



6, bd maréchal Juin  
F-14050 Caen cedex 4

Spécialité Informatique  
2<sup>e</sup> année



Prooftag SAS  
20, Place Prax-Paris  
F-82 000 MONTAUBAN - FRANCE

**ENSICAEN**

Rapport de stage de formation

---

Morphologie floue et transformée  
en ondelettes sur des images de  
Codes à bulles

---

Cédric BLANCHARD

Suivi entreprise :  
Jean-Marc GAUDIN

Mai-Août 2007

Suivi entreprise :  
Jean-Marc GAUDIN

Mai-Août 2007

# Table des matières

Introduction.....	1
1 Environnement de développement et de recherche.....	2
1.1 MFC .....	2
1.2 Algorithme annexes.....	2
2 Approche Morphologie floue.....	3
2.1 Présentation succincte.....	3
2.2 Principe et combinaisons des opérateurs morphologiques de base.....	4
2.3 Choix de l'élément structurant.....	4
2.4 Choix des normes et du complément.....	4
2.5 Applications des ouvertures et fermetures.....	5
3 Approche décomposition et transformée en ondelettes.....	7
3.1 Présentation succincte.....	7
3.2 Choix de l'ondelette.....	8
3.3 Utilisations.....	9
3.3.1 Moyennes et écarts types.....	9
3.3.2 Comparaison avec des textures connues.....	9
3.3.3 De l'utilisation des basses fréquences.....	10
3.3.4 Détermination des centres et rayons des bulles par leurs contours.....	11
4 Utilisation couplée des deux approches.....	13
4.1 Rôles possibles et conséquences de chaque opération.....	13
4.2 Interprétation floue.....	13
5 Élaboration d'une signature.....	15
5.1 Traitements successifs retenus.....	15
5.2 Stockage du résultat sous forme d'une signature.....	17
5.3 Calculs de distances entre les signatures.....	17
5.4 Évaluation de cette solution.....	18
5.5 Réflexion sur la stabilité des signatures.....	19
Conclusion.....	21
Références bibliographiques.....	22

# Introduction

Un Codes à bulles est un code physique réalisé de façon totalement chaotique et non-reproductible, et permettant l'authentification certaine d'un objet ou document. Ce code unique est le produit de la société Prooftag, filiale de Novatec. Le contrôle du code est visuel ou réalisé grâce à un lecteur qui capture des images et compare les codes avec ceux enregistrés dans une base de données.

C'est sur ces images capturées que se porte notre recherche. Le but est ici de trouver différents traitements permettant d'extraire, à partir de ses images, une signature unique pour chaque code rencontré et qui sera donc invariante, quelle que soit l'image prise d'un même code. Pour cela, nous avons employé la morphologie floue et la théorie des ondelettes, deux techniques utilisées aujourd'hui dans de nombreux secteurs du traitement d'image comme la compression, la reconnaissance d'empreintes digitales ou l'analyse de textures.

Ces deux aspects furent inclus dans la réalisation d'une seule et même application de traitement, développée en C++ à l'aide des MFC, et regroupant ainsi l'ensemble des fonctions des deux types ainsi que d'autres fonctions d'ordres plus générales.

# 1 Environnement de développement et de recherche

## 1.1 MFC

L'application qui a été développée au cours de ce stage utilise l'outil MFC qui accompagne Visual C++ et permet le contrôle et la réalisation de l'interface graphique. Il a donc fallu dans un premier temps, grâce à des recherches documentaires et navigations internet, comprendre le fonctionnement des MFC et apprendre à les manipuler afin de construire le programme, application MDI (i.e qui comporte plusieurs fenêtres « filles », une par image ouverte, et un menu général dont les items s'appliquent dans la plupart des cas uniquement à la fenêtre fille active).

Au fil du développement, de nouvelles fonctionnalités des MFC furent donc ajoutées afin d'avoir un outils agréable et efficace pour la recherche de solution. Ces fonctionnalités sont par exemple les barres de défilement sur les images trop grandes, le menu contextuel par clic-droit, les accélérateurs (raccourcis clavier), etc...

## 1.2 Algorithme annexes

Un autre aspect qui fut abordé est le codage d'algorithmes annexes, qui sortent du cadre des MFC et des deux théories de notre recherche.

Il fut notamment réaliser les fonctions de gestion du presse-papier Windows afin de permettre une bonne communication entre le programme et d'autres outils potentiellement utiles.

Une part importante fut également consacrée à la communication avec une base de données My-SQL (locale ou distante). Il est ainsi possible de charger des images à partir d'une base mais également d'intervenir sur cette base afin d'ajouter des résultats tels que des signatures calculées.

Enfin, quelques fonctions élémentaires du traitement d'image classique furent implémentée, dans le but de minimiser l'utilisation d'outils externes (tels que la Demo de CXImage, permettant diverses opérations sur les images). En effet, une dépendance trop forte entraînait dans le processus de recherche des transferts de données répétitifs entre les applications et empêchait par la même occasion de correctement évaluer les temps des traitements réalisés. Il fut donc ajouté , par exemple, des fonctions de tracer de segments (algorithme de Bresenham), de normalisation d'histogramme (algorithme déjà présent dans Cximage) ou encore de segmentation d'images binaires et de binarisation par hystérésis (fonctions personnelles).

## 2 Approche Morphologie floue

### 2.1 Présentation succincte

Comme l'explique [1], la morphologie floue consiste à effectuer une opération proche de la convolution entre une image en niveaux de gris et un filtre flou, appelé aussi élément structurant et composé également de niveau de gris. Cette pseudo-convolution repose sur des comparaisons entre les pixels du filtres et les pixels de l'image.

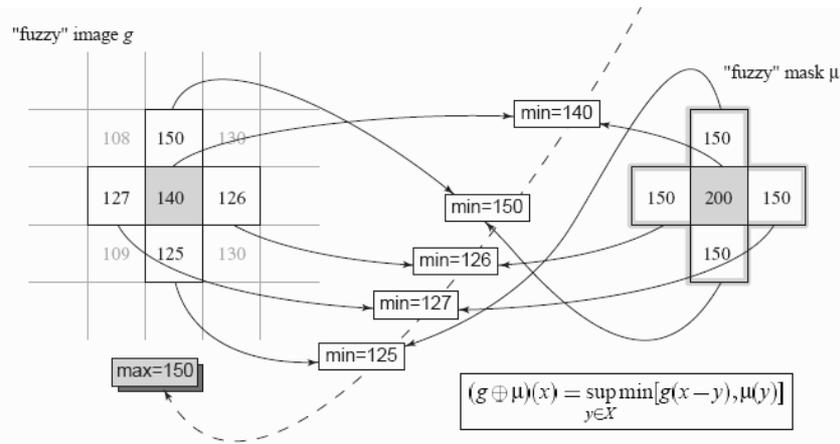


Fig 1: Schéma de la convolution particulière dans le cas de la dilatation. Les fonctions min et max peuvent être remplacées par d'autres normes. (source: [1])

De cette façon, on possède deux opérateurs morphologiques de base qui sont l'érosion et la dilatation.

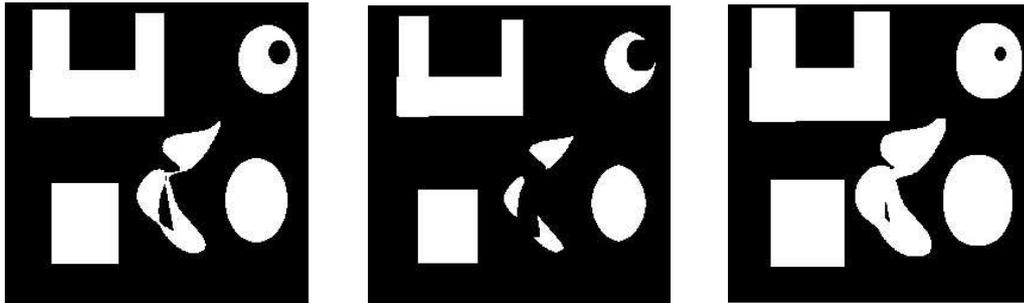
$$\begin{aligned} (g \oplus_B \mu)(x) &= \sup_{y \in X} t[g(x-y), \mu(y)] \\ (g \ominus_B \mu)(x) &= \inf_{y \in X} s[c[g(x-y)], \mu(y)] \end{aligned}$$

Fig 2 : Équations des fonctions d'érosion et dilatation (source: [1])

Ces opérations sont définies grâce à deux normes S et T et une fonction de complémentarité C. L'usage standard est de choisir les fonctions de minima et maxima. Comme nous le verrons par la suite, le choix de ces fonctions et de leurs paramètres éventuels, comme celui de l'élément structurant, influence alors le résultat final.

## 2.2 Principe et combinaisons des opérateurs morphologiques de base

En morphologie mathématique classique et également en morphologie floue, l'érosion diminue la taille des objets, supprime les éléments les plus petits et peut parfois scinder les objet possédant des zones minces. Au contraire, la dilatation augmente la taille globale et fusionne les objets assez proches les uns des autres.



*Fig 3 : Illustrations d'érosion et de dilatation (source: [2])*

Il en résulte les effets suivants pour la combinaison des deux opérateurs: une dilatation suivie d'une érosion influence peu la taille des objets mais supprime les éventuels petits trous qu'ils posséderaient. On appelle alors cette opération une fermeture. L'opération inverse, érosion puis dilatation est appelée ouverture.

## 2.3 Choix de l'élément structurant

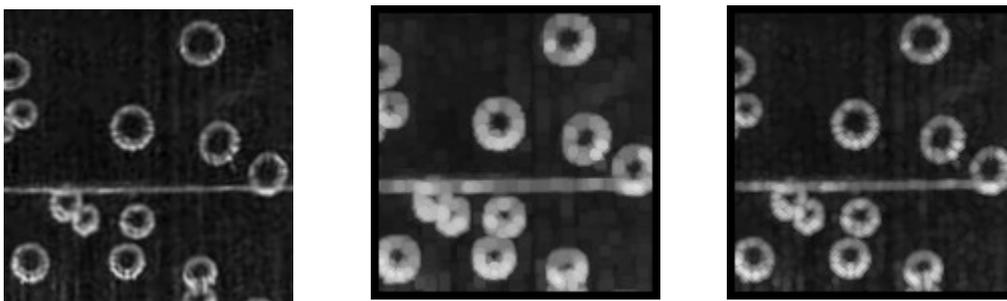
La forme de l'élément sera visible sur les contours des objets modifiés. Ainsi un filtre carré fera apparaître des points angulaires alors qu'un filtre de forme circulaire adoucira les contours. Plus l'élément structurant sera de grande taille, plus les opérations apporteront des modifications visibles sur les objets : Avec l'augmentation de la taille du filtre, on remarque que les objets plus éloigné peuvent fusionner par fermeture et que les éléments de taille moyenne peuvent disparaître par ouverture. Enfin, le choix des niveaux de gris du filtre déterminera la précision de l'opération, le flou qui sera inséré au résultat.

## 2.4 Choix des normes et du complément

Le complément flou est utilisé pour l'érosion. La modification des paramètres du complément permet de jouer sur la largeur et la position de l'histogramme de l'image résultante. Le choix des valeurs du complément dépendra donc de l'intensité du filtre à utiliser.

Le choix des normes a quant à lui une importance capitale. Nous en avons retenues quelques unes parmi celles présentées en [1]. Les normes influencent d'abord le temps de

calcul : les normes standards et algébriques sont les plus rapides. Mais c'est surtout du choix des normes utilisées que dépend la qualité du résultat final. Ainsi l'utilisation de normes « rapides » donnent des érosions et dilatations de taille importantes et où la forme de l'élément structurant apparaît fort visiblement comme un tampon plus ou moins intense sur toutes les zones de l'image. Au contraire, avec les normes de Weber et de Yager, les érosions et dilatations sont plus douces et plus précises. La norme la plus adaptée à l'utilisation morphologique faite ici, notamment pour la fermeture des bulles, semble donc la norme de Weber (la norme de Yager donnant des résultats proches pour un temps de calcul plus conséquent).



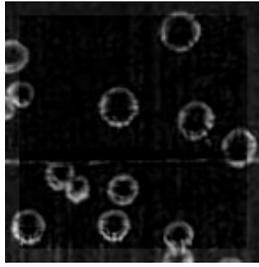
*Fig 4 : Image originale, dilatation avec norme standard et dilatation avec norme de Weber au moyen du même élément structurant.*

## 2.5 Applications des ouvertures et fermetures

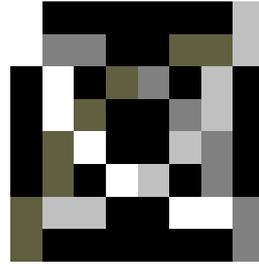
L'ouverture peut être utilisée afin d'atténuer certains éléments de l'image, comme les lignes horizontales. Le filtre utilisé représente un segment de droite horizontal. La précision est suffisante avec la simple norme standard. Le résultat de l'ouverture fait apparaître les zones de présence des lignes. L'atténuation se fera donc par soustraction du résultat à l'image originale. Les bulles, possédant elles aussi des zones horizontales elles seront également légèrement altérées, mais on s'assure ainsi de minimiser l'influence de l'éclairage des lignes, notamment pour la dissociation individuelle des bulles. On remarque également un assombrissement du fond de l'image.

On peut également appliquer une fermeture pour relier au mieux les points constituant les contours de bulles. Pour que la fermeture soit efficace et pour à la fois pour éviter la fusion entre les bulles, on utilise un filtre de petite taille et la norme standard.

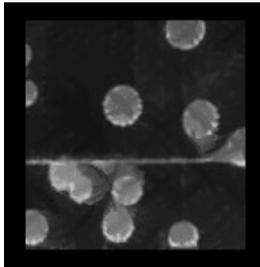
Il peut être également nécessaire de pouvoir fermer la totalité de l'intérieur des bulles dans le but de rendre ces dernières plus visibles et que la direction de l'éclairage influence moins l'aspect visuel global. Pour cela on utilise le filtre de la figure 6 qui détecte avec une pondération différente des portions d'arc de cercles.



*Fig 5 : Atténuation des lignes horizontales de l'image de la figure 4.*



*Fig 6 : Filtre de détection pondérée d'arcs de cercle*

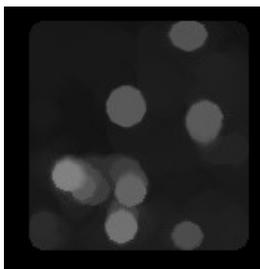


*Fig 7 : Fermeture composée de deux dilations successives puis de deux érosions avec le filtre présenté ci-dessus.*

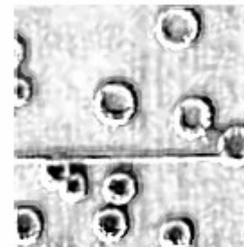


*Fig 8 : Fermeture forte de l'image précédente.*

Il en résulte un étalement des bulles dans le sens diagonal de plus grande pondération. Par simple rotation du filtre, on réalise ensuite un étalement selon la seconde diagonale. La combinaison de ces deux résultats permet de conserver la fermeture de l'intérieur des bulles tout en évitant au mieux les fusions non souhaitées.



*Fig 9 : Détection des bulles complètes par ouverture après fermeture de l'intérieur des bulles..*



*Fig 10 : Autre exemple de manipulation floue : négatif de la différence entre l'image originale et sa fermeture.*

Le cas de la fusion de bulles proches lors des différents traitements peut se révéler être un inconvénient lorsque l'on souhaite segmenter l'image pour dissocier chaque bulle qui la compose, mais il serait envisageable également d'utiliser ses fusions pour obtenir les formes globales des zones de forte présence.

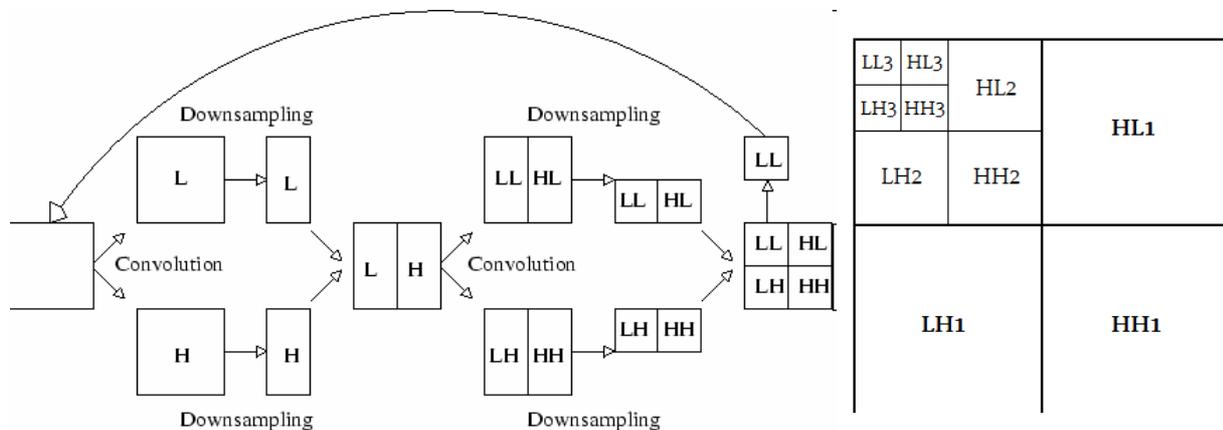
La fermeture de l'intérieur des bulles permet aussi une détection plus efficace des bulles complètes par ouverture.

# 3 Approche décomposition et transformée en ondelettes

## 3.1 Présentation succincte

La transformée en ondelettes d'un signal peut être représentée sous forme de la convolution avec deux filtres (ondelettes) qui scinde le signal en deux signaux distincts correspondant respectivement aux basses et hautes fréquences du signal d'entrée. On peut ensuite appliquer de nouveau les filtres au signal des basses fréquences pour obtenir une séparation fréquentielle supplémentaire.

Ce principe s'étend en deux dimensions, notamment pour les images. Les convolutions sont alors appliquées tour à tour suivant les lignes et les colonnes. Un échantillonnage suivant les deux axes permet de stocker les résultats dans une image de taille égale à l'originale. On obtient alors quatre zones ou sous-images qui sont les basses fréquences, les hautes fréquences, les contours horizontaux et les contours verticaux. L'ensemble de ses principes sont présentés dans [3] et détaillés dans [4].

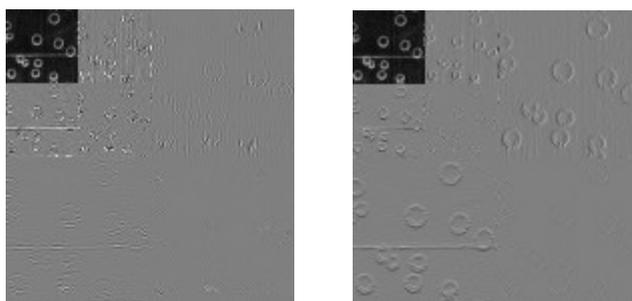


*Fig 11 : Algorithme de décomposition en ondelettes et représentation spatiale des différentes fréquences pour une décomposition de niveau 3 (source: [3])*

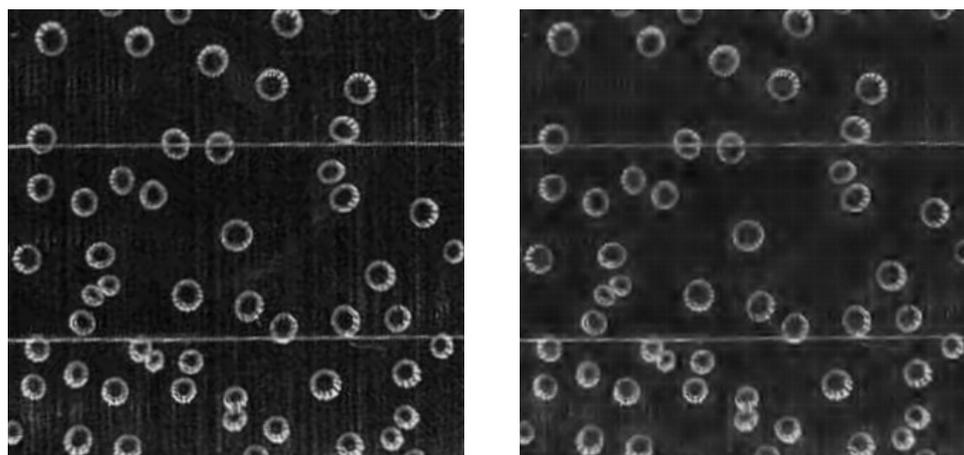
Un processus inverse avec des ondelettes calculées à partir des ondelettes de décomposition permet une reconstruction de l'image.

## 3.2 Choix de l'ondelette

Il existe de nombreuses ondelettes possibles qui séparent différemment les fréquences. Celle de Haar permet par exemple de bien remarquer les contours alors que celle de Daubechies 9/7 donne peu de valeurs en hautes fréquences ce qui est fort utile en vue de travaux de compression : les pixels de faibles valeurs seront approximés par zéro puis un codage entropique de Huffman réduira à de petites tailles les données redondantes comme les zones nulles. Cette approximation sert également à débruiter l'image. Les défauts de reconstruction dépendent aussi de l'ondelette utilisée.



*Fig 12 : Décomposition de niveau 2 par ondelettes de Daubechies 9/7 et par ondelettes de Haar.*



*Fig 13 : Débruitage par seuillage des coefficients inférieurs à 10 de décomposition de Daubechies de niveau 4.*

Image	Taille (Ko)	Compression
Image originale	100	0
Image originale zippée	66,3	34%
Daubechies profondeur 4 zippé	52,0	48%
Daubechies seuillé & zippé	8,24	92%

*Fig 14 : Exemple du résultat de compression d'une image par les ondelettes (image présentée en figure 14) .*



*Fig 15 : Compression d'une image et comparaison visuelle entre l'image originale et l'image compressée.*

### 3.3 Utilisations

Outre le débruitage et la compression entrevues précédemment, la décomposition en ondelettes peut également servir à d'autres fins, notamment en utilisant l'analyse multirésolution (AMR).

#### 3.3.1 Moyennes et écarts types

Il est par exemple possible de caractériser une image ou des zones d'une image par une approche probabiliste sur chaque élément de la transformée (les différentes images de basses et hautes fréquences). L'écart type ou la moyenne peuvent être comparés entre plusieurs images. Mais ces valeurs globales à l'image ne permettent a priori pas de caractériser chaque code à bulles: en effet les différences de valeurs entre deux codes différents ne sont pas significativement plus importantes que les différences qui existent entre deux images d'un même code à l'éclairage changeant. On remarque cependant que l'allure générale des courbes de valeurs des codes à bulles est par exemple très différente de l'allure d'une courbe d'une image d'empreinte digitale.

C'est d'ailleurs au moyen des écart types des éléments de la transformée de fractions de l'image que l'on peut comparer plusieurs images d'empreintes digitales, comme le montre [5]. On peut donc avoir l'idée d'appliquer cette méthode aux Codes à bulles mais le résultat est loin d'être aussi discriminant. Cette voie peut cependant être poursuivie en considérant l'existence éventuelle d'un vecteur caractéristique autre que l'écart type et qui permettrait par cette méthode de définir une certaine « distance » entre deux Codes à bulles, de connaître le degré de similarité entre deux images de codes.

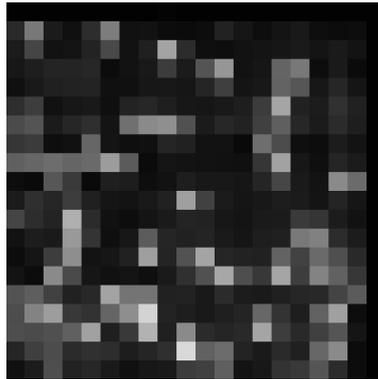
#### 3.3.2 Comparaison avec des textures connues

Une autre possibilité est de déterminer les textures composant une image par comparaison avec certaines textures connues et déjà identifiées. L'utilisation comme texture de référence d'une bulle complète entraîne un temps de calcul conséquent du à la tailles des éléments à comparer et ne permet pas de trouver facilement toutes les zones de présences de bulles. Seul

la bulle prise pour modèle est correctement détectée. Il ne faut donc pas comparer directement les textures, mais bien déterminer un vecteur caractéristique pour les textures, composé des éléments qui caractérisent au mieux chacune d'entre elles.

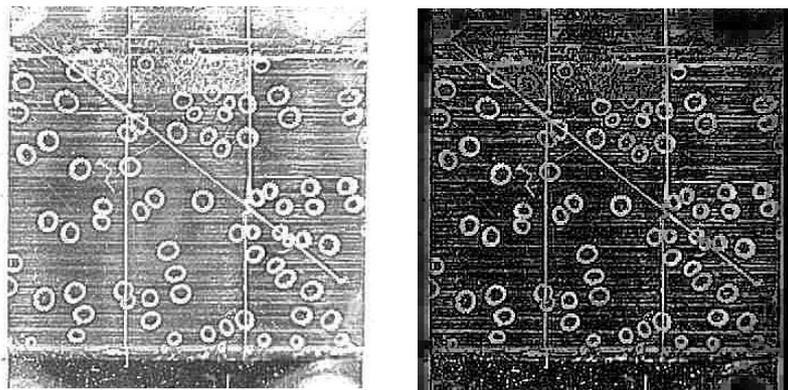
### 3.3.3 De l'utilisation des basses fréquences

La génération du code unique pourrait être approchée en utilisant pour traitement final une décomposition en ondelette dont on garderait uniquement l'image basse fréquence. Cette image correspond en effet à l'aspect globale que l'oeil se fait d'une image.



*Fig 16 : Basses fréquences d'une image de Code à bulles ayant subit quelques traitements intermédiaires..*

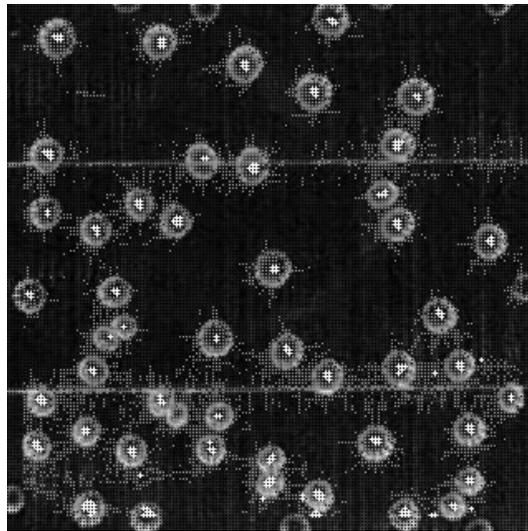
Il est cependant à noter que l'aspect global d'un code à bulle n'est pas toujours identique suivant la photographie faite car l'intensité de l'éclairage du fond peut varier du tout au tout et cela pour quelques zones ou pour l'image entière. On peut donc également envisager dans un premier temps de supprimer les très basses fréquences afin de limiter cet effet désavantageux de l'éclairage global du fond.



*Fig 17 : Image originale et image normalisée de la reconstruction en omettant les basses fréquences de Haar de niveau 4.*

### 3.3.4 Détermination des centres et rayons des bulles par leurs contours

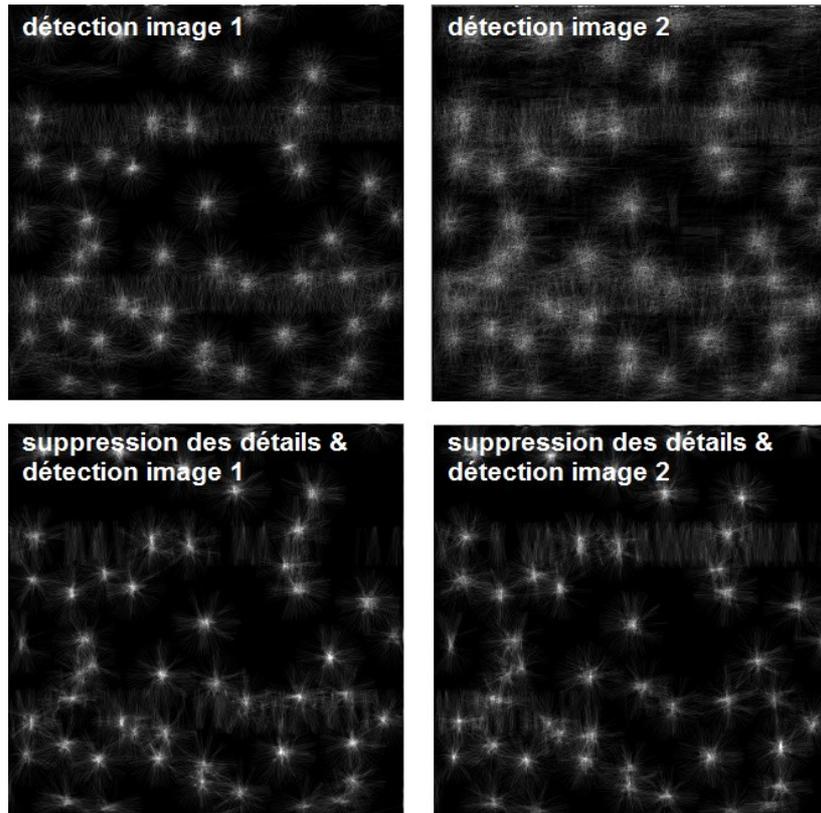
En s'inspirant d'une méthode de détection de cercles par la transformée de Hough exposée en [6], la transformée en ondelette peut permettre aussi de trouver les centres et rayons des bulles présentes. Pour cela on considère les images des contours horizontaux et verticaux (surtout visibles avec la transformée de Haar) et l'on calcule ainsi la direction des vecteurs normales aux contours. On trouve les centres par le recoupement de segments de droites tracés suivant la direction normale des contours. Pour vérifier ensuite qu'il s'agit bien d'un centre, on cherche le rayon pour lequel les contours de la bulle sont détectés suivant quelques axes passant par le point à tester. Ce nouveau test peut être effectué sur l'image basse fréquence (pixels lumineux des bords de bulles) ou sur les images des contours.



*Fig 18 : En gris: points de recoupement des segments de droite.  
En blanc: centres détectés.*

A l'occasion de ce traitement, on remarque sur certaines images parmi les plus nettes que beaucoup trop de contours sont détectés ce qui entraîne le tracé de droites normales parasites. Ce problème est identifié comme la conséquence de détails tels que la distinction des LED d'éclairage dans le reflet sur les bulles, détails superflu et même gênants pour la détection effectuée. Il est donc proposé ici, lors d'une reconstruction de transformée, à l'inverse du traitement précédent, de garder seulement les basses fréquences.

Seulement ce traitement supprime en plus des détails les contours horizontaux et verticaux utilisés dans la détection des cercles. La solution proposée ici est de supprimer les hautes fréquences puis de recréer des contours artificiels de bulles par un agrandissement de l'image au moyen d'une pixelisation de taille 2. La méthode de détection des cercles peut ensuite de nouveau être appliquée.



*Fig 19 : Tracé des normales aux contours sur deux images de codes. La seconde image source est plus nette que la première. Nouveau tracé après suppression des détails superflus suivant la méthode exposée ci-après.*

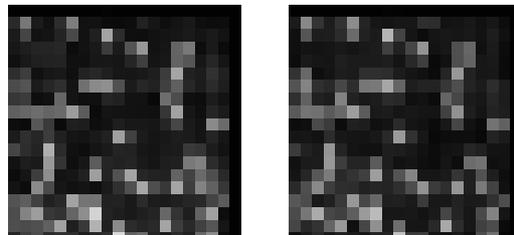
## 4 Utilisation couplée des deux approches

### 4.1 Rôles possibles et conséquences de chaque opération

Dans les images considérées dans notre étude, il est à signaler qu'un redressement préalable à déjà été effectué afin que les codes soient présentés sous un angle et une échelle peu variables.

La morphologie floue peut fermer l'intérieur des bulles ou bien directement servir à segmenter l'image. La fermeture des bulles permet de mieux repérer les bulles à l'oeil mais n'augmente par contre pas la fiabilité de détection des centres car l'opération morphologique supprime ou diminue les contours intérieurs des bulles et rend donc la détection moins facile.

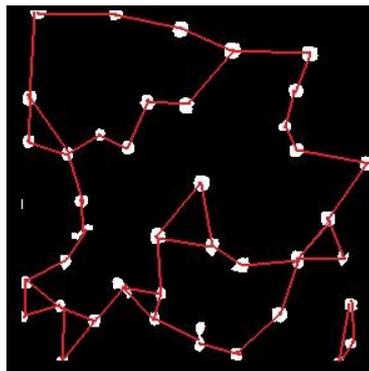
Après l'utilisation de la morphologie on peut également chercher à conserver les basses fréquences de l'image pour obtenir un code 2D.



*Fig 20 : basses fréquences de la fermetures des bulles de deux images d'un même code .*

### 4.2 Interprétation floue

Afin de ne pas dépendre des rotations, translations et zooms de l'image, l'emploi de logique floue semble adaptée.



*Fig 21 : Représentation et détermination visuelle des plus proches voisin (les chemins ont été tracés par nos soins)*

Par exemple, dans la cas où l'image serait segmentée et chaque bulle identifiée, on pourrait considérer l'échelle d'éloignement entre les bulles, écrire un algorithme pour déterminer les n plus proches voisins de chaque bulle, ce qui visuellement est analogue à la considération des formes particulières, des constellations formées par les positions des bulles.

On peut aussi fermer et détecter les bulles par morphologie puis compresser le résultat par ondelettes ou par simple moyennage de zones constituant un quadrillage de l'image. L'interprétation du résultat peut alors reposer sur les intensités résultantes, classées selon quelques intervalles choisis, comme par exemple ceux déterminés visuellement par les différence de familles de couleurs lors d'une considération de l'image niveau de gris comme la composante de teinte de l'espace de couleur HSL.

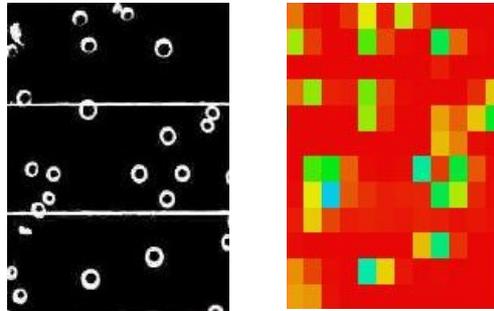


Fig 22: Code et interprétation HSV des valeurs moyennes de zones.

Code original (w035652):	Décalage 1x1 :	Décalage 0x2 :
YGYGGBGGRGGG	<b>R</b> GYGGGGGRGGG	Y <b>G</b> GGGGGRGGG
GBBBBGBGGBBB	GBBBBGBG <b>G</b> B	GBBBBGBG <b>G</b> B
GPPPBGBBYBB	GPPPBGBBY <b>B</b>	GPPPB <b>Y</b> BGRBB
GGBPBBGBBGBB	GGBPBBGBB <b>G</b> B	GGBPBBGBB <b>B</b> B
GBBRBGBBPGBB	GBB <b>P</b> BGBBPGBB	GB <b>P</b> BGBBPGBB
YBBBBBGBYBG	YBBBBBGBY <b>B</b>	<b>R</b> BBBBBGBYBG
GBBGBBBBBGBB	GBBGBBBBB <b>G</b> B	GBBGB <b>G</b> BBGBB
GBBGBBBBBBBB	GBBGB <b>G</b> BBBBBB	GBBGBBBBB <b>B</b> B
GPBGBGGBBBBB	GPBGBGGBBB <b>P</b>	GPB <b>B</b> GBBB <b>P</b> G
GBBBGBGGBBBG	GBBBGBGG <b>G</b> BBG	GBBBGBGG <b>G</b> BG
GBGBYGYGGGBR	GBGB <b>G</b> Y <b>B</b> GGBR	GBG <b>G</b> GR <b>B</b> GGBR

Fig 23: Signature d'un code à partir des couleurs et effet de légères translations sur le résultat.

Mais d'autres interprétations sont peut-être également envisageables dans l'élaboration d'une signature unique de code, notamment la carte des dissimilarités locales (CDL) présentée dans [7] et utilisée pour la reconnaissance de tampons d'imprimerie. En effet cette approche semble elle aussi moins sensible aux altérations qui peuvent rendre assez différentes deux images d'un même objet, d'un même code, ou comme présenté sur la figure ci-dessous, d'un même visage.



Fig 24 : Photographie d'un visage et son érosion par un carré 15x15.  
En couleur : différence classique entre les deux images et CDL.(source : [7])

# 5 Élaboration d'une signature

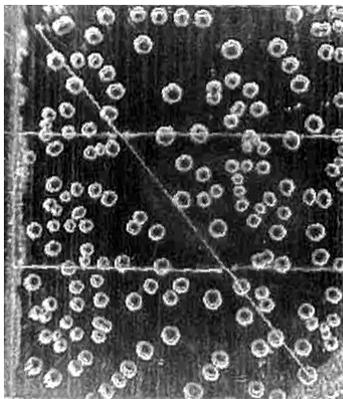
## 5.1 Traitements successifs retenus

Comme entrevu dans le paragraphe précédent et figure 22, la signature est obtenue par une interprétation HSV de valeurs moyennes de zones. Nous énonçons ici plus en détails l'algorithme permettant d'obtenir ce résultat.

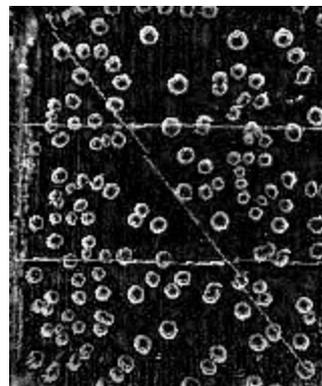
L'image d'origine est chargée soit à partir d'un fichier, soit à partir d'un champs de la base de données des codes déjà acquis. Il est important pour la suite de considérer des images de tailles identiques afin de ne pas risquer de fausser les comparaisons futur. Une fois l'image découpée pour ne garder que le code, nous avons ici des images de 340x400 pixels à l'histogramme normalisé.

On sous-échantillonne d'abord par 2 la taille afin d'accélérer la vitesse de traitement tout en gardant une quantité suffisante d'information utile.

La première étape consiste à supprimer les basses fréquences de l'image par une transformée de niveau 4 de Daubechies suivie d'une reconstruction, ce qui retire quelques effets variables de l'éclairage. On normalise à nouveau l'image obtenue afin de garder une intensité importante sur les bulles.

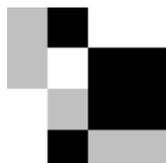


*Fig 24: Image centrée sur le code et normalisée.*



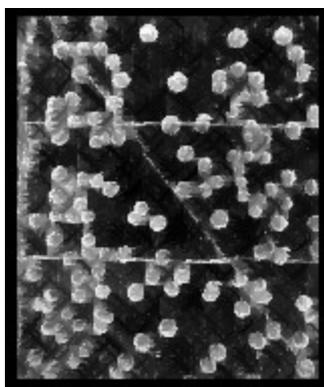
*Fig 25: Image sans les basses fréquences et normalisée.*

Ensuite l'intérieur des bulles est fermé par fermeture morphologique comme expliqué paragraphe 1-5. Une adaptation est cependant nécessaire du fait du sous-échantillonnage de départ : un filtre plus petit à été redéfini pour remplacer celui de la figure 6.

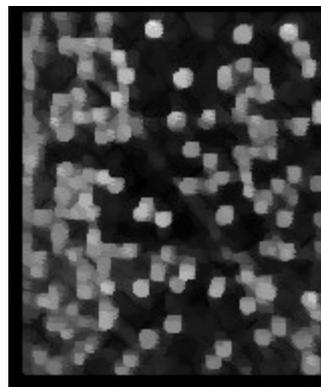


*Fig 26: Nouveau filtre de fermeture de bulles. Il est deux fois plus petit que celui représenté en figure 6 et ne contient plus que 2 quarts de cercles au lieu des 4 du filtre initial. Cette adaptation est celle qui semble la plus efficace après plusieurs essais.*

Les bulles fermées sont ainsi plus facilement détectables, encore une fois à l'aide de morphologie, avec une ouverture par un filtre représentant une bulle fermée. Cette étape permet en outre d'atténuer la présence des lignes.



*Fig 27: Bulles fermées à l'aide du nouveau filtre suivant le principe énoncé paragraphe 1-5.*



*Fig 28: Détection des bulles fermées.*

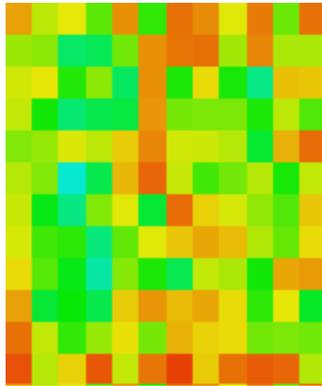
Enfin on obtient la matrice utile par moyennage de l'intensité sur les zones d'un quadrillage 12x12 de l'image.



*Fig 29: Matrice de moyenne de l'image précédente.*

## 5.2 Stockage du résultat sous forme d'une signature

Il s'agit maintenant d'interpréter de façon efficace cette matrice de moyenne. Pour cela, on peut visuellement représenter les intensités des pixels par des couleurs différentes qui sont en réalité une simple interprétation HSV.



*Fig 30: interprétation HSV de la matrice.*

Une intensité faible va correspondre à un pixel rouge tandis qu'une intensité très forte peut engendrer un pixel virant sur le violet. On définit ainsi 5 couleurs, soit 5 intervalles qui segmente les valeurs d'intensités (bornes précisées entre parenthèse) en 5 catégories (nommées par leur couleur) :

Rouge(0-38), Jaune(39-50), Vert(51-110), Bleu(111-180), Violet(181-255).

Le choix de ces intervalles ne répond pas à un soucis d'intervalles réguliers mais est une adaptation de la sensibilité visuelle.

Grâce à ces intervalles on obtient par lecture de la matrice de gauche à droite et de bas en haut (sens de lecture des images) une signature sous forme de chaîne de caractère que l'on peut stocker dans la base de données, dans un champs du code stocké correspondant.

Si un tel type de signature est retenu, il sera possible de fortement compresser sans perte ses données car l'ensemble des caractères possibles ne comporte que cinq éléments.

## 5.3 Calculs de distances entre les signatures

Pour mesurer la distance entre deux images, dans le but de déterminer s'il s'agit ou non d'images d'un même Code à bulles, nous disposons donc de deux signatures. Il faut donc décrire une loi permettant de comparer de façon pertinente ces signatures et de quantifier la différence entre les codes.

Voici donc le principe de mesure de distance entre deux signatures:

- On parcourt les deux signatures caractères par caractères, si un caractère de l'une ne correspond pas exactement au caractère de l'autre, on augmente la valeur de la distance comme suit:
- Si le caractère est d'une couleur proche (par exemple est Jaune ou Bleu au lieu de Vert), la distance est incrémentée de 1.
- Sinon, on cherche dans les caractères voisins (à gauche, à droite, en haut et en bas si on n'est pas sur un bord) si l'on retrouve la couleur exacte qui aurait été traduite par un redressement différent, alors on incrémente la distance de 4.
- Si on ne trouve toujours pas dans les voisins, alors on augmente de façon encore plus significative la distance totale entre les signatures suivant la couleur non trouvée (+8 pour une couleur Rouge, +16 pour le Jaune, +32 pour le Vert, +64 pour le Bleu, +128 pour le violet). Ainsi l'absence totale d'une zone de très forte concentration de bulles -et donc d'intensité élevée- sera fortement pénalisant et augmentera grandement la distance entre les signatures des codes.

## 5.4 Évaluation de cette solution

Il est à remarquer que cette signature n'est pas encore unique à chaque Code à bulles. En effet, la distance entre deux signatures d'un même code n'est jamais nulle, cependant, si l'acquisition d'image est relativement correcte, la distance reste faible comparée à la distance entre des Codes différents.

Appliquée à un échantillon de petite taille, composé de 16 clichés de 3 Codes différents, cette solution permet bien de différencier les codes: Les distances entre codes identiques sont inférieures à 300 (et sauf quelques rares cas dûs semble-t-il à des échelles trop différentes, la distance est même inférieures à 200) (moyenne de 72, écart type de 61). A l'opposée, la distance entre les codes différents s'avère toujours supérieures à 450 (moyenne de 784, écart type de 213). La signature est donc assez discriminante pour l'échantillon considéré.

Grâce aux images déjà présentes en base de données, il a été aussi possible de créer un nombre important de signatures différentes et de mesurer les distances en regroupant les codes par catégories selon le nombre de bulles présentes dans chacun d'eux. La figure 31 montre alors les minima trouvés pour ces distances. Dans aucun des cas rencontrés la distance n'a été inférieure à la valeur de 200.

La signature n'est donc pas unique mais possède un voisinage de reconnaissance. Cette signature est modifiée dans de faibles proportions par les petites translations et rotations de l'image et également par la compression JPEG. Par contre, il est important de signaler une plus forte instabilité en cas de modification du facteur de zoom, i.e. de l'échelle de l'image.

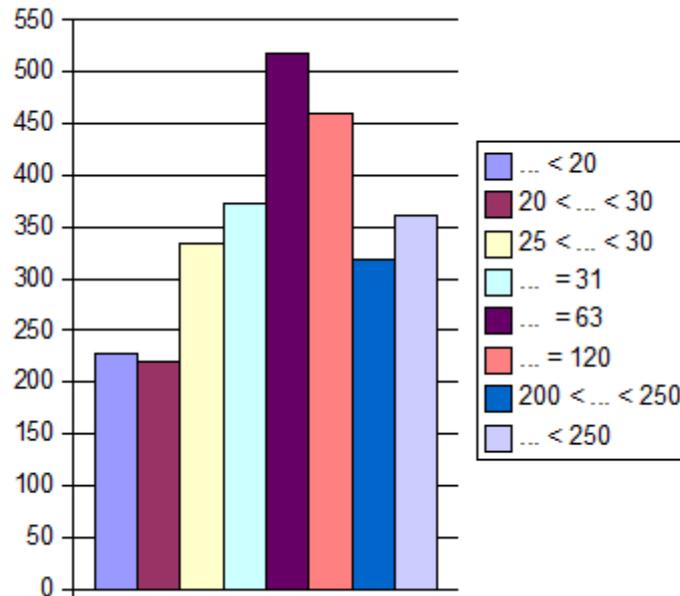


Fig 31: Minima des échantillons suivant le nombre de bulles présentes dans les code

## 5.5 Réflexion sur la stabilité des signatures

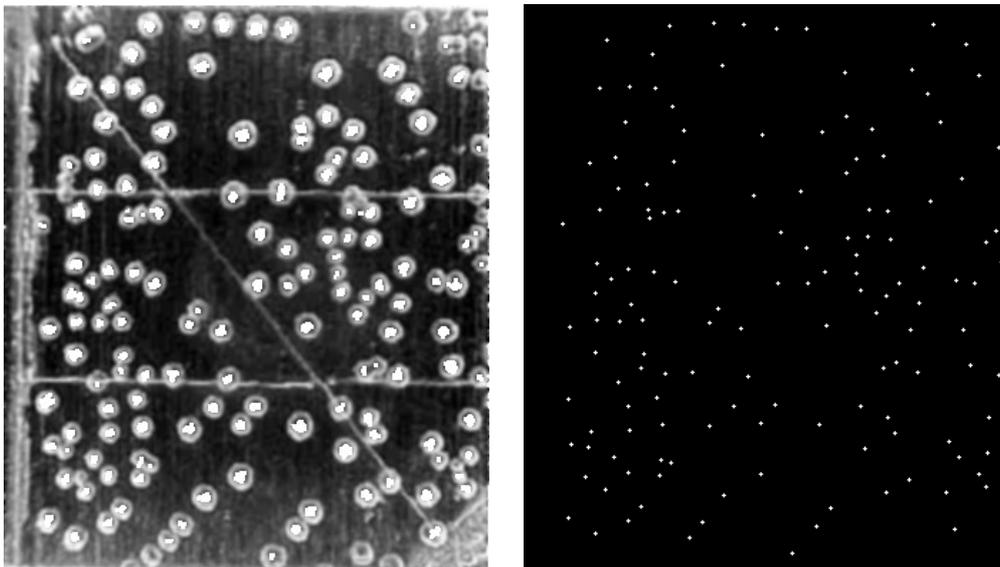
Un problème concernant la stabilité est, nous l'avons vu précédemment, l'influence des rotations, translations, compressions et échelles. Afin de diminuer de façon plus importante les effets de ces variations il semble donc utile d'aller plus loin dans l'interprétation faite de l'image.

Il faudrait donc détecter rapidement l'ensemble des bulles (la position approximative de leur centre) puis ne plus considérer que la position des bulles relativement aux autres (comme entrevu au début de la section 4-2).

Afin de limiter le temps de calcul destiné à rechercher toute correspondance possible entre deux codes, il faut cependant partir des images redressées (même avec une légère différence, nous avons tout de même ainsi l'information importante de l'orientation du code).

De plus, certaines zone d'un code sont plus critiques en ce qui concerne la détection de bulles (reflets latéraux, bulles fictives aux croisements des repères. On peut donc entrevoir le code en deux parties distinctes : une partie centrale, de forme à déterminer où il est normalement plus simple et surtout impératif de trouver exactement toutes les bulles afin d'identifier et d'authentifier ou non le code. Cela constituera le noyau unique de la signature. Le reste de l'image servirait alors à la confirmation ou non de l'authentification avec cette fois-ci une tolérance de comparaison plus importante, permettant notamment des sur-éclairages et des erreurs dans les détections de bulles.

Enfin, dans le codage complet de cette signature, la normalisation des distances par la moyenne des distances mesurées permettrait d'annihiler l'influence du facteur d'échelle.

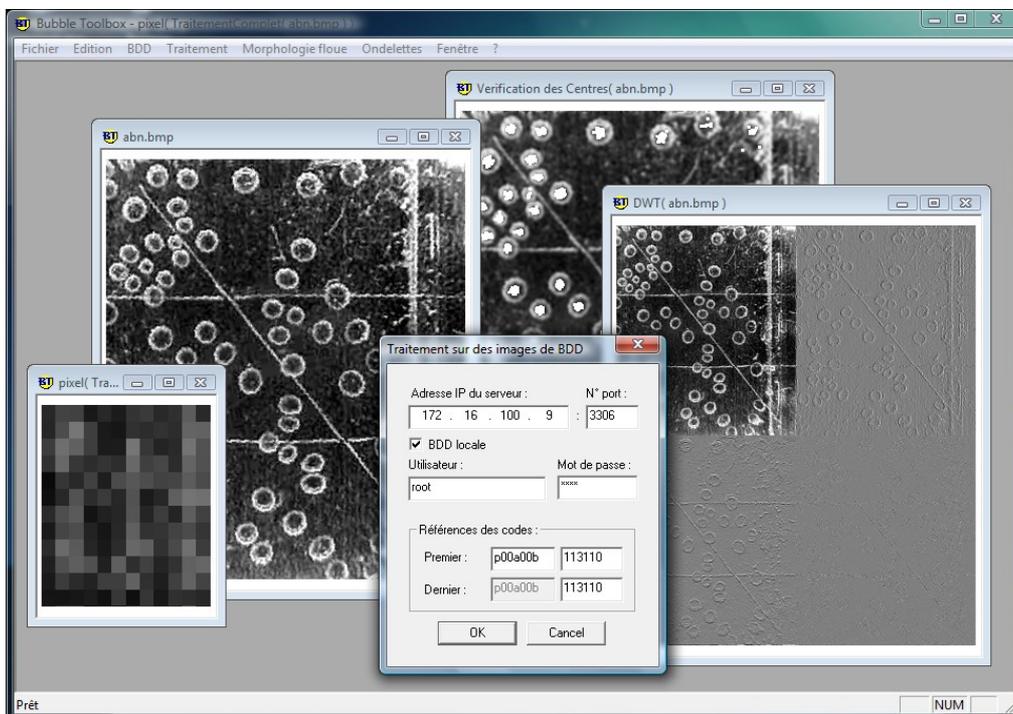


*Fig 32: Exemple de détection de bulles utilisant les ondelettes et une segmentation personnalisée par hystérésis (amélioration du principe présenté section 3-3-4)*

# Conclusion

A travers ce stage j'ai pu découvrir plusieurs éléments du traitement d'image par des méthodes utilisant les ondelettes et d'autres reposant sur la morphologie floue. Ce fut également l'occasion de développer une application dont l'interface graphique réalisée au moyen des MFC mais aussi de coder d'autres algorithmes de traitement général d'image eux-aussi primordiaux.

Il ne fut finalement pas possible de générer une signature réellement unique par code durant la recherche. Cependant, certaines stabilités ont été établies et des axes de réflexions menées et testés afin de bien connaître les possibilités offertes par les deux théories, dans le but d'atteindre une signature unique ou de s'approcher de cette unicité par une recherche de stabilité optimale, peu sensible notamment aux compressions et variations de redressements et d'éclairage.



*Fig 33: Une vue de l'application réalisée*

# Références bibliographiques

- [1] M. KOPPEN, K. FRANKE & O. UNOLD, *A Tutorial on Fuzzy Morphology*, Fraunhofer IPK Berlin & Wroclaw University of Technology (1999)
  
- [2] L. BRUN D'APRES M. COSTER, *Morphologie mathématique : Erosions et Dilations*, laboratoire GREYC de Caen (\_\_\_)
  
- [3] V. PERRIER, *mise en oeuvre des compressions Jpeg et Jpeg2000*, ENSIMAG (2005)
  
- [4] S. MALLAT, *A wavelet tour of signal processing*, École Polytechnique de Paris & Courant Institute, New York University (1998)
  
- [5] M. TICO, E. OMMONEM, P. RAMO, P. KUOSMANEN & J. SAARINEN, *Fingerprint recognition using wavelet features*, DMI Tampere University of Technology, FINLAND (2001)
  
- [6] A. BARGETON, *Détection des cercles dans une image (Hough Circle)*, EPITA (2003)
  
- [7] E. BAUDRIER, *Comparaison d'images binaires reposant sur une mesure locale des dissimilarités: Application à la classification*, UFR des Sciences Exactes et Naturelles de l'université de Reims Champagne-Ardennes (2005)