Fabrice El Gohary **Charles Letaillieur**

Indexation d'objets 3D : approche 2D-3D pour des requêtes à partir d'images

I Objectif

Le but du programme que nous avons développé est de pouvoir retrouver, à partir d'une ou plusieurs vues 2D d'un objet, l'objet correspondant dans une bibliothèque d'objets 3D. On peut imaginer l'application suivante de ce programme dans la vie courante : un visiteur prend une photo (de n'importe quelle direction) d'un objet dans un musée, la soumet à notre programme qui recherche l'objet correspondant dans la base de données du musée et donne le contexte historique propre à cet objet.

Notre programme prend en argument une ou plusieurs images ou même directement un objet 3D. Il compare ces données à la base de données des objets 3D.

II Principes de base

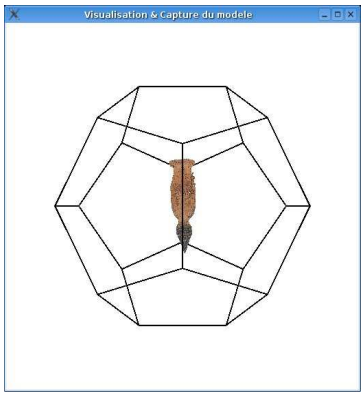
Nous nous basons ici sur les travaux de Chen, Tian, Shen et Ouhyoung qui ont développé une méthode très efficace pour la recherche d'objets 3D. Il s'agit d'une méthode 2D-3D, c'est-à-dire, que même lorsque l'on compare deux modèles 3D, la comparaison se fait via des projections en 2D de ces objets.

Cela permet avec la même méthode d'obtenir un moyen de soumettre, soit des images 2D soit un modèle 3D. Mais surtout, ce passage par la 2D permet une comparaison rapide des objets.

Pour que cette comparaison soit efficace, il ne faut pas, dans ce passage à la 2D, perdre la composante solide de l'objet, c'est-à-dire les relations entre les vues 2D. Ceci se comprend aisément en remarquant qu'une vue de face et de ¾ d'un même objet ne sont pas indépendantes.

Pour comparer les vues 2D entre elles, on utilise des valeurs caractéristiques des images qui sont les moments de Zernike.

Nous allons développer les détails de ces deux aspects, tel que les conçoivent Chen, Tian, Shen et Ouhyoung, d'abord puis, tel que nous les avons implémenté ensuite.



1) Quelles vues choisir pour l'indexation ?

Il faut, en premier lieu, « discrétiser » l'objet 3D. La méthode Chen, Tian, Shen et Ouhyoung considère la notion de « lightfield descriptor » qui peut se traduire par « positionnement des cameras ». Cette technique consiste à englober l'objet 3D dans un dodécaèdre et prendre des vues de l'objet depuis des sommets de ce dodécaèdre.

Cela permet d'obtenir des vues 2D uniformément réparties autour de l'objet. Mais le plus important réside dans la correspondance entre ces projections et le sommet du dodécaèdre dont elles sont issues.

En effet la comparaison de 2 objets 3D consiste à faire tourner ces « lightfield » pour retrouver la bonne orientation. On peut penser qu'un lightfield (un dodécaèdre) par objet suffirait mais pour balayer le mieux possible tout l'espace des vues réalisables de chaque objet, on utilise plusieurs lightfields (une dizaine typiquement) pour décrire chaque objet. Ils sont obtenus par décalage (par rotation) du lightfield de base.

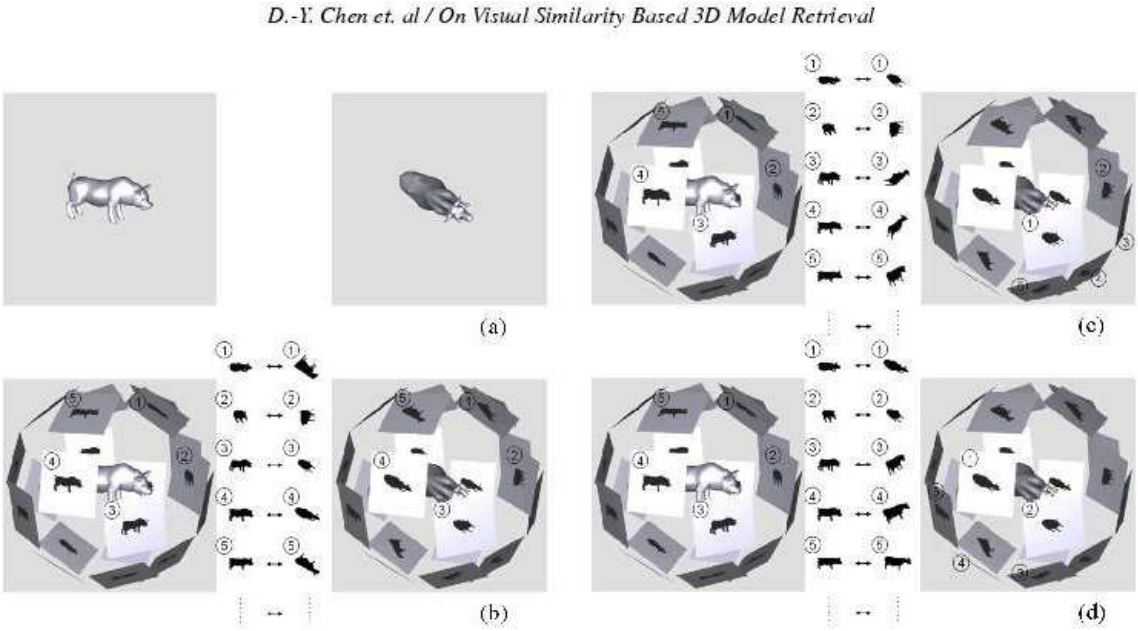


Figure 3: *Comparing LightField Descriptors between two 3D models*

2) Comparaison de 2 images silhouettes

Les vues 2D issues de la « discrétisation » précédente sont binarisées pour ne garder que la silhouette. Il faut ensuite trouver un moyen de caractériser numériquement ces images, pour n'avoir que des comparaisons de valeurs numériques (très rapides à effectuer) lors des comparaisons.

Pour décrire une forme de manière mathématique, il existe 2 types d'approche : l'approche basée sur les contours et une approche basée sur les régions.

L'approche basée sur les contours s'appuie sur les descripteurs de Fournier. Ils sont calculés par application de la transformée de Fourier à une signature de forme de l'image, la distance centroïde par exemple. Ces descripteurs étaient utilisés dans la méthode donnée pour affiner les résultats (et non en première approximation). Nous ne les avons pas utilisé nous ne attarderons pas sur eux. Nous nous concentrerons sur les descripteurs les plus discriminants.

Il s'agit de l'approche basée sur les régions, qui s'appuie sur les moments de Zernike.

Les moments de Zernike sont des moments orthogonaux complexe dont la magnitude possède une invariance par rapport à la rotation. Les moments de Zernike A_{nm} sont définis à l'intérieur du cercle unitaire, en utilisant le polynôme radial de Zernike R_{nm} et la fonction de Zernike V_{nm} :

$$V_{nm}(x, y) = V_{nm}(\rho\cos\theta, \rho\sin\theta) = R_{nm}(\rho)\exp(jm\theta)$$

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}\rho^{n-2s}$$

$$A_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x, y)V^*_{nm}(x, y),\ x^2 + y^2 \le 1$$

avec n et m des entiers strictement positifs respectant les contraintes suivantes : n − |m| est pair et |m| ≤ n.

Les moments de Zernike ont les propriétés suivantes : la magnitude des moments de Zernike est invariante aux rotations ; ils sont robustes au bruit et aux légères variations des formes ; il n'y a pas de redondance d'information car leurs bases sont orthogonales. Les moments de Zernike d'ordre faible représentent la forme globale, alors que les ordres plus élevés représentent les détails de la forme caractérisée. Ces moments sont uniquement invariants par rapport à la rotation.

Pour obtenir une invariance par rapport à la mise à l'échelle et la translation, l'image subie tout d'abord une étape de normalisation. Cette étape vise à nous assurer que ces moments seront calculés de la même manière sur toutes les images Les moments de Zernike, invariants à la rotation, sont ensuite extraits de cette image normalisée. L'invariance par rapport à la mise à l'échelle est obtenue en agrandissant puis en rétrécissant la forme de telle sorte que son moment d'ordre zéro A_{00} soit égal à une valeur p prédéfinie. L'invariance par rapport à la translation est enfin obtenue en déplaçant l'origine vers le centre juste avant de calculer les moments. Pour résumer, une fonction d'image f(x, y) est normalisée par rapport à la mise à l'échelle et la translation en la transformant en une fonction g(x, y) où g(x, y) = f(x/a + x, y/a + y) avec (x, y) le centre de f(x, y) et $a = p/A_{00}$.

L'approche de Chen, Tian, Shen and Ouhyoung prevoit d'utiliser au maximum 35 moments de Zernike d'ordre au plus 10.

Ces descripteurs étant invariants par rotation et symétrie axiale, nous pouvons nous limiter à 10 vues pour chaque lightfield :

- 1 vue par sommet puisqu'il y a invariance par rotation

- 10 sommets (du même côté d'un plan médian) du dodécaèdre seulement puisque les vues de dos et de face d'un objet (dans la même direction) sont identiques (à une symétrie axiale près).

On indexe d'abord toutes les images Ainsi, lorsque l'utilisateur soumettra une ou plusieurs image 2D, ce qui constitue le 2e temps de notre programme, celui-ci n'aura qu'à extraire les caractéristiques de cette image (en utilisant le même module que lors de l'indexation) et comparer ces valeurs à celles des images qui sont indexées dans la bibliothèque. La comparaison de ces valeurs aboutira à un indice de similarité qui indiquera l'objet 3D le plus proche de la requête. Si on soumet un objet 3D, des méthodes simplifiées permettent de réaliser plusieurs étapes de comparaison. Les premières sont grossières et élimines les objets les moins ressemblants.

La méthode de Chen, Tian, Shen et Ouhyoung permet à la fois, une recherche efficace d'un objet 3D à partir d'une vue 2D (via la « discrétisation » de l'espace et l'utilisation des descripteurs) mais surtout une comparaison rapide 3D-3D (via une comparaison subtile d'une série d'images 2D à une autre)

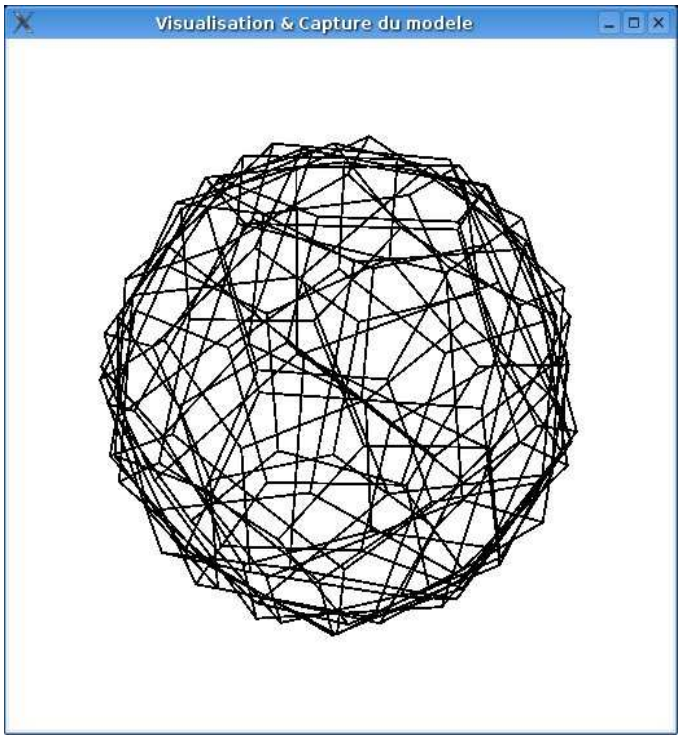
V Nos adaptations

Nous avons implémenté le corps du programme en C++. Nous avons aussi utilisé un programme développé par un élève de l'école pour importer dans OpenGL les objets de la bibliothèque 3D (fichier .tri). Ensuite, grâce aux fonctions d'OpenGL nous avons générer les vues en définissant les coordonnées des sommets des dodécaèdres à considérer ainsi que l'orientation de la camera en ces points.

Ensuite les fichiers sont exportés sous forme de matrice binaire pour les traitements ultérieurs réalisés par d'autres fonctions implémentées en C++.

Nous avons adapté en de nombreux points la méthode Chen & al.

Tout d'abord l'article était assez obscur sur la manière de choisir les « lightfields » (les hexagones légèrement décalés par rapport à celui initial). Nous avons choisi une solution très simple à implémenter, qui consiste à tourner de 30° autour des axes (x,y,z) du repère. Nous observons que cette approche simpliste (une approche plus fine consisterait à determiner les rotations éloignant « le plus possible » le nouveau lightfields des précédents) aboutit à une répartition des points de vue dans l'espace relativement homogène.



Autre point sur lequel nous nous sommes un peu éloignés de la méthode de Chen & Al. : le système de comparaison des différents lightfields. Dans le cas où l'utilisateur soumet une image 3D, la méthode prévoyait pour optimiser la recherche, de ne comparer certaines vues, en première approximation. Cela se faisait en 6 étapes :

1) Comparaison 8 coefficients du moment de Zernike codés sur 4 bits pour 3 images de 2 Light Field Descriptor à toute la bibliothèque. Elimination des plus mauvais candidats.
2) Comparaison 16 coefficients du moment de Zernike codés sur 4 bits pour 5 images de 2 Light Field Descriptor aux candidats restants. Elimination des plus mauvais candidats.
3) Comparaison 16 coefficients du moment de Zernike codés sur 4 bits pour 10 images de 7 Light Field Descriptor aux candidats restants. Elimination des plus mauvais candidats.
4) Comparaison de tous les coefficients du moment de Zernike codés sur 4 bits pour toutes les images de tous les Light Field Descriptor aux candidats restants. Selection des 16 meilleurs comparaisons.
5) Comparaison de tous les coefficients du moment de Zernike codés sur 8 bits pour les 16 meilleurs comparaisons selectionnées précédemment.
6) Comparaison complète (moment de Zernike et descripteurs de Fourier) pour les derniers cas.

Nous avons conserver le principe essentiel de en comparer que des séries de 10 images mais nous ne sommes pas rentrée dans les détails des ces 6 étapes (en particulier nous avons codé tous les moments en float, donc en 32 bits). Ensuite nous avions prévu de suivre, les 4 étapes de comparaison suivantes.

Etape	Nombre d'objets pris en compte dans la Base de Données	Nombre de vues pour chaque objet de la BD	Nombre de vues pour l'objet en entrée	Nombre de comparaisons d'images
1	Tous	1LF (60 positions)	Une position de 1LF	60*10 = 600
2	¾	Tous les LF	Une position de 1LF	8*600 = 4 800
3	½	Tous les LF	1LF	60*4 800 = 288 000
4	¼	Tous les LF	Tous les LF	8* 288 000 = 2 304 000

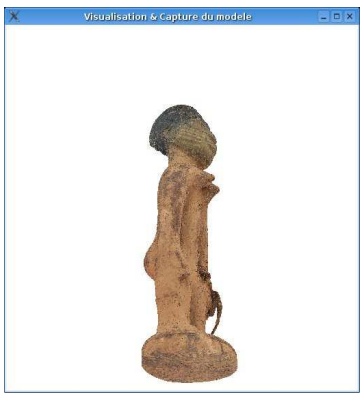
Mais les résultats expérimentaux nous ont montré que les deux premières étapes suffisaient à obtenir de bons résultats (en un temps de calcul très faible).

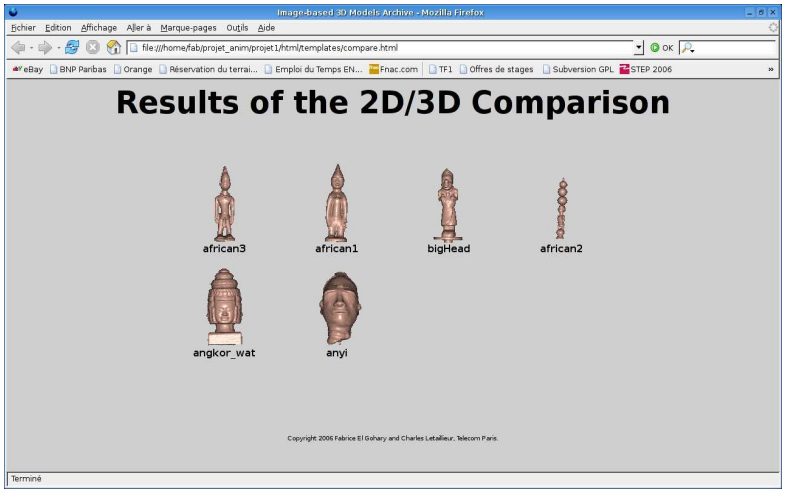
On prend le minimum de la somme des distances euclidiennes des moments de Zernike pour toutes nos comparaisons.

V Nos résultats

Nous avons eu une difficulté récurrentes : il était très difficile de tester les modules indépendamment les uns des autres. En particulier, il est très difficile de trouver des valeurs de moments de Zernike avec les images correspondantes, pour vérifier que le module calculant ces moments est correct.

Mais nous avons obtenu des résultats très satisfaisant à partir d'objet 2D et 3D.





VI Conclusion et perspectives

Nous sommes d'abord satisfait d'avoir réussi à implémenter une forme de la méthode de Chen, Tian, Shen et Ouhyoung. La méthode est très puissant et les résultats obtenues sont assez impressionnants.

Nous avons beaucoup appris sur le traitement d'images en 3D, tant sur la théorie (quaternions, utilisation des dodécaèdres pour « discrétiser » l'espace) que sur l'implémentation (OpenGL). La partie de reconnaissance des formes a aussi été très instructive.

Il reste pourtant de nombreuses fonctions possibles à implémenter : vérifier les résultats sur un plus grand nombres d'objets 3D, réduire le temps de calcul en se rapprochant de la méthode de Chen & Al, le rendre utilisable en ligne (serveur Unix),…

Bibliographie :

- Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen and Ming Ouhyoung, "On Visual Similarity Based 3D Model Retrieval", /EUROGRAPHICS/, Granada, Spain, Sept. 2003.

- Mohamed Chaouch, « Indexation 3D : Descripteurs 2D/3D basés sur les images de profondeur », Avril - Septembre 2005, Université Pierre & Marie CURIE, Master Recherche d'Informatique en Intelligence Artificielle et Décision Thématique Image Multimédia et Sons, Unité de recherche INRIA Rocquencourt.

- Dengsheng Zhang and Guojun Lu, « An Integrated Approach to Shape Based Image Retrieval », Gippsland School of Computing and Information Technology, Monash University Churchill, Australia

http://www.mathcurve.com/polyedres/dodecaedre/dodecaedre.shtml