

Rapport du Projet d'image

Reconnaissance et indexation
de forme

Patrice Coudert et Frédéric Lang

Table des matières

1	Détails des indicateurs	3
1.1	PERIMETRE	5
1.2	DISQUECIRCONS	6
1.3	NBSEG	7
1.4	MAXSEG	7
1.5	VARSEG	8
1.6	ANGLESEG	8
1.7	Distance transformation	9
2	Calcul des images les plus proches	10

Introduction et structure du rapport

Le principe du projet est de créer une signature. Pour cela nous avons créé un certain nombre d'indicateurs que nous allons tout d'abord vous présenter. On en profitera pour s'intéresser à leur robustesse vis à vis de l'occlusion, du bruit, ... Ces propriétés seront étudiées à la fois théoriquement et en pratique.

Pour les résultats pratiques, les scripts bash et les graphiques sont présent dans le dossier pour reproduire ces tests. Même si les graphiques ne seront pas tous dans le rapport, ils sont présent dans le dossier résultat. Les scripts bash servent à reproduire les tests que nous avons effectués.

La suite expliquera comment nous utilisons ces indicateurs pour répondre aux questions. Nous proposons deux options : la première renverra la classe la plus probable et la deuxième renverra les k images les plus proches.

Explications du programme

Nous allons commencer expliquer comment utiliser notre programme. Le programme s'appelle *main*. Il prend deux options. La première est le mode du programme. Les différentes options sont :

- *help* : qui affiche les options et les explications de ce qu'elles font.
- *ND n* : qui renvoie les n images les plus proches au sens de notre distance.
- *indn* : qui renvoie l'indicateur n pour l'image en entrée.
- *disp* : qui crée un fichier *image.eps* qui montre nos constructions. (voir ci-après pour plus de détails)
- *dispDT* : qui crée un fichier *image.eps* qui montre les cercles où la distance au bord est constante.

Pour utiliser notre programme, il faut donc taper le nom du programme, puis le mode d'exécution souhaité, puis si voulue une des options d'affichage et enfin le nom du fichier. Voilà trois exemples :

```
./main -ind1 -disp apple1.pgm  
./main -ind3 -dispDT apple1.pgm  
./main -ND 8 apple1.pgm
```

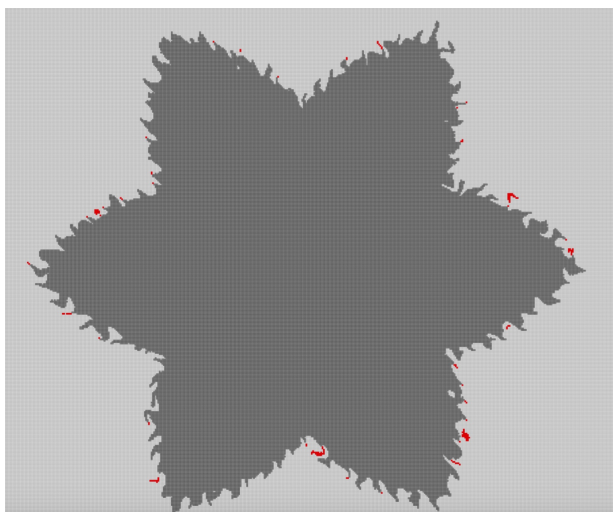
Chapitre 1

Détails des indicateurs

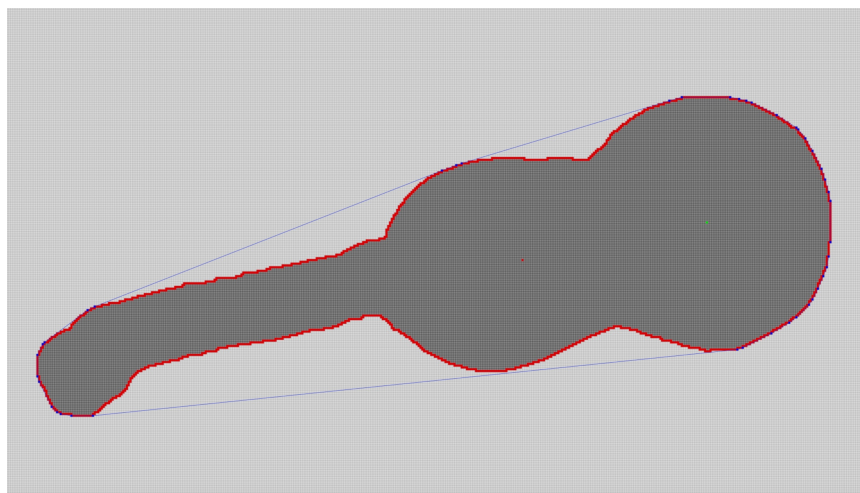
Introduction

Les indicateurs que nous avons choisi essayent de faire ressortir les caractéristiques principales de l'image. Après avoir regarder les images, on a décidé de grandeur caractéristiques simples et assez différentes selon les classes. À partir de la sortie graphique du programme, la cohérence des indicateurs peut être vérifiée visuellement ce qui nous permet de vérifier facilement nos programmes. Les option *-disp* et *-dispd* permet de réaliser ces vérifications visuels.

Nous avons fait le choix de nous intéresser qu'à la composante connexe principale c'est à dire de plus grande taille. Les composantes connexes sont calculées en topologie 4. On donne un exemple du changement d'image qu'impose notre choix sur *device1 - 9.pgm*. La composante principale est en noir et les composantes supprimés sont en jaune. La raison de ce choix est double. Premièrement on enlève des taches qui ne sont pas très caractéristiques de l'image. En effet ces taches peuvent résulter du bruit de notre image ou sont de taille minime et pas caractéristique de la classe. Deuxièmement nos fonctions qui calcule le contour et l'enveloppe convexe ne sont pas résistant à ces problématiques. On discute des améliorations possibles liés à ceci dans le section limitation et améliorations possibles.



L'image ci-dessous est le résultat de notre algorithme pour l'image *guitar* – 1.pgm. Le contour de l'image est en rouge. Celui-ci est pris en topologie 4. Le contour de l'enveloppe convexe apparaît en bleu. Le barycentre apparaît en bleu et le point où la distance au bord est maximal en vert. Ces résultats sont obtenus en ajoutant l'option *-disp* à n'importe quel image. La deuxième image est obtenu sur la même image avec la commande *-dispDT* est montre les cercle où la distance au bord est constant.



Nous allons présenter un par un les dix indicateurs que nous avons choisi. Un script est présent pour calculer un indicateur sur toutes les images de la base de données. Pour l'utiliser il suffit de le lancer avec le numéro de l'indicateur Pour chacun nous allons ensuite justifier théoriquement et en pratique de leur résistance/sensibilité au bruit, à un changement d'échelle et à une rotation. Toute ces sessions ne sont pas développé en détail. Mais, pour chaque indicateur,

trois test ont été réalisé grâce aux scripts résistance et les courbe sont dans le dossier correspondant. Pour refaire ces test, il suffit (après avoir compilé le programme) d'exécuter les script bash par la commande suivante :

```
bash resistance_bruit_1i.sh apple 3 2
```

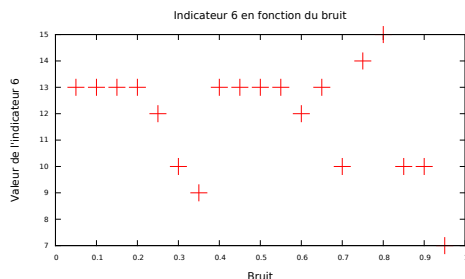
Cette commande lance vingt test sur l'image *apple - 3.pgm* pour regarder l'influence du bruit pour l'indicateur 2. Ensuite pour tracer les courbes il suffit d'utiliser les script gnuplot dans le dossier *resultat/bruit/*. Dans ce fichier, il suffit juste de changer le nom du fichier dont vous souhaitez la courbe.

1.1 PERIMETRE

Cet indicateur mesure la longueur du contour de l'image. Il est destiné à différencier les formes très courbés (*lizzard*) de celle qui se rapproche plus d'un disque (*apple*). Il est calculé en extrayant le contour de la forme avec la fonction *contour* implémentée dans le fichier *utils.cpp*. Cette fonction est celle qu'on a codé lors du TD5. Le contour obtenu correspond a une topologie 4. Pour ne pas le rendre sensible à un changement d'échelle, on divise par la racine de la taille de l'image.

Résistance au bruit

Le contour est par nature assez sensible au bruit. En effet, le bruit se situant majoritairement au bord, c'est le contour qui est le premier modifié. Pour limiter ce problème, on ne garde que la plus grande composante connexe en topologie 4. Ainsi, même si on s'attend à ce que la taille du contour augmente, le fait de supprimer les composantes connexes en topologie enlève un peu le bruit et donc la taille du contour reste acceptable. Les test pratique confirme ce fait on voit un augmentation assez nette de notre indicateur.



Résistance à un changement d'échelle et à une rotation

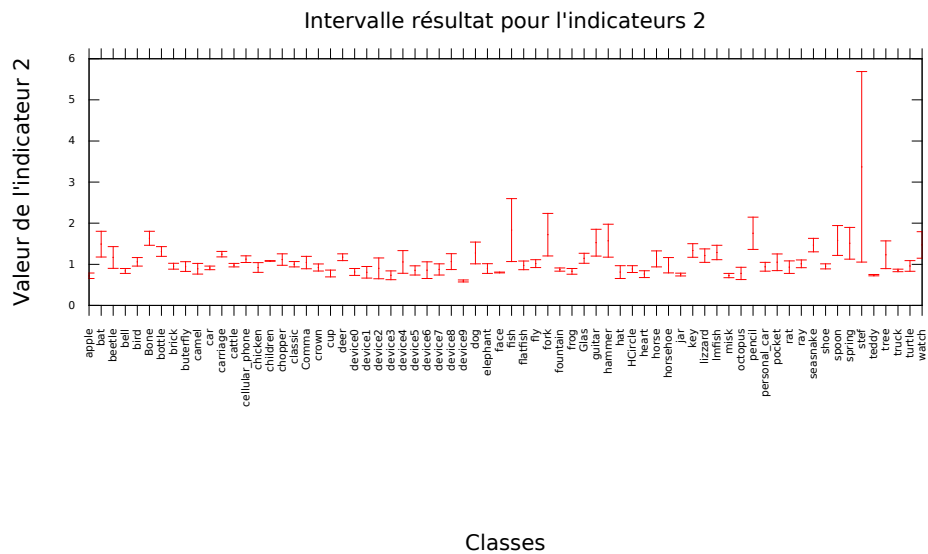
L'orientation de notre figure n'a que très peu d'impact sur notre contour. Lorsque la taille de l'image est assez grande, tourner l'image conserve la taille du contour même si à certain endroit particuliers, le contour varie un tout petit peu. les résultats pratiques confirme ce fait. De même un changement d'échelle

n'influe pas sur cet indicateur du fait de la division par la racine carré de la taille de l'image.

1.2 DISQUECIRCONS

Cet indicateur nous est venu à l'idée lorsque on a vu les classes hammer, lm-fish, glass, ... et les classes fish, apple, ... La longueur de l'image par rapport à sa taille est très différent. Pour ce faire, on utilise le barycentre que l'on calcule. Ensuite on calcule la distance à notre barycentre calculé. Les fonctions *barycentre* et *distFarthestPoint* dans *utils.cpp*. L'indicateur *DISQUECIRCONS* est au final le rapport entre cette plus grande distance (rayon) et la racine carré de la surface. On voit sur le graphique suivant que les intervalles sont très resserrés.

Pour générer ces tests, il faut utiliser la commande suivante :
`bash calcul_ind.sh 2` Le numéro est le numéro de l'indicateur dont on veut calculer la valeur pour toutes les images de la base de donnée. Ce script marche pour tout les indicateurs.



Résistance au bruit

Le barycentre n'est pas sensible car certains points disparaissent et d'autre apparaissent mais le nombre et leur moyenne est quasiment identique. De plus la distance ne change que très peu donc la distance reste quasiment la même. La courbe le montre d'ailleurs parfaitement. Á la fin, le bruit est tel que l'indice explose mais on est très loin de l'image de base.

Résistance à un changement d'échelle et à une rotation

De part la division par la racine de la surface, l'indice est résistant à un changement d'échelle. Le graphique montre ce résultat. La rotation de l'image ne change aussi que très peu cet indice. En effet, les petites différences qu'on observe sont dues au fait qu'on travaille en géométrie discrète. Elles sont minimes (quelques %).

1.3 NBSEG

Le troisième indice est le nombre de segment qui délimitent l'enveloppe convexe de l'image. Les points qui relient ces segments sont calculé par l'algorithme de Graham implémenté par la fonction *convexHull* dans *convexhull.cpp*. Pour la première étape de l'algorithme où sont triés les points selon un ordre lié à un point p_0 , le tri a rencontré des problèmes d'implémentation quand l'angle formé par les trois points était null. Ce problème réglé, l'indicateur fonctionne et est finalement un bon indicateur de classe.

Résistance au bruit

Même si le placement de point dépend en grande partie du bruit, ajouter du bruit ne change que peu le nombre de ces points. L'enveloppe convexe devient juste plus large. Nos résultats pratique confirme ce fait. Le nombre de points augmente mais de façon peu significative au début.

Résistance à un changement d'échelle et à une rotation

Le nombre point ne change pas. La taille et l'orientation des segment est fondamentalement différente mais le nombre lui ne change pas. Les résultats pratique montrent ceci.

1.4 MAXSEG

Cet indicateur mesure la taille du plus grand segment délimitant l'enveloppe convexe. Elle divise cette taille par la taille totale du contour de l'enveloppe convexe. Cet indicateur est très grand pour *bone* ou *glass*. Notre indicateur n'est pas le plus satisfaisant. D'ailleurs, en regardant le graphique, c'est le graphique le moins intéressant.

Résistance au bruit

Le bruit interfère grandement sur cet indicateur car il peut couper notre segment le plus long en deux avec un seul point. Par contre nos résultat pratique montre une forte résistance au bruit. Cela est du au fait que sur l'image de nombreux segment ont cette taille. Donc même si l'un deux est altéré un autre ne l'est pas.

Résistance à un changement d'échelle et à une rotation

L'indicateur n'est pas sensible à une rotation. En tout cas, une rotation ne change pas l'indicateur en profondeur. De même pour un changement d'échelle, l'indicateur est normalisé pour y résister. Les résultats pratiques montrent une variation seulement au quatrième chiffre significatifs ce qui montre une résistance totale.

1.5 VARSEG

Cet indicateur calcule l'écart type de la longueur des segments. La moyenne de la longueur des segment ne nous a pas semblé intéressante au vus des tests effectués. L'écart type elle fait ressortir les petits détails du contour de l'enveloppe convexe. Elle montre à quel point l'enveloppe convexe est complexe. Pour cet indicateur, l'analyse des résultats est plus compliqué car on a pas de représentation graphique sur l'image de cet indicateur.

Résistance au bruit

Théoriquement l'analyse est la même que pour l'indicateur précédent. Les segment sont assez perturbés par le bruit et du coup, certain petit segment apparaissent ce qui augmente fortement la variance. Cette fois les résultats pratiques confirment la vulnérabilité au bruit.

Résistance à un changement d'échelle et à une rotation

De même que précédemment un changement d'échelle ou une rotation ne changent pas en profondeur l'ordure. Les perturbations restent minimales et sur l'enveloppe convexe. Les résultats pratique le montre.

1.6 ANGLESEG

Cet indicateur compte le nombre d'angle supérieur à la moitié de la moyenne. On ne peut pas calculer la moyenne car celle-ci vaut 360 divisé par le nombre d'angle qui est plus ou moins le nombre de point de l'enveloppe convexe.

Résistance au bruit

Le choix de cet indicateur et non de la variance est justement pour que le bruit qui va rajouter de petit angles ne soit pas problématique même si le nombre de point est affectés. On attend à quelque chose de relativement stable. Les résultats théorique prouve ceci. Les écarts sont relativement faibles sauf certains points.

Résistance à un changement d'échelle et à une rotation

La résistance aux rotation et au changement d'échelle est aussi assez concluantes car on travaille avec des angles. Les résultats pratiques confirme cette robustesse.

1.7 Distance transformation

Les quatre derniers indicateur proviennent de la DT (distance transformation). Le principe est de calculer la distance aux bord pour tout les point de l'image. De ceci on sort quatres indicateurs assez simples. MAXDTL2 est le maximum de la distance au bord pour la norme L2. MAXDTL1 est la même chose pour la norme L1. BARYCENTRE-DTL2 est la distance au bord pour la norme L2 du barycentre. BAR-CENTREINSCRIT est la distance entre la barycentre et le point où la distance au bord est maximal.

Chapitre 2

Calcul des images les plus proches

Pour calculer les k images les plus ressemblantes à une image donnée en entrée, le programme appelle la fonction *naiveDistance* dans le fichier *naivedistance.cpp*. L'algorithme est une séquence de 5 étapes distinctes. La première étape est la même que lorsque le programme calcule un indicateur de l'image : La composante connexe principale est extraite, dans le but de calculer le contour, et de ne pas prendre en compte des points éparpillés sur le domaine de l'image par du bruit. La deuxième étape est aussi similaire au programme normal, les objets géométriques utiles au calcul de la signature sont extraits de l'image. La troisième étape est de calculer toute la signature, donc les 10 indicateurs et non juste un seul.

Ensuite, toutes les signatures des 15 images de chacune des 70 classes sont obtenus à partir des résultats stockés dans les fichiers *resultat_liste_indn.dat*. La dernière étape est de calculer la distance entre l'image d'entrée et toutes les images de la base de donnée. Celles qui sont assez proches sont indexées dans un vecteur par ordre croissant. Les k plus ressemblantes images sont présentées par numéro de classe (celui qui est utilisé dans les fichiers *resultat_cat_indn.dat*) et par numéro dans la classe par ordre lexicographique : 10, 11, 12, 13, 14, 1, 20, 2, 3, 4...

La distance entre deux images est définie comme la somme des distances deux-à-deux des indicateurs des deux signatures. La distance entre deux images pour un indicateur donné est simplement la valeur absolue de la différence des valeurs de l'indicateur, divisée par leur moyenne. De cette manière, cette distance n'est pas sensible à l'échelle.