# 行政院國家科學委員會專題研究計畫　成果報告

## 三維模型之建構與其中軸之表現及物件快速擷取(2/2)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-002-015-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：歐陽明

報告類型：完整報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 11 月 4 日

# 行政院國家科學委員會補助專題研究計畫期中進度報告

## 三維模型之建構與其中軸之表現及物件快速擷取(2/2)

計畫編號:NSC92-2213-E-002-015

全程計畫:民國 91 年 8 月 1 日至民國 93 年 7 月 31 日
本年度計畫:民國 92 年 8 月 1 日至民國 93 年 7 月 31 日

計畫主持人:歐陽明 台灣大學資訊工程學系教授

# 計劃中文摘要

關鍵字：三維模型建構，中軸，三維模型搜尋比對，網格最佳化，漸進式網格，隱含平面，輻射基底函數

三維模型資料在現今生活各層面中扮演著重要的角色，也是所有工程摸擬工具與商業動畫及網路電動玩具的基礎。

本計劃為二年期計劃，其主旨為開發一三維模型資料的處理系統，並由此系統建立出來之模型資料進行有關中軸(medial axis)、骨幹(skeleton)建構，與三維模型搜尋的研究。

本三維模型處理軟體函式庫係由以下四個模組所構成：

(1) 建構模組：此模組可將從三維雷射測距儀掃描物體所得到的多個深度影像(range image)，加以對齊、合併以建出原本物體的三維模型。
(2) 最佳化處理模組：此模組將得到的模型做最佳化(optimization)或漸進式(progressive)的處理，以利於壓縮儲存或是網路傳輸。除外，我們利用輻射基底函數(radial basis functions)建立三維模型的隱含曲面(implicit surface)。
(3) 三維模型搜尋比對模組：在中軸、骨幹建構方面，我們利用模型絕對平面函式，以微分幾何的方式建立梯度向量流(gradient vector flow)，並根據此向量流做為物體收縮建立骨幹的依據。在本計劃中，我們將針對不同種類的演算法建立中軸或骨幹並加以比較。在三維模型搜尋方面，利用以物體中軸或是骨幹做為搜尋時的主要關鍵,我們用以發展一搜尋演算法使得運用此的搜尋系統可以針對不同目標（如物體形狀，走向，顏色等）加以合併搜索。就文獻來看,穩定可靠的產生中軸方式為本計劃之特色。而在短時間內可從上千

個三維物件中找到最相關者，則為本計劃之軟體產出。
(4) 特效模組(special effects)：如三維模型輪廓(silhouette)，計算表面曲率等小型輔助模組。

# 計劃英文摘要

Keywords: 3D Model Construction, Medial Axes, 3D Model Search and Retrieval, Mesh Optimization, Progressive Mesh, Implicit Surface, Radial Basis Function

3D models nowadays become an important part of our daily life, and are the fundamental building blocks of engineering simulation tools, commercial animations, and networking computer games.

It is our goal to develop a 3D model information processing system in two years. Besides, using the generated models from the previous system, we will continue the researches on 3D object medial axis generation and its application in 3D model retrieval.

The proposed 3D model information processing system contains the following four software modules:

(1) Model construction module:
This module registers and merges the range images acquired from the 3D laser scanner and construct the original object 3D model.

(2) Mesh optimization module:
This module optimizes the 3D model for compression. Besides, the progressive mode of the 3D model will be constructed.

(3) 3D model retrieval module:
We use radial basis function to construct the implicit surface of a 3D model. In the generation of medial axis and skeleton, we invoke the implicit surface function to build the gradient vector flow, and use this vector flow to be guidelines in the construction of object skeleton. In this project, we will develop algorithms for medial axis and skeleton extraction and analyze them.

In 3D model retrieval, we use the medial axis or skeleton as the primary key of the search system. We will develop an algorithm to search 3D object with

different target (such as topology, shape, color and so on).

According to the survey, reliable and faithful extraction of medial axis or skeleton for 3D objects is a very difficult task, and will be the major feature of our system. The deliverable part of the project is a system that can retrieve from up to thousands of 3D models in reasonably short time, given one target 3D model.

(4) Special effects module:
Modules for special functionality will be developed such as 3D model silhouette finding, curvature calculation.

## 本計畫之目標

本計畫的主要目標是設計一三維模型建立及處理系統，並利用此系統產生的三維模型從事下面各項研究：

A. 中軸及骨架萃取
B. 在大資料庫中搜尋並取得三維模型

## 一、 執行進度

第一年為止，我們已將本計劃的三個主要方法執行完畢：

1. 三維模型建構方法 (3D model construction)。
2. 藉由輻射基底函數之隱含曲面建構方法 (Implicit surface construction with radial basis functions)。
3. 藉由隱含曲面之物體骨架擷取方法 (Skeleton extraction with a implicit surface)。

第二年的進度，

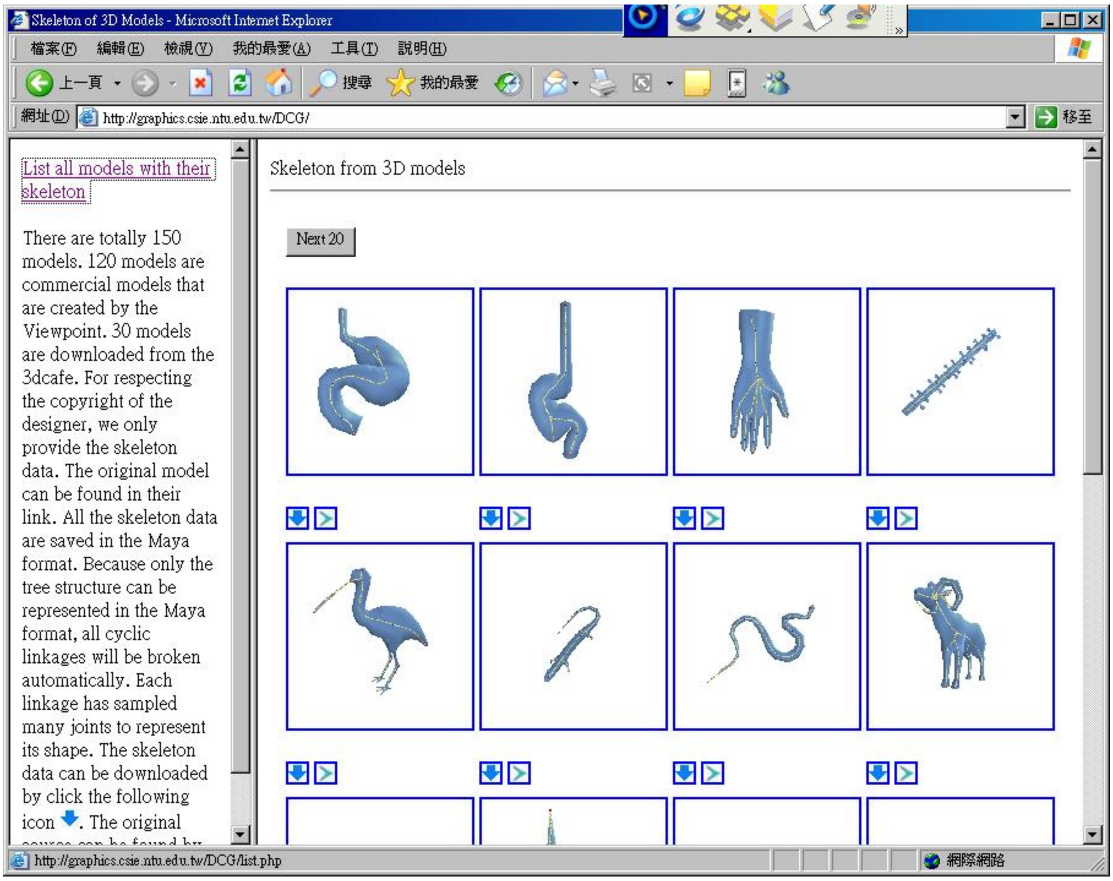
1. 中軸及骨架萃取

我們現在已建立一個網站 http://www.cmlab.csie.ntu.edu.tw/~joyce/skeleton/ 將一百五十多個已經產生中軸的三維模型加上對映的中軸以著名三維建模動畫繪圖軟體 Maya 之格式，對外發表，並有一篇論文已投稿至 The Visual Computer。

Fu-Che Wu, Wan-Chun Ma, Rung-Huei Liang, Bing-Yu Chen, Ming Ouhyoung,

"Domain Connected Graph: the Essential Skeleton of a 3D Shape", submitted to The Visual Computer, 2004.

上述網頁之結果，如下圖。論文則附在本報告之後。




2. 在大資料庫中搜尋並取得三維模型，

我們已發展一套系統名為“3D Model Retrieval System”見 http://3d.csie.ntu.edu.tw。此系統包含全世界所有的網路上可公開的 3D Model，剔除重複的，約有一萬一千多個 3D Mesh Model。可在「側面剪影」模式下，0.1 秒找到按相似度排序之目標，或直接由“從 3D Object 找 3D Model”模式下，2 秒之內完成搜尋。其結果之精確度，比現今 State of the art 的 Princeton 大學 Thomas Funkhouser 團隊(ACM Tr. Graphics and Visualization, Feb. 2003)之成果還超過 23% 。也因此本人受邀與其合作，共同設立 3D model test bed，結合兩校之資料庫，成為世界最完整之 3D Model 資料庫與測試平台，同時本人也利用 sabbatical leave 至 Princeton Univ.作 Visiting Fellow 一學期（9/2003～12/2003）。

發表之三篇論文如下，其中第一篇已被今年的 SIGGRAPH paper (ACM Transaction on Graphics)所引用，另外含 IEEE Transaction on Multimedia 等之其他論文所引用。論文也附在本報告之後。

(1) Ding-Yun Chen, Ming Ouhyoung, Xiao-Pei Tian, Yu-Te Shen, "On Visual Similarity Based 3D Model Retrieval", Computer Graphics Forum 2003, pp. 223 - 232, (also to appear in EuroGraphics2003, Spain). (SCI)

(2) Fu-Che Wu, Bing-Yu Chen , Rong-Hui Liang, Ming Ouhyoung, "Prong Features Detection of a 3D Model Based on the Watershed Algorithm", in ACM SIGGRAPH2004 Sketches, Los Angeles, California, USA., August, 2004.

(3) Shuen-Huei Guan, Ming-Kei Hsieh, Chia-Chi Yeh, and Bing-Yu Chen. Enhanced 3D Model Retrieval System through Characteristic Views using Orthogonal Visual Hull. ACM SIGGRAPH 2004 Conference Abstracts and Applications (Posters Program), Los Angeles, California, USA., 2004.

第一年之成果發表之論文如下:

W.-C. Ma, F.-C. Wu and M. Ouhyoung. *Skeleton Extraction of 3D Objects with Radial Basis Functions*. Shape Modeling International 2003, Seoul, Korea.

在前述論文中，沒有完全包含之部分，分述如後三節。

## 二、 三維模型建構方法 (3D model construction)

在三維模型的建構方面，我們採用了在國科會計劃 NSC 90-2213-E-002-089 所發展的技術來進行三維資料擷取、網格對齊 (mesh registration)、網格接合 (mesh integration) 與網格最佳化 (mesh optimization)。詳細演算法請參見國科會報告書。

## 三、 藉由輻射基底函數之隱含曲面建構方法 (Implicit surface construction with radial basis functions)

### 3.1 隱含曲面

目前有越來越多的研究單位對於隱含曲面有更深的探討[TURK99][YNGVE99]。隱含曲面提供了建構複雜的幾何物件的另外一種選擇，由於隱含曲面為一可微分之數學式，在應用上的價值遠比單用三角形與節點來描述一個物體形狀來的大的多。我們定義一個隱含曲面為一個三維的函式

$$f(x) = c \text{ ， } x \in \Re^3$$

我們可以將空間中任何的一點代入到此函式中，假設計算出來的結果等於 $c$ 的話，則我們可得知此點是位在此隱含曲面所代表的物體上。一個簡單的隱含曲面例子：

$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

此函式即代表著一個在三度空間中一個以座標 (0,0,0) 為原點，半徑長為 1 的球。簡單形狀的隱含曲面可以用人來計算，但是如何建立複雜形狀的隱含曲面則

是一大難題。目前許多研究朝向以利用非線性內差 (non-linear interpolation) 來建立任意物體的隱含曲面。而在其中，以輻射基底函數 (radial basis functions, RBFs) 來進行非線性內差則為最常見的方法。

## 3.2 利用輻射基底函數來建立隱含曲面

主要的計算過程如下：

1. 將三維模型上面所有的點 (vertices) 當作輸入。假設此模型有 $N$ 個點，則我們建立出來的隱含曲面有著以下的結構：

$$f(x) = \sum_{i=1}^{2N} d_i \phi(|x - x_i|) + s(x) \text{ ， } x \in \Re^3$$

2. 在模型上的點 (on-surface point) 集合 $V$ 與每個點其法向量集合分別定義為：

$$M = \{m_1, ..., m_N\} \text{ ， } V = \{V_1, ..., V_N\}$$

3. 藉由 V 與 M，我們定義在模型外的點 (off-surface point) 為：

$$L = \{l_1, ..., l_N\} = V + w * M \text{ ， } w > 0$$

4. 我們定義 $x = \{V, M\}$ 為隱含曲面的邊界限制(boundary constraints)。
5. 對於基底函數的選擇，有以下幾種：

$$\phi(r) = r^2 \log(r)$$
$$\phi(r) = \exp(-cr^2)$$
$$\phi(r) = 1/\sqrt{r^2 + c^2}$$

6. 隱含曲面會滿足以下的矩陣運算式：

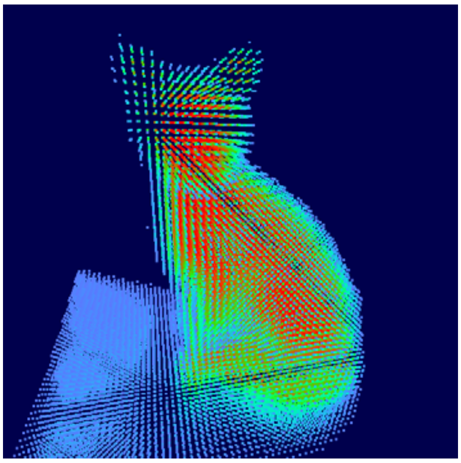
$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} d \\ s \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

$$A_{i,j} = \phi(|x_i - x_j|)$$
$$P_i = x_i$$
$$x_i \in V \rightarrow f_i = 0$$
$$x_i \in M \rightarrow f_i = 1$$

我們利用已知的 A, P, f 解未知的輻射基底函數參數 d 及一次項參數 s。當 d 與 s 都知道之後，則此隱含曲面：

$$f(x) = \sum_{i=1}^{2N} d_i \phi(|x - x_i|) + s(x)$$

即建立出來。我們可以將任意的三維空間點 $p = (x, y, z)$ 代入公式所得到的值 $c = f(p)$ 來判斷說此三維空間點是在模型的表面、內部、或是外部。圖為一波斯貓模型的隱含曲面採樣結果。

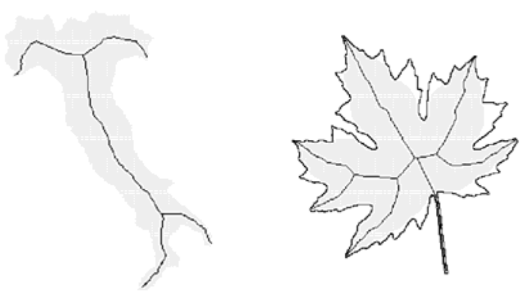


圖：波斯貓模型的隱含曲面。圖中的點為三維的採樣點（此圖是在 $1\times1\times1$ 的正立方體空間中，以每邊切 40 等份來做採樣）。點的顏色越接近藍色代表求出來的值越接近 0，越往紅色代表值越大。

## 四、 藉由隱含曲面之物體骨架擷取方法 (Skeleton extraction with an implicit surface)

### 4.1 骨架簡介

骨架的概念是由 H. Blum [BLUM73] 根據中軸轉換 (Medial Axis Transform, MAT) 和對稱軸轉換 (Symmetry Axis Transform, SAT) 所提出。中軸轉換為物體上的每一點找出最接近它的邊界點（可能不只一個），若一個內部點能找到兩個以上邊界點，則此內部點就在骨架上。骨架化的目的是萃取出一物體地域性的形狀特徵。此外，骨架應該要用代表一物體最小維度的表示法。例如，二維圖片的骨架應該是一維的線段，三維模型的骨架則應該由一維的線段和二維的表面組成。

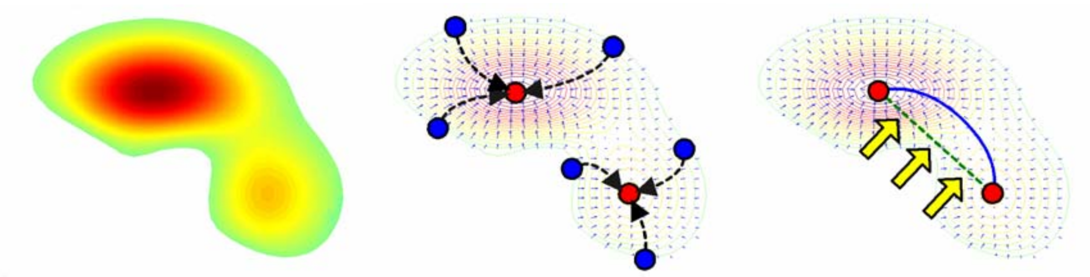


圖：（左）義大利領土圖形與其中軸。（右）楓葉圖形與其中軸。

由於從中軸轉換所建立的骨架通常是十分雜亂的(中軸轉換會受到物體表雜訊的影響，所以我們在這邊藉由不同的想法來提出我們對於骨架的想法。許多研究報告，如 [LAZARUS99] 等，利用所謂的 Reeb graph [REEB46] 來產生物體的一維骨架。Reeb graph 骨架保留了物體大致的幾何形態與拓樸關係。因此，在最近幾年，Reeb graph 骨架是一個十分熱門的課題。我們所提出的骨架概念，除了相同的保留物體的拓樸關係，更進一步的比 Reeb graph 保留更多的幾何資訊。以下將簡介我們所提出的骨架擷取演算法。

4.2 骨架擷取演算法

本段將敘述由我們所提出之骨架擷取演算法，主要的內容已發表為國際會議論文 [MA03]。本演算法分為三個主要步驟：A. 利用梯度下降法 (gradient descent) 取得局部極小值，B. 建立局部極小值之相鄰關係，C. 利用利用主動輪廓模型 (Active Contour Model) 建立骨架。以下我們將分別介紹這部分：



圖：骨架擷取演算法之說明簡圖。(左)隱含曲面，顏色越深代表隱含曲面值高。(中)對於每個在物體表面的點，利用梯度下降法來取得局部極小值位置。(右)建立局部極小值之相鄰關係，並利用主動輪廓模型 (Active Contour Model) 來連結局部極小值。

**A. 利用梯度下降法來取得局部極小值位置**

藉由上述提及的隱含曲面可微分之特性，我們可以建立其梯度場：

$$g(x) = \nabla f(x)$$

我們利用所有在三維模型上的節點 (vertices) 當作起始點，我們利用此梯度場進行梯度下降：

$$x' = x + g(x) \times step$$

當所有的節點都收斂完畢的時候，此步驟即停止。這些因收斂而得到的局部極小值的位置，在我們的研究中被視為骨架上的重要節點。這些位置通常會有兩種在骨架上的意義：一是關節點 (branching point)，一是端點 (termination point)。

**B. 建立局部極小值之相鄰關係**

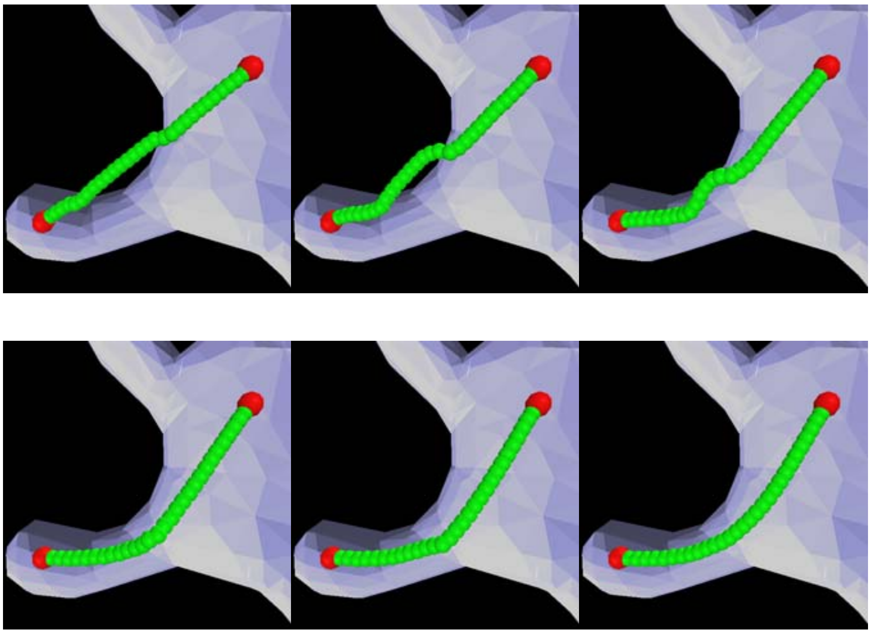
當我們獲得了這些局部極小值位置後，接下來我們就要判斷兩個局部最小值是否有相鄰關係。我們定義了一個簡單直覺的方法：假設 $P_A$ 與 $P_B$ 為三維模型上的兩個獨立節點（收斂起始點），其對應的局部最小值分別為 $M_A$ 與 $M_B$，則如果 $P_A$ 與 $P_B$ 有相鄰關係（即 $P_A$ 與 $P_B$ 之間有一三角形邊相連），$M_A$ 與 $M_B$ 也會有相鄰關係（如果 $M_A$=$M_B$，則也算相鄰）。對於每個局部最小值，我們都去做這樣的分析，即可建立出一個骨架圖 (skeleton graph)。

**C. 利用主動輪廓模型 (Active Contour Model) 來連結局部極小值**

當我們擷取出骨架圖之後，最後的工作就是讓骨架圖裡的每個連結都變形成其所屬區域的形狀。在此我們利用主動輪廓模型 [KASS87] 來做形變的動作。主動輪廓模型為一個將能量最小化 (energy minimization) 的演算法。主要的構想是，將每個連結都視為一個彈簧，在此彈簧上的每個點同時受到兩種力量的拉扯：外部力 (external force) 與內聚力 (internal force)。外部力為彈簧外部力場所影響的力量，在我們這邊為則就是隱含曲面的梯度場。內聚力則為彈簧兩兩相連的點的引力，我們這邊將相連的點的內聚力利用虎克定律來描述。所以整個彈簧的能量為：

$$E_{acm}(x) = w_1 \times E_{external}(x) + w_2 \times E_{internal}(x)$$

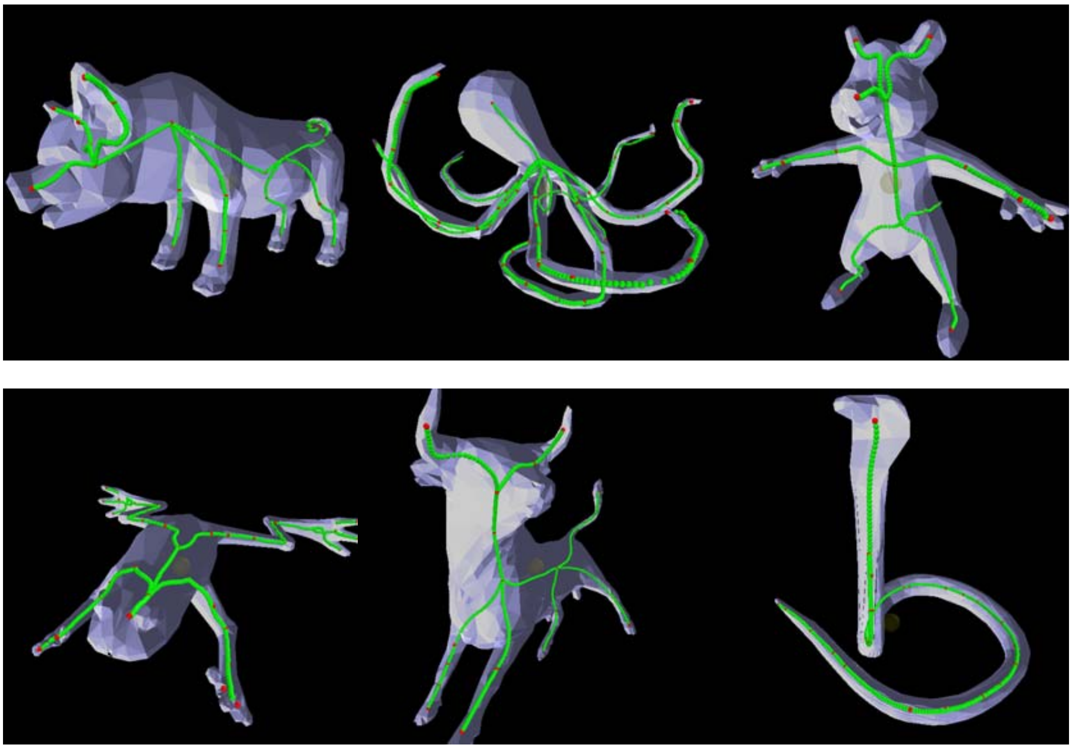
將此彈簧的能量調整到最低，此彈簧的位置即可被認為是最後骨架連結變形出來的結果。我們將每個骨架連結都利用主動輪廓模型來做變形的動作。最後的結果即符合我們所提出的骨架。



圖：主動輪廓模型收斂過程的連續圖。

## 4.3 相關結果

再此我們列出六個利用本演算法所計算出來的骨架。我們將原本的三維模型繪製成藍色半透明狀、骨架為綠色、所有的局部極小值點為紅色。我們可以見得我們提出的骨架保留了物體大部分的幾何資訊與形態 (morphological) 資訊。但是目前的結果仍有改進的地方，例如由於我們是利用輻射基底函數來建立隱含曲面，但是利用此方法建立出來的曲面跟原本的三維模型間會有一定的誤差(尤其是在物體較細微的部分)，所以在一些較精細的部分，比如老鼠的手的部分，通常這邊的幾何資訊會因為隱含曲面的問題而具有不小的誤差，因此這邊所建立的骨架沒有辦法完整的表達其形態上的資訊。



圖：利用本演算法所擷取出來的骨架資訊。

本演算法的執行資訊如下：

| 模型名稱 | 總邊界<br>限制數 | RBF 半徑 | 建立<br>隱含曲面<br>(秒) | 梯度下降<br>(秒) | 主動輪廓<br>模型變形<br>(秒) |
|---|---|---|---|---|---|
| 豬 | 2170 | 0.1 | 74.83 | 74.28 | 294.30 |
| 章魚 | 2004 | 0.1 | 57.44 | 41.25 | 344.62 |
| 鼠 | 1912 | 0.1 | 51.05 | 44.80 | 182.45 |
| 蛙 | 1890 | 0.06 | 48.14 | 21.93 | 312.27 |
| 牛 | 1982 | 0.2 | 55.93 | 79.96 | 210.33 |
| 蛇 | 1752 | 0.2 | 39.17 | 62.13 | 110.04 |

## 五、 研究成果

本研究成果至今已發表了一篇國際期刊論文以及三篇國際會議論文。其中有兩篇在今年的 SIGGRAPH 會議中，以 Sketch paper 及 Poster paper 的方式呈現。另外，還有一篇國際期刊論文已投稿在審查中。

隨著本計畫，我們還建立了兩個網站，一個做為三維搜尋引擎(http://3d.csie.ntu.edu.tw)，另一個做為三維骨架(中軸)成果之展現(http://www.cmlab.csie.ntu.edu.tw/~joyce/skeleton/)。 此兩個網站，都受到國際相關研究人員之重視，也常常放入他們發表論文的 reference 中。上述論文擇其要者附在本報告之後。

## 六、 參考文獻

[BLUM67] H. Blum. *A Transformation for Extracting New Descriptors of Shape*, pp. 362－380. MIT Press, 1967.

[KASS87] M. Kass, A. Witkin and D. Terzopoulos. *Snakes: Active Contour Models*. International Journal of Computer Vision, 1:321–331, 1987.

[LAZARUS99] F. Lazarus and A. Verroust. *Level Set Diagrams of Polyhedral Objects*. Proceedings of the ACM Symposium on Solid Modeling and Applications, pp. 130–140, 1999.

[MA03] W.-C. Ma, F.-C. Wu and M. Ouhyoung. *Skeleton Extraction of 3D Objects with Radial Basis Functions*. Shape Modeling International 2003, Seoul, Korea.

[REEB46] G. Reeb. *Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numerique*. Comptes Rendus Acad. Science Paris, 222:847–849, 1946.

[TURK99] G. Turk and J. F. O'Brien. *Variational Implicit Surfaces*. Technical Report GIT-GVU-99-15, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, 1999.

[YNGVE99] G. Yngve and G. Turk. *Creating Smooth Implicit Surfaces from Polygonal Meshes*. Technical Report GIT-GVU-99-42, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, 1999.

[WML04] Fu-Che Wu, Wan-Chun Ma, Rung-Huei Liang, Bing-Yu Chen, Ming Ouhyoung, *Domain Connected Graph: the Essential Skeleton of a 3D Shape*, submitted to The Visual Computer, 2004.

[COT03]Ding-Yun Chen, Ming Ouhyoung, Xiao-Pei Tian, Yu-Te Shen, *On Visual Similarity Based 3D Model Retrieval*, Computer Graphics Forum 2003, pp. 223 - 232, (also to appear in EuroGraphics2003, Spain). (SCI)

[WCL04] Fu-Che Wu, Bing-Yu Chen , Rong-Hui Liang, Ming Ouhyoung, *Prong Features Detection of a 3D Model Based on the Watershed Algorithm*, in ACM SIGGRAPH2004 Sketches, Los Angeles, California, USA., August, 2004.

[GHY04] Shuen-Huei Guan, Ming-Kei Hsieh, Chia-Chi Yeh, and Bing-Yu Chen. *Enhanced 3D Model Retrieval System through Characteristic Views using Orthogonal Visual Hull*. ACM SIGGRAPH 2004 Conference Abstracts and Applications (Posters Program), Los Angeles, California, USA., 2004.

# On Visual Similarity Based 3D Model Retrieval

Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen and Ming Ouhyoung

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
{dynamic, babylon, edwards}@cmlab.csie.ntu.edu.tw, ming@csie.ntu.edu.tw

**Abstract**

*A large number of 3D models are created and available on the Web, since more and more 3D modelling and digitizing tools are developed for ever increasing applications. The techniques for content-based 3D model retrieval then become necessary. In this paper, a visual similarity-based 3D model retrieval system is proposed. This approach measures the similarity among 3D models by visual similarity, and the main idea is that if two 3D models are similar, they also look similar from all viewing angles. Therefore, one hundred orthogonal projections of an object, excluding symmetry, are encoded both by Zernike moments and Fourier descriptors as features for later retrieval. The visual similarity-based approach is robust against similarity transformation, noise, model degeneracy etc., and provides 42%, 94% and 25% better performance (precision-recall evaluation diagram) than three other competing approaches: (1)the spherical harmonics approach developed by Funkhouser et al., (2)the MPEG-7 Shape 3D descriptors, and (3)the MPEG-7 Multiple View Descriptor. The proposed system is on the Web for practical trial use (http://3d.csie.ntu.edu.tw), and the database contains more than 10,000 publicly available 3D models collected from WWW pages. Furthermore, a user friendly interface is provided to retrieve 3D models by drawing 2D shapes. The retrieval is fast enough on a server with Pentium IV 2.4GHz CPU, and it takes about 2 seconds and 0.1 seconds for querying directly by a 3D model and by hand drawn 2D shapes, respectively.*

Categories and Subject Descriptors (according to ACM CCS): H.3.1 [Information Storage and Retrieval]: Indexing Methods

## 1. Introduction

Recently, the development of 3D modeling and digitizing technologies has made the model generating process much easier. Also, through the Internet, users can download a large number of free 3D models from all over the world. This leads to the necessities of a 3D model retrieval system. Although text-based search engines are ubiquitous today, multimedia data, such as 3D models, usually lacks meaningful description for automatic matching. The MPEG group aims to create an MPEG-7 international standard, also known as "Multimedia Content Description Interface", for the description of multimedia data [11]. However, little description is about 3D models. The need of developing efficient techniques for content-based 3D model retrieval is increasing.

To search 3D models that are visually similar to a queried model is the most intuitive way. However, most methods concentrate on the similarity of geometric distributions rather than directly searching for visually similar models. The geometric-based approach is feasible since much appearance for an object is controlled by its geometry. In this paper, however, we present a novel approach that matches 3D models using their visual similarities, which are measured with image differences in light fields. We take this approach to better fit the goal of comparing models that appear to be similar to a human observer. The concept of the visual similarity-based approach is similar to that of the image-driven simplification, proposed by Lindstrom and Turk [14].

The geometry-based approach is broadly classified into two categories: shape-based and topology-based matching. The shape-based approach uses the distribution of vertices or polygons to judge the similarity between 3D models [1, 2, 4, 5, 6, 7]. The challenge of the shape-based approach is how to define shape descriptors, which need to be sensitive, unique, stable, efficient, and robust against similarity transformations of various kinds of 3D models. The topology-based approach utilizes topological structures of 3D models to measure the similarity between them [3]. The difficulties of the topology-based approach include automatic topology extraction from all types of 3D models, and the discrimination between topologies from different categories. Each of the two approaches has its inherent merits and demerits. For example, a topology-based approach leads to high similarity between two identical 3D models with different gestures, whereas a shape-based approach cannot. On the other hand, a shape-based approach results in high similarity between 3D models with different connections among parts, whereas a topology-based approach cannot. For instance, both a finger and the shoulder of a human model are parts of a human body. The topologies are quite different whether the finger does or does not connect to a human body, but the shapes are similar.

Most previous works of 3D model retrieval focused on defining suitable features for the matching process [1∼13], and were based on either statistical properties, such as global shape histograms, or the skeletal structures of 3D models. Osada et al. [2] proposed and analyzed a method for computing shape signatures of arbitrary 3D polygonal models. The key idea is to represent the signature of a 3D model as a shape distribution, which is a histogram created from the distance between two random points on a surface for measuring global geometric properties. The approach is simple, fast and robust, and could be applied as a pre-classifier in a complete 3D model retrieval system.

Funkhouser et al. [1] proposed a practical web-based search engine that supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. For 3D shape queries, a new matching algorithm that uses spherical harmonics to compute similarities is developed. It does not require repair of model degeneracy or alignment of orientations. In their system, a multimodal query is applied to increase the retrieval performance by combining features such as text and 3D shapes. It is also fast enough to retrieve from a repository of 20,000 models in less than one second.

Hilaga et al. [3] proposed a technique in which the similarity between polyhedral models is accurately and automatically calculated by comparing the skeletal and topological structure. The skeletal and topological structure decomposes a 3D model to a one-dimensional graph structure. The graph is invariant to similarity transformations, robust against simplification and deformation caused by changing posture of an articulated object, etc. In their experimental results, the average search time from 230 3D models is about 12 seconds with a Pentium II 400MHz processor. Another 3D model retrieval system [10] , having 445 models in the database, is extended from the work of Hilaga et al., and takes about 12 seconds on a server with Pentium IV 2.4 GHz processor.

In this paper, a novel visual similarity-based approach for 3D model retrieval is proposed, and the system is also available on the web for practical trial use. The proposed approach is robust against similarity transformations, noise and model degeneracy, etc. There are more than 10,000 3D models in our database, and a user-friendly interface is provided for 3D model retrieval by drawing 2D shapes, which are taken as one or more projection views.

In general, a retrieval system contains off-line feature extraction and on-line retrieval processes. We introduce the *LightField Descriptor* to represent 3D models, which is detailed in Section 2, as well as the feature extraction. In Section 3, comparing 3D models is represented for the on-line retrieval process. The experimental results and the performance evaluations are shown in Section 4. Section 5 concludes the write up.

## 2. Feature Extraction for Representing 3D Models

The proposed descriptor used for comparing the similarity among 3D models is extracted from 4D light fields, which are representations of a 3D object. The phrase light field describes the radiometric properties of light in a space and was coined by Gershun [23]. A light field (or plenoptic function) is traditionally used in image-based rendering and is defined as a five dimensional function that represents the radiance at a given 3D point in a given direction [24,25]. For a 3D model, the representation is the same along a ray, so the dimension of the light field around an object can be reduced to 4D [25,14]. Each 4D light field of a 3D model is represented by a collection of 2D images, which are rendered from a 2D array of cameras. The camera positions of one light field can be put either on a flat surface [25] or on a sphere [14] in the 3D world. The light field representation has not only been used in image-based rendering, but also in image-driven simplification by Lindstrom and Turk [14], whose approach uses images to decide which portions of a model to simplify.

In this paper, we extract features from the light fields rendered from cameras on a sphere. The main idea of using the approach to get the similarity between two models is introduced in Section 2.1. To reduce size of the features and speed up the matching process, the cameras of the light fields are distributed uniformly and positioned on vertices of a regular dodecahedron. We name the descriptor *LightField Descriptor*, and describe it in Section 2.2. In Section 2.3, we show that one 3D model is represented by a set of *LightField Descriptors* in order to improve the robustness against rotations while comparing between two models. One that is also important is the image metric used, and is detailed in Section 2.4. Finally, the flow chart of extracting the *LightField Descriptors* for a 3D model is summarized in Section 2.5.

### 2.1. Measuring similarity between two 3D models

The main idea comes from the following statement, "If two 3D models are similar, they also look similar from all viewing angles." Accordingly, the similarity between two 3D models can be measured by summing up the similarity from all corresponding images of a light field. However, what

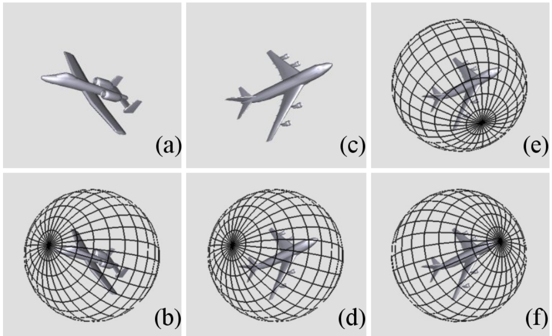



**Figure 1:** *The main idea measuring similarity between two 3D models*

must be considered is the transformation, including translation, rotation and scaling. The translation and the scaling problems are discussed in Section 2.5 and ignored by our image metric described in Section 2.4. As for rotation, the key to this problem is visual similarity. The camera system surrounding each model is rotated until the highest overall similarity (cross-correlation) between the two models from all viewing angles is reached. Take Figure 1 as an example, where (a) and (c) are two different airplanes with inconsistent rotations. First, for the airplane in Figure 1 (a), we place the cameras of a light field on a sphere, as shown in Figure 1 (b), where cameras are put on the intersection points of the sphere. Then, cameras of this light field can be applied, at the same positions, to the airplane in Figure 1 (c), as shown in Figure 1 (d). By summing up the similarities of all pairs of corresponding images in Figure 1 (b) and (d), the overall similarity between the two 3D models is obtained. Next, the camera system in Figure 1 (d) can be rotated to a different orientation, such as Figure 1 (e), which leads to another similarity value between the two models. After evaluating similarity values, the correct corresponding orientation, in which the two models look most similar from all corresponding viewing angles, can be found, such as Figure 1 (f). The similarity between the two models is defined as summing up the similarity from all corresponding images between Figure 1 (b) and (f).

However, the computation will be very complicated and impractical to a 3D model retrieval system using current processors. Therefore, the camera positions of a light field are distributed uniformly on vertices of a regular dodecahedron, such that reduced camera positions are used for approximation.

### 2.2. A *LightField Descriptor* for 3D models

To reduce the retrieval time and the size of features, the light field cameras can be put on 20 vertices of a regular dodecahedron. That is, there are 20 different views, which are distributed uniformly, over a 3D model. The 20 views can

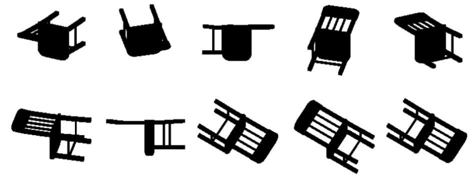


**Figure 2:** *A typical example of the 10 silhouettes for a 3D model*

roughly represent the shape of a 3D model, as been applied similarly in previous works. Huber and Hubert [27] proposed an automatic registration algorithm, which is able to reconstruct real world objects from 15 to 20 various viewpoints from a laser scanner. Lindstrom and Turk [14] also employ the 20 views in comparing 3D models for image-driven simplification.

Since their applications are different from the retrieval system, the requirements of rendering image and the image metric used are also different. First, lighting is different while rendering images of an object. We turn all lights off, so that the rendered images will be silhouettes only, which enhance the efficiency and the robustness of image metric. Second, orthogonal projection is applied in order to speed up the retrieval process and reduce the size of features. Therefore, ten different silhouettes are produced for a 3D model, since the silhouettes projected from two opposite vertices on the dodecahedron are identical. Figure 2 shows a typical example of the 10 silhouettes of a 3D model. In our implementation, the rendered image size is 256 by 256 pixels. Consequently, the rendering process can filter out high-frequency noise of 3D models, and also make our approach reliable from degeneracy of meshes, such as those missing, wrongly-oriented, intersecting, disjoint and overlapping polygons.

Since the cameras are placed on the vertices of a fixed regular dodecahedron, we need to rotate the camera system 60 times (to be explained below), so that the cameras can be switched onto different vertices, while measuring the similarity between descriptors of two 3D models. The dissimilarity, $D_A$, between two 3D models is defined as:

$$D_A = \min_i \sum_{k=1}^{10} d\left(I_{1k}, I_{2k}\right), \qquad i = 1..60 \qquad (1)$$

where $d$ denotes the dissimilarity between two images, defined in Section 3.1, and $i$ denotes different rotations between camera positions of two 3D models. For a regular dodecahedron, each of the 20 vertices is connected by 3 edges, which results in 60 different rotations for one camera system (mirror mapping is not available). $I_{1k}$ and $I_{2k}$ are corresponding images under $i$-th rotation.

Here is a typical example to explain our approach. There are two 3D models, a *pig* and a *cow*, in Figure 3 (a), both

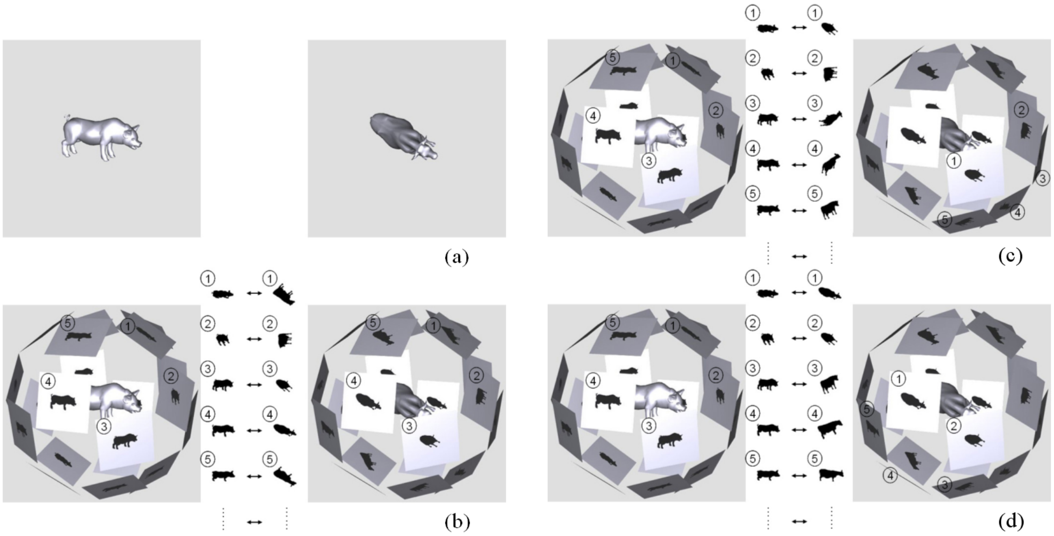



**Figure 3:** *Comparing LightField Descriptors between two 3D models*

rotated randomly. First, 20 images are rendered from vertices of a dodecahedron for both the 3D model. As shown in Figure 3 (b), we compare all the corresponding 2D images from the same viewing angles, such as, the order 1∼5 between *pig* and *cow* model. Thus we get a similarity value under this rotation of camera system. Then, we map the order 1∼5 differently as in Figure 3 (d), and get another similarity value. After repeating this process, we find a rotation of camera positions with the best similarity (cross-correlation being highest), as shown in Figure 3 (d). Therefore, the similarity between the two models is the summation of the similarities among all the corresponding images.

Consequently, the *LightField Descriptor* is defined as the basis representation of a 3D model, and is defined as features of 10 images rendered from vertices of dodecahedron over a hemisphere. A *LightField Descriptor* somehow eliminates the rotation problem, but this is not exact enough. Therefore, a set of light fields is applied to improve the robustness.

### 2.3. A set of *LightField Descriptors* for a 3D model

To be robust against rotations among 3D models, a set of *LightField Descriptors* is applied to each 3D model. If there are $N$ *LightField Descriptors*, which are created from different camera system orientations for both 3D models, there are $(N \times (N-1)+1) \times 60$ different rotations between the two models. Therefore, the dissimilarity, $D_B$, between two 3D models is then defined as:

$$D_B = \min D_A(L_j, L_k), \qquad j, k = 1..N \tag{2}$$

where $D_A$ is defined in Equation (1), and $L_j$ and $L_k$ are light field descriptors of two models, respectively.

The relationship of the $N$ light fields needs to be carefully set to ensure that all the cameras are distributed uniformly and able to cover different viewing angles to solve the rotation problem effectively. The approach of generating the evenly distributing camera positions of the $N$ light fields comes from the idea in relaxation of random points proposed by Turk [15]. The process can be pre-processed, and then all 3D models use the same distributed light fields to generate corresponding descriptors. In our implementation, we set $N = 10$, as shown in Figure 4, that is, the similarity between two 3D models is obtained from the best one of 5,460 different rotations. Therefore, the average maximum error of rotation angle between two 3D models is about 3.4 degree in longitude and latitude. That is:

$$\frac{180^\circ}{x} \times \frac{360^\circ}{x} = 5460 \;\Rightarrow\; x \cong 3.4^\circ \tag{3}$$

which is small enough for our 3D model retrieval system according to our experimental results. Of course, the number

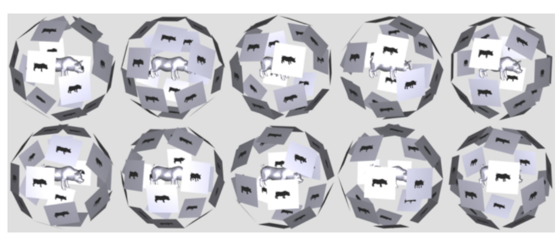


**Figure 4:** *A set of LightField Descriptors for a 3D model*

*N* can be bigger than 10, and we will evaluate in the future the saturation effect when *N* becomes bigger.

### 2.4. Image metric

An image metric is a function measuring the distance between two images. Recently, Content-Based Image Retrieval (CBIR) has become a popular research, and different image metrics have been proposed [19~22]. Many approaches of image metrics are robust against transformations such as translation, rotation, scaling, and image distortion.

One of the important features of images is the shape descriptor, which can be broadly classified into region-based and contour-based descriptor. The use of a combination of different shape descriptors has been proposed recently in order to improve the retrieval performance [21, 22]. In this paper, we adopt an integrated approach proposed by Zhang and Lu [21], which combines a region shape descriptor (Zernike moments descriptor) and a contour shape descriptor (Fourier descriptor). The Zernike moment descriptor is also used in MPEG-7, which is named *RegionShape* descriptor [11]. Different shape signatures have been used to derive Fourier descriptor, and the retrieval using Fourier Descriptors derived from the centroid distance has significantly higher performance than those of the other methods. These were also compared by Zhang and Lu [20]. The centroid distance function is expressed by the distance to boundary points from the centroid of the shape. The boundary points of a shape are extracted through a contour tracing algorithm, proposed by Pavlidis [31]. Figure 5 shows a typical example of the centroid distance. Figure 5 (a) shows a 2D shape rendered from a viewpoint of a 3D model, and the contour tracing result is shown in Figure 5 (b). Figure 5 (c) shows the centroid distance of (a).

Sometimes, however, a 3D model might be rendered into several separated 2D shapes, as shown in Figure 6 (a). When this situation occurs, the following two stages are applied to connect. First, we apply Erosion operation [32] from one to several times to connect the separated parts, as shown in Figure 6 (b). Second, a thinning algorithm [32] is applied in order to connect the separated parts, as shown in Figure 6 (c). Note that the pixels of rendered 2D shape cannot be removed during the thinning algorithm. The separated parts are then connected, and the high-frequency noise will be filtered out by the Fourier descriptor. But if there are still separated parts after running the Erosion operation for several times, a bounding box algorithm will replace the first stage above.

There are 35 coefficients for Zernike moment descriptor and 10 coefficients for Fourier descriptor. Each coefficient is quantized to 8 bits in order to reduce the size of descriptors and accelerate the retrieval process. Consequently, the approach is robust against translation, rotation, scaling and image distortion, and is very efficient.

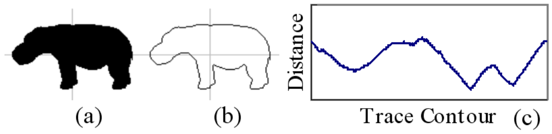



**Figure 5:** *A typical example of the centroid distance for a 2D shape*

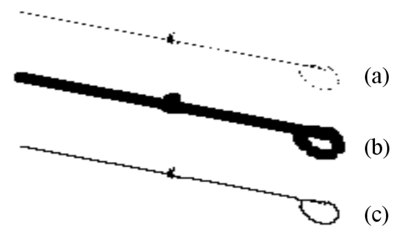



**Figure 6:** *Connection of different parts of 2D shapes*

### 2.5. Steps of extracting *LightField Descriptors* for a 3D model

The steps of extracting the *LightField Descriptors* for a 3D model are summarized in the following.

(1) Translation and scaling are applied first to ensure that 3D model is entirely contained in rendered images. The input 3D model is translated from the center of the model to the origin of world coordinate system. The axis is then scaled such that the maximum length is 1.

(2) Render images from the camera positions of light fields, as described in Section 2.3.

(3) For a *LightField Descriptor*, 10 images are represented for 20 viewpoints, and are in a pre-defined order for storage. For a 3D model, 10 descriptors are created, so that totally 100 images should be rendered.

(4) Descriptors for a 3D model are extracted from the 100 images, as in Section 2.4.

## 3. Retrieval of 3D Models Using *LightField Descriptors*

In the off-line process mentioned in last section, the *LightField Descriptors* of each 3D model in the database are calculated and stored for 3D model retrieval. This section details the on-line retrieving process, which compares the descriptors of the queried one with all the other 3D models in the database. Comparing the *LightField Descriptors* within two models is described in Section 3.1. Those who are greatly dissimilar to the queried model will be rejected early in the process, detailed in Section 3.2, which accelerates the retrieval with a large database. Practically, when a user wants to retrieve 3D models, he/she can upload a 3D model as a query key. However, early experiences of Funkhouser et al. [1] suggest that even a simple gesture interface, such as Teddy system [26], is still too hard for novice and casual users to learn quickly. They proposed that drawing 2D shapes with a painting program to retrieve 3D models is intuitive for users. In this paper, a user-friendly drawing interface for 3D model retrieval is also provided. The approach of comparing 2D shapes with 3D models is described in Section 3.3.

### 3.1. Similarity between *LightField Descriptors* of two 3D models

The retrieving process can be referred as calculating the similarity one by one between the queried one and each of the models in the database and showing those similar to the queried one. The similarity between two models is defined as summing up the similarity from all the corresponding images, as described in Section 2.3. The comparison of two descriptors is as Equation (2). When comparing the dissimilarity, $d$, of corresponding images, we use simple $L1$ distance to measure:

$$d(J,K) = \sum_i |C_{1i} - C_{2i}| \tag{4}$$

where $C_1$ and $C_2$ denote coefficients of two images, and $i$ denotes the index of their coefficients. There are 45 coefficients for each image, each quantized to 8 bits. To simplify the computation, a table is created and stored for the value of $L1$ distance from 0 to 255. Thus, a table-look-up method is used to speed up the retrieval process.

### 3.2. Retrieval of 3D models from database with large number of models

For a 3D model retrieval system, a database with a large number of models should be considered. For example, there are over 10,000 3D models in our database. To efficiently retrieve 3D models from an enormous database, an iterative early-jump-out method is applied in our 3D model retrieval system. First, when comparing the queried model with all the others, only parts of the images and coefficients are used. This can remove almost half of the models. The threshold of removing models is set as the mean of the similarity rather than the median, since the calculation of the mean is simpler. Then, compare the queried model to the remainder models using more images and coefficients. Repeat the above steps in several times. All iterations are detailed as follows.

(1) In the initial stage, all 3D models in the database are compared with the queried one. Two *LightField Descriptors* of the queried model are compared with ten of those in the database. Three images of each light field are compared, and each image is compared using 8 coefficients of Zernike moment. Each coefficient is quantized to 4 bits.

(2) In the second stage, five *LightField Descriptors* of the queried model are compared to ten of the others in the database. Five images of each light field are compared, and each image is compared using 16 coefficients of Zernike moment. Each coefficient is quantized to 4 bits.

(3) Thirdly, seven *LightField Descriptors* of queried model are compared with ten of the others in the database. The other is the same as the second stage, while another five images of each light field are compared.

(4) The fourth stage is the same as full comparison, but only the Zernike moment coefficients, quantized to 4 bits, are used. In addition, the top 16 of the 5,460 rotations are recorded between the queried one and others.

(5) Each coefficient of Zernike moment is quantized to 8 bits, and the retrieval uses the top 16 rotations, recorded from the $4^{th}$ stage, rather than 5,460 rotations.

(6) In the last stage, each coefficient of Fourier descriptor is added to the retrieval.

The approach speeds up the retrieval process by early rejection of non-relevant models. The query performance and robustness of each step will be evaluated in the future.

### 3.3. A user friendly interface to retrieve 3D models from 2D shapes

Creating a queried model for retrieval is not easy and fast for general users. Thus, a user-friendly interface, a painting program, is provided in our system. Furthermore, users can again utilize the retrieved model to find more specific 3D models, since 2D shapes carry less information than a 3D model. To sum up, our 3D model retrieval system is easy for a novice user.

Recognizing 3D objects from single 2D shape is an interesting and difficult problem, and has been researched long time ago. Dudani et al. [16] identified aircrafts with moment invariants derived from the boundary of 2D shapes. They captured 2D images of 3D objects from every 5 degrees of azimuth and roll angle. 3,000 images for 6 aircrafts are used for comparison with an input 2D shape. A 3D aircraft recognition algorithm of Wallace and Wintz [17] used 143 projections to represent an aircraft over the hemisphere, and performed the recognition using normalized Fourier descriptors of the 2D shape boundary. Cyr and Kimia [18] proposed 3D object recognition by generating "equivalent view" from different positions on the equator for 65 3D objects. They recognized an unknown 2D shape by comparing all views of 3D objects using shock matching, which takes about 5 minutes. Recently, Funkhouser et al. [1] proposed a search engine for 3D models, which also provides a 2D drawing interface for 3D model retrieval. The boundary contours are rendered from 13 viewpoints for each 3D model, and then additional 13 shape descriptors are created. In our 3D model retrieval system, it is intuitive and direct to compare 2D shapes with 3D models, since the descriptors for 3D models are composed of features of 100 2D shapes over the hemisphere, as

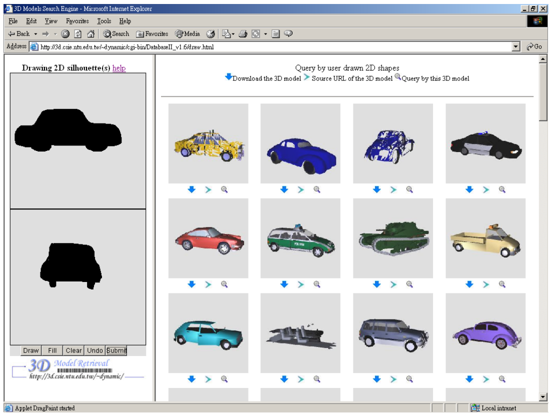


**Figure 7:** *Retrieval results from user drawn 2D shapes*

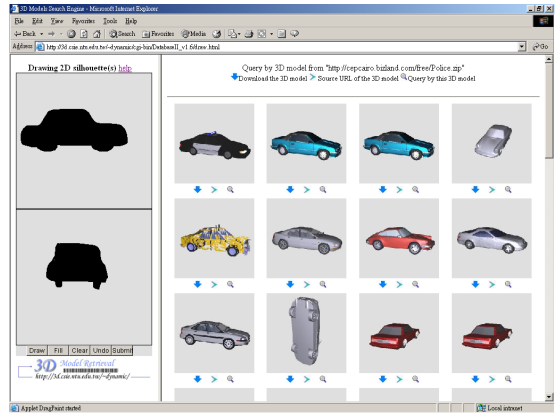


**Figure 8:** *Retrieval results from interactively searching of selecting a 3D model from Figure 7*

described in Section 2.3. The image metric we used is defined in Section 2.4.

## 4. Experimental Results

In Section 4.1, the proposed 3D model retrieval system is demonstrated. The performance and robustness of the approach are evaluated in Section 4.2 and 4.3, respectively.

### 4.1. The proposed 3D model retrieval system

The 3D model retrieval system is on the following web site for practical trial use: *http://3d.csie.ntu.edu.tw*. There are 10,911 3D models in our database now, all free downloaded via the Internet. Users can query with a 3D model or drawing 2D shapes, and then search interactively and iteratively for more specific 3D models using the first retrieved results. Models are available for downloading from the hyperlink of their original downloaded path listed in the retrieval results. Figure 7 shows a typical example of a query with 2D drawing shapes, and Figure 8 shows the interactive search by selecting a 3D model from Figure 7.

The system consists of off-line feature extraction in preprocessing and on-line retrieval processes. In the off-line process, the features are extracted in a PC with a Pentium III 800MHz CPU and GeForce2 MX video card. On the average, each 3D model with 7,540 polygons takes 6.1 seconds to extract features, detailed in Table 1. Furthermore, the average time of rendering and feature extraction for a 2D shape takes 0.06 seconds. Extracting features are suitable for both 3D model and 2D shape matching. No extra effort should be done for 2D shapes. In the on-line process, the retrieval is done in a PC with two Pentium IV 2.4GHz CPUs. Only one CPU is used for the query at one time, and the retrieval takes 2 and 0.1 seconds with a 3D model and two 2D shapes as the query keys, respectively.

### 4.2. Performance Evaluation

Traditionally, the diagram of "Precision" vs "Recall" is a common way of evaluating performance in documental and visual information retrieval. Recall measures the ability of the system to retrieve all models that are relevant. Precision measures that the ability of the system to retrieve only models that are relevant. They are defined as:

$$Recall = \frac{relevant\ \ correctly\ \ retrieved}{all\ \ relevant}$$

$$Precise = \frac{relevant\ \ correctly\ \ retrieved}{all\ \ retrieved}$$

In general, the recall and precision diagram requires a ground truth database to assess the relevance of models with a set of significant queries. Test sets are usually large, but only a small fraction of the relevant models are included [30]. Therefore, a test database with 1,833 3D models is used for evaluation. The test database contains free 3D models from 3DCafe [34], downloaded in Dec. 2001, but removes several models with failed formats in decoding. One student independent of this research, regarded as a human evaluator, classified the models according to functional similarities. The test database was clustered into 47 classes including 549 3D

| | Average | Standard Deviation | Minimum | Maximum |
|---|---|---|---|---|
| Vertex | 4941.6 | 13582.8 | 4 | 262882 |
| Polygon | 7540.7 | 21003.8 | 2 | 519813 |
| Time | 6.11 sec | 4.38 sec | 2.32 sec | 48.93 sec |

**Table 1:** *Vertex and polygon number of the 10,911 3D models and the feature extraction time from a PC with a Pentium III 800 MHz CPU*

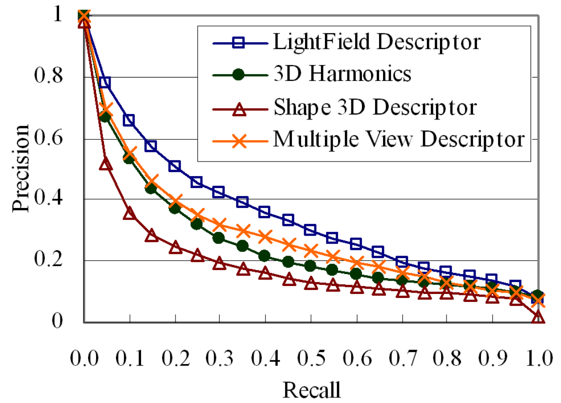



**Figure 9:** *Performance evaluation of our approach,* LightField Descriptor*, and those of others.*

models mainly for vehicle and household items (such as categories of airplane, car, chair, table, etc.), and all the other 1,284 models are classified as "miscellaneous".

To compare the performance with others systems, three major previous works are implemented as follows:

(1) 3D Harmonics: This approach is proposed by Funkhouser et al. [1], and outperforms many other approaches, such as Moments [12], Extended Gaussian Images [8], Shape Histograms [9] and D2 Shape Distributions [2], which are evaluated in their paper. The source code of SpharmonicKit 2.5 [35], also used in their implementation, is used for computing the spherical harmonics.

(2) Shape 3D Descriptor: The approach is used in MPEG-7 international standard [11], and represents a 3D model with curvature histograms.

(3) Multiple View Descriptor: This method aligns 3D objects with Principal Component Analysis (PCA) [33], and then compares images from the primary, secondary and tertiary viewing directions of principal axes. Descriptor of the viewing directions is also recorded in MPEG-7 international standard [11], but does not limit the usage of image metrics. To get better performance, integration with different image metrics described in Section 2.4 are used. Furthermore, for calculating PCA correctly from vertices, each 3D model is resampled first to ensure that vertices are distributed evenly on the surface.

Figure 9 shows the comparison of the retrieval performance of our approach, *LightField Descriptors*, with those of the others. Each curve plots the graph of "recall and precision" averaged over all 549 classified models in the test database. Obviously, *LightField Descriptor* performs better than the others. The precision values are 42%, 94% and 25% higher than those of 3D Harmonics, Shape 3D Descriptor and Multiple View Descriptor, respectively, after comparing and averaging over all the "recall" axis.

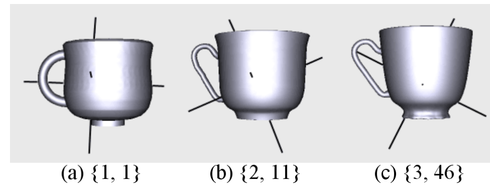



**Figure 10:** *Three similar cups with their principal axes, orienting the models in different directions. Retrieval results of querying are done by the model in (a). The first number in bracket shows the queried number by our method, and the second number shows the Multiple View Descriptor.*

However, in our implementation of 3D Harmonics, the precision is not as good as that indicated in the original paper [1], shown in Table 2. Evaluating by different test database is one possible reason, and another one may lie in a small amount of different details between our implementation and original paper, even if we try to implement the same as the original paper. The test database used in the original paper is also purchased by us, and will be evaluated in the future. As for PCA applied to Multiple View Descriptor, Funkhouser et al. [1] found that principal axes are not good while aligning orientations of different models within the same class, and also demonstrated this problem using 3 mugs. Retrieval with similar examples in our test database is shown in Figure 10. Clearly, our approach works well against this particular problem of PCA.

### 4.3. Robustness evaluation

All the classified 3D models in the test database are applied to the following evaluation in order to assess the robustness. Each transformed 3D model is then used for queries from the test database. The average recall and precision of all 549 classified models are used for the evaluation. The robustness is evaluated by the following transformation:

(1) Similarity transformation: For each 3D model, seven random numbers are applied to x-, y-, and z-axis rotations (from 0 to 360 degree), x-, y- and z-axis translations (-10∼+10 times of the length of the model's bounding box), and scaling (a factor of -10∼+10).

(2) Noise: Each vertex of 3D model is applied three ran-

| Recall | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|---|
| Our approach | 0.51 | 0.42 | 0.36 | 0.30 | 0.25 | 0.20 |
| 3D Harmonics | 0.37 | 0.27 | 0.22 | 0.18 | 0.16 | 0.14 |
| 3D Harmonics with different test database [1] | 0.41 | 0.33 | 0.26 | 0.20 | 0.17 | 0.14 |

**Table 2:** *Precision of 3D Harmonics in the original paper for comparison.*

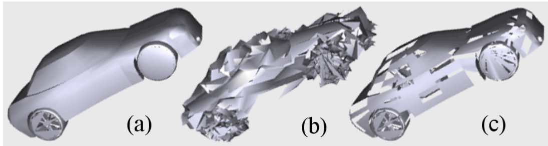



**Figure 11:** *Robustness evaluation of (b) noise and (c) decimation from (a) original 3D model*

dom number to x-, y- and z-axis translation (-3%~+3% times of the length of the model's bounding box). Figure 11 (b) shows a typical example of the effect.

(3) Decimation: For each 3D model, randomly select 20% polygons to be deleted. Figure 11 (c) shows a typical example of the effect.

Experimental result of the robustness evaluation is shown in Figure 12. Clearly, our approach is robust against similarity transformation, noise and decimation.

## 5. Conclusion and Future Works

In this paper, a 3D model retrieval system is proposed based on visual similarity. The new metric based on a set of *Light-Field Descriptors* is proposed for matching among 3D models. The visual similarity-based approach is robust against translation, rotation, scaling, noise, decimation and model degeneracy etc. A practical retrieval system that includes more than 10,000 3D models is available on the web for expert and novice users, and the retrieval can be done in less than 2 seconds on a server with Pentium IV 2.4 GHz CPU. A friendly user interface is also provided to query by drawing 2D shapes. The experimental results demonstrate that our approach outperforms 3D Harmonics, MPEG-7 Shape 3D Descriptor and Multiple View Descriptor.

In future work, several investigations are described as follows. First, other image metric for 2D shapes matching may be evaluated and included to improve the performance. In addition, the image metric for color and texture [11] can also be included to retrieval 3D model using more visual features. Second, different approaches ("cocktail" approach) can be combined to improve the overall performance. Third, the mechanism of training data or active learning [12, 13] may be used to adjust the weighting among different features. Finally, partial matching from several objects takes a long time to compute in general, and is also an important and difficult research direction in the future work [28, 29].

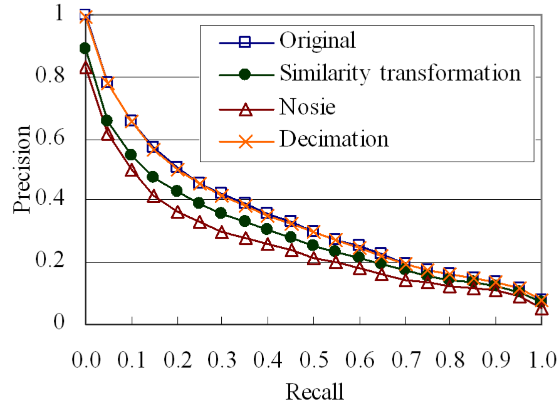



**Figure 12:** *Robustness evaluation of similarity transformation, noise and decimation*

## Acknowledgements

We would like to thank Miss Wan-Chi Luo, a graduate student, to help us manually classify the 1833 3DCafe objects into 47 classes plus another class "miscellaneous". We also appreciate Jeng-Sheng Yeh to set up the web server for our 3D model retrieval system. This project is partially funded by the National Science Council (NSC, Taiwan) under the grant number NSC91-2622-E-002-040 and by the Ministry of Education under the grant number 89-E-FA06-2-4-8.

## References

1. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin and D. Jacobs, "A Search Engine for 3D Models", *ACM Transactions on Graphics*, **22**(1):83-105, Jan. 2003.
2. R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, "Shape Distributions", *ACM Transactions on Graphics*, **21**(4):807-832, Oct. 2002.
3. M. Hilaga, Y. Shinagawa, T. Kohmura and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", *Proc. of ACM SIGGRAPH*, 203-212, Los Angeles, USA, Aug. 2001.
4. E. Paquet, M. Rioux, A. Murching, T. Naveen and A. Tabatabai, "Description of Shape Information for 2-D and 3-D Objects", *Signal Processing: Image Communication*, **16**:103-122, Sept. 2000.
5. R. Ohbuchi, T. Otagiri, M. Ibato and T. Takei, "Shape-Similarity Search of Three-Dimensional Models Using Parameterized Statistics", *Proc. of 10th Pacific Graphics*, 265-273, Beijing, China, Oct. 2002.
6. D. V. Vranic and D. Saupe, "Description of 3D-Shape using a Complex Function on the Sphere", *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 177-180, Lausanne, Switzerland, Aug. 2002.
7. I. Kolonias, D. Tzovaras, S. Malassiotis and M. G. Strintzis, "Fast Content-Based Search of VRML Models based on Shape Descriptions", *Proc. of Interna-*

*tional Conference on Image Processing (ICIP)*, 133-136, Thessaloniki, Greece, Oct. 2001.

8. B. K.P. Horn, "Extended Gaussian Images", *Proceedings of the IEEE*, **72**(12):1671-1686, 1984.
9. M. Ankerst, G. Kastenmuller, H.-P. Kriegel and T. Seidl, "3D Shape Histograms for Similarity Search and Classification in Spatial Databases", *Proc. of 6th International Symposium on Advances in Spatial Databases (SSD)*, Hong Kong, China, 207-228, 1999.
10. D.-Y. Chen and M. Ouhyoung, "A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph", *Proc. of Computer Graphics Workshop*, 16, Tainan, Taiwan, June 2002.
11. S. Jeannin, L. Cieplinski, J. R. Ohm and M. Kim, *MPEG-7 Visual part of eXperimentation Model Version 7.0, ISO/IEC JTC1/SC29/WG11/N3521*, Beijing, China, July 2000.
12. M. Elad, A. Tal, and S. Ar. "Content Based Retrieval of VRML Objects - An Iterative and Interactive Approach", *Proc. of 6th Eurographics Workshop on Multimedia*, 97-108, Manchester UK, Sept. 2001.
13. C. Zhang and T. Chen, "An Active Learning Framework for Content-Based Information Retrieval", *IEEE Transactions on Multimedia Special Issue on Multimedia Database*, **4**(2):260-268, June 2002.
14. P. Lindstrom and G. Turk, "Image-Driven Simplification ", *ACM Transactions on Graphics*, **19**(3):204-241, July 2000.
15. G. Turk, "Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion", *Computer Graphics (Proc. of ACM SIGGRAPH)*, **25**(4):289-298, July 1991.
16. S. A. Dudani, K. J. Breeding and R. B. McGhee, "Aircraft Identification by Moment Invariants", *IEEE Transactions on Computers*, **C-26**(1):39-46, Jan. 1977.
17. T. P. Wallace and P. A. Wintz, "An Efficient Three-Dimensional Aircraft Recognition Algorithm Using Normalized Fourier Descriptors", *Computer Graphics and Image Processing*, **13**:99-126, 1980.
18. C. M. Cyr, B. B. Kimia, "3D Object Recognition Using Shape Similarity-Based Aspect Graph", *Proc. of International Conference on Computer Vision (ICCV)*, 254-261, Vancouver, Canada, July 2001.
19. C. E. Jacobs, A. Finkelstein, D. H. Salesin, "Fast Multiresolution Image Querying", *Proc. of ACM SIGGRAPH*, 277-286, Los Angeles, USA, Aug. 1995.
20. D. S. Zhang and G. Lu. "A comparative Study of Fourier Descriptors for Shape Representation and Retrieval". *Proc. of 5th Asian Conference on Computer Vision (ACCV)*, 652-657, Melbourne, Australia, Jan. 2002.
21. D. S. Zhang and G. Lu. "An Integrated Approach to Shape Based Image Retrieval". *Proc. of 5th Asian Conference on Computer Vision (ACCV)*, 652-657, Melbourne, Australia, Jan. 2002.
22. D. Heesch and S. Ruger, "Combining Features for Content-Based Sketch Retrieval - A Comparative Evaluation of Retrieval Performance", *Proc. of 24th BCS-IRSG European Colloquium on IR Research Glasgow (LNCS 2291)*, UK, Mar. 2002
23. A. Gershun, "The Light Field", Moscow, 1936. Translated by P. Moon and G. Timoshenko in *Journal of Mathematics and Physics*, **18**:51-151, MIT, 1939.
24. L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", *Proc. of ACM SIGGRAPH*, 39-46, Los Angeles, USA, Aug. 1995.
25. M. Levoy and P. Hanrahan, "Light Field Rendering", *Proc. of ACM SIGGRAPH*, 31-42, New Orleans, USA, Aug. 1996.
26. T. Igarashi, S. Matsuoka and H. Tanaka, "Teddy: A Sketching Interface for 3D Freeform Design", *Proc. of ACM SIGGRAPH*, 409-416, Los Angeles, USA, Aug. 1999.
27. F. Huber and M. Hebert, "Fully Automatic Registration of Multiple 3D Data Sets", *Proc. of IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS)*, Kauai, Hawaii, USA, Dec. 2001.
28. A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(5):433-449, May 1999.
29. S. M. Yamany and A. A. Farag, "Surfacing Signatures: An Orientation Independent Free-Form Surface Representation Scheme for the Purpose of Objects Registration and Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(8):1105-1120, Aug. 2002.
30. A. E. Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, Inc., 1999.
31. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, 1982.
32. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Pub. Co., 1992.
33. I. T. Jolliffe, *Principal Component Analysis*, 2nd edition, Springer, 2002.
34. 3DCAFE, http://www.3dcafe.com
35. SpharmonicKit 2.5: Fast spherical transforms. http://www.cs.dartmouth.edu/~geelong/sphere/

# Domain Connected Graph: the Essential Skeleton of a 3D Shape

**Fu-Che Wu, Wan-Chun Ma, Rung-Huei Liang, Bing-Yu Chen, Ming Ouhyoung**

**Abstract** In previous research, three main approaches were used to solve the skeleton extraction problem: Medial Axis Transform (MAT), Generalized Potential Field, and Decomposition based methods. The three different approaches focus respectively on surface variation, inside energy distribution and the connectivity of parts. Since a 3D object may be composed of the above three properties, we combine these three ideas to form a novel structure inside an object to represent the object's essential skeleton. In this paper, a skeleton can be composed by end points, connection points and joint points, which are the most important positions to depict an object. In order to maintain stability, we introduce a essential domain ball and a level iso-surface function based on repulsive force field and also define a neighborhood relationship inherited from the surface to describe the connecting relation of these positions. Based on this relation, we construct a Domain Connected Graph (DCG), which preserves the topology information of a 3D shape. The proposed DCG is a concise representation and less sensitive to the perturbation of shapes than that of MAT. Moreover, from the results of complicated 3D models consisting of thousands to millions of polygons, it is also meaningful and more conformed to human perception.

## 1 Introduction

There are many ways to represent a 3D shape, using boundary representations (b-rep), Constructive Solid Geometry (CSG), implicit surfaces, skeletal surfaces, volume-based, point-based and parameterization-based methods. According to Pizer et al.'s suggestion [30], the skeletal surface method is a good way to represent shape because there are many applications directly related to a skeleton, such as object matching [16], animation [35], collision detection, and mesh editing [23]. In general, the skeleton-based method embeds local information and conforms more directly to human perception than other shape descriptions. Therefore, it is suitable for handling a significant component as a unit and because of its concise property, it is more efficient than other representations in computation [30].

However, the requirements for a skeleton differ with each application. For example, object matching may need skeletons to preserve principal features such as morphology (e.g. branching and termination) and geometrical information. Therefore, we can regard these skeletons as major keys and use them to find similarities among objects[16]. On the other hand, object reconstruction needs a kind of skeleton that can reconstruct complete geometrical information of the original surface[14]. The requirements have to address the needs of various applications: when comparing almost identical shapes that have only minor differences, it is acceptable to have the same skeleton; for surface representations, two different shapes need to have different skeletons.

Although skeleton related applications have attracted much attention, it is not easy to extract a robust, meaningful, and manageable skeleton from an arbitrary shape. Extracting a skeleton from a 3D object is more difficult than from a 2D object; the complexity has been extended by another dimension. Here are more situations from different kinds of source models. For example, a model may be a non-manifold model, or it may be constructed by volume , surface and line; a mesh may contain holes or consist of many parts; the data may be very large. We still have not solved all the bad conditions in our proposed method, a pre-process procedure is required to provide a suitable result. Recently, many tools have been developed to repair a bad model, such as RapidForm2004 and PolyMender[18].

In MAT, the skeleton can be the locus of the inscribing sphere centered inside a 3D object. A MAT consists of many spheres with different radii to approximate an object. Consequently, if the curvature of the object surface varies, the MAT needs to describe this variation precisely and thus produces many links for its skeleton. By definition, MAT is sensitive to the variation of boundaries, or noise on the boundaries [7]. A MAT skeleton consists of medial surfaces, curves and lines, and is a complex structure that is hard to handle as a control skeleton. Our goal is to provide a skeleton that is suitable for further manipulation; in our opinion, there are certain properties that need to be considered.

- Simplicity
  Generally, an extracted skeleton is used in applications such as animation by control skeleton, so, it must be simple and easy enough to control.
- Stability
  Two similar shapes should also have two similar skeletons. If a shape is varied only a little, its skeletal structure should remain almost exactly the same.
- Meaningfulness

The skeleton will be simplified to consist of only major prong parts without completely representing the details of a shape, such as width, surface noises, etc. Therefore, a skeleton which is conformed to human perception is preferred.

– Precision
A skeleton can be a shape descriptor; therefore, the embedded information should be invariant under different kinds of transformations. A skeleton and its surface binding information (eg. distance to the surface) should be enough to reconstruct the original shape.
– Hierarchical
Each part in the skeleton has a different priority. According to the priority of each node in the skeleton, we can apply the skeleton's hierarchical properties to the problems of progressive refinement in transmission or of level-of-detail.

The main idea of DCG is to represent a skeleton with a few linked points: a minimal number of points is selected to cover the whole surface, which is similar to a decomposition process, where each part is represented as a meaningful unit in an object. Based on this idea, a main part is represented by a joint point, a prong part is modeled by an end point, and a variation on the skeleton is described by a connection point. The idea is depicted in Fig. 1.

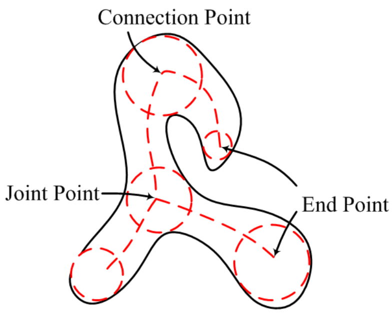



**Fig. 1** Three main point types in a DCG skeleton: joint point, end point, and connection point.

## 2 Previous work

Previous research about skeleton extraction could be divided into three categories: MAT, Generalized Potential Field and Decomposition based approaches. These methods define different kinds of skeletons, and we have combined their ideas to construct a novel representation that avoids their respective disadvantages. In an abstract sense, MAT focuses on shape variation; therefore, it is suitable to depict the detail of the surface, and we can use it to identify prong parts of a 3D shape. The Potential Field method focuses on the inside volume of an object, and we can use it to point out where the main body is. The Decomposition approach can divide an object into many parts that are connected with some linked points; we can choose a few meaningful connection points to describe the variation of skeleton. In the following section, we will briefly explain each idea.

### *2.1 Medial Axis Transform*

The notion of a skeleton was first introduced by Blum [4] in 1967. The MAT based skeleton is a kind of reduced representation and provides region-based shape features. Since its symmetric property is derived from its surface, it is suitable to reconstruct the original shape [36].

It is common to use the Voronoi diagram to construct the MAT skeleton of a 2D shape [26,29], and the idea is then directly extended into a 3D space. Based on this concept, Edelsbrunner and Mücke [12] introduced the α-shape to represent an object. According to the idea of α-shape, Amenta et al. [1] proposed the "power crust" algorithm for MAT approximation and 3D surface reconstruction.

Since the skeleton extracted by the Voronoi diagram is too noisy and therefore not always acceptable, methods for pruning and bifurcation reduction are proposed to solve these problems[6,9,24]. Choi and Seidel[8] suggested a MAT pruning method by bounding the one-sided Hausdorff distance of MAT with respect to the boundary perturbation. Ogniewicz [28] proposed a residual function to measure the significance of skeleton edges. Attali and Montanvert [2] showed that noises on the boundary can be modeled by a parameter graph, suggesting that noisy skeleton branches could be characterized by a small maximal separation angle or thickness. Tam and Heidrich [34] proposed an approach for MAT that removes unwanted artifacts while preserving fine features. Their method prunes the axis by using an intuitive global threshold based on noise size, and then automatically reconstructs the fine features by extending the remaining branches.

### *2.2 Generalized Potential Field*

Another skeleton extraction approach is to use the distance field. To construct the distance field and to extract the skeleton, voxel-based methods are frequently used. The basic methodology consists of constructing the distance field of an arbitrary object, finding the local maximums of the distance field, and connecting these local maximums to generate the skeleton.

Leymarie and Levine [22] implemented a 2D MAT (or Grassfire Transformation) by minimizing the distance field energy of an active contour model whose initial position is at the boundary of a given shape. Zhou and Toga [38] proposed a voxel-coding method, which performs a recursive voxel-by-voxel propagation, then use the Euclid and geodesic distance fields of an object to generate a skeleton. Bitter et al. [3] proposed a penalized-distance algorithm to extract a skeleton from volumetric data. Based on the distance field, the algorithm finds the farthest voxels as the centerline, which forms an initial skeleton.The skeleton is then refined recursively by discarding voxels near the centerline. Wade and Parent [35] also presented an automated algorithm for a control skeleton based on voxel data.

In addition to utilizing voxel data, there are also some approaches to extract a skeleton directly from the boundary. Sherbrooke et al. [31] and Culver et al. [11] proposed an approach that finds a seam as the starting point and then searches for the path of the medial axis. Grigorishin et al.[15] proposed an electrostatic field to depict the inside energy, and some corner points on the boundary are chosen as starting points to track the skeleton path. Chung et al. [10] proposed a similar approach. They assumed that each face of a polygonal object produces a force field; each point on the boundary is pushed by this force field and finally converges to a balanced position, which is labeled as a potential skeleton. In the end, all of these potential skeletons are connected to produce the main skeleton. Siddiqi et al. [32] proposed the Hamilton-Jacobi skeleton: method that obtains the internal medial axis of an object, given its bounding curve or surface, by simulating the grassfire flow as a Hamilton-Jacobi equation on the Euclidean distance function.

Our previous work [25] proposed a thinning-like process that uses the radial basis function as the distance function for object surface thinning to form the skeleton. The result shown in Fig. 2 is similar to our current approach. However, our previous approach was not stable in some cases such as the 'donut-like' model, since there is no guarantee of constructing a radial basis function that keeps the monotonically increasing property well inside an object.

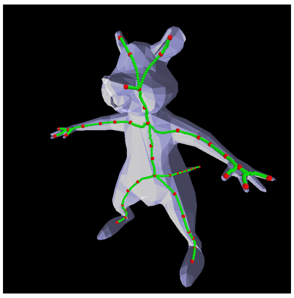


**Fig. 2** Our previous result based on radial basis functions.

There are also methods for skeleton representations. Li et al. [23] directly simplified the object mesh to get the skeleton. Siddiqi et al. [33] proposed a shock graph, where structural descriptions are derived from the singularities of a curve evolution process acting on bounding contours. Zhu [39] proposed a stochastic algorithm for computing skeleton in two phases. The first phase is to find some line segments in favor of parallelism and symmetry. The second phase process computes axes and junctions.

A skeleton based on these approaches is generally simpler than one from MAT, since these approaches usually include a thinning or a shrinking process. They focus on where the most important part inside an object is, and in doing so, ignore the shape variation. However, determining the point that indicates the end of a skeleton is a problem which needs to be solved.

### *2.3 Decomposition Based Method*

This type of method has received much attention recently; it is based on the Reeb graph to extract a one-dimensional skeleton. The idea of the Reeb graph is to use a continuous function, usually a height function, to describe the topological structure and to reveal the topological changes (such as merging or splitting). Based on the continuous function, the target object can be divided into many parts. Connecting the center position of all parts forms the main skeleton. However, it can easily be seen that using a height function to construct the Reeb graph does not guarantee the result to be affine-invariant, which is very important in shape analysis. Since different functions may generate different Reeb graphs, Lazarus and Verroust [21] replaced the height function with the Geodesic Distance Function which is calculated by the Dijkstra algorithm; the generated skeleton is called a "Level Set Diagram" (LSD). Unfortunately, LSD is dependent upon its source points so that different LSDs may be produced from the same model. Mortara and Patané [27] chose the source points in high curvature regions to guarantee that the result is affine-invariant. Hilaga et al. [16] proposed a technique called "topology matching" that compares the Multiresolutional Reeb Graphs (MRGs) of 3D objects in order to evaluate their similarities. Katz and Tal [19] introduced an angular distance to measure the shape variation in order to identify a meaningful part. However, this approach is not suitable for a 'donut-like' shape either.

## 3 Overview

Our goal is to find a concise skeleton that is suitable for animation control and editing. To avoid the complex structure of a MAT skeleton, we wish for the representation to be composed of a set of important points with linkages. There are three main stages for constructing a control skeleton.

- **Feature detection:** This initial procedure detects which position is suitable to represent a skeletal point of a 3D model.
- **Connection construction:** To preserve the topology relation of the original mesh, the neighborhood relationship of each feature point is inherited from surface edge connectivity.
- **Refinement for shape variation:** Two feature points connected by a straight line cannot suitably describe the variation of shape. We introduce a force field to force the skeleton path to follow the shape's varied form.

In each the procedure, we need to evaluate whether a position is appropriate or not; a cost function based on the visibility repulsive force field is introduced. Fig. 3 shows different stages in the algorithm. Essential domain balls are first extracted, and then a prong feature detection procedure is introduced to indicate where an end point is. Other positions are adjusted into more stable locations based on the visibility repulsive force field, and then connectivity relationships are constructed (details are described in the next section). Finally, the connections are adjusted as shape variation.

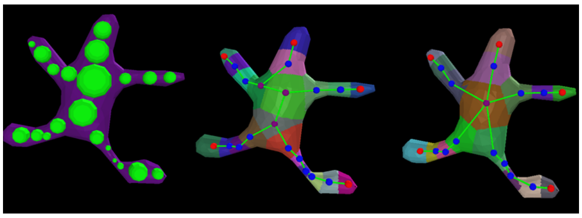


**Fig. 3** Different stages in our algorithm

**Summary of DCG**: To select the most important feature points as the essential domain points, we develop a procedure to evaluate which are suitable for end features, connection features, and joint features. Initially, a Voronoi diagram is constructed to locate initialized candidates. An end feature is locally far from the main part: a geodesic distance is measured on the surface, and then a watershed algorithm is applied to detect where an end feature is. A visibility repulsive force field is introduced, and all other features except end features are adjusted to the local minima that are connection points. Following this, a neighborhood relationship from mesh connectivity is determined; a point that has more than two connection linkages is a joint point. Finally, a snake algorithm also based on the repulsive force field is applied, and all linkages are adjusted to fit the shape variation. The final result, a model with a skeleton, is exported into a Maya format for further application.

## 4 End point

In this section, the initial candidates are constructed from the Voronoi diagram, and a prong-detection algorithm is demonstrated to identify the end points.

### *4.1 Voronoi diagram*

To determine the initial candidates for the skeleton points, we need to find positions that represent more surface range. A MAT locates the position at the axis that is a good candidate. In previous work, most researches are based on the Voronoi diagram, which, for a discrete set of points, identifies for each point, the region of space closer to that point than to any of the other points. The regions are called Voronoi cells. For 2D points the cells are convex polygons; for 3D points they are convex polyhedra. Each face of a cell is equidistant between two points in the discrete set. Thus, a set of points on these faces is roughly lying on the medial axis [17]. Based on Bradshaw's implementation[5], we pick up these Voronoi nodes as the initial candidates called domain balls.

However, the node directly extracted from the Voronoi diagram is still too noisy. For stability purpose, we need to remove the effects from small variation on the shape. There are many previous works that address this problem[8, 13]. Based on their ideas, the solution can be divided into two main concepts: one is to ignore the small variation on the surface, another one is to find the principal data from the noisy preliminary result. In the former concept, Choi [8]developed the one-sided stability theory to enlarge the range of coverage of each skeleton point, and thus, the small variation is ignored. In the latter concept, Foskey [13]defined the $\theta$-simplified medial axis that is based on the separation angle to extract the principal part from the MAT.

### *4.2 Essential domain ball*

In our experiences, more possible points results in more noise; fewer possible points results in the loss of significant positions. To avoid the noise effect, in our approach, we only leave the essential feature points in the first stage. Then, we iteratively recover other necessary points to preserve the variation of the original shape. We locate the good feature points at the axial position and ignore the noise effect on the surface. There are three important properties that need to be considered: radius, separation angle and prong features. Larger radius and separation angle mean more range of coverage. However, we also need to preserve the prong features, so a domain distance based on the geodesic distance is introduced. in the following procedures, we will use the three properties to define the domain ball first, and then introduce a priority sequence to extract the essential domain ball.

The priority of the domain balls is determined by their importance. We can define a priority function $f(p_1, p_2, \cdots)$, in which the sequence of the parameters is also sorted by its priority. Thus, if $p_1(x) > p_1(y) \rightarrow f(x) > f(y)$ . If $p_1(x) = p_1(y), p_2(x) > p_2(y) \rightarrow f(x) > f(y)$ , and so on. We choose radius, domain distance and the maximal separation angle as our primary importance parameters. Let $\rho_p, D_p, A_p$ be the radius, domain distance and the maximal separation angle of $B_D(p)$, respectively. The domain distance is defined as follows.

**Definition 1 (Domain Distance)** *Let $S_p$ be a regular sampled points set defined on shape $S$ . Let $p, q \in S_p$ , $g(p, q)$ be the geodesic distance. A global geodesic distance can be defined as $G(p) = \sum_{\forall q \in S_p} g(q, p)$ . A domain distance of an essential domain ball $B_D$ is defined as $D(B_D) = \max\{G(p), p \in B_D \cap S_p\}$ .*

We increase the radius of the domain ball to raise its covering range, so the tolerable visible domain will have more intersections with the input shape. Among these intersecting positions we can find a maximal separation angle. A small tolerance results in less intersections with the input shape. As Fig. 4 shows, a domain ball has intersections with a plane. To calculate the maximal separation angle correctly, the ball is shifted toward the inside in order to avoid too many intersections with a plane. Moreover, if the normals of the intersection faces do not vary too much, we assume there is only one intersection.

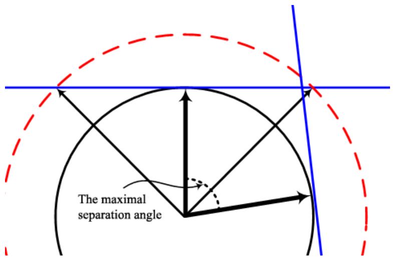



**Fig. 4** The maximal separation angle.

Furthermore, we define a priority sequence of domain balls under shape $S$ and label each domain ball by the sequence number as follows. If $f(\rho_1, D_1, A_1) > f(\rho_2, D_2, A_2) > \cdots > f(\rho_n, D_n, A_n) > \cdots$ , then $B_{D_1}, B_{D_2}, \cdots B_{D_n}, \cdots$ is a domain sequence.

**Definition 2 (The Essential Domain Ball)** *Let $B_{D_1}, B_{D_2}, \cdots B_{D_n}$, $\cdots$ be a domain sequence. The essential domain ball can be defined by inserting a domain ball iteratively based on its priority keeping no intersection. $B_E = \{B_{D_i} \mid \bigcap_{i=1}^{n} B_{D_i} = \emptyset$, where n is the maximal number $\}$ .*

After calculating all radii of the domain balls, we choose the essential domain balls iteratively based on the magnitude of their radii. Once a maximal ball is chosen, it will delete all its neighbor balls that possibly are touching. In order to decrease the number of balls, we also introduce a minimal distance constraint between each ball. Fig. 5 is a result showing the essential domain balls.

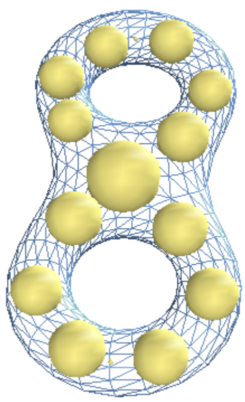


**Fig. 5** The essential domain balls for a 3D model in the shape of the digit eight.

*4.3 Prong feature*

There are many domain balls which are distributed inside an object. Since we are only interested in the main skeleton and the end position of a prong, we extract the most important part as the essential domain ball. Therefore, we develop a prong-features detection algorithm [37] and define a priority sequence to identify which ball is more important.

First, we construct a normalized scale function on the input mesh to describe the position of a potential prong. The scale function is calculated by the sum of the geodesic distance from each node among the mesh to the evaluated node[21]. The scale function is defined in the following equation:

$$f(v) = \sum_{\forall v_i \in S} Geodesic(v, v_i),$$

where $v$ is the vertex on the shape $S$. To ensure that the calculation of the geodesic distance approximates the real value, a remesh pre-process is required. The result is shown in the following:

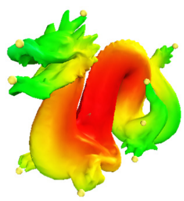


**Fig. 6** From the scale function, a higher value indicates a prong feature that is denoted by the green color.

In order to find these prong features we apply a watershed algorithm. First, all vertices are sorted by the scale function value. Then, the maximum value of the vertex is picked up as the first prong feature and marked as traversed. As we have to focus on the feature detection, the water level decreases iteratively from the maximum value. The decreasing level affects the sensitivity of a prong feature. The prong shape's valley value divided by its peak value is defined as the smoothness ratio. If the smoothness ratio is larger than the threshold, the result is a small prong and will be ignored. Each non-traversed vertex in the sorting sequence will reset the water level, ie. its function value is multiplied by the smoothness ratio. Consequently, we can mark all vertices as traversed if their function value is above the water level and if they are a neighbor of the traversed set. If the vertex is not in the traversed set, it is a new prong feature. Thus, the nearest essential domain point to a prong feature is an **end point**.

## 5 Connection point

The next step would be to locate the connection points inside an object. Since we want to identify the most important part inside a shape, a generalized potential field based on the level iso-surface function is introduced. The basic idea is shown in Fig. 7.

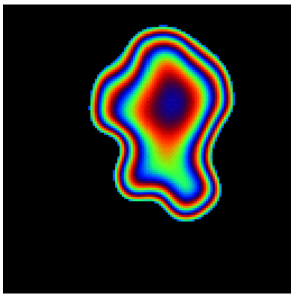


**Fig. 7** The level iso-surface function.

### 5.1 Level iso-surface function

A level iso-surface function is a distance function on a shape $S$ . If such function $f : R^3 \to R$ is defined on a shape $S$ , all points on the boundary have maximal values. Moreover, a location near the boundary has a higher value. For example, let $d$ be the inverse of a shortest distance function; it satisfies the following properties:

1. If $p \in S$ , $\forall q \in \overset{\circ}{S} \to d(p) > d(q)$ , where $\overset{\circ}{S}$ is the interior of $S$.
2. If $p, q$ are two points inside a shape $S$ , $\|p\| < \|q\| \to d(q) > d(q)$ , $\|p\|$ is the shortest distance from position $p$ to the boundary.

The balanced positions of the function $d$ are the candidates of MAT, which consist of some faces and lines. Since the shortest distance is determined based on only one position on the boundary, the result is unstable and sensitive to noise. We want to construct a function that has only one global minimal point in a convex shape. Thus, a repulsive force field is defined as following. $F(x) = \int \frac{\overrightarrow{\mathbf{r}}}{r^n} d\theta$ , where $r$ is the distance from $x$ to the boundary in the direction $\overrightarrow{\mathbf{r}}$. A discrete form is

$$\mathbf{F}(p) = \sum_{0 \le \alpha < 2\pi; 0 \le \beta < \pi} \frac{\overrightarrow{\mathbf{u}}_{\alpha,\beta}}{\|p\|^n},$$

where $\overrightarrow{\mathbf{u}}_{\alpha,\beta}$ is a unit sample ray, and $\alpha, \beta$ are angles in the spherical coordinate. A level iso-surface function $f(x)$ can be defined as $\|\mathbf{F}(x)\|$ . When $n \to \infty$ , the repulsive force field is dominated by the shortest distance, and thus, $f(x) \propto d(x)$ . In some sense, $n$ is a term for a smoothing effect. Each position receives more boundary influence for a smaller $n$ . As Fig. 8 shows, the inner level of the iso-surface function is smoother with a smaller $n$. In our implementation, we adopt $n = 1$.

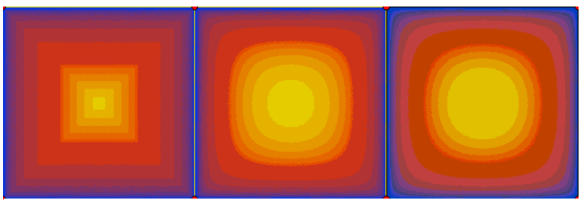


**Fig. 8** The different $n$ effects on the level iso-surface function. where $n = 100, 1$,and 0.01, respectively.

### 5.2 Repulsive force field

In the previous section, we have constructed the essential domain balls and end points. All domain points except end points are candidates that need to be refined. The refining process is based on the repulsive force field. For each point $p_i$, we use the repulsive force to push it to a new position $p_{i+1}$:

$$p_{i+1} = p_i + \mathbf{F}(p_i) \times step$$

The iterative process stops when the position reaches a local minimum $\|\mathbf{F}(p_i)\| < \|\mathbf{F}(p_{i+1})\|$. In practice, all intermediate points will converge to a location within a small range, which depends on the iterative step. To solve this problem simply and without loss in general, we merge these final positions in a given range to get a single domain point, which is regarded as a **connection point**. Fig. 9 shows the shrinking process.

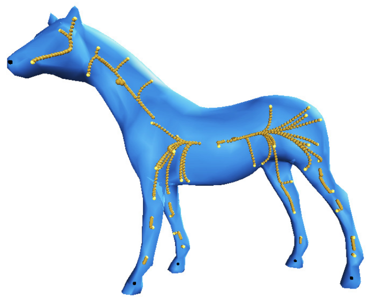


**Fig. 9** The essential domain points are shrunk under the repulsive force field to form the connection points.

## 6 Joint point

A joint feature potentially connects more than two skeleton linkages. Thus, we need to recover the topology relationship first, which shows how things are connected. To recover the connectivity of domain points, we need to determine the relationship between surface and domain points first. Thus, based on the connectivity of surface, we can induce the connectivity of domain points.

**Definition 3 (Face Connectivity)** *Connectivity is a neighborhood relation denoted as* $\underset{c}{\Leftrightarrow}$ *. If two faces share an edge, the two faces have connectivity. Let a face A be bounded with some boundary edges* $\partial A$ *. Thus, for two faces* $A_i, A_j$ *, if* $\partial A_i \cap \partial A_j \neq \emptyset \to A_i \underset{c}{\Leftrightarrow} A_j$ *.*

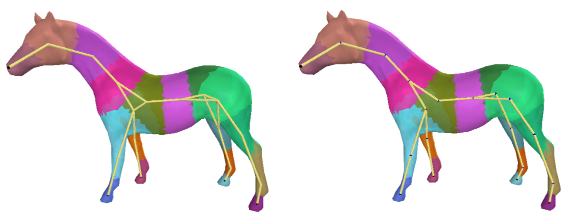
**Definition 4 (Domain Surface)** *DS is a surface region which belongs to a domain point DP. Each domain point has a domain surface which can be described by this point. A domain point can find at least one shortest distance of point on the surface. A face that contains this point is called a primitive visible face denoted as* $A_{pv}$ *.*

1. $A_{pv} \in DS(p)$
2. *If* $A_i \underset{c}{\Leftrightarrow} A_j, A_j \in DS(p) \to A_i \in DS(p)$ .

*3. If $A_i \in DS(p) \rightarrow \forall q \in DP, \|A_i p\| < \|A_i q\|$*

**Definition 5 (Domain Point Connectivity)** *Let $DS(p), DS(q)$ represented as domain surface of domain point $p, q$ , respectively. If $\exists A_i \underset{c}{\Leftrightarrow} A_j, A_i \in DS(p), A_j \in DS(q) \rightarrow p \underset{c}{\Leftrightarrow} q$ .*

Based on previous definitions, we could implement the algorithm as follows. First, we need to determine to which domain point each face on the shape belongs. We have found the touching position on the shape of each domain ball in previous processes. Faces containing this position are primitive visible faces, so they belong to this domain point. Then, for the other unsolved faces, we will search all domain points to find the shortest distance of domain point that must be visible as their potential target. If a face potentially belongs to a domain point and its neighbor also belongs to the same point, this face is tagged with this domain point. Finally, there are faces that are not visible to any domain points. These faces are defined with their neighbor faces using the same domain point iteratively. If all faces have been defined with their domain points, edges shared by different domain faces have connectivity relationship between their domain points. We connect the two domain points with a connection link. A connection link is constructed by many skeletal points as control points for a curve. A connectivity relationship constructed directly from the surface; Fig 10(a) shows it is possible that a cycle linkage is formed. A path cost is evaluated by the mean force under the repulsive force field. A larger force of linkage will be deleted, because it is far from the main skeleton. If this linkage is construed from a ring on the surface, that is a genus case, and we will leave the cycle. Fig. 10(b) shows the result.



(a) (b)

**Fig. 10** A cycle linkage is deleted based on the mean force of the linkage under the repulsive force field.

From the previous definition, the connectivity of domain point is inherited from connectivity of domain surface. After constructing the connectivity relationship among domain points, we could define a connectivity counting function $C(DP)$ to report each domain point containing how many connectivity relationships. The **joint point** is defined as $E = \{DP|C(DP) > 2\}$

## 7 Refinement

To describe the variation of the shape faithfully, a refinement procedure on these skeletal points among a linkage is required.

**Definition 6 (Skeletal Point)** *Let $P = \{x|\langle a,x\rangle - D = 0\}$ be a cutting plane. A cutting plane can divide an object into two parts. Each part contains one domain point. Thus, the cutting plane contains at least one skeletal point. $SP = \{p|\min\{f(p)\}$ , $\langle a,p\rangle - D = 0\}$*

For two parts connected in the shape, there must be a skeletal point to connect the two parts. If two domain points have connectivity relationship, there is a link between these two points. We can generate initial points to represent this link. At first, some points are sampled on a straight line to represent a connection link. Here, we invoke a Snake [20] algorithm (Active Contour Model) to adjust these points so that they correspond to a skeletal curve. An active contour model is generated by adjusting a contour to minimize an energy function:

$$E_{Snake} = E_{Internal} + E_{External},$$

where $E_{Internal}$ and $E_{External}$ are internal and external energy respectively. Internal energy is the part that depends on intrinsic properties of the link, such as length or curvature. $E_{External}$ means external energy, and its value depends on the generalized potential field. The position of each point is adjusted on the constrained plane which is perpendicular to the original straight line. When the energy of the active contour model is minimized, the result skeleton is obtained. Fig. 11 shows the deforming process of a connection link.

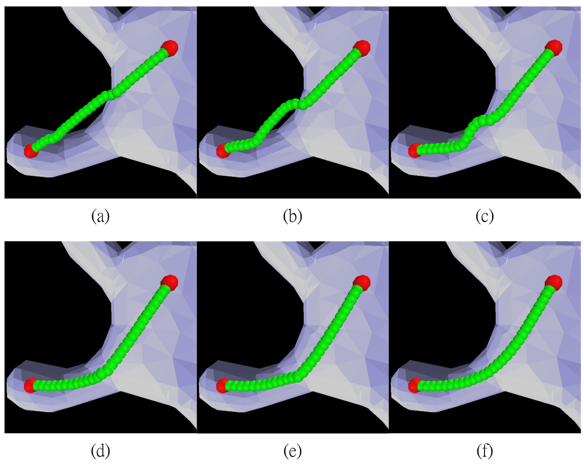


**Fig. 11** The deforming process of a link, where the green points between two red points (end point and joint point) are pushed by the generalized force field to the final position.

## 8 Result

Fig. 13 shows the DCG of nine typical 3D models, and Table 1 shows the execution time for different stages. The timing

| Model | Face number | Essential domain ball (min.:sec.) | Joint point (min.:sec.) | Refinement (min.:sec.) | Convert to Maya format (min.:sec.) |
|---|---|---|---|---|---|
| dragon | 7315 | 00:37 | 00:24 | 05:01 | 00:01 |
| eight | 1536 | 00:08 | 00:03 | 01:05 | 00:01 |
| octopus | 8648 | 01:05 | 01:18 | 07:58 | 00:01 |
| dinosaur | 5605 | 00:33 | 00:35 | 02:58 | 00:01 |
| bunny | 6343 | 00:27 | 00:14 | 06:14 | 00:01 |
| hand | 9936 | 01:03 | 01:35 | 14:40 | 00:01 |
| rhino | 7678 | 00:50 | 00:25 | 02:50 | 00:01 |
| cheetah | 5345 | 00:30 | 00:20 | 04:53 | 00:02 |
| rabbit | 2174 | 00:16 | 00:04 | 01:47 | 00:01 |

**Table 1** Execution time statistics for nine 3D models on a Pentium 2.4 GHz PC

.

statistics are from a notebook with Intel Pentium-4 2.4 GHz processor and 512 Mbytes memory. Currently, the refinement procedure consumes most of the time, and the whole process is not optimized yet. To avoid long computation time, the original models have been simplified first into one to five thousands of polygons. Fig. 14 shows more results. Their skeleton data in Maya format are available at
http://www.cmlab.csie.ntu.edu.tw/g/DCG/.

We also demonstrate the further application in Maya by using current result to produce an animation sequence. Fig. 12 shows the sequence.

## 9 Future Work and Applications

We have formed the definition of end point, joint point and connection point of a skeleton. We could also find a result based on the current definition. However, since our main concern is to extract the principal part of a shape, some smaller parts or details of the shape will be ignored. To solve this problem, we could iteratively refine it by decomposing the object into many parts[19]. A small feature in a whole object becomes a big feature when the sub-part is scaled into a unit. The decomposition process cuts an object with a cutting plane. On the cutting plane we can find a connection point in order to connect the two parts. Thus, a decomposed part must have a connection point on its shape to connect to the original skeleton.

Based on the current result, each meaningful part is extracted and represented as a concise form. Thus, it is suitable for many applications. In animation, each domain point is an easily controllable point. In object matching, the result skeleton could be a primary key for comparing the similarity of 3D objects. In mesh editing, each domain surface could be handled as a unit. In mesh compression, since the residual between domain point and its described surface is small and the connectivity relationship is simple, it is very possible to have an efficient compression. Furthermore, the level of detail and the progressive display are also possible based on our result skeleton. In some applications, if a very concise form is not needed, connecting the essential domain balls at the first stage will quickly generate a prototype. However, the symmetry property of a symmetric skeleton is an issue that needs more consideration in the future.

## 10 Conclusion

In this paper a new representation for the skeleton of an arbitrary shape is proposed. The nodes of the graph are named domain points (significant points inside the shape) which can represent the local properties of a shape. The domain points are classified into three categories: joint points, end points and connection points. In this paper, we introduce the concept and the definition of the DCG. Moreover, we propose an algorithm to generate the graph, and also describe the difference between our definition and the traditional MAT skeleton explaining the advantages with DCG.

Given an arbitrary 3D shape, the proposed skeletonization method can extract both topological and geometrical information. The result skeleton can converge in limited computational time and is completely inside the boundary of a 3D shape. This is a concise, stable and meaningful representation of a general 3D object, which can avoid the noise-sensitive problem encountered in the MAT. Furthermore, there is no restriction on the types of 3D models, a problem often faced by the Radial Basis Function based approaches.

## 11 Acknowledgements

We thank 3DCAFE[1] to provide all 3D models used in this paper. This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC92-2622-E-002-002, NSC92-2213-E-002-015, NSC92-2218-E-002-056 and 89E-FA06-2-4-8 and Silicon Integrated Systems (SiS) Education Foundation.

## References

1. N. Amenta, S. Choi, and R. Kolluri, “The Power Crust”, *Proceedings of ACM Symposium on Solid Modeling and Applications*, PP.249-260, (2001).

[1] http://www.3dcafe.com

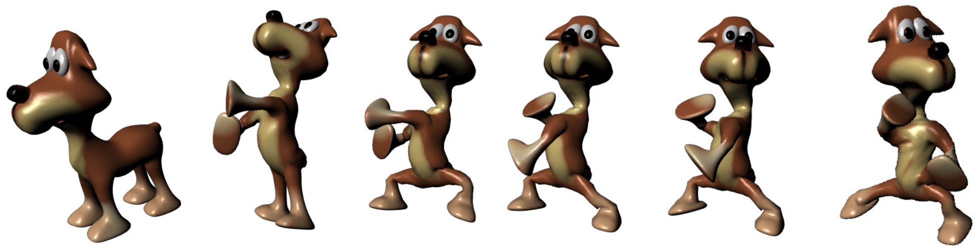



**Fig. 12** Using our result, a animation sequence is produced from Maya.


2. D. Attali, A. Montanvert, "Modeling Noise for a Better Simplification of Skeletons", *Proceedings of International Conference on Image Processing*, **3**, PP.13-16, (1996).
3. I. Bitter, A. E. Kaufman, and M. Sato, "Penalized-Distance Volumetric Skeleton Algorithm", *IEEE Transaction on Visualization and Computer Graphics*, **7**(3), PP.195-206, (2001).
4. H. Blum, "A Transformation for Extracting New Descriptors of Shape", *MIT Press*, PP.362-380, (1967).
5. G. Bradshaw, C. O'Sullivan,"Adaptive Medial-Axis Approximation for Sphere-Tree Construction",*ACM Transactions on Graphics*,**23**(1),PP.1-26,(2004).
6. G. Borgefors, I. Nyström, "Efficient shape Representation by Minimizing the Set of Centres of Maximal Discs / Spheres", *Pattern Recognition Letters*, **18**, PP.465-472, (1997).
7. H.I. Choi, S.W. Choi, and H.P. Moon, "Mathematical Theory of Medial Axis Transform",*Pacific Journal of Mathematics*. **181**(1) PP.57-87. (1997).
8. S.W. Choi, H. P. Seidel, "One-Sided Stability of Medial Axis Transform", *Lecture Notes in Computer Science*, **2191**, PP.132-139, (2001).
9. W.-P. Choi, K.-M. Lam., and W.-C. Siu, "Extraction of the Euclidean Skeleton based on a Connectivity Criterion", *Pattern Recognition*, **36**, PP.721-729, (2003).
10. J.-H. Chung, C.-H. Tsai, and M.-C. Ko, "Skeletonization of Three-Dimensional Object Using Generalized Potential Field", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **22** (11), PP.1241-1251, (2000).
11. T. Culver, J. Keyser, and D. Manocha, "Accurate Computation of the Medial Axis of a Polyhedron", *Proceedings of ACM symposium on Solid Modeling and Applications*, PP.179-190, (1999).
12. H. Edelsbrunner, E. Mucke, "Three-Dimensional Alpha Shapes", *ACM Transactions on Graphics*, **13** (1), PP.43-72, (1994).
13. M.Foskey, M.C. Lin, D.Manocha,"Efficient Computation of A Simplified Medial Axis", *Proceedings of ACM Symposium on Solid Modeling and Applications*, (2003).
14. P. Giblin, B.B. Kimia, "A Formal Classification of 3D M.A. Points and their Local Geometry", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **26**(2), PP.238-251, (2004).
15. T. Grigorishin, G. Abdel-Hamid, and Y. H. Yang, "Skeletonization: An Electrostatic Field-Based Approach", *Pattern Analysis and Applications*, **1**, PP.163-177,(1998).
16. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", *SIGGRAPH 2001 Conference Proceedings*, PP.203-212, (2001).
17. P.M. Hubbard, "Approximating polyhedra with spheres for time-critical collision detection", *ACM Transactions on Graphics*, **15**(3), PP.179ąV210,(1996).
18. T. Ju, "Robust Repair of Polygonal Models", *SIGGRAPH 2004 Conference Proceedings*, PP.888-895, (2004).
19. S. Katz , A. Tal, "Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts", *SIGGRAPH 2003 Conference Proceedings*, PP.954-961, (2003).
20. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models", *International Journal of Computer Vision*, **1**, PP.321-331, (1987).
21. F. Lazarus, A. Verroust, "Level set diagrams of polyhedral objects", *In ACM Symposium on Solid Modeling and Applications*, PP.130-140, (1999)
22. F. Leymarie, M.D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **14** (1), PP.56-75, (1992).
23. X.-T. Li, T.-W. Woon, T.-S. Tan, and Z.-Y. Huang, "Decomposing Polygon Meshes for Interactive Applications", *Proceedings of ACM Symposium on Interactive 3D Graphics*, PP.35-42, (2001).
24. F. Nilsson, P-E. Danielsson, "Finding the Minimal Set of Maximum Disks for Binary Objects", *Graphical Models and Image Processing*, **59**(1), PP.55-60, (1997).
25. W.-C. Ma, F.-C. Wu, and M. Ouhyoung, "Skeleton Extraction of 3D Objects with Radial Basis Function", *Proceedings of Shape Modelling International 2003*, PP.207-215, (2003).
26. N. Mayya, V. T. Rajan, "Voronoi Diagrams of Polygons: A Framework for Shape Representation", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, PP.638-643, (1994).
27. M. Mortara, G. Patane. "Affine-Invariant Skeleton of 3D Shapes", *Proceedings of Shape Modeling International 2002*, PP.245-278, (2002).
28. R. Ogniewicz, "Automatic Medial Axis Pruning by Mapping Characteristics of Boundaries Evolving Under the Euclidean Geometric Heat Flow onto Voronoi Skeletons", *Technical Report 95-4*, Harvard Robotics Laboratory , (1995).
29. R. Ogniewicz, M. Ilg, "Voronoi Skeletons: Theory and Applications", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, PP.63-69, (1992).
30. S. M. Pizer, A. L. Thall, and D. T. Chen, " M-Reps: A New Object Representation for Graphics", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, PP.638-643, (1994).
31. E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson, "Computation of the Medial Axis Transform of 3D Polyhedra", *Proceedings of ACM Symposium on Solid Modeling and Applications*, PP.187-199, (1995).
32. K. Siddiqi, S. Bouix, A. R. Tannenbaum, and S. W. Zucker. "Hamilton-Jacobi Skeletons",*International Journal of Computer Vision*, **48**(3), PP.215-231, (2002).

33. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. "Shock Graphs and Shape Matching", *International Conference on Computer Vision*, PP.222-229, (1998).
34. R. Tam, W. Heidrich, "Feature-Preserving Medial Axis Noise Removal", *ECCV2002*, PP.672-686, (2002).
35. L. Wade and R. E. Parent, "Automated Generation of Control Skeletons for Use in Animation", *The Visual Computer*, **18**(2), PP.97-110, (2002).
36. F.-E. Wolter, "Cut Locus and Medial Axis in Global Shape Interrogation and Representation", *Technical Report*, (1993).
37. F.C. Wu, B.Y. Chen, R.H. Liang, M. Ouhyoung, "Prong Features Detection of a 3D Model Based on the Watershed Algorithm",*ACM SIGGRAPH2004 Sketches*, (2004).
38. Y. Zhou, A. Toga. "Efficient Skeletonization of Volumetric Objects", *IEEE Transaction on Visualization and Computer Graphics*, **5**(3), PP.195-206, (1999).
39. S.-C. Zhu, "Stochastic Jump-diffusion Process for Computing Medial Axes in Markov Random Fields", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **21**(11), PP. 1158-1169, (1999).

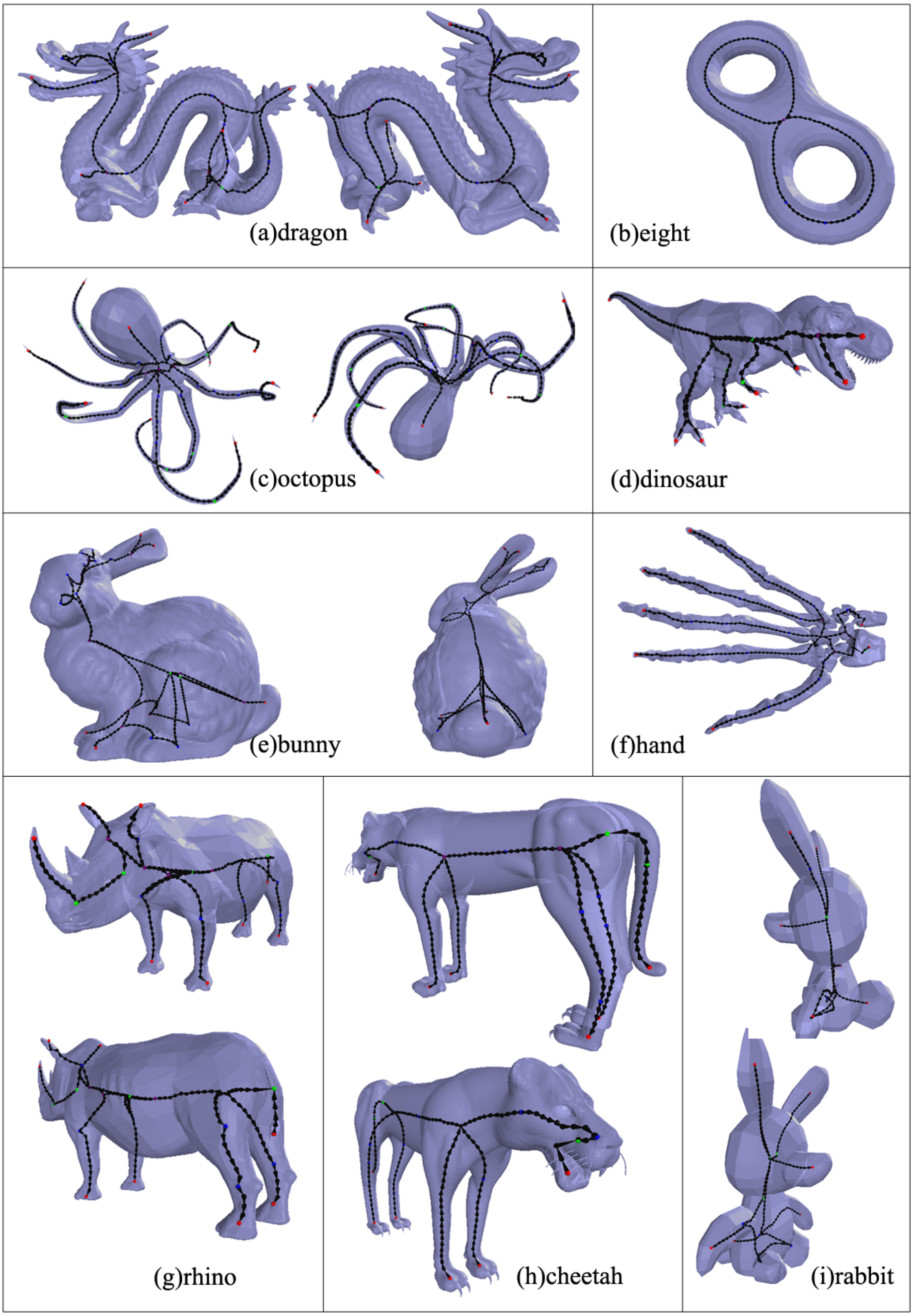
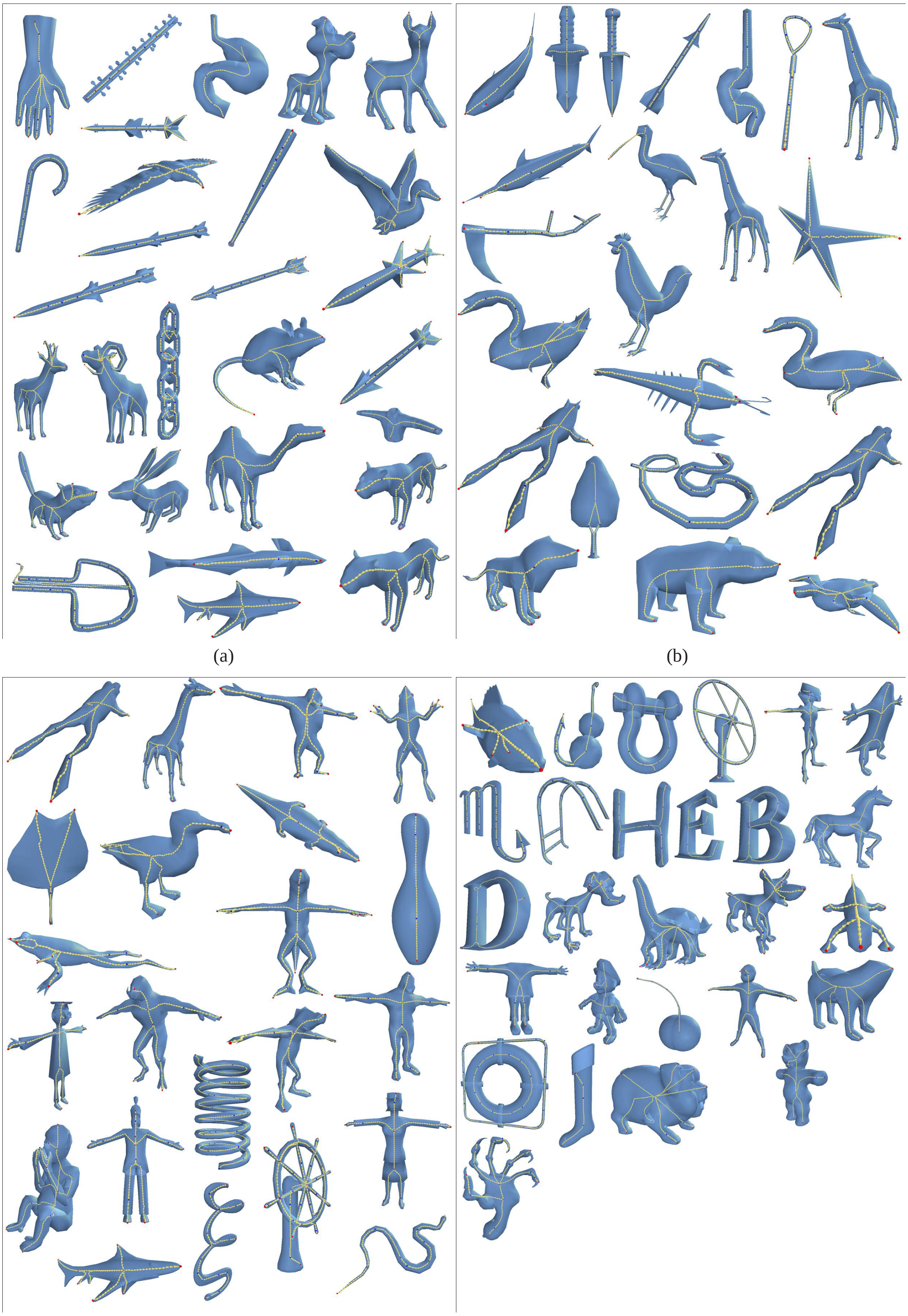



**Fig. 13** There are results from different kinds of models. For performance consideration, the models have been simplified firstly. The number of faces are under ten thousand.



(a) (b)

(c) (d)

**Fig. 14** More results with skeleton data are available at **http://www.cmlab.csie.ntu.edu.tw/g/DCG/**.

# Skeleton Extraction of 3D Objects with Radial Basis Functions

Wan-Chun Ma, Fu-Che Wu, Ming Ouhyoung
Communication and Multimedia Laboratory
Dept. of Computer Science and Information Engineering
National Taiwan University
{firebird, joyce}@cmlab.csie.ntu.edu.tw, ming@csie.ntu.edu.tw

## Abstract

*Skeleton is a lower dimensional shape description of an object. The requirements of a skeleton differ with applications. For example, object recognition requires skeletons with primitive shape features to make similarity comparison. On the other hand, surface reconstruction needs skeletons which contain detailed geometry information to reduce the approximation error in the reconstruction process. Whereas many previous works are concerned about skeleton extraction, most of these methods are sensitive to noise, time consuming, or restricted to specific 3D models.*

*A practical approach for extracting skeletons from general 3D models using radial basis functions (RBFs) is proposed. Skeleton generated with this approach conforms more to the human perception. Given a 3D polygonal model, the vertices are regarded as centers for RBF level set construction. Next, a gradient descent algorithm is applied to each vertex to locate the local maxima in the RBF; the gradient is calculated directly from the partial derivatives of the RBF. Finally, with the inherited connectivity from the original model, local maximum pairs are connected with links driven by the active contour model. The skeletonization process is completed when the potential energy of these links is minimized.*

## 1 Introduction

Surface reconstruction and object recognition are all concerned about the shape description. Intuitively, the skeleton of an object is regarded as a good description to be applied to the above problems. Hence, how to extract skeletons, especially from arbitrary 3D objects, becomes an important subject in recent years. Medial Axis Transform (MAT), which is introduced by Blum [5] and has been widely adopted in the computer vision area, is a way to extract reliable skeletons from given shapes. The medial axis, as the locus of the inscribed circle centers inside a 2D object, is a kind of reduced representation and provides region-based shape features.

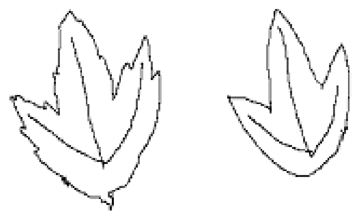


**Figure 1. Two arbitrary shapes and their skeletons**

Nevertheless, different applications may require different skeleton descriptions. To give a circumstantial account of above statement, we take two arbitrary shapes in Figure 1 for example. These two leaves have high similarities if we compare their skeletons with each other. However, if we use the left skeleton to reconstruct its own object, the error may not be acceptable since the left skeleton loses lots of the boundary information.

We take MAT as another example. A MAT skeleton performs well in surface reconstruction because it preserves all the boundary information of the given shape. However, if we try to apply a MAT skeleton to object recognition, sometimes it may contain too many branches of minor importance and using it as a comparison key may have undesired recognition results. In this paper, we try to focus on the skeleton which is able to represent the object both morphologically and geometrically. A 3D object can be transformed into an implicit surface, which has the following form:

$$f(x) = h$$

where $x$ is a point in $\Re^3$ and $h$ is a level value of a designated surface, usually we use $h = 0$ to indicate the object surface. The function can be regarded as a distance function [4]; it returns a higher value as the input point $x$ gets deeper away from the object surface. Given an implicit surface of a 3D object, we can extract the skeleton from it by using techniques such as gradient descent and active contour model.

## 2 Previous Works

A good skeleton should be able to fulfill the following requirements:

1. Morphology: The skeleton must retains the morphology of the original shape, such as branching and termination.

2. Geometry: Force the skeleton to be in the middle of an object to preserve the region based geometry features (e.g. surface bends).

3. Affine-invariant: The skeleton should be invariant under affine transformations, especially translation and rotation.

Many previous works focus on 2D shape skeletons. Traditionally, there are two kinds of approaches to construct the skeleton of an object. One is to compute the skeleton directly from the object surface points, such as using Voronoi diagram. The other is to extract the skeleton by constructing a distance field of an object.

**Direct computation method:** Voronoi diagram is widely adopted to construct 2D skeletons [21, 24], and this idea is straightforward to be extended into 3D space. The medial axis in 3D, which is the locus of the inscribed sphere centers inside a 3D object by definition, has already been discussed in many studies [23, 31]. With this concept, Edelsbrunner and Mücke [13] introduce $\alpha$-shape to formalize an intuitive shape representation for the scattered data points. The $\alpha$-shape is a generalization of convex hull and a subgraph of the Delaunay triangulation. The real number $\alpha$, which represents the radius of a sphere with $0 < \alpha < \infty$, controls the desired level of detail of the shapes.

Following the idea of $\alpha$-shape, Amenta et al. [2] propose the power crust algorithm for MAT approximation and 3D surface reconstruction. The algorithm computes the Voronoi diagram of the scattered data points first and then retrieves a set of polar balls by selecting candidates from the Voronoi balls that have maximal distance to the sampled surface. After labeling these polar balls with power diagram, the object described by the data points is now transformed into the polar ball representation. Connecting these polar ball centers forms a good approximation to the MAT.

A comparatively up-to-date direct computation method is to use the Reeb graph. The definition of the Reeb graph is introduced by Reeb [25]. The Reeb graph is regarded as a one dimensional skeleton. Lazarus and Verroust [18] use the idea of Reeb graph to construct the skeleton called the level set diagram (LSD). Unfortunately, LSD is dependent on its source points, so different LSDs may be produced from the same model. Mortara and Patané [22] choose the source points in high curvature regions to guarantee the result is affine-invariant.

**Distance field method:** Building a 3D Voronoi diagram needs at least $O(n^2)$ in time complexity where $n$ is the input point number. Due to this reason, direct computing method is not suitable for the condition of large number of input points. Another approach is to use the distance field of an object to extract the skeleton. The basic methodology is as follows:

1. Construct the distance field of an arbitrary object.

2. Find the local maxima of the distance field.

3. Connect these local maxima to generate the skeleton.

Several distance field methods are proposed to produce skeletons. Leymarie and Levine [19] implement a 2D MAT (also called grassfire transformation) by minimizing the distance field energy of an active contour model. Zhou and Toga [33] propose a voxel-coding method which is based on recursive voxel propagation. The algorithm starts from a set of seed voxels, and then uses a coding schema to construct connectivity relationship and distance fields. Bitter et al. [3] propose a penalized-distance algorithm to extract skeleton from volumetric data. The algorithm first computes distance and gradient vector fields for locating the initial skeleton. Then, the skeleton is refined iteratively by discarding its redundant voxels. Chuang et al. [9] use each face in a polygonal object as a force field source, then several designated points, which are regarded as skeleton candidates, move to balanced positions in the force field. By tracing the locus of the candidates and connecting them together, a MAT skeleton can be constructed. Recently, Wade and Parent [30] use a discrete distance field to generate 3D skeletons for animation use.

## 3 Problem Analysis

Let $S$ be a surface. A skeleton of surface $S$ is a set of points denoted as $M(S)$. For each point $q$ on surface $S$ we can find a point $p$ which belongs to skeleton $M(S)$, and we denote it as $p = M(q)$. For each $p$ on skeleton, we can also find an inverse of the skeleton point denoted as $M^{-1}(p)$, which is a set of points on the surface. Then, if $M(S)$ is a well-defined skeleton of surface $S$, we propose that $M(S)$ should satisfy the following hypotheses:

1. Neighborhood property: if $p_1$ and $p_2$ are neighbors in $M(S)$, $M^{-1}(p_1)$ and $M^{-1}(p_2)$ are neighbors in $S$.

2. Uniformity property: Let $q \in S$, the distance between $q$ and $M(q)$ is denoted as $\rho(q)$. Let $p \in M(S)$, for all

$q_i \in M^{-1}(p)$, $\rho(q_i)$ should be the same (according to the definition of MAT)

3. Compactness property: The size of $M(S)$ is minimal if the difference between $M^{-1}(M(S))$ and $S$ is under a given error tolerance.

Algorithms based on property 2 produce symmetrical skeleton preferentially. Leyton [20] has inferred the Symmetry-Curvature Duality. The theorem indicates that each curvature extremum can be assigned a unique symmetry axis leading to the extremum. In another word, if the shape has more curvature variation, it may generate more skeleton branches. It explains that property 2 conflicts with property 3. MAT prefers property 2 so that it is suitable for surface reconstruction. However, we prefer property 3 since we want to find major skeleton points to preserve the primary shape structure. Due to this reason, we modify the property 2 by not limiting $\rho(q_i)$ be the same value but trying to minimize the variant of $\rho(q_i)$.

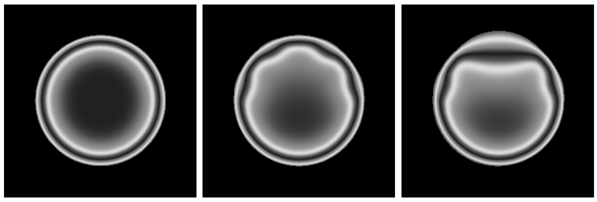
$$\min\left(\int_{\forall q \in M^{-1}(p)} \|\rho(q) - Average(\rho(q))\| dA\right)$$

One benefit from the modified uniformity property is that a skeleton point now can embed an arbitrary shape instead of regular spheres (or circles) in MAT to represent part of the object.

Let $\Omega_0$ be a connected bounded domain in $\Re^3$. A point $p$ on the $S$ can be denoted as $p \rightarrow S$. If there exists a distance function $D_s(x)$ that calculates the distance between point $x$ and surface $S$, a gradient operator $\nabla D_s(x)$ can be applied and for $p \rightarrow S$ we can apply the gradient descent $G(p)$: $p^0 = p$, $p^{n+1} = p^n + \nabla D_s(p^n)$. If $D_s(p^n) = 0$, $p^{n+1}$ is a local maximum and $G(p) = p^{n+1}$. We then denote $M$ as the set of local maxima. In our definition, the local maximum $m_i \in M$ is a representative of the set $P(m_i) = \{p_j \mid G(p_j) = m_i\}$. We can complete the skeleton graph by linking local maximum pair $m_i$ and $m_j$ if $P(m_i)$ and $P(m_j)$ are adjacent. The linking process conforms to the neighborhood property.

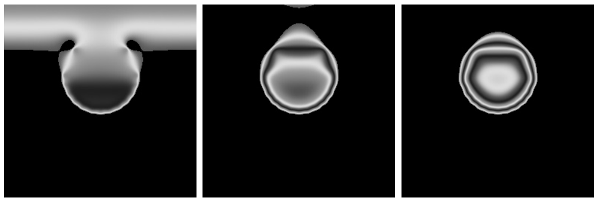
# 4 Implicit Surfaces with Radial Basis Functions

Many previous studies have been done for implicit surfaces and RBFs [6, 12, 27, 28, 29]. Given a set of constraint points which define the shape of a 3D object, an implicit surface can be created by using 3D scattered data interpolation. Turk and O'Brien construct implicit surfaces with RBFs by solving a linear system [29]. This method is widely adopted for height interpolation or deformable models. In this paper, we invoke radial basis functions as height (distance) interpolation for building a distance field.



(a) (b) (c)

**Figure 2. Cross section pictures of RBF level set coded in grayscale for different models. (a) Level set from a 10x10 sampled sphere. (b) Reduce the vertex density of a upper hemisphere by 50%. (c) Reduce the vertex density of a upper hemisphere by 75%.**



(a) (b) (c)

**Figure 3. Cross section pictures of RBF level set with different RBF radius. (a)** $c = 0.1$ **(b)** $c = 0.2$ **(c)** $c = 0.3$**. The model is the same with Figure 2 (c). The radius of the sphere is 1.0.**

## 4.1 Implicit Surface Construction with Radial Basis Functions

An implicit surface can be defined by a set of RBFs. The implicit surface function has the form:

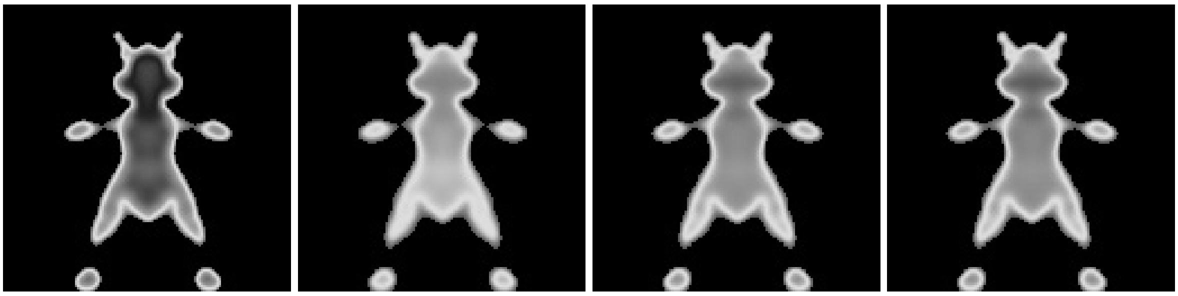
$$f(x) = \sum_{j=1}^{n} d_j w(x - c_j) + p(x)$$

where $c_j$ are the surface constraints, $d_j$ are weight coefficients, $w(x)$ is a basis function and $p(x)$ is a degree one polynomial for the linear and constant terms of $f(x)$. To solve the weights and polynomial parameters we need to assign interpolation constraints as follows:

$$h_i = f(c_i)$$

then

$$h_i = \sum_{j=1}^{n} d_j w(c_i - c_j) + p(c_i)$$



(a) (b) (c) (d)

**Figure 4. Cross section pictures from a mouse model. (a) Exterior constraint $\lambda$ ($\lambda_{ex}$) = 0.0, surface constraint $\lambda$ ($\lambda_{surf}$) = 0.0, (b) $\lambda_{ex}$ = 1.0, $\lambda_{surf}$ = 1.0, (c) $\lambda_{ex}$ = 1.0, $\lambda_{surf}$ = 0.1 and (d) $\lambda_{ex}$ = 1.0, $\lambda_{surf}$ = 0.01. The darker the color, the higher the RBF value is. The RBF level set seems to have better distribution when proper $\lambda$ values are applied. The control radius of all the above RBF level set is 0.05.**

Assume that there are $n$ interpolation constraints, the equation above can be formulated as a linear system:

$$\begin{bmatrix} W & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} D \\ P \end{bmatrix} = \begin{bmatrix} H \\ 0 \end{bmatrix}$$

where

$$\begin{array}{ll} C_i = (1, c_i^x, c_i^y, c_i^z) & i = 1, ..., n \\ W_{ij} = w(c_i - c_j) & i, j = 1, ..., n \\ D = (d_1, ..., d_n)^T & \\ P = (p_0, p_1, p_2, p_3)^T & \\ H = (h_1, ..., h_n)^T & \end{array}$$

The system is symmetric and positive semi-definite, so there is always a unique solution for $D$ and $P$. We may solve this linear system to get $D$ and $P$ using LU factorization [10] or singular value decomposition (SVD) [11].

As for the interpolation constraints, we apply the normal constrains method described in [29]. There are two kinds of constraints assigned to the linear system in this method:

1. Surface constraints: We use the vertex set $V$ of a arbitrary 3D model as surface constraints. For each vertex $v_i \in V$, we set the value $h_i = 0$.

2. Normal constraints: For each vertex $v_i$, we create the paired positive-valued normal constraint $r_i = v_i + k \times (-\overrightarrow{n})$, where $\overrightarrow{n}$ is the normal at vertex $v_i$. For each $r_i$, we set the value $h_i$ a positive value, usually $h_i = 1$. The value of $k$ depends on the size and resolution of an object. In our case, we set $k = 0.01$ for the 3D objects which are scaled within a $2 \times 2 \times 2$ cube.

We can choose many basis functions for the interpolation, usually the appropriate basis function to use is $w(r) = |r|^3$. Dinh et al. [12] proposed a multiple order basis function that can control the smoothness of the implicit surface by changing basis function parameters. The basis function we invoke here is $w(r) = (|r|^2 + c^2)^{-0.5}$, where $c$ is the control radius. This function has a simple form and similar behavior to the multiple order one. Notice that the basis function must be continuous.

## 4.2 Properties of Radial Basis Functions

In this section, we will discuss about how the RBF parameters, such as interpolation constraints, radius and regularization parameters influence the RBF level set.

### 4.2.1 Interpolation Constraints

The RBF level set behavior is concerned with constraint distribution. The RBF of a sphere with uniform constraint distribution is prefect. All the level set surfaces are concentric spheres, which are shown in Figure 2 (a). As the reduced sphere models in Figure 2 (b) and (c), distortion happens in the RBF level set although the reconstructed surface boundary still seems the same. The RBF level set relaxes at the locations without the constraints, shown in the top region of the sphere in Figure 2 (c).

### 4.2.2 RBF Radius

Figure 3 shows the effect of the RBF control radius when applied to a reduced sphere model. As shown in Figure 3 (a), RBF surface with small control radius $c$ sometimes may become out of control. If we increase the control radius, the result will be better and smoother (Figure 3 (b) and (c)). Using large radius to construct a RBF surface, however, may lose some geometry details due to the global smoothing effect.

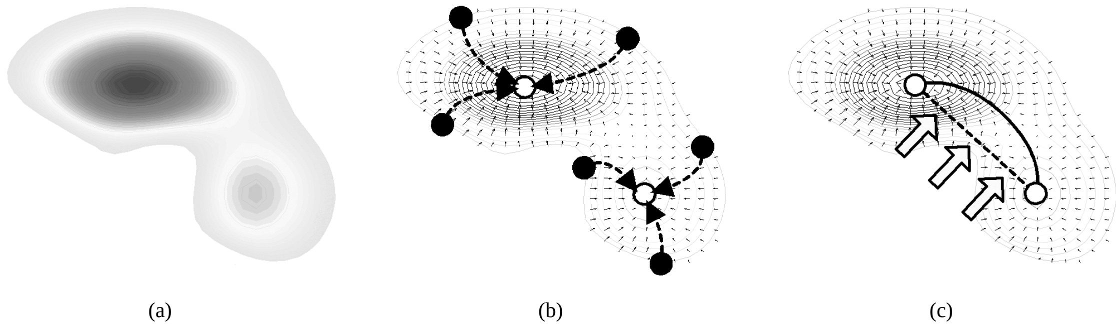



**Figure 5. Illustrations of the skeletonization algorithm, where color representation is the same as in Figure 4. (a) The object and its RBF level set. (b) A gradient vector field (the small arrows) is constructed with the partial derivatives of RBF. Surface points (the black points) shrink to the local maxima (the white points) enforced by the gradient vector field. (c) After the local maxima being gathered, we link two local maxima with connectivity relationship together with a Snake (active contour model, the black dash line). The arrows indicate the external force that pushes the Snake toward the direction that decreases the potential energy. The Snake itself deforms continuously until the potential energy is minimized (the black solid line), and a skeleton is found.**

#### 4.2.3 Regularization Parameter

We apply the method proposed by Dinh et al. [12] for the RBF surface smoothing. Several $\lambda$ values can be added to the diagonal elements in the RBF system matrix, which becomes:

$$\begin{bmatrix} w(r_{11})+\lambda_1 & \cdots & w(r_{1n}) & 1 & c_1 \\ \vdots & \ddots & \vdots & 1 & \vdots \\ w(r_{n1}) & \cdots & w(r_{nn})+\lambda_n & 1 & c_n \\ 1 & 1 & 1 & 0 & 0 \\ c_1 & \cdots & c_n & 0 & 0 \end{bmatrix}$$

The smoothness of the surface is proportional to the $\lambda$ values. As the $\lambda$ values become larger, the RBF surface becomes smoother. From the viewpoint of the RBF level set behavior, the level set seems to distribute better if we apply this technique, which is shown in Figure 4. The level set distribution behavior is important because it affects the RBF gradient vector field, which directs the convergence of the surface points.

## 5 Skeletonization Algorithm

Our skeletonization algorithm can be stated as follows:

1. Compute the RBF level set from a given 3D model.
2. For each point on the model, apply the gradient descent to it until a local maximum position is reached.
3. Cluster the local maxima in a given range.
4. Link any two maxima which have connectivity relationship together with the Snake (active contour model) method [17] and minimize the Snake energy.
5. Output the final positions of the Snakes as skeleton.

### 5.1 Surface Gradient Descent

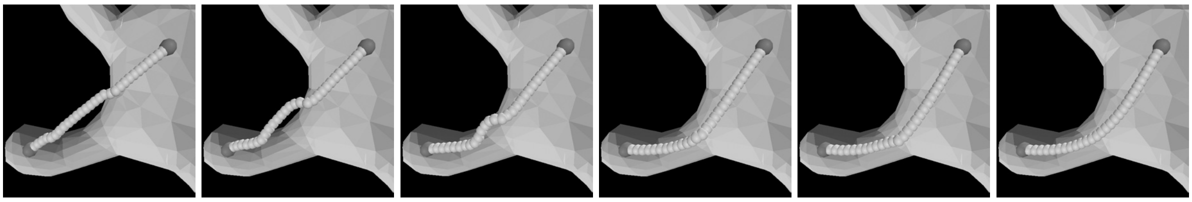
As previously discussed, RBF $f(x) = h$ is a continuous level set where $h$ indicates a height value for a designated surface. An interior point has a higher RBF value than an exterior one. With this idea, we can regard a cross section plane of RBF as a height contour map shown in Figure 5 (a). The most interior points are the highest peaks. Each point from the surface moves inside an object along the gradient direction until a corresponding peak (local maximum) is found.

A gradient vector field can be easily built by only substituting the partial derivatives of the basis function for the original one without re-computation. We choose the vertices of the model as the surface points to do the gradient descent without losing generality. Sometimes the vertices of an arbitrary model may not be well distributed. Methods such as dense sampling or random points relaxation [26] can be applied to make the surface points somewhat equally spaced.

For a surface point $v_i$ in the $i_{th}$ iteration of the gradient descent process, we calculate the new position of $v_{i+1}$ by:

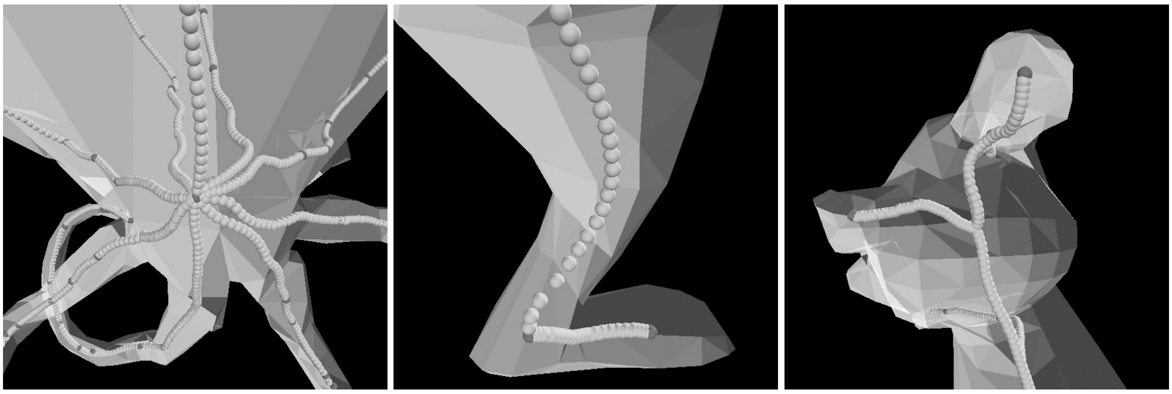
$$v_{i+1} = v_i + gradient(v_i) \times step$$

where $step$ is a small discrete factor. The iteration stops when the RBF value of $v_{i+1}$ is smaller than that of $v_i$ and



(a) (b) (c) (d) (e) (f)

**Figure 6. The skeleton deforming process, where it deforms successively until an energy-minimized position is found (From left to right).**



(a) (b) (c)

**Figure 7. Featured details of the selected skeleton. (a) The connection joint of the head and the eight tentacles of an octopus model. (b) The leg of a mouse model. (c) The head of a mouse model.**

the final position of $v$ is recorded as a local maximum. The whole process stops when all of the points remain in their corresponding local maxima. Figure 5 (b) illustrates the shrinking process.

### 5.2 Clustering

In practice, the final positions of the surface points may be close to, but not exactly the same as where the local maxima are. The range between the final position and the local maximum depends on the precision of gradient descent step. Due to this reason, we merge the final positions of these points in a given range to become a single local maximum point.

### 5.3 Connecting

As we use the vertices of the 3D model to be the gradient descent points, it is intuitive that the edges of a model become the connectivity relationship. For two shrinking points $v_i$ and $v_j$ and their corresponding local maxima position $m_i$ and $m_j$, the connectivity relationship is defined as if $v_i$ and $v_j$ are adjacent in the original 3D model, then $m_i$ and $m_j$ have connectivity relationship (either $m_i = m_j$ or $m_i$ and $m_j$ should have a link between them). We employ a Snake (active contour model) method to link two local maxima which have connectivity relationship. An active contour model is generated by adjusting a contour to minimize an energy function $E_{Snake}$:

$$E_{Snake} = E_{Internal} + E_{External}$$

where $E_{Internal}$ and $E_{External}$ represent the internal and external energy respectively. The internal energy depends on the intrinsic properties of the snake itself, such as its length or curvature. The external energy depends on the extrinsic factors, such as distance field in this case. Initially, we connect two local maxima as a straight line. We sample several points on the straight line as Snake candidates. The position of each candidate is adjusted on the constrained plane which is perpendicular to the original straight line (described in Appendix A). When the energy of the active contour model is minimized, we regard the Snake itself as a part of the skeleton (Figure 5 (c)). Figure 6 shows the deforming process of a Snake.

| Model | Constraint number | RBF radius | RBF construction (sec.) | Shrinking (sec.) | Skeleton construction (sec.) |
|---|---|---|---|---|---|
| Mouse | 1,912 | 0.1 | 51.05 | 44.80 | 182.45 |
| Bull | 1,982 | 0.2 | 55.93 | 79.96 | 210.33 |
| Octopus | 2,004 | 0.1 | 57.44 | 41.25 | 344.62 |
| Cobra | 1,752 | 0.2 | 39.17 | 62.13 | 110.04 |
| Frog | 1,890 | 0.06 | 48.14 | 21.39 | 312.27 |
| Pig | 2,170 | 0.1 | 74.83 | 74.28 | 294.30 |

**Table 1. Execution time statistics on a desktop computer with Intel Pentium-4 processor running at 1.5 GHz and with 256 Mbytes RAMBUS memory. We apply MathWork MATLAB C++ Library for the RBF construction matrix computation.**

## 6 Results

Detailed views of several models are shown in Figure 7. Six results from different models are shown in Figure 8 in colors. The execution time statistics for each model in Figure 8 is shown in Table 1. The error evaluation algorithm is described in Appendix B, and the skeleton we generated as compared to the ideal medial axis has a deviation from 2.9% to 8.5%.

Some detailed features, such as the hands of the mouse and the frog are not obviously because we apply larger RBF radius to make the level set smoother and the global smoothing effect has been applied to the fine features of the model. However, significant shape features are preserved well in the skeleton representation, and these are important to be used in object retrieval algorithms.

## 7 Conclusions and Future Work

Creating implicit surfaces with RBFs provides us a simple and robust method to generate arbitrary 3D object distance field, which is needed for building gradient vector field and then, surface point gradient descent. Our skeletonization method extracts both morphological and geometrical information. Significant geometry details, such as arms and legs in the model, are well preserved. There are many works need to be done in the future. From the viewpoint of research, we will keep working on the complicated relationship between the implicit surface, constraint distribution, control radius and regularization parameters. As for the basis functions, currently we only use the basis function $w(r) = (|r|^2 + c^2)^{-0.5}$ for all the RBF level set generation. In the future, we may continue to explore the RBF level set property with different basis functions. Computation efficiency is also an important task to be optimized. Meanwhile, since how to build a good RBF for our purpose still needs more consideration, recently we also try to use a different method which uses visibility and a repulsive force model to extract 3D skeletons. [32].

Besides, from the viewpoint of applications, there are several projects related to this method. For example, a 3D object retrieval system is going to take the skeletons as main comparison keys. We still need to evaluate the results compared with the method proposed by Hilaga et al. [16], Funkhouser et al. [14] and Chen et al. [7, 8]. We also want to apply the skeleton to the original model for animation use. We may extract a skeleton from a 3D model and select several points on the skeleton as joints. The pose of the 3D model can be changed by adjusting the skeleton. With this idea, we want to develop a system for artists to simply edit the skeleton movements and accurately animate the models with skin mesh method. This may become a useful tool for key frame animation.

## Acknowledgements

We would like to thank the reviewers for their valuable comments. We also thank 3DCAFE [1] to provide all 3D models used in this paper. This work was partially supported by grants from National Science Council, Taiwan, R.O.C. under NSC91-2213-E-002-066 and Silicon Integrated Systems (SiS) Education Foundation.

## Appendices

## A Coordinate Transformation

To adjust a point which moves in a plane, we can define a point as $\mathbf{r} = (p_x, p_y, 0)_p$ in plane coordinate. The normal of the plane is as $\mathbf{n} = (n_x, n_y, n_z)$ that is also the straight line between two local maxima. A rotation axis is $\hat{\mathbf{n}} = \mathbf{z} \times \mathbf{n} = (-n_y, n_x, 0)$. A rotation angle is $\Phi = \cos^{-1}(\mathbf{z} \cdot \mathbf{n})$. As Figure 9 shows, we can find the relation between before and after rotation[15].

$$\mathbf{r}' = \overrightarrow{ON} + \overrightarrow{NV} + \overrightarrow{VQ}$$

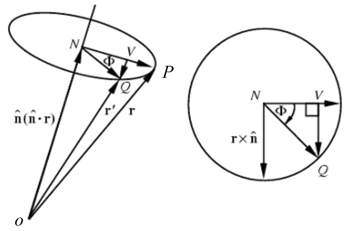



**Figure 8. Coordinate rotation**

$$= \mathbf{r}\cos\Phi + \hat{\mathbf{n}}(\hat{\mathbf{n}}\cdot\mathbf{r})(\mathbf{1}-\cos\mathbf{\Phi}) - (\mathbf{r}\times\hat{\mathbf{n}})\sin\mathbf{\Phi}$$

let $\hat{\mathbf{n}} = (x, y, z)$, then

$$\mathbf{r}' = \mathbf{M}\cdot\mathbf{r}$$

where $\mathbf{M}$ equals to

$$\begin{bmatrix} x^2+\cos\Phi(1-x^2) & xy(1-\cos\Phi)-z\sin\Phi & zx(1-\cos\Phi)+y\sin\Phi \\ xy(1-\cos\Phi)+z\sin\Phi & y^2+\cos\Phi(1-y^2) & yz(1-\cos\Phi)-x\sin\Phi \\ xz(1-\cos\Phi)-y\sin\Phi & zy(1-\cos\Phi)+x\sin\Phi & z^2+\cos\Phi(1-z^2) \end{bmatrix}$$

$$= \begin{bmatrix} n_y^2+\cos\Phi(1-n_y^2) & -n_xn_y(1-\cos\Phi) & n_x\sin\Phi \\ -n_xn_y(1-\cos\Phi) & n_x^2+\cos\Phi(1-n_x^2) & n_y\sin\Phi \\ -n_x\sin\Phi & -n_y\sin\Phi & \cos\Phi \end{bmatrix}$$

Based on this relation, we can transform a point in plane coordinate into world coordinate $\mathbf{p_w}$ as follows:

$$\mathbf{p_w} = \mathbf{M}\cdot\mathbf{r} + \mathbf{T}$$

where $\mathbf{T}$ is the position of a shrinking candidate.

## B Error Evaluation

Here we describe the error evaluation method used in this paper:

1. The surface of the input model is densely sampled using $n$ points (In our evaluation, the models are sampled by more than 20,000 points).

2. For each surface point $v$, find the nearest point $k$ on the skeleton then define $v \in M^{-1}(k)$ and the distance between $v$ and $k$ is $\rho(v)$.

3. For each $k$, sort each $v \in M^{-1}(k)$ and find the smallest $\rho(v)$ and define it as $r(k)$, which is the zero error position. For each surface point $v$ we calculate the error $e(v) = \frac{\rho(v)}{r(k)}$.

4. The total MSE equals to $\sqrt{\frac{\sum e^2(v)}{n}}$

## References


[1] http://www.3dcafe.com

[2] N. Amenta, S. Choi, and R. Kolluri. The Power Crust. *Proceedings of the sixth ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001.

[3] I. Bitter, A. E. Kaufman, and M. Sato. Penalized-Distance Volumetric Skeleton Algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):195–206, 2001.

[4] J. Bloomenthal, C. Bajaj, J. Blinn, M.-P. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill. *Introduction to Implicit Surfaces*, page 9. Morgan Kaufmann, 1997.

[5] H. Blum. *A Transformation for Extracting New Descriptors of Shape*, pages 362–380. MIT Press, 1967.

[6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. *SIGGRAPH 2001 Conference Proceedings*, pages 67–76.

[7] D.-Y. Chen. http://3d.dhs.org/∼dynamic/

[8] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On Visual Similarity Based 3D Model Retrieval. *Paper submitted to Eurographics 2003*.

[9] J.-H. Chuang, C.-H. Tsai, and M.-C. Ko. Skeletonization of Three-Dimensional Object Using Generalized Potential Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1241–1251, 2000.

[10] B. N. Datta. *Numerical Linear Algebra and Applications*, pages 111–132. Brooks/Cole, 1995.

[11] B. N. Datta. *Numerical Linear Algebra and Applications*, pages 551–599. Brooks/Cole, 1995.

[12] H. Q. Dinh and G. Turk. Reconstructing Surfaces by Volumetric Regularization. *GVU-00-26, College of Computing, Georgia Tech*, December 2000.

[13] H. Edelsbrunner and E. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[14] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A Search Engine for 3D Models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.

[15] H. Goldstein. *Classical Mechanics*, pages 164–166. Addison-Wesley, 1980.

[16] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. *SIGGRAPH 2001 Conference Proceedings*, pages 203–212.

[17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1:321–331, 1987.

[18] F. Lazarus and A. Verroust. Level Set Diagrams of Polyhedral Objects. *Proceedings of the fifth ACM Symposium on Solid Modeling and Applications*, pages 130–140, 1999.

[19] F. Leymarie and M. Levine. Simulating the Grassfire Transform Using an Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):56–75, 1992.

[20] M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38:327–341, 1987.

[21] N. Mayya and V. T. Rajan. Voronoi Diagrams of Polygons: A Framework for Shape Representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–643, 1994.

[22] M. Mortara and G. Patané. Affine-Invariant Skeleton of 3D Shapes. *Proceedings of Shape Modeling International 2002*, pages 245–278, 2002.

[23] L. R. Nackman and S. M. Pizer. Three-Dimensional Shape Description Using the Symmetric Axis Transform I: Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):187–202, 1985.

[24] R. Ogniewicz and M. Ilg. Voronoi Skeletons: Theory and Applications. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 63–69, 1992.

[25] G. Reeb. Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numerique. *Comptes Rendus Acad. Science Paris*, 222:847–849, 1946.

[26] G. Turk. Generating Textures on Arbitrary Surfaces Using Reaction Diffusion. *SIGGRAPH 1991 Conference Proceedings*, pages 289–298.

[27] G. Turk, J. F. O'Brien. Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics*, 21(4):855–873, 2002.

[28] G. Turk and J. F. O'Brien. Shape Transformation Using Variational Implicit Surface. *SIGGRAPH 1999 Conference Proceedings*, pages 335–342.

[29] G. Turk and J. F. O'Brien. Variational Implicit Surfaces. *GVU-99-15, College of Computing, Georgia Tech*, May 2000.

[30] L. Wade and R. E. Parent. Automated Generation of Control Skeletons for Use in Animation. *The Visual Computer*, 18(2):97–110, 2002.

[31] F.-E. Wolter. Cut Locus and Medial Axis in Global Shape Interrogation and Representation. *MIT Sea Grant Report*, 1992.

[32] F.-C. Wu, W.-C. Ma, P.-C. Liou, R.-H. Liang, and M. Ouhyoung. Automatic Skeleton Generation by Domain Connected Graph. *Paper submitted to Eurographics 2003*.

[33] Y. Zhou and A. Toga. Efficient Skeletonization of Volumetric Objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):195–206, 1999.

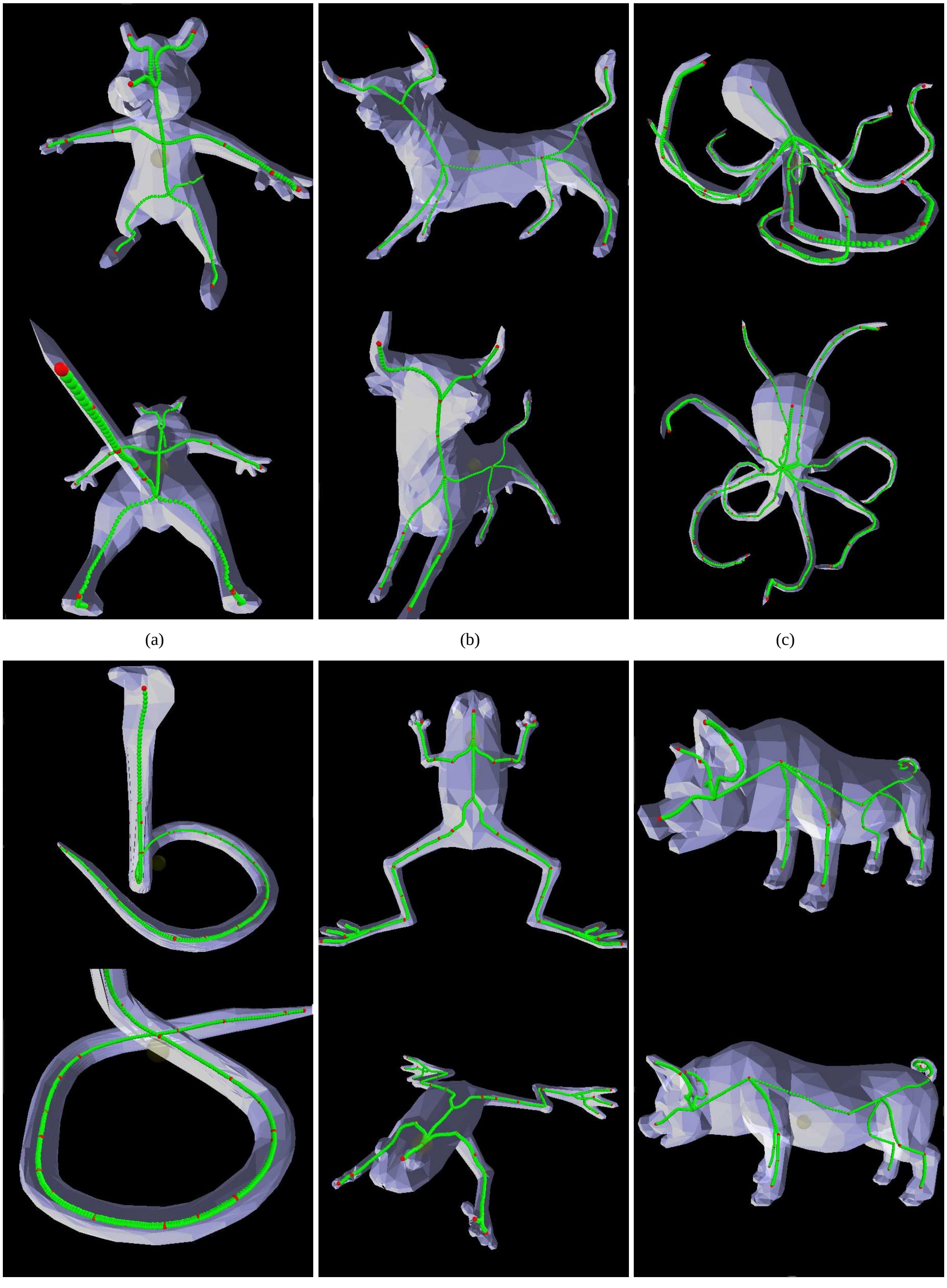


Figure 8. Skeletons extracted from 3D models. (a) Mouse, (b) Bull, (c) Octopus, (d) Cobra, (e) Frog, (f) Pig. Models are rendered in semi-transparent style. The red and green spheres represent local maxima and skeleton points respectively.

# Prong Features Detection of a 3D Model Based on the Watershed Algorithm

Fu-Che Wu* Bing-Yu Chen† Rung-Huei Liang* Ming Ouhyoung‡

National Taiwan University

## 1 Introduction

In this paper, a simple and robust prong features detection algorithm is proposed. A prong feature is an assisting feature that can be used in many applications. For instance, it can be used to identify a model that consists of several prong parts for model decomposition. It represents a useful feature for skeleton extraction as well as a comparable feature for object matching. In addition, it could also be a fast alignment feature for model alignment and morphing. Furthermore, it is an invariant feature for mesh simplification.

In traditional application, a watershed algorithm is often applied to solve segmentation problems, such as [Mangan and Whitaker 1999]. In this paper, we outline an approach based on mesh connectivity relationships which apply to watershed algorithms on a 3D data to detect some meaningful features. If we could find other suitable continuing scale functions on a 3D data, they would also be suitable to detect or to segment other features.

## 2 Algorithm

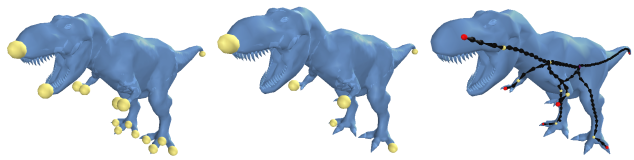
First, we construct a normalized scale function on the input mesh to describe the position of a potential prong. The scale function is calculated by the sum of the geodesic distance from each node among the mesh to the evaluated node [Lazarus and Verroust 1999]. The scale function is defined in the following: $f(v) = \sum_{\forall v_i \in S} Geodesic(v, v_i)$, where $v$ is the vertex on the shape $S$. To ensure the calculation of the geodesic distance approximates the real value, a remesh pre-process is required. The result is shown in the following:

Figure 1: From the scale function, a higher value indicates a prong feature that is denoted by the green color.

In order to find these prong features we apply a watershed algorithm. First, all of the vertexes are sorted by the scale function value. Then, the maximum value of the vertex is picked up as the first prong feature and marks it as traversed. As we have to focus on the feature detection, the water level decreases iteratively from the maximum value. The decreasing level effects the sensitivity of a prong feature. The prong shape's valley value divided by its peak value is defined as the smoothness ratio. If the smoothness ratio is larger than the threshold, the result is a small prong and will be ignored. Each non-traversed vertex in the sorting sequence will reset the water level, i.e., its function value is multiplied by the smoothness ratio. Consequently, we can mark all vertexes as traversed if their function value is above water level and if they are a neighbor of the traversed set. If the vertex is not in the traversed set, it is a new prong feature. The pseudo code is as follows:

```
Function watershed algorithm
V=sort (S); //if v_i ∈ V, f(v_0) > f(v_1) ... > f(v_n)
T ← v_0;    //insert v_0 into the traversed set
P ← v_0;    //v_0 is the first prong feature.
for each (v_i ∈ V .and. v_i ∉ T) {
 water_level=f(v_i)× smoothness_ratio;
 do{
   for each ( v_j ∈ Neighbor(T) .and. v_j ∉ T)
     if (f(v_j) > water_level)
       T ← v_j;
   }while (there exists a new traversed vertex)
 if ( v_i ∉ T)
   T ← v_i; P ← v_i;
}
```

For a model with about 8,000 triangles, the execution time is 22 seconds using a notebook PC with an Intel Pentium 4 2.4GHz CPU and 512 Mb memory. About 75% of execution time is needed in order to calculate the scale function. The following graph shows the demonstration of an application which is based on the detected prong features to construct a skeleton.



(a) (b) (c)

Figure 2: The results are shown in different decreasing levels. (a) Fifteen features are detected when the smoothness ratio is set to 1.0. (b) Six features are detected when the smoothness ratio is set to 0.85. (c) A skeleton is constructed based on these prong features

## 3 Conclusion

We have presented an algorithm in order to detect the prong features on a 3D mesh. The procedure includes a smoothness ratio that can be adjusted for different applications. According to our experience, setting the smoothness ratio to 0.85 is suitable to detect significant prong features for most models. We also have demonstrated that prong features are a useful information in skeleton construction.

## References

LAZARUS, F., AND VERROUST, A. 1999. Level set diagrams of polyhedral objects. In *ACM Solid Modeling and Applications 1999 Conference Proceedings*, 130–140.

MANGAN, A. P., AND WHITAKER, R. T. 1999. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics 5*, 4, 308–321.

*e-mail:{joyce, liang}@cmlab.csie.ntu.edu.tw
†e-mail:robin@ntu.edu.tw
‡e-mail:ming@csie.ntu.edu.tw

# Enhanced 3D Model Retrieval System through Characteristic Views using Orthogonal Visual Hull

Shuen-Huei Guan* Ming-Kei Hsieh* Chia-Chi Yeh* Bing-Yu Chen†

National Taiwan University

## 1 Introduction

The introduction of the Princeton Shape Benchmark database (PSB) [Shilane et al. 2004] has given rise to extensive research into ways to accurately perform 3D matching. Among the extant methods proposed, the LightField Descriptor (LFD) [Chen et al. 2003] based method currently is the most accurate, but it suffers from a deficiency in on-line matching speed. The contribution of this research is twofold. First, fewer characteristic views are used instead of huge images to boost the on-line retrieval time. Second, depth information is employed to keep the retrieval accuracy.

## 2 Description

In Chen et al.'s system [Chen et al. 2003], several images, taken from different views around, are used to represent a 3D model, which are then transformed into descriptors. Thereafter, the matching of any two 3D models is achieved by matching two groups of 2D images by those descriptors. 100 images for each model are required, represented by 4 descriptors, which are Fourier descriptor for contour region, Zernike moment descriptor for shape region, circularity and eccentricity.

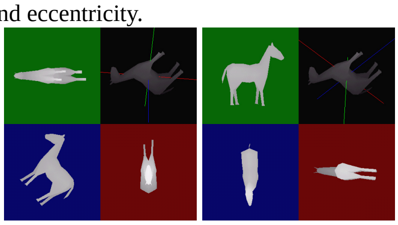


Figure 1: Different pose estimations by PCA (left) and OVH (right).

Our proposal to improve on-line speed is based on reducing the number of images required for each 3D model. Some other research made use of three principal axes by Principal Component Analysis (PCA), applied in pose estimation phase. Although PCA has nice meaning in statistics, it does not fit most cases in 3D model retrieval and its resulting views do not fit user's input of 2D sketches, as illustrated in Figure 1.

Hence, we propose Orthogonal Visual Hull (OVH) to establish pose estimation. Visual hull is an algorithm to reconstruct a 3D model from several views. Given a 3D model and images taken around, since visual hull does not capture concavities, the volume of the reconstructed model is always greater. In other words, the smaller volume we get, the closer approximation it fits the original model. There are two criteria to determine the number and mutual relation of views: 1) The views can be transformed into characteristic axes without obstruction and 2) the visual hull by these views is very close to the original model. Therefore, we chose three orthogonal views because there are three characteristic axes in pose estimation, and relatively orthogonal views carve much space in most cases. Then the next task is to get the views forming the least volume. The global minimum is extracted by the heuristic: (i) find local extrema from widely varying starting orientation of the 3D model by an iterative stochastic method or downhill simplex algorithm as Figure 2, (ii) pick the most extreme of these. The method is slow, but off-line and we are exploring to use faster algorithms, like conjugate gradient descent. Sometimes, the extreme is not unique. In our experiment of randomly picking one extrema to 3D model retrieval system [Chen et al. 2003], the resulting retrieval accuracy is nearly the same.

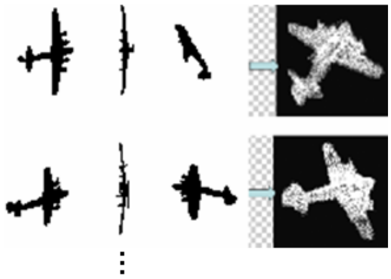


Figure 2: Heuristically pick the characteristic views that form the least volume.

Using fewer images raises the possibility of losing information necessary for 3D matching. By employing the Generic Fourier Descriptor (GFD) [Zhang and Lu 2002], we are able to compensate the information loss because it supplies additional depth information. Therefore, for each image, its contour, shape and depth information are all covered. In order to make use of GFD, the system takes 2 images for each characteristic axes. Consequently, the total number of images taken from each 3D model is reduced from 100 to 6. The number of pair-wise image matching is reduced from 5460 to 24. And the retrieval accuracy is nearly the same to the original system.

In summary, this research enhances the extant best matching method [Chen et al. 2003] by two means: 1) improving the on-line speed 93 times faster at the cost of 5 times deceleration of the off-line process, and 2) keeping the retrieval accuracy. The proposed OVH is also a brand new idea for pose estimation. Additionally, the shapes of the resulting characteristic views generated by OVH more closely fit human visual perception.

| | Storage Size (bytes) | Off-line Generate Time (s) | On-line Match Time (ms) |
|---|---|---|---|
| LFD | 4,700 | 3.25 | 1.300 |
| LFD with OVH | 2,592 | 15.72 | 0.014 |

## References

CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3d model retrieval. *Conputer Graphics Forum 22*, 3, 223–233. (Proc. EG 2003).

SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The princeton shape benchmark. In *Proc. SMI 2004*.

ZHANG, D., AND LU, G. 2002. Generic fourier descriptor for shape-based image retrieval. In *Proc. ICME 2002*, vol. 1, 425–428.

*e-mail:{drake, lionkid, id5}@cmlab.csie.ntu.edu.tw

†e-mail:robin@ntu.edu.tw