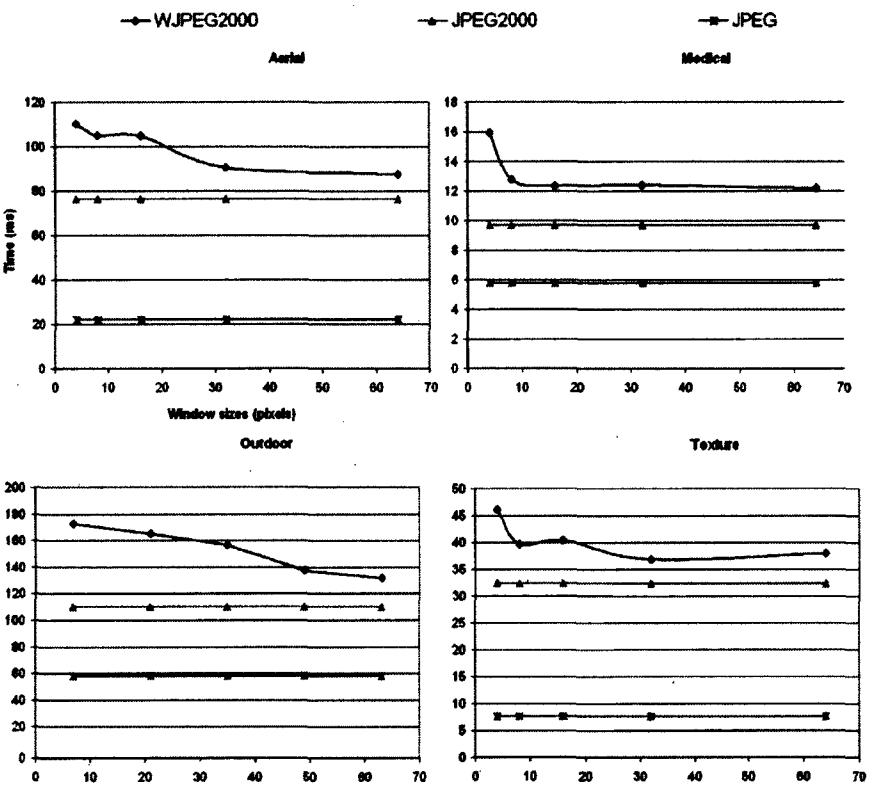


Fig. 7. Compression time for JPEG, JPEG2000 and WJPEG2000 for each class using CR = 0.03.

4 Conclusion

Most of compression standard comparisons do not take into account image content. In this paper, we advocate in favor of an add-on for the regular compression standard quality assessment framework. We use the image content as an additional criterion in the assessment. By splitting the dataset among various classes and analyzing each one independently, we provide some insight on the differences between JPEG2000 and JPEG. Further, this method allowed us to bring out an interesting result. Namely, that windowing the wavelet transform in JPEG2000 increases its quality in terms of PSNR and SSIM for medical images. Futur work includes gathering a larger and more structured dataset in order to compare various compression standards. Moreover, the windowing of a wavelet transform for compression purposes needs further exploration.

References

1. Jasper jpeg2000 codec. http://www.ece.uvic.ca/ mdadams/jasper/
2. Jpeglib jpeg codec. http://www.ijg.org/
3. A. J. Ahumada, J.: Computational image quality metrics: a review. In: SID International Symposium, Digest of Technical Papers. vol. 24, pp. 305–308 (1993)
4. Aycibas, I., Sankur, B., Sayood, K.: Statistical evaluation of image quality measures. Journal of electronic imaging 11, 206—223 (2002)
5. Ebrahimi, F., Chamik, M., Winkler, S.: JPEG vs. JPEG2000: an objective comparison of image encoding quality. Proceedings of SPIE applications of digital image processing 5558, 300308 (2004)
6. Eskicioglu, A.M.: Quality measurement for monochrome compressed images in the past 25 years. In: Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on. vol. 6, p. 1907 (2000)
7. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 20th International Conference on Pattern Recognition, 2010. Proceedings. vol. 4 (2010)
8. Santa-Cruz, D., Grosbois, R., Ebrahimi, T.: JPEG 2000 performance evaluation and assessment. Signal Processing: Image Communication 17(1), 113–130 (Jan 2002)
9. Simone, F.D., Ticca, D., Dufaux, F., Ansorge, M., Ebrahimi, T.: A comparative study of color image compression standards using perceptually driven quality metrics (2008), http://infoscience.epfl.ch/record/125933
10. Wang, Z., Bovik, A., Lu, L.: Why is image quality assessment so difficult? In: Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on. vol. 4, pp. IV–3313–IV–3316 vol.4 (2002)
11. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on 13(4), 600–612 (2004)

Chapitre 3

Détection de visages rapide dans des images compressées

La majorité des techniques de détection de visages se font sur une image non compressée, donc dans le domaine des pixels. Dans certaines situations, l'image sous forme de pixels n'est pas disponible, seule sa version compressée est accessible. Par ailleurs décompresser l'image impliquerait un temps de calcul et une exigence en quantité de mémoire qui seront parfois inacceptables. Il faut alors être capable de mener à bien la détection en utilisant des coefficients du domaine compressé. Dans le cas présent on s'intéresse au problème de la détection de visage à partir d'images JPEG compressées, en utilisant le détecteur de Viola-Jones. Le format JPEG utilise les coefficients de la transformée en cosinus discrète (DCT) faite sur des blocks de 8×8 pixels. Cela pose de nombreux problèmes pratiques lors de l'extraction de caractéristiques en particulier car les techniques de multi-résolution classiques ne sont plus utilisables. Pour éviter ces problèmes nous proposons d'utiliser une décompression spécialisée qui permet de décompresser les coefficients DCT et de les mettre directement en forme pour le calcul des caractéristiques. L'extraction de caractéristiques et l'ajustement de leur contraste demande de connaître la moyenne et la variance de la région d'extraction. Cette information peut être rapidement obtenue à partir des images intégrales simple et carrée. Les techniques de calcul des images intégrales se basent sur l'information des pixels. L'originalité de ce travail vient du fait qu'il propose une manière de calculer les images intégrales à partir de l'information des coefficients DCT. Ce faisant elle permet d'effectuer la détection de visages dans l'espace DCT. Dans cette étude, l'idée de traiter la détection de visages dans le domaine DCT et d'utiliser le détecteur de Viola-Jones vient de D. Ziou. La supervision a été faite par D. Ziou et M.F. Auclair-Fortier. Enfin, l'idée d'utiliser une décompression spécialisée, les méthodes de calcul utilisées pour ce faire et la réalisation des expériences sont dues à moi-même. Cette article a été soumis à la conférence Digital Signal Processing 2011 et est en cours d'évaluation par les organisateurs de la conférence.

FACE DETECTION IN A COMPRESSED DOMAIN

Guido Manfredi Djemel Ziou Marie-Flavie Auclair-Fortier

Centre MOIVRE
Universite de Sherbrooke
Sherbrooke(QC), Canada

ABSTRACT

We focus on the Viola-Jones face detector in the Discrete Cosine Transform (DCT). In order to avoid typical pitfalls due to features in the DCT domain we propose to merge the integral image stage with the decompression process, thus saving time for both operations. The proposed method saves 128 additions and 224 multiplications on an 8×8 block for a simple integral image. We propose a similar method for fast feature contrast adjustment by computing the squared integral image using DCT and simple integral image coefficients. These methods are faster and more transmission error resilient than classical methods.

Index Terms— DCT, integral image, Viola-Jones, face detection, compressed domain

1. INTRODUCTION

Face detection is the action of finding the spatial location of faces in an image. It is used as a preprocessing stage in many advanced face processing systems such as face recognition [1], expression recognition [2] and face tracking [3]. For practical purposes, face detection algorithms must be fast. Moreover, the precision of the whole system depends strongly on the detection's precision.

In order to achieve both speed and precision many approaches have been used. They can be classified into two groups: template based methods and training based methods. Template based methods use hypothesis on face proportions or morphological structure of the face [4]. They do not need training so they are easy to set up. However, their precision and robustness are limited [5]. Training based methods need thousands of images and a long training but they provide good scores with high detection rates and few false alarms. Moreover, depending on the database used for training, they can be robust to illumination, skin color or face orientation. These training based methods can be further divided into two groups: implicit features based methods and explicit features based methods. Using implicit methods, it is not apparent which set of features has been used, as in the case of neural networks [6]. Using explicit methods, only a subset of the total feature pool defines the face model. This is the case for classifier cascades [7, 8]. The feature set used is another important element in face detection techniques. Their choice is crucial as they hold the discriminative power between face and "non-face" regions. The most popular features are histograms [9], Haar-like features [7], Local Binary Patterns (LBP) [10] and coefficients from various transforms like wavelets [7] or Fourier transforms [8].

Because of its high scores, we choose to work on the Viola-Jones face detector, a training based method which uses explicit Haar-like features. Usually, this algorithm is implemented in spatial domain, and it runs on uncompressed images or videos [7]. However, most images are in compressed format. These compressed formats mainly use two types of transforms: the wavelets transform or the Discrete Cosine Transform (DCT). In order to save time, some authors proposed to discard the decompression stage and compute face detection directly on the transform coefficients [8]. However, working in transform spaces imposes some constraints on the processing such as finding adapted features or spatial precision issues.

The main contribution of our work is a solution to avoid these problems using a specialized decompression scheme instead of transform domain features. As it is largely spread through JPEG, we focus on the 8×8 blocks DCT transform, which will be referred for simplicity as DCT in the following. The idea is to merge the integral image step of Viola-Jones algorithm with the DCT decompression. This approach brings two gains: a complexity reduction due to the merge of two steps in one and the possibility of working from DCT coefficients without classical limitations. This paper is organized as follows. In the next section we will see the constraints of implementing Viola-Jones detector in the DCT domain and how to bypass these issues by merging the decompression and the integral image stage. In section 3, we present the mathematical theory behind this fusion. Section 4 shows a complexity comparison between the classical Viola-Jones detector and our DCT Viola-Jones detector. We conclude by proposing a way to improve this work by the factorization of the transform.

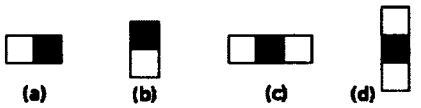


Fig. 1. The feature values are computed by subtracting the sum of pixels over the dark area from the sum over the white area.

Table 1. Number of features for each type, in a 24 × 24 pixel window.

Feature type	Base size (w/h)	Maximum scale factor (S_w/S_h)	Number of features
Fig. 1(a)	2/1	12/24	43,200
Fig. 1(b)	1/2	24/12	43,200
Fig. 1(c)	3/1	8/24	27,600
Fig. 1(d)	1/3	24/8	27,600

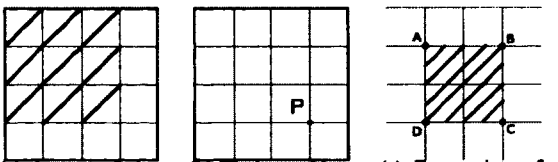
2. FEATURES IN A COMPRESSED DOMAIN

The Viola-Jones detector owes its performances to an ingenious combination of techniques [7]. It starts with the feature extraction stage. The algorithm slides windows of various sizes across the image. For each window, Haar-like features [7] are extracted from the region. These features are computed by luminance differences between areas (Fig. 1). This stage is made of four iteration loops: on the feature scale along width, on the feature scale along height, on the feature position along width and on the feature position along height. This process is done for each feature type. In one sentence, each feature type is computed at each pixel of the window and at each possible scale.

For a window of size $W \times H$ and for a feature type of base size $w \times h$, the total number T of features is

$$T = S_w \times S_h(W + 1 - w\frac{S_w + 1}{2})(H + 1 - h\frac{S_h + 1}{2}),$$

where $S_w = W/w$ and $S_h = H/h$ are respectively the maximum scale factor along the width and the height. The first term means that features are extracted at each scale. The second term represents the extraction at each pixel position along the window's width, except some positions for which the feature is too large to fit in. The third term represents this same quantity along the window's height. Table 1 lists the number of features for each feature type in a 24 × 24 pixels window. This yield 141,600 different features. In order to allow fast computation of this large feature set, Viola and Jones [7] propose a tool called integral image. The integral image allows fast computation of sums of pixels over an area. It is to say each value of the integral image represents the mean luminance between the considered pixel and the origin of the image (Fig. 2(a) and 2(b)).



(a) An area of pixels (b) The integral image. (c) Four points of the integral image.

Fig. 2. The value of P, in 2(b), is equal to the sum of pixels over the dashed area in 2(a). The sum of pixels S over any area, in 2(c), can be computed with four points of the integral image: $S = C + A - B - D$.

With the values of four points of the integral image, the mean luminance over any area of any size can be known (Fig. 2(c)). This is useful for extracting mean luminance related features at constant time regardless of the scale [7]. Indeed, a combination of four integral image coefficients gives the sum of pixels over an area. For a type (a) feature (*cf.* Fig. 1), at a given scale and position, four coefficients are needed for the dark area and four more for the white one. So, eight integral image coefficients combined give the value of one feature. More details on the integral image will be provided in the next section.

Feature extraction is the first step of the algorithm. A learning process allows to use only a subset of the total feature set. The values of these particular features are extracted and compared to learned values in the classification state. The region final label, face or "non-face", depends on a combined decision from a cascade of classifiers called the attentional cascade. The first classifier extracts some features. If these features correspond to a face, the region is analyzed by the next classifier. If not, the region is classified as "non-face". If all classifiers decide that a region is a face, then it is labeled as a face. This attentional cascade concentrates the computational effort on the regions which hold faces most likely. Indeed, most of "non-face" regions are discarded by the first classifiers, so few features are analyzed for these regions. Moreover, this architecture multiplies the detection and false alarm rate of each classifier thus allowing a high level of precision with simple features. For example, if there are ten classifiers with each a detection rate of 99% and a false alarm rate of 50% then the total detection rate is $0.99^{10} = 0.9$ (=90%) and the false alarm rate is $0.5^{10} = 0.001$ (=0.1%). The Viola-Jones detector runs on uncompressed images. That is why, when processing a compressed image, there is a decompression stage before the face detection stage. However, working with compressed information can sometimes be convenient, for example to avoid a costly decompression.
The aim of this work is to allow the Viola-Jones detector to successfully process images in the DCT domain. In order to work in the compressed domain we could simply delete the decompression. This means that the detector would have to

work with compressed domain features. We did not choose this strategy since various underlying problems have to be tackled.

First of all, the 8×8 blocks DCT offers a good frequency resolution but a low spatial resolution, which is divided by eight along the two dimensions in comparison with the spatial domain. This means that an image cannot be processed pixel by pixel; it must be processed by 8×8 pixel blocks. The consequences are precision problems. If the window can only be moved eight pixels by eight pixels, some faces can remain unnoticed. Moreover, the window needs to be scaled in order to detect larger faces. Again, the scaling can be done at least by a multiple of eight. If the window is too small compared to faces size, the considered part of the face will be too small for detection. On the other hand, if the window is too large, background will introduce noise which may prevent detection.

Even when a face is detected, its location would have a eight pixel uncertainty. Sometimes a face is present among some blocks, but its exact position is unknown. This problem can be solved by decompressing only the detected blocks. In the case of a little face in a large image the complexity will be greatly reduced as only a small part of the image will be decompressed. In the case of a large face it will be equivalent to decompressing the whole image, so no gain is obtained.

The last, and maybe the most important problem is the scale invariance of features. Indeed, to locate faces of various scales, features must be scale invariant. In spatial domain, features of one scale can easily be compared with larger or smaller features. However, in the DCT domain this rule does not work. Finding a scale invariant feature in the DCT is a complex problem. Solutions were proposed in [11, 8]. However these solutions result in using only the first coefficients of the transform, which can be processed in the same way as pixels.

Using features in the DCT domain poses many unsolved problems. Therefore we suggest bypassing them with a different approach. Shen *et al.* [12] merge a convolution process with the inverse DCT (IDCT) stage rather than using a blind decompression, thus obtaining a specialized decompression for fast convolution. We propose to proceed in a similar way by merging the integral image and the IDCT, thus saving costs in both operations. This results in a specialized decompression for fast integral image computation. As we operate upstream of the sliding window operation, spatial precision problems are avoided. Furthermore, the merged operations provide an integral image identical to the one obtained by a spatial domain integral image, so spatial features can be used.

3. FEATURE EXTRACTION FROM DCT COEFFICIENTS

As seen in the last section, the Viola-Jones detector uses the integral image for fast feature extraction. Please note that the integral image coefficients are not features, they are tools for fast multiresolution feature computation. In Viola-Jones detector, the integral image is computed from pixels. However, as mentioned earlier, sometimes it is convenient to be able to compute the integral image in compressed domain. The first contribution of this work is a method to compute the integral image from DCT coefficients. This method is described in the next subsection.

In the classical Viola-Jones detector, features suffered from contrast differences between different extraction windows. To overcome this problem, Lienhart *et al.* [13] propose a contrast stretching method. It introduces a normalization process for features. Their values are normalized depending on the window they are extracted from. If F is a feature value, then the normalized feature F_n is defined as

$$F_n = \frac{F - \mu}{2\sigma},$$

where μ and σ are respectively the mean and variance luminance over the extraction window. So, two unknowns, μ and σ, must be estimated in order to normalize a feature. The mean μ can be obtained from the simple integral image. However, the variance cannot be obtained from the integral image alone. Indeed,

$$\sigma = \epsilon - \mu^2,$$

where ϵ is the mean energy over a region. For fast computation of ϵ, a variant of the integral image is needed: the squared integral image. This one represents the energy over different regions at different scales. Once again, it can be convenient to do this normalization step in the compressed domain. As we will see further, the second contribution of this work is a method which allows computation of the squared integral image from DCT coefficients instead of pixels.

In what follows, we present two methods to compute the simple and squared integral images from DCT coefficients.

3.1. Simple integral image

The integral image is a fast way to compute a convolution at different scales and at constant time. When convolving an image with a filter we have the following operation:

$$h = I * f,$$

where h is the product of the convolution, I is the image and f is a filter. A linear operation can be applied to I if its inverse is applied to f. Because the integral operation is linear, if we can integrate I and derive f we obtain the same result h. Then,

$$h = \left(\int I\right) * f',$$

where f' denotes the derivative of f. This operation is very convenient when the derivative of f is sparse and can be easily computed. This is the case when f is a uniform rectangular

filter. For such a filter, its derivative is a sum of four impulse functions. So the convolution result can be computed in four array references, regardless of the size of the convolution mask. This result is used in the Viola-Jones detector to compute the mean luminance over an area at different scales. Indeed, as denoted earlier, for a one-dimensional vector of size n, we have

$$\mu = \frac{h_n}{n},$$

where μ is the mean value of the vector and h_n is the n^{th} integral image coefficient. Indeed, h_n represents the sum of all pixels over the vector. What follow shows how to obtain the integral image coefficients from DCT coefficients. For simplicity, in the remaining of this section, the problem is considered in one dimension. It can be easily extended to two dimensions as the DCT and the integral image are both separable transforms.

Consider a vector of pixels f. Each pixel value f_k is denoted by its position k. The goal is to compute the integral image coefficients h_i, where i is the position in the integral image vector, such as

$$h_i = \sum_{k=0}^{i} f_k. \tag{1}$$

However, since the image is in the DCT domain, the coefficients of the transform are the only information available. As the considered DCT works on eight pixels windows, let us consider the image by blocks of eight pixels. We will see further how to generalize to various blocks. For a eight pixels vector, the relationship between DCT coefficients and pixels is

$$f_k = \sum_{l=0}^{7} \gamma(l) C_{k,l} F_l \tag{2}$$

where k is the spatial position, l is the frequency position, F_l is a DCT coefficient value, $\gamma(l) = 1/\sqrt{8}$ if $l = 0$ and $1/2$ otherwise, and $C_{k,l} = \cos\left(\frac{(2k+1)l\pi}{16}\right)$. Combining Equations (1) and (2) we obtain a direct relationship between integral image coefficients and DCT coefficients:

$$h_i = \sum_{l=0}^{7} \gamma(l) W_{i,l} F_l, \tag{3}$$

where $W_{i,l} = \sum_{k=0}^{i} C_{k,l}$. This means the integral image can be obtained from the cosines of the DCT transform. Moreover, $h_7 = \sqrt{8} F_0$. Which allows us to rewrite Equation (3) as

$$h_i = \frac{i+1}{8} h_7 + \sum_{l=1}^{7} \gamma(l) W_{i,l} F_l.$$

It can be written in a recurrent form,

$$h_i = \begin{cases} h_{i+1} - \frac{h_7}{8} - \sum_{l=1}^{7} \gamma(l) C_{i+1,l} F_l & \forall\, i \in [0,6] \\ \sqrt{8} F_0 & \text{if } i = 7. \end{cases} \tag{4}$$

Equation (4) is valid inside an eight pixel block. In order to calculate the integral image over various blocks, the DC coefficient of the previous block must be added to the one of the currently considered block. Noting h_{i7} the seventh coefficient and F_{i0} the DC coefficient of the i^{th} block, $h_{i7} = \sqrt{8} \sum_{n=0}^{i} F_{n0}$. This is equivalent to computing the integral image on the DC coefficients. Afterwards, the DC coefficient gives h_7. The other values of a block are derived from h_7. This allows our method to obtain the same result as a classical integral image made on the whole vector.

3.2. Squared integral image

In the upgraded face detector proposed by R. Lienhart [13], features need to be contrast-adjusted. This process asks for the variance over the extraction window. As the extraction window can have different sizes, the squared integral image is used for fast calculation of variance over windows of different sizes. Indeed, as seen earlier, the variance σ can be obtained from the mean and the energy of a region. The mean is obtained from the simple integral image, the energy can be obtained from the squared integral image as follows:

$$\epsilon = \frac{H_n}{n^2},$$

where ϵ is the energy over a vector of size n and H_n is the n^{th} squared integral image coefficient of this vector. By computing the squared integral image, one can retrieve the variance over windows of various scales for fast contrast adjustment. In this section we show how the squared integral image can be obtained from the DCT coefficients. The basic formula of the squared integral image is

$$H_i = \sum_{k=0}^{i} f_k^2. \tag{5}$$

For a block size of eight pixels, H_7 can be expressed as a function of the DCT coefficients:

$$H_7 = \sum_{l=0}^{7} F_l^2. \tag{6}$$

Moreover, energy conservation in DCT gives the following relationship

$$\sum_{l=0}^{7} F_l^2 = \sum_{k=0}^{7} f_k^2. \tag{7}$$

With this term, in a similar way as for the simple integral image, we can compute the other terms of the block. The coefficients of the simple integral image allow simplifying Equa-

tion (5),

$$\begin{aligned} H_i &= \sum_{k=0}^{i} f_k^2 \\ &= H_{i+1} - f_{i+1}^2 \\ &= H_{i+1} - (h_{i+1} - h_i)^2. \end{aligned}$$

By processing the integral images from DCT coefficients the face detection process is made possible from compressed domain coefficients. As only the integral image is modified, and as it is the first step of the face detection process, the following operations remain unchanged. To summarize, the integral image is first computed only on the DC coefficients of each block, over the whole image. Then the coefficients of each block are derived from the DC coefficient. This results in the integral image. Then a window of varying size is slided along the image. For a given window, features are extracted at each possible position and scale. Given a window, a position and a scale, features are extracted by combining simple integral image coefficient. Finally, the simple and squared integral image coefficients give the mean and variance over the extraction window for feature normalization.

4. COMPLEXITY ANALYSIS

In order to highlight the beneficial effects of our approach the following section provides a complexity analysis of the classical Viola-Jones scheme against the DCT Viola-Jones provided by our method.

Let us start with the complexity of the classical scheme which includes a decompression and an integral image computation. For a eight pixels vector, this imply 56 additions and 64 multiplications for the inverse DCT and 1 addition per pixel for the integral image. This represents a total of 64 additions and 64 multiplications. For an 8×8 pixel block, the cost is multiplied by 16. Indeed, the operation is repeated eight times along the rows and eight times along the columns, which yields for a $n \times n$ pixels image

$$A = \frac{n \times n}{8 \times 8} \times 16 \times 64 = 16n^2,$$
$$M = \frac{n \times n}{8 \times 8} \times 16 \times 64 = 16n^2,$$

where A and M are the number of additions and multiplications required. For the squared integral image, a eight pixels vector requires 56 additions and 64 multiplications per block for the inverse DCT plus 1 addition and 1 multiplication per pixel for the integral image. This results in 64 additions and 72 multiplications. For a $n \times n$ image this yields

$$A = \frac{n \times n}{8 \times 8} \times 16 \times 64 = 16n^2,$$
$$M = \frac{n \times n}{8 \times 8} \times 16 \times 72 = 18n^2.$$

Table 2. Number of multiplications for the classical and DCT Viola-Jones integral image stage, for a $n \times n$ pixels image.

Integral image type		Classical	DCT
Simple	A	$16n^2$	$14n^2$
	M	$16n^2$	$12.5n^2$
Squared	A	$16n^2$	$5.5n^2$
	M	$18n^2$	$3.75n^2$

Now we do the same complexity analysis for the DCT Viola-Jones. For a one-dimensional eight pixels vector, the computational complexity is eight additions and seven multiplications for seven pixels of the vector. Plus one addition to update the DC coefficient and a multiplication per vector to obtain h_7. This gives a total of 56 additions and 50 multiplications per vector. For a $n \times n$ pixels image, the calculus is as follows:

$$A = \frac{n \times n}{8 \times 8} \times 16 \times 56 = 14n^2,$$
$$M = \frac{n \times n}{8 \times 8} \times 16 \times 50 = 12.5n^2.$$

For the squared integral image over a eight pixels vector, eight multiplications and seven additions give H_7, other pixels are computed with two additions and one multiplication. A total of 22 additions and 15 multiplications per vector allow computing the squared integral image using only DCT and simple integral image coefficients. For a $n \times n$ pixels image the cost is of

$$A = \frac{n \times n}{8 \times 8} \times 16 \times 22 = 5.5n^2,$$
$$M = \frac{n \times n}{8 \times 8} \times 16 \times 15 = 3.75n^2.$$

This approach for integral image is more efficient than the classical scheme based on decompression followed by integral image. Table 2 sums up the performances of both methods presented here. Apart from the complexity reduction, our method is more robust against noise from transmission, for example during a streaming. Indeed, each block is decompressed independently from the others. So if an AC coefficient get corrupted, only the values of its block will be contaminated, but others blocks will not suffer from this detrimental effect.

5. EXPERIMENTAL RESULTS

In order to confirm the complexity analysis, the described methods has been tested on a set of 100 images. Ten images have been taken randomly from the internet. Their widths equals their heights and both are multiple of eight, for implementation reasons. The initial 10 images have been resized to ten different resolutions ranging from 32×32 to 1024×1024 pixels. Indeed, the speed of our method is only resolution

dependent. So, for a given resolution the speed will be the same whatever the image content. Fig. 3(a) and 3(b) show the mean speed in milliseconds, for a given resolution in pixels. The results confirm the complexity analysis for an image size superior to 256 × 256 pixels. The greater the resolution, the greater the speed gain when using our method.

6. CONCLUSION

In this work, we presented the problem of features in DCT domain and showed an original method to avoid them. The presented technique allows saving operations compared to a naive decompression algorithm. Though, the IDCT has been improved [14] and advanced decompression techniques allow decompression with 29 additions and 11 multiplications for a eight pixels vector. These works use particular factorizations of the DCT transformation matrix to improve speed. However the $W_{i,l}$ terms derived in this article are very similar to the ones of the classical transform $C_{i,j}$. The transformation matrix derived from the $W_{i,l}$ has the same structure and symmetries as the classical DCT matrix. Future works will try to show that the same factorization used in [14] can be used with the terms derived in this work, thus further increasing the speed up.

7. REFERENCES

[1] Ziad M. Hafed and Martin D. Levine, "Face recognition using the discrete cosine transform," *Int. J. Comput. Vision*, vol. 43, no. 3, pp. 167–188, 2001.

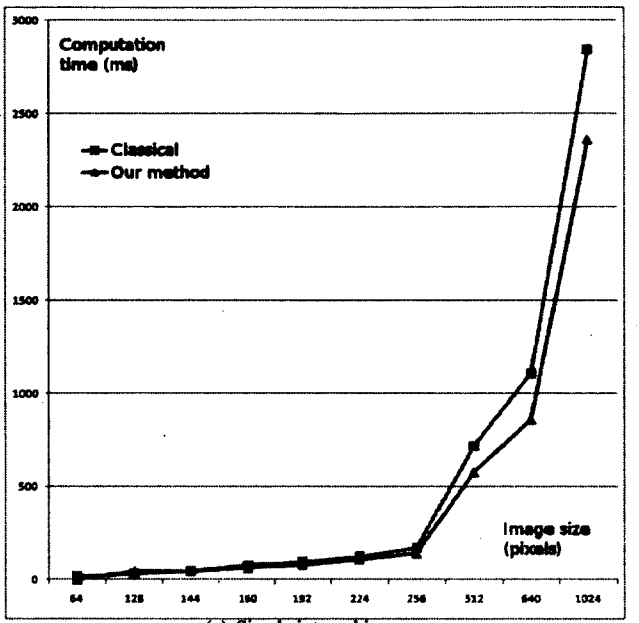
[2] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, and Javier R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction.," *Computer Vision and Pattern Recognition Workshop*, vol. 5, pp. 53, 2003.

[3] Ragini Choudhury Verma, Cordelia Schmid, and Krystian Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1215–1228, 2003.

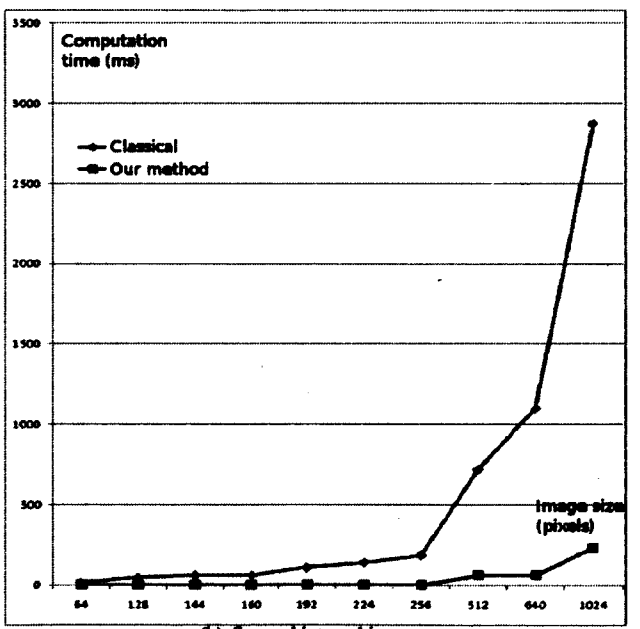
[4] Satyanadh Gundimada and Vijayan Asari, "Face detection technique based on rotation invariant wavelet features," *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 2, pp. 157–158 Vol.2, 2004.

[5] Erik Hjelms and Boon Kee Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, Sept. 2001.

[6] H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on*



(a) Simple integral image.



(b) Squared integral image.

Fig. 3. Results of classical method and our method for integral image computation.

Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[7] P. Viola and M. Jones, "Robust real-time face detection," *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 747, 2001.

[8] Chen Lei and Zhou Guo-fu, "A detection strategy of multi-pose face in compressed domain," *Wuhan University Journal of Natural Sciences*, vol. 9, no. 5, pp. 845–850, 2004.

[9] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR(Cat.*, Hilton Head Island, SC, USA, 2000, pp. 746–751.

[10] Yann Rodriguez, *Face Detection and Verification using Local Binary Patterns*, Ph.D. thesis, Ecole Polytechnique Fdrale de Lausanne, 2006.

[11] P. Fonseca and J. Nesvadha, "Face detection in the compressed domain," *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 3, pp. 2015–2018 Vol. 3, 2004.

[12] Bo Shen, I.K. Sethi, and V. Bhaskaran, "DCT convolution and its application in compressed domain," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 8, pp. 947–952, 1998.

[13] Rainer Lienhart and Jochen Maydt, "An extended set of Haar-Like features for rapid object detection," in *IEEE ICIP 2002*, 2002, pp. 900–903.

[14] C. Loeffler, A. Ligtenberg, and G.S. Moschytz, "Practical fast 1-d dct algorithms with 11 multiplications," *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pp. 988 –991 vol.2, may. 1989.

Conclusion

Le but de ce projet était de déterminer le domaine compressé le plus approprié pour la détection de visage et de montrer que l'algorithme de Viola-Jones peut fonctionner sur des images compressées. Bien que la première approche de trouver des caractéristiques propres au domaine compressé n'ai pas abouti, une solution répondant au problème de la détection de visage à partir d'une image compressée a été proposée.

Bien que non cité dans le chapitre deux, à cause de problèmes expérimentaux, durant ces travaux, le lien entre la qualité des images et la détection automatique de visages a été mis en évidence et un format s'est avéré de meilleure qualité pour réaliser la détection à partir de données compressées : le format JPEG. Nous avons, montré qu'il est possible d'éviter les principaux problèmes de la détection en domaine compressé grâce à la fusion de la décompression et de la première étape de la détection. Ce faisant, une méthode a été proposée qui permet de détecter des visages dans les images compressées au format JPEG à l'aide du détecteur de visages de Viola-Jones.

La méthode proposée a l'avantage d'impliquer des modifications minimes dans la chaine de traitement résultante, toute en permettant la détection de visages à partir d'informations compressées.

Bibliographie

[1] M. BINDEMANN et A.M. BURTON.
« The Role of Color in Human Face Detection ».
Cognitive Science, 33(6):1144–1156, 2009.

[2] F. FLEURET et D. GEMAN.
« Fast Face Detection with Precise Pose Estimation ».
Proceedings of ICPR2002, 1:235—238, 2002.

[3] P. FONSECA et J. NESVADHA.
« Face detection in the compressed domain ».
Image Processing, 2004. ICIP '04. 2004 International Conference on, 3:2015–2018 Vol. 3, 2004.

[4] L. GARRIDO, B. DUCHAINE et K. NAKAYAMA.
« Face detection in normal and prosopagnosic individuals ».
Journal of Neuropsychology, 2(Pt 1):119–140, mars 2008.
PMID : 19334308.

[5] S. GUNDIMADA et V. ASARI.
« Face detection technique based on rotation invariant wavelet features ».
Information Technology : Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, 2:157–158 Vol.2, 2004.

[6] J.V. HAXBY, E.A. HOFFMAN et M.I. GOBBINI.
« The distributed human neural system for face perception ».
Trends in Cognitive Sciences, 4(6):223–233, juin 2000.
PMID : 10827445.

[7] R-L HSU, M. ABDEL-MOTTALEB et A.K. JAIN.
« Face detection in color images ».

IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):696–706, 2002.

[8] C. LEI et Z. GUO-FU.
« A detection strategy of multi-pose face in compressed domain ».
Wuhan University Journal of Natural Sciences, 9(5):845–850, 2004.

[9] M.B. LEWIS et A.J. EDMONDS.
« Face detection : Mapping human performance ».
Perception, 32(8):903–920, 2003.

[10] R. LIENHART, E. KURANOV et V. PISAREVSKY.
« Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection ».
In DAGM 25th pattern recognition symposium, pages 297—304, 2003.

[11] C. PAPAGEORGIOU, M. OREN et T. POGGIO.
« A General Framework for Object Detection ».
Proceedings of the Sixth International Conference on Computer Vision, page 555, 1998.

[12] C.A. PEREZ et J.I. VALLEJOS.
« Face Detection using PSO Template Selection ».
Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on, 5:4220–4224, 2006.

[13] Y. RODRIGUEZ.
« Face Detection and Verification using Local Binary Patterns ».
(79), 0 2006.
PhD Thesis #3681 at the Ecole Polytechnique Federale de Lausanne.

[14] H.A. ROWLEY, S. BALUJA et T. KANADE.
« Neural network-based face detection ».
IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23–38, janvier 1998.

[15] H. SCHNEIDERMAN et T. KANADE.
« A statistical method for 3D object detection applied to faces and cars ».
Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, 1:746 –751 vol.1, 2000.

[16] K. SHINJIRO et T. NOBUJI.
« Real-Time Detection of Between-the-Eyes with a Circle-Frequency Filter. ».
IEICE Transactions on Information and Systems, Pt.2 (Japanese Edition), J84-D-2(12):2577–2584, 2001.

[17] C. SIAGIAN et L. ITTI.
« Biologically-Inspired Face Detection : Non-Brute-Force-Search Approach ».
Computer Vision and Pattern Recognition Workshop, 5:62, 2004.

[18] P. VIOLA et M. JONES.
« Robust real-time face detection ».
Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, 2:747, 2001.

[19] M. YANG, D. ROTH et N. AHUJA.
« A SNoW-Based Face Detector ».
Advances in Neural Information Processing Systems 12, pages 855–861, 2000.