

國立交通大學

資訊工程系

碩士論文

具幾何與材質保留之網格化簡

Mesh Simplification with Geometry and Material
Preserving



研究生：廖彥塵

指導教授：莊榮宏 博士

中華民國九十三年十月

具幾何與材質保留之網格化簡

Mesh Simplification with Geometry and Material Preserving

研究生：廖彥塵

Student : Yen-Chen Liao

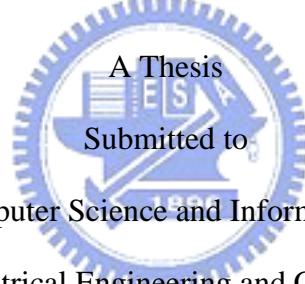
指導教授：莊榮宏 博士

Advisor : Dr. Jung-Hong Chuang

國立交通大學

資訊工程學系

碩士論文



Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Computer Science and Information Engineering

October 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年十月

具幾何與材質保留之網格化簡

研究生：廖彥塵

指導教授：莊榮宏 博士

國立交通大學資訊工程學系

摘要

在三維計算機圖學的應用軟體中需要使用複雜且高精細度之模型，然而在即時描繪的需求之下，為了加速繪製速度通常以低複雜度的模型代替高精細度的。網格化簡技術使用區域化簡操作去減低模型複雜度，並維持模型的可觀度，許多已經被提出的技術測量化簡前後的兩個網格在三維空間中距離的差距來引導化簡的順序，但確無法考量到模型上的各種屬性。在本論文中我們提出一新的網格化簡方法，以射線去測量化簡前後模型之視覺相似度來引導化簡，維持網格之幾何與材質呈現在螢幕上的可觀性。

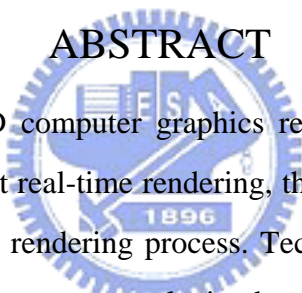
Mesh Simplification with Geometry and Material Preserving

Student : Yen-Chen Liao

Advisor : Dr. Jung-Hong Chuang

Department of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

The logo of National Chiao Tung University is a circular emblem. It features a central shield with a book and a torch, surrounded by the university's name in Chinese and English. The year '1896' is inscribed at the bottom of the shield. The entire emblem is set against a blue background with a gear-like border.

Many applications in 3D computer graphics require complex, highly detailed models. Under the requirement real-time rendering, the complex models usually need to be reduced to speed up the rendering process. Techniques of mesh simplification are applied to reduce the geometry complexity by performing local simplification operation sequentially, and maintain its fidelity. Most methods measure 3D distance to guide the simplification, but they don't consider the attributes, such as color, normal, and texture contents. In this thesis, we propose a new mesh simplification algorithm that preserves the geometric and material appearance of a given mesh. The algorithm uses rays to measure the visual similarity of simplified mesh and original mesh to preserve appearance of texture models.

誌謝

首先要感謝是我的指導教授莊榮宏博士，由他以經驗和專業知識引導整個論文的完整性，以及對於論文敘述方法的指導，才有這一篇論文的誕生。此外特別感謝 danki 學長在整個論文完成的過程中給予完全的關心和建議，ksl 和 dis 在二年研究所其間給予我在課業上的幫助，wylee 在我身體健康建議和心理方面開導，suen、peng、jhlu、jeng 在生活上、計畫上的幫助，讓我全力準備論文，chihchun 在程式方面的幫助，有緣能認識 foldpinna、charlice、yglin、juan、dandan 也在生活上給予我協助。

在研究期間，許多支持並關心我的好友家人給我鼓勵和支持，包括爸媽和東華大學的同學們，這是支持我繼續做這項研究的動力。父母在整個研究所生涯之中全力的支持，讓我無後顧之憂，由衷地感謝以上讓我感到溫暖的朋友家人同學教授。

■

目錄

圖表目錄

1	緒論.....	1
1.1	研究動機與目標.....	1
1.2	論文組織.....	3
2	相關研究.....	4
2.1	區域化簡操作.....	4
2.1.1	邊線摺疊法(Edge collapsing).....	4
2.1.2	頂點叢集法(vertex clustering).....	6
2.1.3	多精細度取樣法(Multi-resolution Re-sampling).....	6
2.2	化簡成本估計.....	7
2.2.1	二次方誤差評量.....	7
2.2.2	外觀保持式化簡法.....	10
2.2.3	影像導向式化簡法.....	12
3	以射線為基礎測量視覺相似度.....	15
3.1	化簡演算法概要.....	15
3.1.1	化簡演算法.....	15
3.1.2	暫緩估計化簡成本(lazy evaluation).....	17
3.2	化簡成本計算.....	18
3.2.1	取樣視點.....	20
3.2.2	取樣射線.....	21
3.2.3	局部視覺相似度化簡成本.....	23

3.2.4	整體視覺相似度化簡成本.....	26
3.2.5	視覺相似度化簡成本結合全邊線摺疊之作法.....	28
4	結果與討論.....	31
4.1	以視覺相似度化簡成本計算之效果.....	31
4.1.1	視覺相似度化簡成本與外觀保持式化簡法比較.....	35
4.1.2	視覺相似度化簡成本與影像導向式化簡法比較.....	37
4.2	視覺相似度化簡成本使用全邊線摺疊效果.....	39
5	結論與未來發展方向.....	41
5.1	結論.....	41
5.2	未來發展方向.....	42
	參考資料.....	43



圖表目錄

圖 2.1 邊線摺疊區域操作[HDD+ 93]。.....	5
圖 2.2 (a)網格對摺[XEV97](b)非流型形格。.....	5
圖 2.3 頂點叢集法[LE97]。.....	6
圖 2.4 多精細度取樣法[EDD+95]。.....	7
圖 2.5 在五維空間中由p,q,r所形成三角形， e_1 和 e_2 是區域性座標軸[GH98]。.....	9
圖 2.6 以二維的例子說明貼圖變量的計算方式。.....	10
圖 2.7 以方形盒來限制最大的貼圖變量[COM98]，以二維為例子，(a)化簡前各三角形已有至原始模型的貼圖變量，(b)進行化簡後以新的邊界方形來限制到原始模型最大的貼圖變量。.....	11
圖 2.8 影像導向式化簡[LT00]，固定在某些視點繪製影像，比較化簡前後每張影像的差距做為化簡成本。.....	12
圖 3.1 半邊線摺疊操作。每條邊線 e 會有兩個方向的摺疊，箭頭代表摺疊的方向，從 v_{start} 摺疊至 v_{end} ，而顏色代表化簡後所影響網格改變的區域。.....	16
圖 3.2 (a)對紅箭頭方向進行的半邊線摺疊，(b)所有鄰近化簡影響區域的半邊線(藍箭頭表示)都必需重新計算其化簡成本。.....	17
圖 3.3 以二維為例，測量視覺相似度。在化簡區域的周圍平均取樣一些視點。....	20
圖 3.4 以二維為例，從每個視點射出一些射線得到的顏色差異 d_{RMS} ，做為視覺相似度的誤差。.....	21
圖 3.5 掃描線演算法，落在三角形之中的點即為取樣點， Δ 為取樣的單位長度。.....	22
圖 3.6 以2D為例。(a)顏色代表貼圖的內容，當化簡區域是在貼圖變化的地方，射線的顏色差異 d_{RMS} 會較大(b)當化簡區域為輪廓邊時，會有沒有交點的射線，必需對這些射線給予適當的處罰值。.....	23
圖 3.7 (a)模型 M_0 包含化簡區域的三角形，(b)進行化簡的區域，(c)進行化簡後的區域。.....	26
圖 3.8 邊線化簡前後在貼圖域的變化。.....	28
圖 3.9 全邊線摺疊。.....	29
圖 3.10 使用全邊線摺疊導致射線不常用只相交於一個網格。.....	29
表 4.1 以豬模型比較局部和整體幾何誤差保留能力。.....	32

圖 4.1 (a)豬原始模型。使用局部(b)和整體(c)視覺相似度化簡成本，由上而下分別為化簡至 2000,1000,700 個三角形。.....	33
圖 4.2 比較四種方法化簡模型的結果，左列為 1000 個三角形，右列為 400 個(a)二次方誤差評量(b)本論文方法(c)影像導向式化簡法(d)外觀保持式化簡法。.....	34
圖 4.3 在 400 個三角形的豬模型上，視覺相似度成本計算較可保留其外觀。...	35
圖 4.4 兔子模型原始三角形數為 10000，以(a)視覺相似度化簡成本，(b)外觀保持式化簡法，化簡至 700 個三角形.....	36
圖 4.5 外觀保持式與視覺相似度化簡成本比較，紅色箭頭代表在貼圖空間中貼圖拉扯嚴重的區域。.....	37
圖 4.6 影像導向式化簡法所不能保留的化簡區域，(a)(b)同為化簡至 2000 三角形的模型，(c)(d)為化簡至 400 三角形。.....	38
表 4.2 使用豬模型比較全邊線與半線摺疊對於局部視覺相似度化簡成本計算的幾何誤差。.....	39
圖 4.7 使用全邊線摺疊之局部視覺相似度化簡成本計算。.....	40



第一章

緒論

1.1 研究動機與目標

多邊形網格模型(polygonal mesh model) 是目前在 3D 電腦圖學領域中最常見的模型描述與儲存方式，並且其顯像運算有 3D 繪圖硬體的支援。但此種格式的一大缺點是在描述複雜且精密的模型時，需要大量的多邊形資料，因而嚴重影響 3D 繪圖硬體所支援的顯像速度。因應電腦遊戲產業的特性，使用者的硬體配備由高階至低階都有，因此遊戲對電腦資源的使用必需非常的節制且需能調整。少量多邊形網格模型和多層精細度模型(level of detail modeling)在 3D 遊戲程式中非常常見，並用貼圖來彌補其精細度之不足。

多精細度模型簡化技術已經有許多方法被提出，無論是模型簡化的方法或是化簡成本的計算，基本上都是以移除非特徵處的幾何資料，來獲得少多邊形數目高品質的模型。簡化的方法最廣被接受的有頂點叢聚集法(vertex clustering)[LE97]、頂點剔除法(vertex decimation)[SZL92]、邊線摺疊法(edge collapsing)[Hoppe96]及多精細度取樣法[EDD+95]，都是可以減少多邊形數目的區域化簡操作(local simplification operation)。然而計算化簡成本的機制是模型化簡中最重要的部分，化簡成本決定進行化簡的順序，一般是挑選化簡成本小的區域來化簡網格。模型化簡的結果完全會依化簡成本測量特性的不同而有所不同。

當模型在化簡時，要保持與原始模型的相似度，應減少化簡之幾何誤差以維持化簡後之模型品質是很直覺的方式，因此目前已經有許多各式各樣的幾何化簡成本

演算法被提出來，如計算頂點到平面的距離當作其頂點剔除化簡方法的成本 [SZL92]，利用 Hausdorff 距離測量化簡後模型和原始模型的幾何差距 [KLS96]，以能量函式(energy function)引導其邊緣摺疊的進行順序，而能量函式以幾何距離計算為基礎來制定 [HDD+93]，以化簡區域的三角形的長度和夾角來決定頂點叢集時需要叢集到哪一頂點 [RB93]，封套式簡化法(Simplification Envelope)將模型放大及縮小，化簡時不能超過此封套，因此可以保證化簡的誤差不超過封套 [CVM+96]，用多層次分析(multi-resolution analysis)去化簡任意的網格 [EDD+95]，以二次方誤差評量(quadric error metric)來計算一頂點至平面的距離做為幾何誤差之估計 [GH97]。

目前的模型不只有幾何空間中的三維位置而已，為了模型的外觀，有愈來愈多的資訊，例如貼圖、法向量、顏色等。幾何距離的化簡成本計算方法沒有考慮這些資訊在化簡後之改變，用這些幾何測量做出來的化簡模型往往不能保留模型的外觀。Garland et al.以三維位置延伸加上貼圖座標或法向量或顏色等等的外觀屬性形成一高維度的向量，在高維度的空間之中計算幾何的誤差，因此化簡時可以考量三維空間及貼圖座標等屬性 [GH98]。Cohen et al.透過貼圖座標的參數化來尋找化簡前後網格上對應的點，以此兩點的距離做為化簡的成本，透過幾何誤差來確保貼圖的內容不會在三維空間中移動太多 [COM98]。

然而以上提到保持外觀屬性的方法都沒有真正考量化簡對貼圖內容(texture content)的影響，Lindstrom et al.以比對化簡前後模型影像差異做為化簡的成本 [LT00]，因此對於貼圖內容是可以完全呈現在最後的影像之上以供比較，化簡會依照貼圖內容的不同而有不同的化簡結果。只考量貼圖座標的化簡成本計算，其成效和參數化的好壞有很大的關係，常會使得較多的頂點被保留在對外觀上不重要的地方。

針對以上的研究，我們提出視覺相似度(Visual Similarity)之化簡成本計算，以射線為基礎去測量網格在化簡前後視覺相似度，期望達成目標為：

1. 能依據模型上貼圖內容的變化做適當的化簡
2. 可保留模型上比較明顯的幾何特徵

3. 對於可化簡的區域不論在模型的內部外部，皆給予平均的機會計算其化簡成本。

1.2 論文組織

第二章將會介紹模型化簡中的區域化簡操作和化簡成本之計算，化簡成本計算將介紹維持幾何的成本計算法和維持外觀的成本計算法，第三章會詳細說明我們提出的視覺相似度化簡成本如何計算，包括所使用的區域操作和化簡演算法流程，第四章以視覺相似度化簡成本計算的化簡結果，與其他的方法做比較，強調本論文方法的優勢之處，第五章則做個總結以及提出一些未來的研究方向。



第二章

相關研究

多精細度模型簡化技術已經有許多方法被提出，無論是模型簡化的方法或是化簡成本的計算，基本上都是以移除非特徵處的幾何資料，來達到少多邊形數目高品質的模型。2.1 節將介紹幾種模型簡化的區域化簡操作(local simplification operation)，包括頂點叢聚集法、邊線折疊法及多精細度取樣法。計算化簡成本的機制是模型化簡中最重要部分，化簡成本決定進行化簡的順序，而模型化簡的結果完全會依化簡成本測量特性的不同而有所不同，化簡成本的計算大致分為以幾何距離測量成本或以保持模型外觀測量成本的不同方式。2.2 節中將介紹三種化簡成本的計算方法，包括二次方誤差評量、外觀保持式化簡法、影像導向式化簡法，其中後兩者化簡成本計算可以維持貼圖模型的外觀。本論文以射線為基礎計算化簡成本，並試圖去克服這兩種方法的缺點。

2.1 區域化簡操作

在這節中將介紹一般多層次精細度模型技術中所常用的區域化簡操作，每一種區域化簡操作都可減少小部分的網格複雜度。

2.1.1 邊線摺疊法(Edge collapsing)

Hoppe 首先提出用邊線摺疊做為化簡網格的區域化簡操作，每次的邊線摺疊最多可以減少一個點和兩個三角形，將邊線(v_1, v_2)摺疊至單一頂點 v ，如圖 2.1 所示。

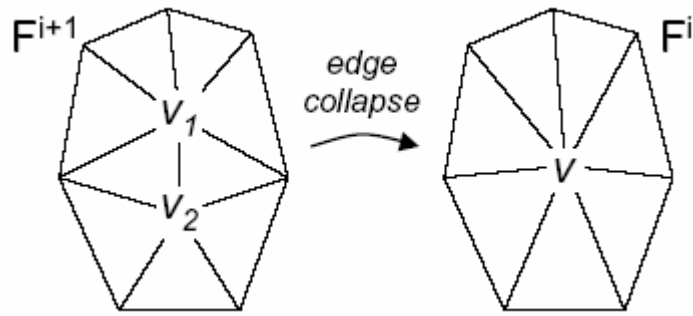


圖 2.1 邊線摺疊區域操作[HDD+ 93]。

邊線摺疊的相反操作就是頂點分裂(vertex split)，很容易增加多邊形的數目。因此這個特性，邊線摺疊目前被廣泛的使用，包括漸進式網格(progressive mesh) [Hoppe96]、視點相依之網格化簡(View-dependent simplification) [XV96] [Hoppe97]。邊線摺疊分為兩種，半邊線摺疊(half edge collapsing)和全邊線摺疊(fully edge collapsing)，在本論文中兩者都有使用，在第三章中會詳細介紹，兩者的差別在於有無產生新的頂點。邊線摺疊相非常容易實作，但要注意網格對摺(Mesh foldover)和會造成非流形(non manifold)的問題，兩者都使得網格的外觀改變非常的劇烈。網格對摺可以經由測試摺疊前後網格上的法向量改變的角度，一般改變超過約 90 度即為網格對摺情況，如圖 2.2(a)。許多化簡演算法是只能在流形的模型上進行化簡，因邊線摺疊造成非流形，不但對視覺和化簡演算法都造成影響，因此化簡時需注意化簡後一條邊線不能被二個以上的三角形相鄰，如圖 2.2(b)。

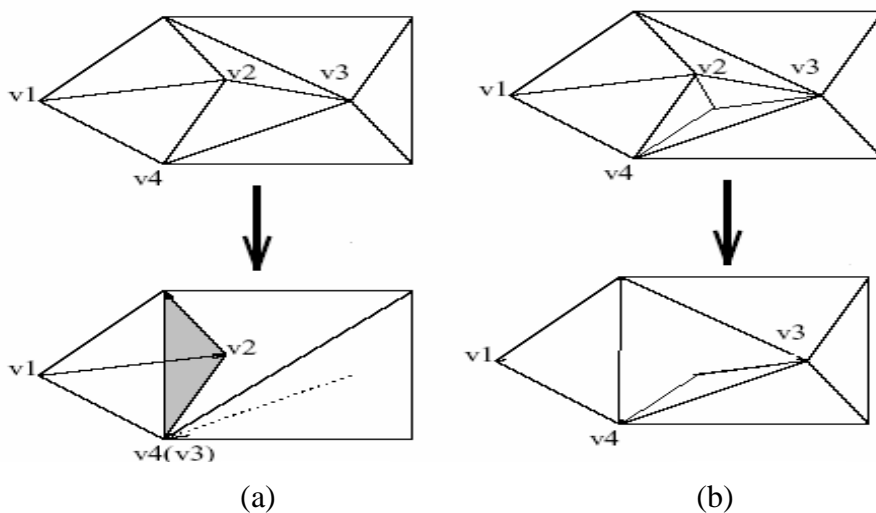


圖 2.2 (a)網格對摺[XEV97](b)非流形網格。

2.1.2 頂點叢集法(vertex clustering)

頂點叢集法[RB93]基本上是一種重新取樣頂點的行為。概念上而言，當一個物體遠離視點的時候，原始模型上有許多頂點事實上被投影到同一個像素上，因此將模型上一些足夠接近的點直接用一個頂點代替，並不會影響輸出的品質。基於這個觀念，頂點叢集法對模型的方形盒(bounding box)做空間分割(spatial subdivision)，將模型所佔據三維空間分成好幾個小細格(cell)，然後在每一個細格間選擇一個重要性最高的代表頂點代表同細格內所有的其他頂點，如圖 2.3 所示。分割細格的方法有平均分割(uniform)[RB93]、八分樹分割(octree)[LE97]等分割法，頂點叢集法保證每一個頂點被移動的距離絕對不超過細格的對角線長度，而模型化簡結果由細格大小來決定。要注意此方法可以很快的降低模型的複雜度，但不保證可以維持模型原始的拓樸。

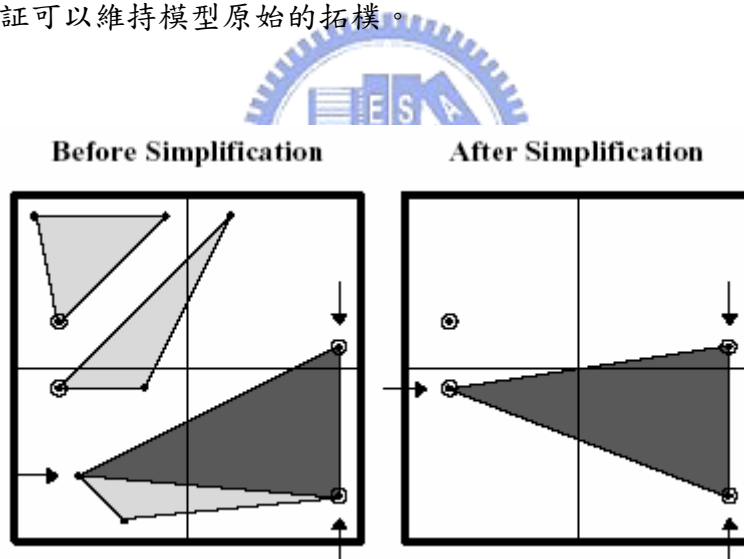


圖 2.3 頂點叢集法[LE97]。

2.1.3 多精細度取樣法(Multi-resolution Re-sampling)

Eck et al.提出經三角形分裂來得到所需要精細度之模型[EDD+95]，將離散取樣型態的多邊形大致性地分割為數個區域(patch)，每一個區域重新參數化回歸成連續的參數曲，在各個參數曲面重新做取樣得到取樣頻率較低的多邊形模型。圖 2.4

由左至右分別為原始模型、分割成數個區域後得到的最粗糙模型和以粗糙模型上每個三角形為 1-4 細分後得到的複雜度較低的網格，Lee et al.也以類似的方法得到多精細度模型[LSS+98]，不同的是找分割區域方法和參數化方法。

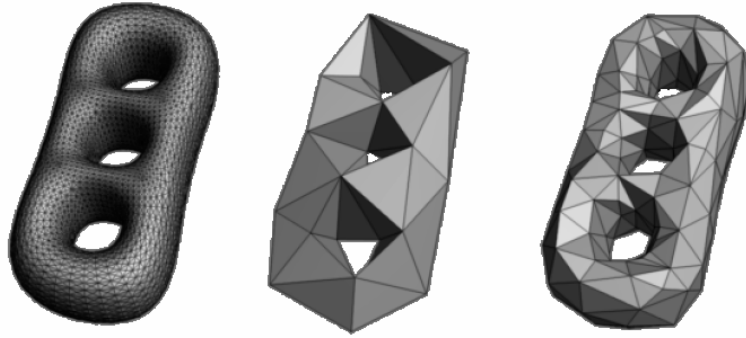


圖 2.4 多精細度取樣法[EDD+95]。

2.2 化簡成本估計

2.2.1 二次方誤差評量



Garland et al.提出以二次方誤差評量(Quadric error metric, QEM)來定義化簡成本 [GH97]，和 Schroeder et al.所提計算幾何的方法類似[SZL92]，是以頂點到平面的距離計為基礎。Schroeder et al.先計算出被剔除頂點周圍的頂點(1-ring neighbor)的平均平面(average plane)，以被剔除頂點到此平均平面的距離平方和做為化簡成本。Garland et al.以邊緣摺疊為化簡方法，與[SZL92]不同之處在於，取代其計算化簡前的頂點到平面的距離，Garland et al.是利用化簡後的頂點到平面的距離，而邊緣摺疊造成的誤差就是摺疊後頂點到所有化簡前三角形的距離，將頂點至三角形的距離取代為頂點到平面的距離，計算頂點與平面的距離計算複雜度比頂點至三角形為小，並導出了可以累加的距離平方和計算方法，使得其二次方誤差評量可以計算原始模型至化簡後模型的總體誤差(global error)，而[SZL92]成本計算方法只計算區域誤差(local error)，因此[GH97]考量了品質和效率提出了二次方誤差評量法。

假如我們用一向量 $p = [a \ b \ c \ d]^T$ 來代表平面 $ax + by + cz + d = 0$ ，其中

(a, b, c) 是單位化(normalized)的平面法向量，而這裡所謂的平面就是網格上的一三角形所處的平面。一個頂點 $v = [v_x \ v_y \ v_z \ 1]^T$ 到一個平面 p 的距離為 $p^T v$ ，而非負的平方距離則為 $(p^T v)^2$ 。一個頂點 u 被合併到頂點 v 所產生的誤差 $\Delta(v)$ 為

$$\Delta(v) = \Delta([v_x \ v_y \ v_z \ 1]^T) = \sum_{p \in \text{planes}(u)} (p^T v)^2 \quad (2.1)$$

其中 $\text{planes}(u)$ 代表頂點 u 的鄰近三角形所形成的平面集合，而 Garland et al. 將式子 2.1 重新改寫為

$$\begin{aligned} \Delta(v) &= \sum_{p \in \text{planes}(u)} (p^T v)^2 = \sum_{p \in \text{planes}(u)} (v^T p)(p^T v) \\ &= \sum_{p \in \text{planes}(u)} v^T (pp^T) v = v^T \left(\sum_{p \in \text{planes}(u)} K_p \right) v \end{aligned} \quad (2.2)$$

其中 K_p 稱為頂點 u 的基本二次方誤差(fundamental error quadric)

$$K_p = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (2.3)$$

對於任一個頂點 u 定義 Q 為二次方誤差(error quadric)， Q 代表在頂點鄰近所有平面的基本二次方誤差之相加總合， $Q = \sum_{p \in \text{planes}(u)} K_p$ 。因此當原始模型要用二次方誤

差評量做為化簡成本時，需在化簡前對於模型上所有的頂點計算其二次方誤差 Q ，所有誤差 Δ 的計算只以單一的矩陣來代表一個平面的集合。當兩個頂點 v_1 、 v_2 要進行邊緣摺疊至 \bar{v} 時，計算 \bar{v} 至平面集合的距離，而實際上平面的集合應該為 $\text{planes}(v_1) \cup \text{planes}(v_2)$ ，但為了簡化，不去除重覆之平面，進行摺疊後 \bar{v} 化簡成本為 $\Delta(\bar{v}) = \bar{v}^T (Q_1 + Q_2) \bar{v}$ ， \bar{v} 的二次方誤差為 $\bar{Q} = Q_1 + Q_2$ 。

二次方誤差的化簡成本計算方法，提供非常有效率的化簡，並且儲存 Q 所使用的記憶體只需 10 個浮點數的空間，因為 Q 是一對稱矩陣。而且以其累加的計算方法，可以計算化簡後模型至原始模型的幾何距離，所以很適合整體誤差的計算。雖然每個 Q 之間會有重覆的平面被計算多次，故會有誤差，但其優勢在於計算速度非常快速。目前的模型不只有幾何空間中的三維資料，而且有其他材質資料如貼圖與法向量，幾何誤差的估計方法沒有考量化簡對這些材質資料的影響，用這些幾何測量做出來的化簡模型往往不能保留模型的外觀。

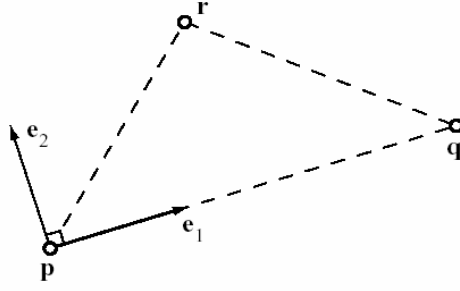


圖 2.5 在五維空間中由 p, q, r 所形成三角形， e_1 和 e_2 是區域性座標軸[GH98]。

Garland et al.改進二次方誤差評量加入貼圖座標的考量[GH98]，將貼圖座標加上三維位置座標共五維，在高維度空間中計算幾何誤差，並找出最佳的頂點使得摺疊後幾何誤差最小。由 p, q, r 三頂點所構成的三角形 T ，頂點 $p=[p_x p_y p_z p_s p_t]^T$ ，另二點也相同，由於在三角形內貼圖座標是線性內插的，所以這三個五維的頂點在 R^5 所形成的三角形也是二維的平面，先計算兩個向量 e_1 和 e_2 為此三角形的區域性座標軸：

$$e_1 = \frac{q-p}{\|q-p\|} \quad e_2 = \frac{r-p - (e_1 \cdot (r-p))e_1}{\|r-p - (e_1 \cdot (r-p))e_1\|} \quad (2.4)$$

任意一個頂點 $v \in R^5$ ，要計算 v 到平面 T 的距離平方 D^2 為化簡成本，令 $u = p - v$ ，而 $\|u\|^2$ 可以分解成 $\|u\|^2 = (u \cdot e_1)^2 + \dots + (u \cdot e_5)^2$ ，可以重寫成以下式子：

$$(u \cdot e_3)^2 + \dots + (u \cdot e_5)^2 = \|u\|^2 - (u \cdot e_1)^2 - (u \cdot e_2)^2 \quad (2.5)$$

式子 2.5 等式左邊是 u 向量中所有垂直平面 T 的距離平方，也就是說 v 到平面 T 的距離平方就為 $D^2 = \|u\|^2 - (u \cdot e_1)^2 - (u \cdot e_2)^2 = u^T u - (u^T e_1)(e_1^T u) - (u^T e_2)(e_2^T u)$ ，因此 D^2 可以寫成以下式子：

$$\begin{aligned} D^2 &= v^T v - 2p^T v + p \cdot p \\ &\quad - v^T (e_1 e_1^T) v + 2(p \cdot e_1) e_1^T v - (p \cdot e_1)^2 \\ &\quad - v^T (e_2 e_2^T) v + 2(p \cdot e_2) e_2^T v - (p \cdot e_2)^2 \end{aligned} \quad (2.6)$$

式子 2.6 是一個二次方誤差(error quadric)形式，可以改寫成 $D^2 = v^T A v + 2b^T v + c$ 其中：

$$\begin{aligned}
A &= I - e_1 e_1^T - e_2 e_2^T \\
b &= (p \cdot e_1) e_1 + (p \cdot e_2) e_2 - p \\
c &= p \cdot p - (p \cdot e_1)^2 - (p \cdot e_2)^2
\end{aligned} \tag{2.7}$$

A 是 5×5 矩陣， b 是 5 維向量，而二次方誤差 $Q=(A,b,c)$ ，而任一頂點到 Q 所代表平面集合的平方距為 $Q(v) = v^T A v + 2b^T v + c$ ，Garland et al. 找到一頂點 \bar{v} 使得 $Q(\bar{v})$ 是最小的值，其中 $\bar{v} = -A^{-1}b$ ，當邊緣摺疊後以頂點 \bar{v} 替代摺疊掉的邊緣可以使得化簡成本最小。[GH98] 成功的將貼圖座標視為幾何距離當做化簡成本，並可以同時計算出新頂點 \bar{v} 的三維座標和貼圖座標。

2.2.2 外觀保持式化簡法

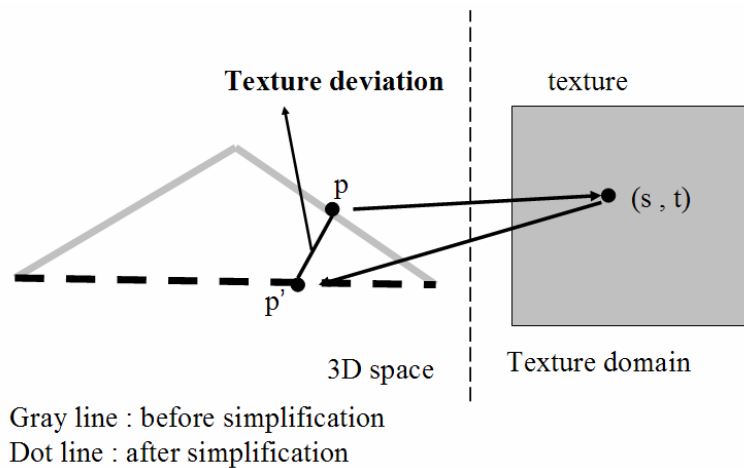


圖 2.6 以二維的例子說明貼圖變量的計算方式。

Cohen et al. 提出的外觀保持式化簡法 (Appearance Preserving Simplification, APS) [COM98]，考慮到貼圖對於外觀的影響，計算貼圖變量 (texture deviation) 當做化簡的成本。基本上這是純幾何的誤差計算，也是頂點對頂點的距離計算，但其關鍵在於如何尋找化簡前後兩種模型上的對應頂點。APS 是延續在 [CMO97] 被提出的方法，兩種方法主要不同之處在於尋找頂點的對應 (mapping) 方式不同。後者利用垂直投影的方式將化簡前後的網格投影至平面之上，在平面上有相同位置的頂點即為其對應。前者取代用投影的方式，改為從貼圖參數化的二維空間中尋找對應頂點，即每一個對應頂點有相同的貼圖座標，也就是說在繪製化簡前後的模型時，對應點繪製在螢幕上會是相同的貼圖顏色。換個方式說明就是測量同一個

貼圖像素(texel)繪製在三維空間中的幾何距離，當然這距離愈短表示化簡對於模型外觀的影響會愈小。

圖 2.6 以簡單二維網格來說明貼圖變量的計算方式。假設兩個網格 M_a 和 M_b ， M_a 代表化簡前的網格， M_b 代表化簡後的網格，而貼圖變量 $TD(p, M_b)$ 定義為

$$M_a \quad TD(p, M_b) = d(p, F_b^{-1}(F_a(p))) \quad (2.8)$$

其中函式 $d()$ 為計算三維空間中兩點的尤拉距離，函式 $F_i(p): M_i \rightarrow T$ 為三維空間的網格 M_i 上的點 p 在二維貼圖空間 T 的對應，可以說是以參數化表示貼圖座標，而 $F_i^{-1}(s, t): T \rightarrow M_i$ 則是相反的對應。圖 2.6 的例子說明了貼圖變量其實就是 p 至 p' 的距離計算，由式子 2.8 可以得到同樣的結果

$$TD(p, M_b) = d(p, F_b^{-1}(F_a(p))) = d(p, F_b^{-1}(s, t)) = d(p, p') \quad (2.9)$$

對每個邊線化簡取其網格上所有的頂點中貼圖變量最大的距離為其化簡成本，亦即 $\max_{p \in M_a} TD(p, M_b)$ 。

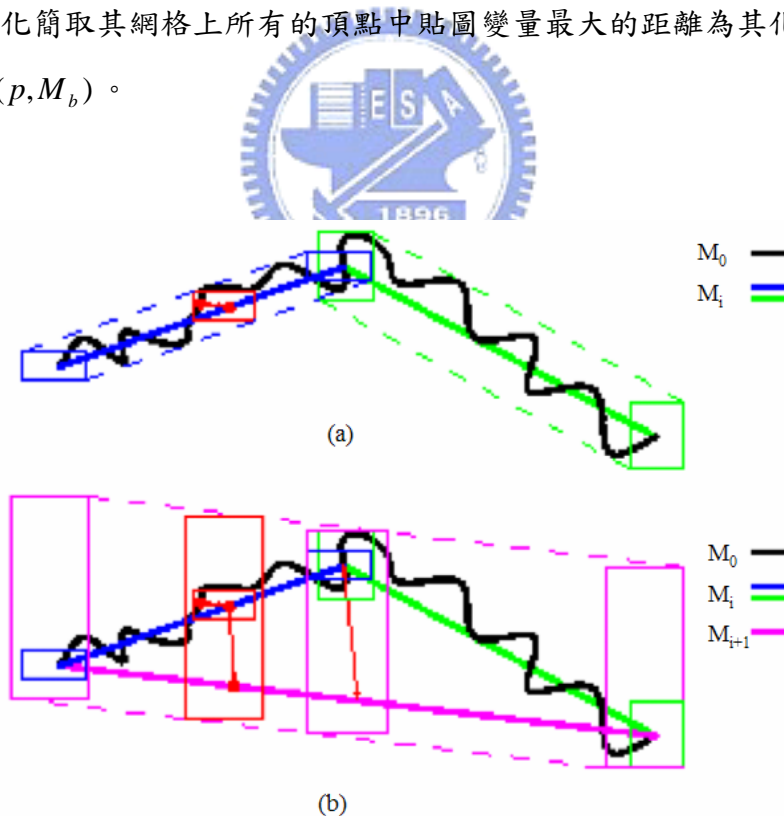


圖 2.7 以方形盒來限制最大的貼圖變量[COM98]，以二維為例子，(a)化簡前各三角形已有至原始模型的貼圖變量，(b)進行化簡後以新的邊界方形來限制到原始模型最大的貼圖變量。

Cohen et al.中不只計算局部貼圖變量，利用方形盒(bounding box)來限制每次化簡的最大貼圖變量，並且在化簡時不斷的擴大方形盒去包圍最大局部貼圖變量與在化簡區域上累積的貼圖變量，來達到整體誤差計算。圖 2.7(a)中黑線代表原始網格 M_0 ，藍和綠代表經過 i 次化簡的網格 M_i 上的三角形，而藍和綠方形代表化簡至 M_i 後各三角形所產生至 M_0 最大的貼圖變量。圖 2.7(b)中粉紅色代表 $i+1$ 次的化簡之後的網格 M_{i+1} ，而粉紅色的方形盒必需包含區域中最大的貼圖變量(即圖中右邊的紅線)以及化簡前模型上的方形盒，以此方形盒的大小做為化簡的成本。

計算貼圖變量並沒有考慮到貼圖本身的內容，只考慮一像素材質在空間中移動之距離，化簡貼圖變量小的邊緣，並且其貼圖變量的計算受到參數化或是貼圖座標指定的影響很大，不好的參數化會使得最後模型在幾何上的誤差會變大，以方形盒來累加局部的誤差，也會造成所計算出的幾何誤差高於實際貼圖變量的距離。

2.2.3 影像導向式化簡法

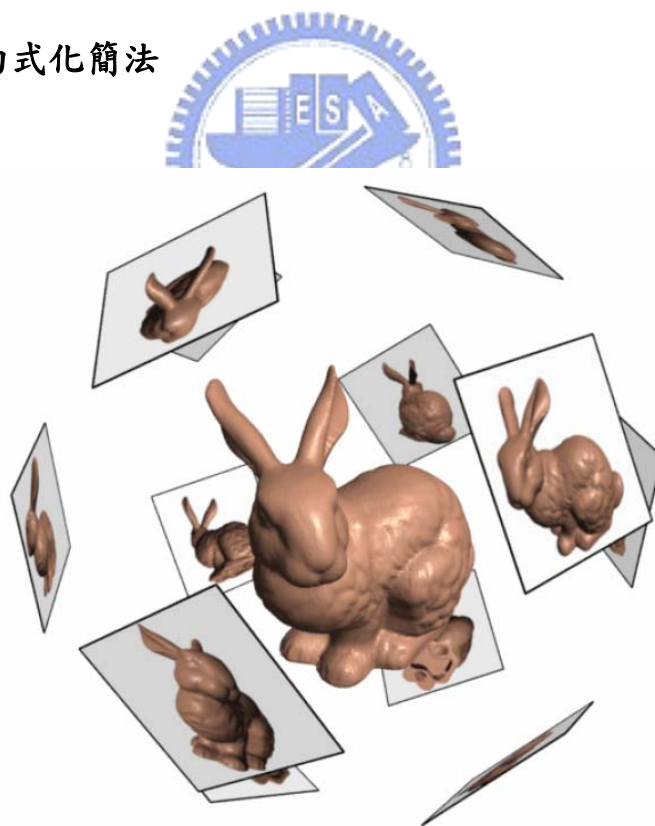


圖 2.8 影像導向式化簡[LT00]，固定在某些視點繪製影像，比較化簡前後每張影像的差距做為化簡成本。

Lindstrom et al.提出以影像導向式化簡法(image-driven simplification, IDS) [LT00]

引導化簡流程的進行。和其他所有的誤差評量法不同，它不在幾何空間(object space)也不在屬性空間(attribute space)來估計誤差，而是直接由對人類視覺最有關係的影像來估計誤差，即化簡成本就是化簡後的影像與化簡前的影像做均方根(root mean square)。直接將化簡誤差訴諸於最後繪製的結果，Lindstrom et al.避免了在結合幾何誤差評量和屬性誤差評量的權重問題，用影像比對完全不用考慮幾何、顏色、貼圖、法向量的扭曲，因為最終這些因化簡被扭曲的地方最後會繪製到螢幕上造成誤差。因此不用和其他的誤差評量一樣，不是沒有考慮模型的屬性，不然就是以幾何計算的方式將屬性當做幾何誤差來計算。影像導向式化簡法的優勢在於保留相當好的模型輪廓，保留貼圖內容在模型上的外觀，並且對於模型的內部區域會劇烈的化簡。但代價是在目前多層精細度的演算法之中，影像導向式化簡法算是速度最慢的化簡。

影像導向式化簡的核心在於如何比較兩張化簡前和化簡後的影像之間的差距。Lindstrom et al.以簡單的像素與像素(pixel-by-pixel)間比對出影像的差距。一般影像都是以RGB三個頻道來儲存，而對於人類視覺系統來說，RGB的差距和人類視覺是不成正比的，因此將RGB轉換成較符合人類視覺的亮度頻道(luminance channel) Y 來比對影像。最終中評量就是以均方根計算兩張影像中每個像素顏色上的差距，兩張亮度的影像 Y^0 和 Y^1 ，皆有 $m \times n$ 個像素，則兩張影像均方根的差距為

$$d_{RMS}(Y^0, Y^1) = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{ij}^0 - y_{ij}^1)^2} \quad (2.10)$$

IDS 以邊線摺疊進行化簡，對於每個可能進行邊線摺疊的地方估計其化簡的成本。如圖 2.8，每個化簡的成本計算都要先從多個視點繪製多張影像，並且將這些影像與原始模型在同樣視點上所繪製的影像計算其均方根差距，而影像差距的總合就是化簡成本。

影像導向式化簡有兩個主要的問題。由於是繪製影像來比對，化簡的速度非常的慢。繪製影像當然以硬體來加速產生影像，並且以漸近式的方法更新影像，只針對邊線摺疊對於影像有影響的地方才要繪製，也就是只繪製被影響的三角形在螢幕空間上的邊界方形(screen-space bounding box)區域即可。為了快速找出邊界方

形中的三角形，對於每個視點都要建立 hash table，儲存每個像素和三維空間中三角形的對應關係。因此和一般的多精細度模型化簡的技術來比較，為了增加速度所以 IDS 以各種軟體資料結構和硬體繪製的方法加速。而另一個問題為品質問題，即要取樣多少視點來繪製影像，以及每張影像的解析度為多少。這兩個關鍵都是速度與品質的取捨，Lindstrom et al.以 20 個取樣視點和 256×256 或 512×512 的影像解析度。由於取樣的視點是固定在模型的周圍，因此模型的凹洞內部區域或視點無法觀察到的區域可能會被大量化簡掉的。



第三章

以射線為基礎測量視覺相似度

在這章中我們介紹如何用射線為基礎去測量模型化簡前後視覺上的相似度，以此相似度來決定一邊線化簡後的成本。3.1 節首先概述一般的邊線摺疊化簡演算法的流程，說明模型化簡方法以及暫緩估計化簡成本的更新機制，3.2 節描述如何透過取樣的視點和射線去測量模型化簡所產生視覺差異，並提出兩種化簡成本的計算方式，即局部(local)以及整體(global)的視覺相似度(visual similarity)之化簡成本來引導化簡的順序。

3.1 化簡演算法概要



許多模型化簡演算法對模型進行一連串的邊線摺疊(edge collapse)或其他的區域性的化簡操作(local simplification operation)，以逐漸減少輸入模型的複雜度。對於整個化簡品質影響最大的是模型化簡的順序。所有的模型化簡演算法會計算每個區域化簡操作的成本，再依照成本由低至高進行其區域化簡操作。這些成本估計方法依其保留模型上所要的特性之多少計算其成本，當化簡造成其上的特性改變較大則估計出來的成本會增加。所有可能進行的區域化簡操作基本上會儲存在一優先權佇列(priority queue)中，並且會持續的更新佇列以反應化簡的結果，並確保下一化簡操作是成本最小的。這種化簡流程的設計用在許多目前已被提出的化簡演算法，包括 [Hoppe96,GH97]，並且常加上暫緩化簡成本(lazy evaluation)[CMO97]之更新機制以加速整個化簡的流程。

3.1.1 化簡演算法

我們的視覺相似度成本估計演算法依照上述的化簡流程，加上視覺相似度之成本

估計的計算，使得邊緣摺疊對模型視覺相似度的影響是最小的。我們選擇使用半邊緣摺疊(half edge collapsing)做為區域化簡操作，對於每一條邊緣 e 可以有兩個方向性的摺疊方向。如圖 3.1 所示，半邊緣摺疊有兩個可能，對每一個可能都要計算其化簡的成本，所以在初始的優先權佇列之中會有模型邊緣數兩倍的節點 (node)，每一個節點代表一個半邊緣摺疊的區域化簡操作。當開始進行化簡時，選擇一個成本最小的節點執行其半邊緣摺疊。每執行一次半邊緣摺疊會消除掉兩個三角形和此三角形所包含的六條半邊緣。摺疊後必需從佇列中去除掉這些節點，並且對於化簡區域周圍的邊緣都要更新其成本，因為進行一次摺疊後，其周圍的邊緣所代表的幾何已經被改變，必需重新計算得到新的、正確的成本。

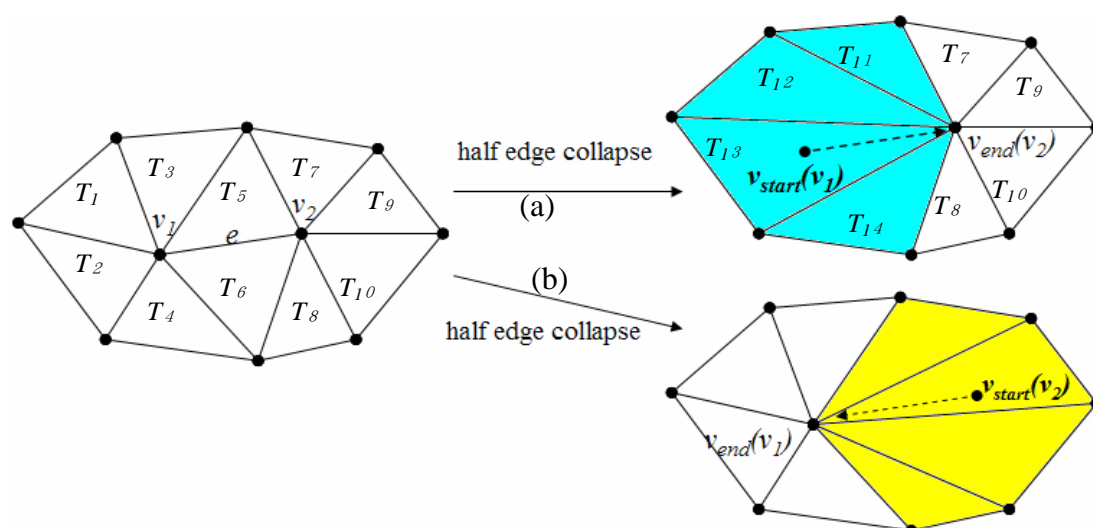


圖 3.1 半邊緣摺疊操作。每條邊緣 e 會有兩個方向的摺疊，箭頭代表摺疊的方向，從 v_{start} 摺疊至 v_{end} ，而顏色代表化簡後所影響網格改變的區域。

使用半邊緣摺疊的優點在於容易維持其資料結構。圖 3.1 中可以觀察到當摺疊後對於化簡區域不用重新進行三角化，直接使用原本的拓樸即可。而且所需要改變的資訊只有包含 v_{start} 的邊緣，將含 v_{start} 資訊改成 v_{end} 。因為而化簡後所有的點都是在原始模型上相同位置的點，這使得在化簡前與化簡後的兩種不同模型上尋找對應區域較容易。而進行半邊緣摺疊時，網格將會被改變的區域即定義為化簡區域，即 v_{start} 的周圍一圈三角形所形成的區域。如圖 3.1 為例，若進行 (a) 的半邊緣摺疊則化簡區域為 T_1 至 T_6 ，進行 (b) 半邊緣化簡則化簡區域則為 T_5 至 T_{10} ，在化簡區域之外的網格不會受到半邊緣摺疊的影響。

一般的化簡演算法依以下步驟來進行：

1. 對所有可能進行區域化簡的半邊線，計算其初始的化簡成本，根據化簡成本放到優先權佇列中。
2. 從佇列中挑選並移除化簡成本最小的半邊線，進行半邊線摺疊，此操作會將佇列中六個半邊線去除。
3. 對於化簡區域周圍所有的半邊線重新計算其化簡成本。
4. 重覆步驟 2，3，4 的步驟，直到化簡至所需要的模型多邊形數目為止。

一般的化簡演算法步驟會花太多計算時間在無用的化簡成本之更正計算上，暫緩估計化簡成本的機制能避免這無用的計算。

3.1.2 暫緩估計化簡成本(lazy evaluation)

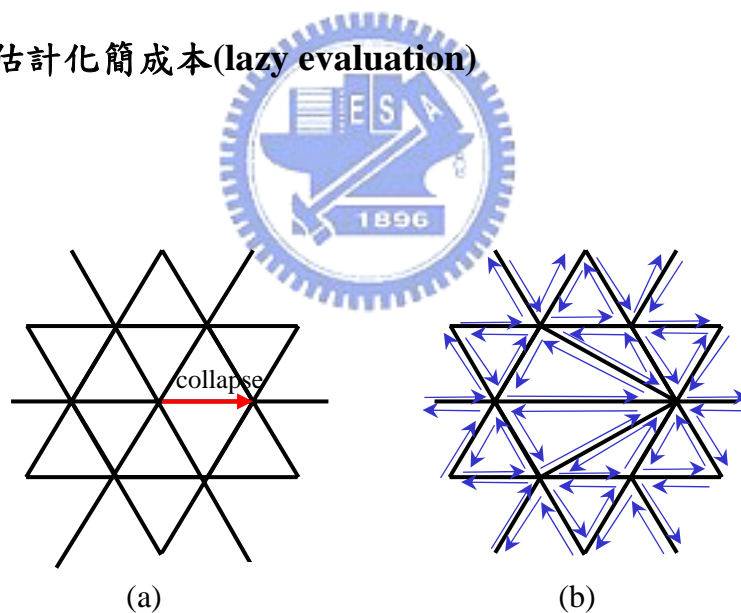



圖 3.2 (a)對紅箭頭方向進行的半邊線摺疊，(b)所有鄰近化簡影響區域的半邊線(藍箭頭表示)都必需重新計算其化簡成本。

在一般化簡流程中，如圖 3.2(a)做一次半邊線摺疊後，必需把圖 3.2(b)中所有藍色的半邊線從優先權佇列中取出並重新計算其化簡成本，若模型的平均維度 (degree) 為 6 時，每進行一次的化簡需重新計算化簡成本 54 次，這樣的計算量相

當的龐大。

進行一次的化簡通常會微量增加周圍邊線的化簡成本，而周圍這些被影響到的邊線由於跟被化簡的邊線是在相同的區域，這些邊線的化簡成本也應在佇列的較前面的部分。當化簡一直進行時，我們會一直重新計算這些成本，而之前計算的成本值還沒被使用到，又要重新更新其成本，造成重覆計算某些半邊線的成本太多次。為了避免重覆計算成本，暫緩估計化簡成本之機制[CMO97]將這些被化簡後影響的半邊線(圖 3.2(b)中藍色邊線)標記為 dirty 且不計算其新的化簡成本。dirty 表示其化簡成本不是最精確的數值，相對於 dirty 的標記則為 clean，表示其化簡成本是精確的數值。標記為 dirty 的半邊線不重新計算其成本，直到這些 dirty 的半邊線被排序至佇列最前面時再去計算其實際的成本，將其標記 clean 並在佇列中重新排序。

加上上述的暫緩估計化簡成本的機制，我們所使用的化簡演算法為：

- 
1. 對所有可能進行區域化簡的半邊線，計算其初始的化簡成本，並設定每個半邊線為 clean，並將其優先權佇列中做排序。
 2. 從佇列中挑選並移除化簡成本最小的半邊線，若此半邊線為 dirty，則重新計算半邊線化簡成本並設成 clean，並在佇列中重新排序，重覆步驟 2 直到選到 clean 的半邊線。
 3. 將步驟 2 選出來的半邊線進行半邊線摺疊，此操作會將佇列中六個半邊線去除。
 4. 對於化簡區域周圍所有的半邊線設定為 dirty，表示其化簡成本非最精確的數值。
 5. 重覆步驟 2-5，直到化簡至所需要的模型多邊形數目為止。

3.2 化簡成本計算

我們提出以透過射線來量測模型化簡後的視覺相似性。與影像導向式化簡法[LT00]不同的是觀察模型變化所取樣的視點位置，影像導向式化簡法在模型包圍

圓(bounding sphere)半徑兩倍的圓之上取樣視點，並且從每個視點往圓心方向繪製影像，對於原始模型和化簡後的模型各繪製一張，計算出兩張影像的差距。因為影像導向式的取樣視點是在整個模型的外圍，在繪製影像時對於凹的區域或模型的內部區域可能繪製不到或是只能繪製到一點點，在這些區域對於最後的影像僅只有小部分或跟本沒有影響，由於對影像貢獻很小因此造成這些地方化簡成本相當低，凹的區域可能因此被填平。這些區域可能只對某些角度的視點來說是可見的，但因取樣視點關係，讓此區域完全被化簡掉。對於較小的幾何特徵，影像導向式化簡法將無法保留，因為品質完全取決於影像之解析度，當其所投影至影像的像素較少時，則很容易就被化簡掉，影響物體的外觀。而影像導向式化簡法最大的缺點在於化簡速度很慢，因此必需在軟體技術和硬體技術中來加快其化簡速度。

由於每個化簡的區域操作都只是區域性的，影響的範圍也僅只於區域性而已。我們的視覺相似度成本計算只觀察化簡的區域，如圖 3.1 中右邊化簡後影響的區域為有上顏色的區域即為因化簡被影響的區域，在這區域之外的網格並不會改變。我們取樣視點僅在化簡區域的周圍，只觀察化簡前後區域性的變化。與影像導向式化簡法相比，我們對於每一個化簡區域操作給予更平均的化簡機會。我們是透過射線測量模型外觀的顏色變化，一個邊緣化簡若造成貼圖內容在模型上變化過大，則取樣射線在化簡前後網格上所得到的交點上的顏色差異會較大，因此挑顏色差異小的區域操作來化簡網格即可避免化簡造成貼圖內容在網格上變形。

我們的視覺相似度化簡成本的計算，就是在化簡區域周圍平均地取樣一些固定數目的視點，並由這些視點觀察化簡前後網格在各種角度的變化。從視點往化簡區域上射出一些射線，射線跟化簡前與化簡後的網格各有一個交點，交點之顏色即代表著網格最後繪製在螢幕上所呈現的顏色，兩個交點的顏色差異愈大表示化簡後在螢幕上所呈現出來差異愈大。面積愈大的化簡區域必需取樣多一點的射線才可觀察出化簡的影響，射線取樣過少可能無法完全的反應出網格的變化，因此射線數目要和面積成正比。一般而言不論面積大小，較平坦區域或貼圖內容無變化區域所得到的平均顏色差異會較小，曲度較大區域和貼圖內容變化大之區域所得到的平均顏色會較大，期望是挑選前者先進行化簡。若視覺相似度以平均各射線顏色差異值做為化簡成本，會造成保留太多的小三角形在曲度大和貼圖變化大的

地方，就算化簡至很低的三角形數目這些小三角形會一直保留著，造成最後的模型化簡不平均。我們選擇使用累加(sum)所有射線的颜色差異做為化簡成本。使用累加會導致面積小但曲度大或貼圖變化大的化簡區域被化簡掉，因面積小所取樣的射線數目會較小，累加起來數值可能會小於面積大但平坦區域(面積大會取樣較多射線)。但面積小對於視覺上的感觀或最後投影在螢幕上的像素數目都是影響比較小，所以就算化簡掉對於最後所呈現出來的結果是微小的差異。累加可以確保化簡後模型上的三角形較平均的分佈，但在面積相似的化簡區域中又可以挑選較平坦或貼圖變化較小的區域先進行化簡。

以下小節將對於取樣視點、取樣射線、定義化簡成本做更詳細的說明。3.2.1 節介紹視點取樣的方法，3.2.2 節定義射線取樣的方法，3.3.3 節和 3.3.4 節定義兩種化簡成本，分別為局部的化簡成本以及整體化簡成本。

3.2.1 取樣視點

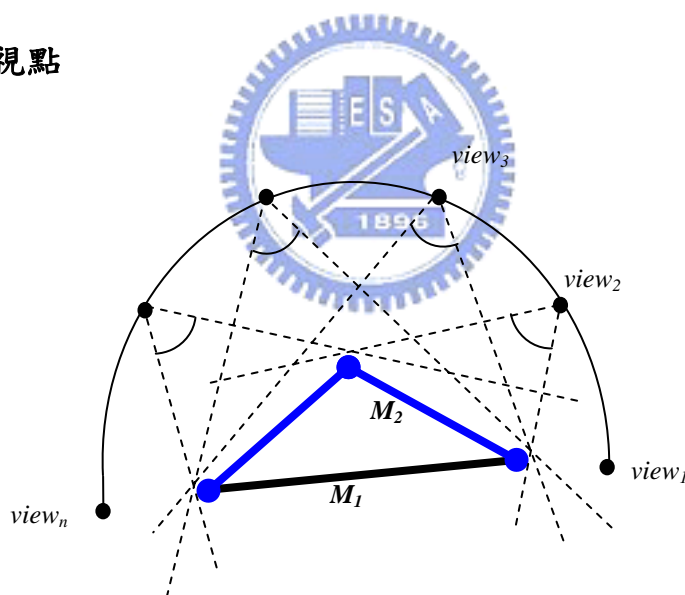


圖 3.3 以二維為例，測量視覺相似度。在化簡區域的周圍平均取樣一些視點。

在測量視覺相似度時不太可能對於所有角度的視點、各種不同顏色不同亮度的光源情況之下都抓得到其在視覺上的差異，因此我們在比較模型化簡後的視覺差異時必需有一些假設。在模型化簡時，因都是區域性的化簡操作，所以不論視覺上和幾何上影響的區域都是模型上的小部分，會產生差異的地方就是區域化簡操作的地方，我們測量視覺差異性只需要對於化簡會影響到的區域進行量測。對每個

化簡區域，我們在它的法向量方向上的半圓上取樣視點，為了使得這些視點可以完全觀察到各個角度上網格的變化，因此半圓的圓心訂為化簡區域內頂點之平均中心，且半徑則為此平均中心至最遠頂點距離之二倍，在半圓上的取樣視點只要是平均地(uniform)取樣即可。如圖 3.3 以二維為例所示， M_1 和 M_2 代表化簡前後兩個不同的網格，而 $view_1 \sim view_n$ 則為我們取樣的視點。

3.2.2 取樣射線

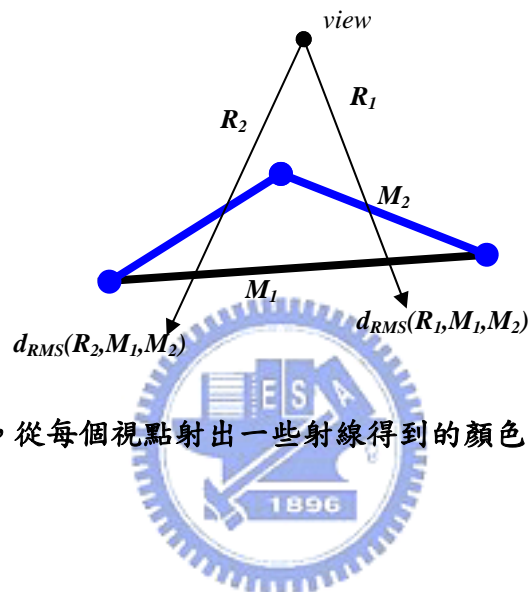


圖 3.4 以二維為例，從每個視點射出一些射線得到的顏色差異 d_{RMS} ，做為視覺相似度的誤差。

當我們取樣好視點之後，接下來就由這些視點觀察化簡後網格的變化，必需從視點往化簡前網格上的三角形射出射線，得到射線在化簡前與後網格交點上顏色的誤差值。由每個取樣的視點對化簡區域上之網格平均射出一些射線(如圖 3.4)，每一條射線與化簡前後兩個網格的交點可以得到兩個顏色，若化簡造成貼圖或外觀上有很大的改變的話，所有射線上的兩個交點顏色的平均差距會很大，因此我們利用這些由視點射出的射線上交點的顏色差做為視覺相似度的依據。

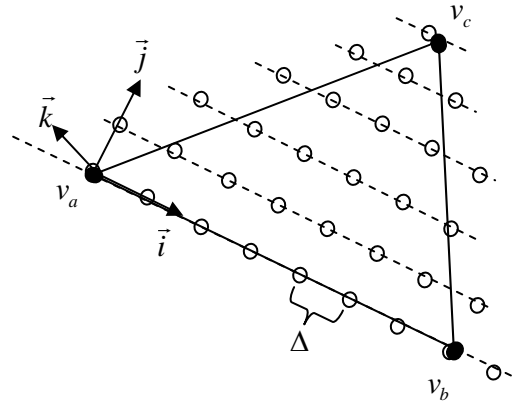


圖 3.5 掃描線演算法，落在三角形之中的點即為取樣點， Δ 為取樣的單位長度。

當模型化簡後，三角形的面積會愈來愈大，將來繪製到螢幕上會佔較多的像素，佔的像素多影響人類視覺就較大，因此當化簡區域的面積一直在成長時，我們取樣射線的數目也必需成正比跟著增加，也可以避免取樣不足導致抓不到貼圖顏色的變化。要測量其化簡之成本，在化簡區域上的每個三角形上依以下方式細分(subdivision)一些取樣點，如圖 3.5，我們使用類似二維掃描線(scan line)的演算法，從視點對每個化簡區域中三角形上的取樣點各射出一條射線。

取樣的方式為給予一個三角形 $T=\{v_a, v_b, v_c\}$ ，找出區域性座標軸(local reference coordinate) $\{\vec{i}, \vec{j}, \vec{k}\}$ 三個向量，其中：

$$\vec{i} = \frac{\overrightarrow{v_a v_b}}{\left\| \overrightarrow{v_a v_b} \right\|}, \quad \vec{k} = \frac{\overrightarrow{v_a v_b} \times \overrightarrow{v_a v_c}}{\left\| \overrightarrow{v_a v_b} \times \overrightarrow{v_a v_c} \right\|}, \quad \vec{j} = \vec{k} \times \vec{i}$$

(\vec{i}, \vec{j}) 平面和 T 是共面， \vec{k} 是三角形的法向量。掃描線演算法依照兩個方向 \vec{i} 和 \vec{j} 在三角形上取樣點，每條所產生的掃描線是在 (\vec{i}, \vec{j}) 平面上並且平行於 \vec{i} ，每個取樣點間的距離 Δ 決定取樣的密度。其中 Δ 為：

$$\Delta = \sqrt{2 \times A_{avg}}$$

其中 A_{avg} 為原始模型上所有三角形平均面積，使得初始模型上每個三角形平均至少可以被取樣到三個點。這個方式可以平均地在三角形上產生取樣點，並且取樣點數目與三角形的面積成正比例。

3.2.3 局部視覺相似度化簡成本

有了取樣視點和取樣射線的方法後，我們開始介紹如何定義視覺相似度化簡成本，先介紹局部誤差，即單一化簡步驟前後的視覺相似度。

計算一條射線與模型網格交點後可以同時得到交點上的法向量和貼圖座標，這兩個資訊是模型在一般繪圖流程中計算畫到螢幕上的顏色所需要的資訊。我們視覺相似度的成本計算就是依法向量和貼圖座標所得到的顏色做為誤差的計算，其中法向量會影響光影的亮暗，貼圖座標可以查出射線與模型的交點上貼圖內容的顏色。對於一般的化簡成本的計算而言，我們無法對於模型在各種不同光源顏色或位置之下計算其化簡成本，因化簡時並不會知道模型會在什麼情況之下被繪製出來，所以在進行化簡成本計算時使用單一白色光源，並將光源放在取樣視點的相同位置，使得由視點往化簡區域觀察時所看到的網格是亮的。

給予一個多邊形網格 M ， M 包含網格所有的點、法向量、貼圖座標、拓樸，定義 R 為一條射線，包含起始點(即為視點)和射線的方向，我們用 $Cost(R, M)$ 代表任一條射線 R 和網格 M 的交點所呈現出來的顏色，顏色的計算方法為：

$$Cost(R, M) = (C_{texture}) \times (I \cdot N) \quad (3.1)$$

其中 $C_{texture}$ 為交點上的貼圖顏色， N 則為交點上的單位法向量， I 則是光源的向量，為交點往射線起始點的單位向量。

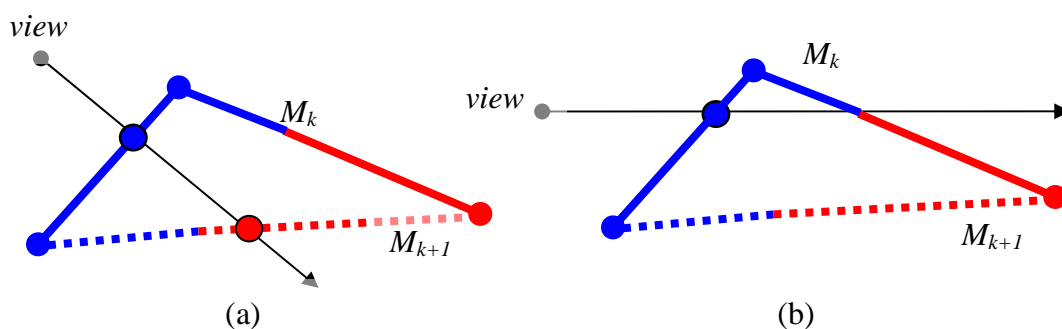


圖 3.6 以 2D 為例。(a) 顏色代表貼圖的內容，當化簡區域是在貼圖變化的地方，射線的顏色差異 d_{RMS} 會較大 (b) 當化簡區域為輪廓邊時，會有沒有交點的射線，

必需對這些射線給予適當的處罰值。

此成本計算的方法對於維持模型上的貼圖內容和幾何誤差有一定的效果。圖 3.6(a)中模型上的顏色代表貼圖的顏色，圖中實線與虛線各代表 M_k 與 M_{k+1} ，此圖說明了當化簡的區域是貼圖變化的區域時，會有許多射線是圖 3.6(a)的情況，由於射線和兩個模型的交點顏色差異 d_{RMS} 大，因此這個區域若進行半邊線摺疊會造成視覺上很大的差異，因此這個區域的化簡順序會比沒有貼圖顏色變化的區域的順序更晚被化簡到。由於取樣視點時是平均的取樣，以模擬各種角度來觀察化簡區域，當化簡的區域為幾何差異較大時，很容易出現圖 3.6(b)的情況，當這種情況出現時射線和網格並無交點，因此式子 3.1 必需改為：

$$Cost(R, M) = \begin{cases} 0.5 & \text{若射線} R \text{與網格} M \text{沒有交點} \\ C_{texture} \times (l \cdot N) & \text{其他} \end{cases} \quad (3.2)$$

當 R 與 M 無交點，得不到顏色，因此我們給予一個處罰的值 0.5 代表灰色，為最強亮度的一半，由這個處罰值來維持模型的幾何誤差。

由圖 3.3 很容易可以觀察出來，由 $view_1$ 和 $view_n$ 所射出的射線很容易與化簡後網格 M_1 沒有交點，而較靠近半圓頂部的 $view_3$ 較不容易，當幾何誤差很大時才會使得 $view_3$ 產生交點之射線。由這個觀察，我們可以將 $Cost$ 函式中沒交點的處罰值依照不同視點給予不同的值，當視點愈靠近半圓頂部給予其處罰值大於視點在半圓的底部，使得幾何變化愈大，處罰值也會加大，以期使得化簡後模型的幾何誤差更小。

要比較模型化簡前後的差異，從視點往化簡的區域射出射線，分別對模型在化簡之前與化簡之後兩個不同的模型之上會有兩個交點。而我們的成本估計利用此兩個交點所得到的顏色差異做為視覺相似度的測量標準。因此對於一條射線 R ，由原始模型的網格 M_0 經過 k 次化簡的網格 M_k ，以及要在 M_k 上進行半邊線化簡後的網格 M_{k+1} ，定義其非負的視覺相似度差異 d_{RMS} 為：

$$d_{RMS}(R, M_k, M_{k+1}) = \sqrt{(Cost(R, M_k) - Cost(R, M_{k+1}))^2} \quad (3.3)$$

此公式使用均方根值(root mean square)來計算兩個顏色的差異得到非負的誤差。一般在顏色的計算和顯示上都是使用 RGB(紅綠藍)三個頻道，每個頻道各有 256

階的明暗變化來顯示螢幕上的各種顏色，但在 RGB 上的顏色變化和人類的視覺並不成線性的變化，因此測量視覺相似度時必需採用其數值變化符合人類視覺的表示方法，我們採用亮度(luminance)單一頻道來代替 RGB 三頻的表示法，所以 $C(R,M)$ 得到的 RGB 表示法依一般的 RGB 轉換成亮度係數計算出其亮度，範圍為 0~1。

從每個視點往化簡區域射出足夠的射線，而每條射線和化簡前後的網格可能會有顏色的差異 d_{RMS} ，將所有射出的射線所得到的 d_{RMS} 累加後做為我們化簡的成本。當顏色的差異總和愈大表示，此化簡對於模型外觀視覺上的影響愈大。

對於一 M_k 上可進行半邊線摺疊的半邊線 he ，以及經過 he 化簡後的模型 M_{k+1} ，可以定義 M_k 和 M_{k+1} 兩個網格間局部視覺相似度 $Local_Cost$ 為

$$Local_Cost(he, k) = \sum_{i=0}^v \sum_{j=0}^r d_{RMS}(R_{ij}, M_k, M_{k+1}) \quad (3.4)$$

由於是半邊線摺疊，化簡區域僅限於 M_k 中 he 的 v_{start} 周圍的三角形，以 $T_k \in M_k$ 代表這些周圍的三角形集合。射線 R_{ij} 的起始點(取樣視點)是 T_k 上所有頂點平均位置為中心的半圓之上的第 i 個視點，而取樣視點總共為 v 個。 R_{ij} 的終點則是以掃描線演算法在 T_k 所得到所有取樣點中的第 j 個， r 為三角形集合 T_k 掃描後所有取樣點的總數。 $Local_Cost$ 即為所有射線所得誤差的和。

綜合以上，當一個半邊線 he 需要計算其局部視覺相似度化簡成本時，用以下的虛擬碼來描述所要計算的步驟：

```

Local_Cost( half edge he , Current level mesh  $M_k$  )
  Local_Cost = 0.0
   $M_{k+1}$  = mesh  $M_k$  after collapsing he
   $V$  = Sample_Viewpoint_around_simplify_region(he)
  For each  $v \in V$ 
  do
     $T_k$  = Triangles around he on  $M_k$ 
    For each  $t \in T_k$ 
    do
       $R$  = Sample_Ray( $v$ ,  $t$ )
      For each  $r \in R$ 
      Do
         $T_{k+1}$  = Triangles on  $M_{k+1}$  changed by collapsing he on  $M_k$ 

```

```

Local_Cost += d_{RMS}(r, T_k, T_{k+1})
End
End
End

```

由於射線與網格的交點測試的計算時間是跟網格上的三角形數目成正比，所以在實做時射線只與化簡區域上的三角形做測試即可以找到交點，在上述虛擬碼中以 T_k 和 T_{k+1} 代表在 M_k 與 M_{k+1} 和上的化簡區域上的三角形，以圖 3.1(a) 的半邊線化簡為例， T_k 代表 $\{T_1, T_2, T_3, T_4, T_5, T_6\}$ 三角形結合，而 T_{k+1} 則代表 $\{T_{11}, T_{12}, T_{13}, T_{14}\}$ 三角形結合。

3.2.4 整體視覺相似度化簡成本

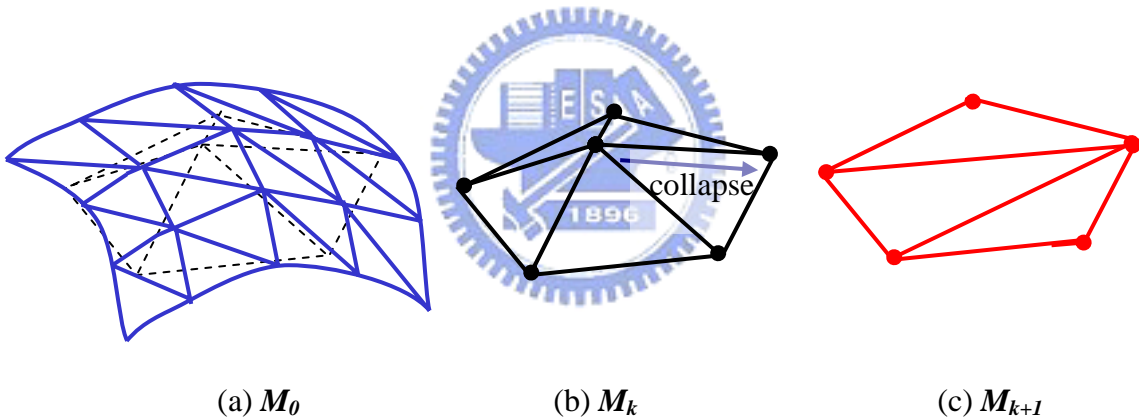


圖 3.7 (a) 模型 M_0 包含化簡區域的三角形，(b) 進行化簡的區域，(c) 進行化簡後的區域。

在 3.2.3 節中介紹的視覺相似度化簡成本只針對化簡前後的兩個網格做比較，就是圖 3.7(b) 與 (c) 兩個網格來做成本計算，這是屬於局部的誤差 (local error) 計算方法。而 M_k 也是由 M_0 化簡而來的， M_0 為原始網格，所以更精確來說每個化簡都是要由原始模型網格 M_0 (圖 3.7(a) 藍色網格) 和化簡後網格 M_{k+1} (圖 3.7(c) 紅色網格) 來計算化簡成本，因此整體性視覺相似度化簡成本計算可以寫為：

$$Global_Cost(he, k) = \sum_{i=0}^v \sum_{j=0}^r d_{RMS}(R_{ij}, M_0, M_{k+1}) \quad (3.5)$$

相較於式子 3.4，這式子是整體的誤差(global error)計算方法，而其取樣射線和取樣視點的方法都一樣，

實作上，取樣的射線不可能與整個模型網格做相交測試，必需找一組三角形集合 T_0 和 T_{k+1} 代表網格 M_0 和 M_{k+1} 可能會與射線相交的三角形。 T_{k+1} 在化簡後的網格上很容易找到。但很難直接得到在 M_0 上的 T_0 。在許多視點相依多重精細度模型 (view-dependent level-of-detail modeling) 的化簡演算法之中如 [XV96, Hoppe97, LE97]，可以利用其化簡時建立的頂點階層 (vertex hierarchy) 來找到在原始模型上對應的區域。透過貼圖的對應我們可以透過 T_{k+1} 在貼圖空間中所涵蓋區域，在 M_0 上找到涵蓋相同區域的三角形集合。定義函式 $\Pi_i(T)$ ：

$$\Pi_i(T): M_i \rightarrow \text{Region in texture domain} \quad (3.6)$$

將在網格 M_i 上三角形集合 T 對應到貼圖空間中的區域，而函式 Π_i^{-1} 是將貼圖空間

中的區域對應回 M_i 中的三角形集合。因此可以定義 $T_0 = \Pi_0^{-1}(\Pi_{k+1}(T_{k+1}))$ ，並以 T_0 和 T_{k+1} 和來做視覺相似度化簡成本計算。半邊線 he 需要計算其整體性化簡成本時，用以下的虛擬碼來描述所要計算的步驟：

```

Global_Cost( half edge  $he$  , Current level mesh  $M_k$  )
  Global_Cost = 0.0
   $M_{k+1}$  = mesh  $M_k$  after collapsing  $he$ 
   $V$  = Sample_Viewpoint_around_simplify_region( $he$ )
  For each  $v \in V$ 
  do
     $T_k$  = Triangles around  $he$  on  $M_k$ 
    For each  $t \in T_k$ 
    do
       $R$  = Sample_Ray( $v, t$ )
      For each  $r \in R$ 
      Do
         $T_{k+1}$  = Triangles on  $M_{k+1}$  changed by collapsing  $he$  on  $M_k$ 
         $T_0 = \Pi_0^{-1}(\Pi_{k+1}(T_{k+1}))$ 
        Global_Cost +=  $d_{RMS}(r, T_0, T_{k+1})$ 
      End
    End
  End
End
End
  
```

在尋找 T_0 時必需要有加速機制，而這機制適用於半邊線摺疊和全邊線摺疊。將每個三角形 t 聯結一三角形集合 S_t ，代表 $S_t = \Pi_0^{-1}(\Pi_{k+1}(t))$ ， S_t 中的三角形皆屬於原始模型。我們可以使用漸近式的方法在化簡邊線時去更新 S_t ，使得在 S_t 中的三角形在貼圖域所含蓋的區域包含 t 在貼圖域中的區域。在還未化簡的網格 M_0 上的每個三角形 t 所聯結的 S_t 中只存本身三角形， $S_t = \{t\}$ ，每經過一次邊線摺疊之後，經摺疊而改變過的三角形 $t_{k+1} \in M_{k+1}$ ，假設在 M_k 中有 $t_1 \sim t_n$ 三角形在貼圖域中包含 t_{k+1} 在貼圖域的範圍，即 $\{t_1, \dots, t_n\} = \Pi_k^{-1}(\Pi_{k+1}(t_{k+1}))$ ，我們更新 t_{k+1} 所聯結的三角形集合 $S_{t_{k+1}} = S_{t_1} \cup S_{t_2} \cup S_{t_3} \cup \dots \cup S_{t_n}$ ，用這種漸近的更新方法使得可以很容易的得到 T_0 。

圖 3.8 是 M_k 化簡至 M_{k+1} 過程，而 M_{k+1} 上灰色三角形為其 S 必需更新為包含 $S_c \cup S_i \cup S_h$ ，因黃色三角形在貼圖域中是被三角形 c, h, i 所涵蓋。

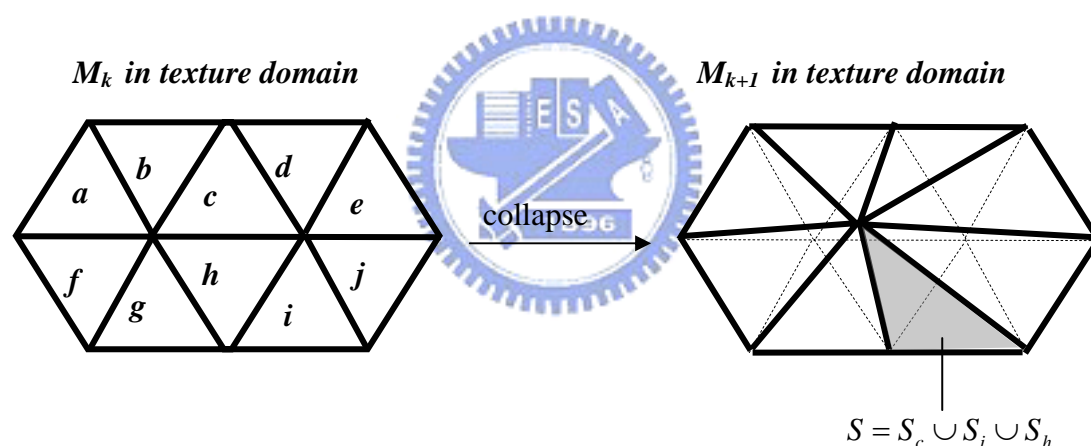


圖 3.8 邊線化簡前後在貼圖域的變化。

3.2.5 視覺相似度化簡成本結合全邊線摺疊之作法

全邊線摺疊(fully edge collapse)是當摺疊一邊線時，邊線的兩端點會合併到一新的頂點之上，這新的頂點可以在任意的位置甚至不在原本的網格之上，和半邊線摺疊不同之處在於化簡所影響的區域擴大了。我們選擇[GH98]提出之二次方誤差評量找出全邊線摺疊後的新頂點，詳細的做法在 2.1 節後面已介紹，此方法可以直接算出新頂點 \bar{v} 之三維座標及貼圖座座標。本節以局部視覺相似度化簡成本

為例說明半邊線與全邊線摺疊不同之處，整體視覺相似度化簡成本是類似的，不再多做說明。

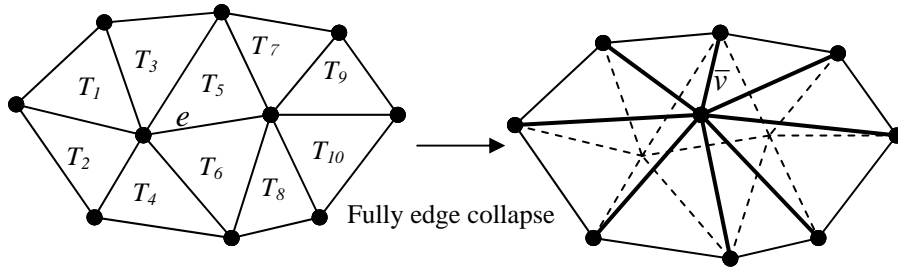


圖 3.9 全邊線摺疊。

圖 3.9 是邊線 e 的全邊線摺疊示意圖，和半邊線摺疊不同之處在於在摺疊時全邊線摺疊的化簡區域較大，只要在邊線兩端頂點周圍的三角形都會改變其網格。局部視覺相似度化簡成成本重寫改寫為

$$Local_Cost(e, k) = \sum_{i=0}^v \sum_{j=0}^r d_{RMS}(R_{ij}, M_k, M_{k+1}) \quad (3.7)$$

由於化簡區域改變，因此 T_k 則改變為邊線 e 兩端點的周圍三角形集合，圖 3.7 為例就是 $T_k = \{T_1, \dots, T_{10}\}$ ，以 T_k 之頂點平均中心為圓心之半圓上平均取樣視點，並且在 T_k 上所有三角形都要進行掃描線演算法去取樣點，由視點到所有三角形上取樣點形成的射線收集在 R_{ij} 中，而 T_{k+1} 則為 M_{k+1} 中 \bar{v} 的周圍三角形集合，而其他則與半邊線摺疊無異。

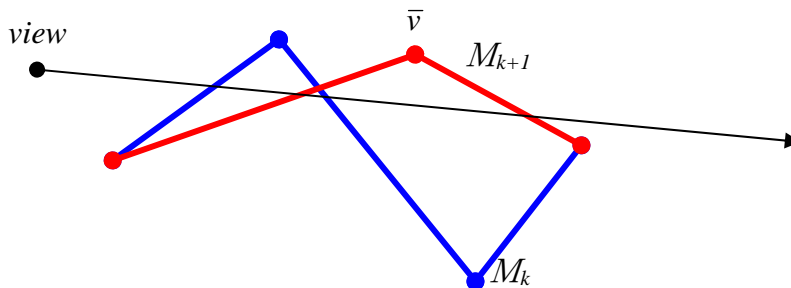


圖 3.10 使用全邊線摺疊導致射線不常用只相交於一個網格。

這樣的結合是以幾何計算方法找出新頂點，並以視覺相似度化簡成本做為化簡的引導，因此和半邊線摺疊相比，全邊線摺疊可以得到較好的幾合誤差。我們所提

之成本化簡是以處罰值來維持幾何的特徵，但使用全邊線化簡會使得射線與化簡後的網格很容易有交點，如圖 3.10，因此幾何就算變化很大，因為新頂點位置的關係導致無法將處罰值反應在成本之上，較無法維持細部的幾何特徵，而會被新計算出的頂點漸漸 smooth 掉了特徵的地方。



第四章

結果與討論

本論文的系統實作平台是配備 AMD Athlon XP 1.5GHz 中央處理器、512MB RAM、GeForce FX5200 顯示卡的個人電腦。4.1 節以本論文提出的視覺相似度化簡成本計算和二次方誤差評量、外觀保持式化簡法、影像導向式化簡法對於有貼圖的模型進行化簡，展示本論文所提的方法確實可以保持模型上貼圖的可觀性。而外觀保持式化簡法和影像導向式化簡法雖然對於貼圖也可維持其可觀性，這兩個方法都會導致幾何上的某些缺點。在 4.1.1 節和 4.1.2 節中分別討論這兩種方法與本論文方法相比無法做到的地方。4.2 節中將討論視覺相似度化簡成本使用全邊線摺疊與半邊線摺疊的差異。

4.1 以視覺相似度化簡成本計算之效果

具有貼圖的模型在化簡時必需考量到貼圖的內容，才會使得化簡後的模型其外觀與化簡前是相似的。在這一節裡將以本論文的局部視覺相似度化簡成本、整體視覺相似度化簡成本與相關研究中之已提出之方法做比較，這些方法分別為只考慮幾何化簡成本的二次方誤差評量[GH97]，將貼圖變量視為幾何誤差的外觀保持式化簡法[COM98]，以最後投影在螢幕上影像間的差異做為化簡成本的影像導向式化簡法[LT00]，而化簡的模型為有條紋貼圖的兔子(bunny)和豬(pig)。

圖 4.1 為豬的模型分別以局部視覺相似度化簡成本和整體視覺相似度化簡成本兩種不同成本計算方法化簡模型，展示出不論模型如何化簡，對於貼圖的維持是相當不錯的。透過射線所得到的顏色差異，使得化簡集中在貼圖內容變化小的區域，保留多一點的頂點在貼圖內容變化大的區域。而當射線射到曲度較大的網格時，會有許多射線不會與化簡後的網格相交，並無法得到顏色，給予 0.5 的處罰

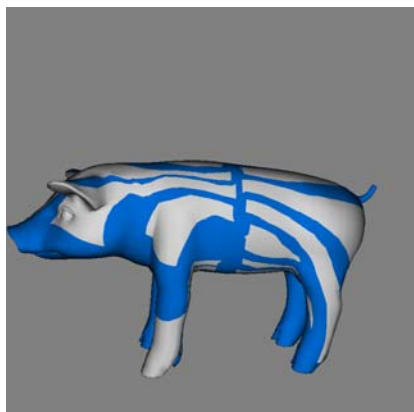
值使得視覺相似度對於幾何也有一定的維持效果，不會因為只維持貼圖的可觀性而使得幾何變的較差，尤其對於較尖銳的網格，視覺相似度化簡成本會比某些計算幾何誤差的化簡成本保留的更好。

圖 4.1 中(b)(c)分別代表以局部和整體視覺相似度成本計算。在外觀上而言，兩者的差異並不會很大，對於經過k次化簡的模型 M_k ，局部以觀察 M_k 與 M_{k+1} 的變化，整體則是觀察 M_{k+1} 及 M_0 間的變化。整體視覺相似度對於模型的幾何誤差的保留較局部成本計算來得佳。整體成本計算是觀察較精確完整的網格，而局部成本計算所觀察只是多邊形數較少的 M_k ，而 M_k 上的多邊形內部也是以線性內插出其顏色，而內插出的顏色不能精確的代表原始模型上網格幾何變化。因此整體成本計算對幾何誤差保留的較好，表 4.1 以Metro[CRS98]測量兩個方法化簡後的豬模型與原始模型的Hausdroff距離，大部分情況之下整體性視覺相似度成本計算得到較佳的幾何誤差。

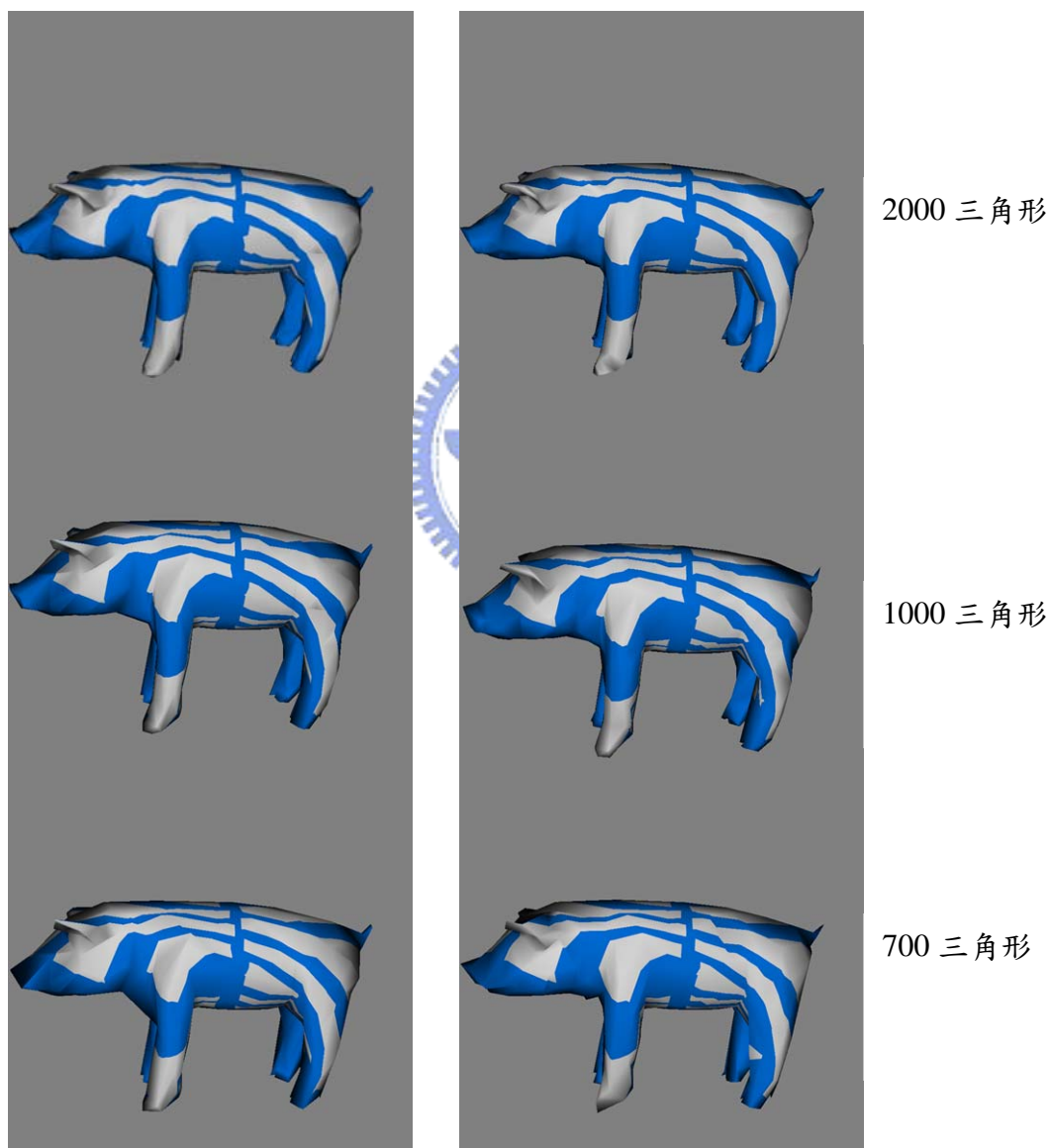
三角形數目	2000		1000		700		400	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
局部	0.071	1.012	0.142	1.190	0.180	1.334	0.278	2.134
整體	0.072	0.908	0.126	1.135	0.162	1.170	0.273	2.21

表 4.1 以豬模型比較局部和整體幾何誤差保留能力。

圖 4.2 中比較一些化簡成本計算，圖 4.2(a)所使用的二次方誤差評量只有考慮到幾何距離誤差，並不適用在含有貼圖座標的模型，因此最後的結果導致貼圖是完全的變形。圖 4.2(c)(d)分別是影像導向式化簡法和外觀保持式化簡法，兩種方法都對於含貼圖座標的模型都可以維持其外觀，所以在圖中基本上看不出太大的差別，然而在細部的幾何特徵上，這兩個方法皆有其缺點，將在下面章節做討論。



(a)原始模型 7164 三角形



(b)局部視覺相似度化簡成本 (c) 整體視覺相似度化簡成本

圖 4.1 (a)豬原始模型。使用局部(b)和整體(c)視覺相似度化簡成本，由上而下分別為化簡至 2000,1000,700 個三角形。

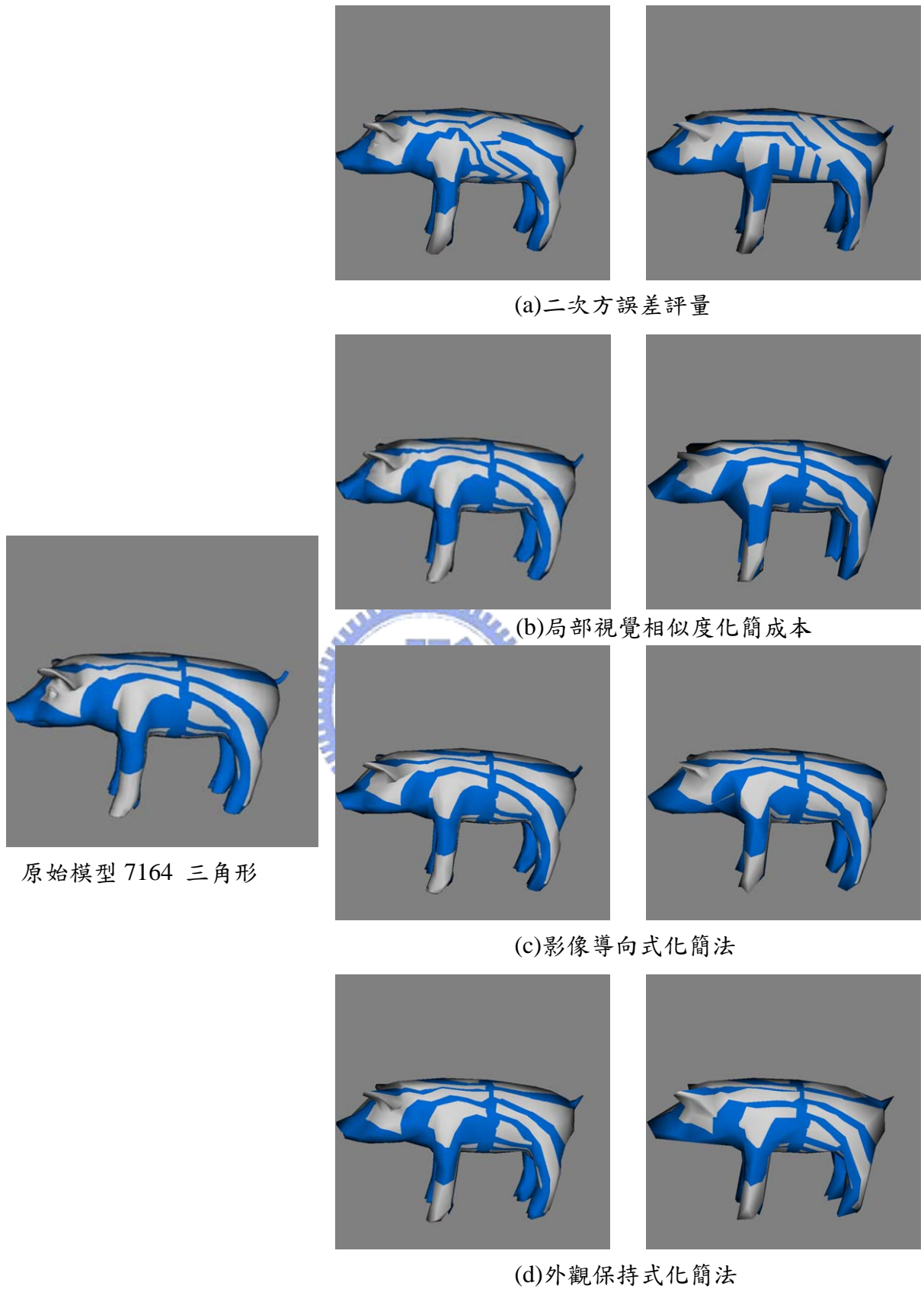


圖 4.2 比較四種方法化簡模型的結果，左列為 1000 個三角形，右列為 400 個(a)二次方誤差評量(b)本論文方法(c)影像導向式化簡法(d)外觀保持式化簡法。

4.1.1 視覺相似度化簡法與外觀保持式化簡法比較

外觀保持式化簡法以幾何距離誤差計算一貼圖像素在三維空間中的移動距離，基本上是一計算幾何誤差的成本估計，外觀保持式化簡法完全依模型參數化結果來決定其化簡的順序，參數化的好壞會影響其化簡之品質。它是一個計算幾何誤差的化簡成本，化簡出來的模型在幾何方面有一定的維持程度，貼圖變量也量一貼圖像素在三維空間中移動距離，因此貼圖也可以維持的不錯。

外觀保持式化簡法用包圍方形(bounding box)來限制貼圖變量的距離，然而用包圍方形會使得所計算的成本高估(over estimate)實際的成本。當化簡的層次愈來愈多時，會使得化簡成本無法實際代表真正的幾何誤差，在低多邊形數目時，多觀保持式化簡法無法對於較細微或對人類視覺影響較大的地方，如圖 4.3 即為這種情況。視覺相似度成本計算在多邊形數目不多時依然能保留豬腿上的蹄。

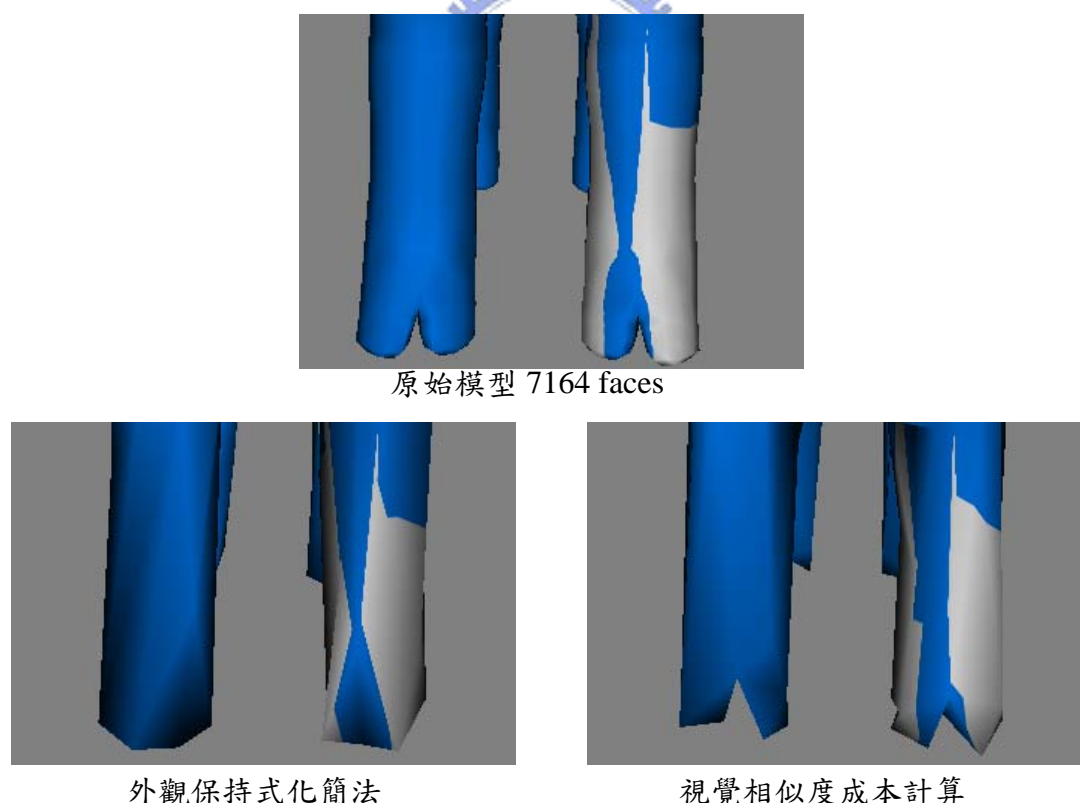
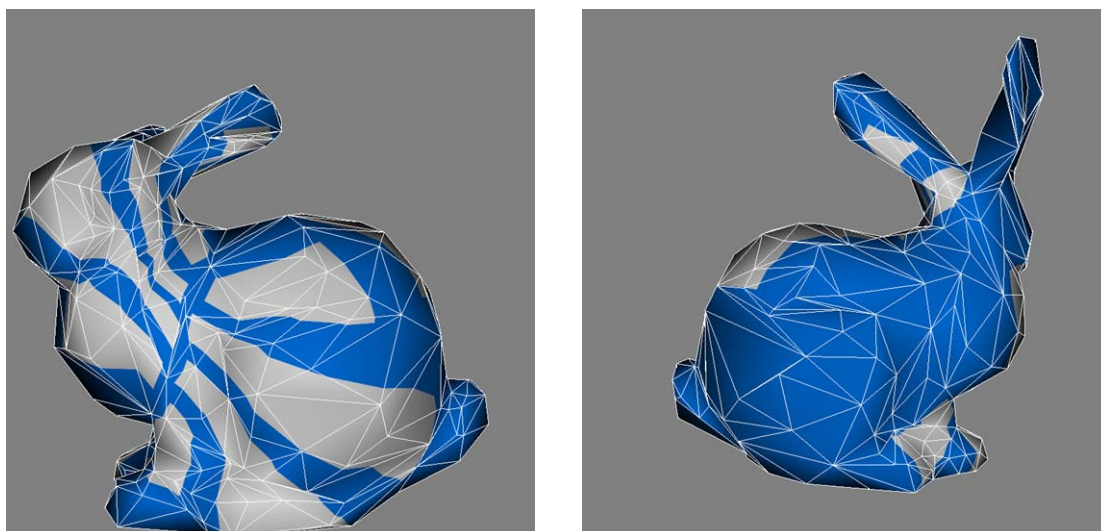
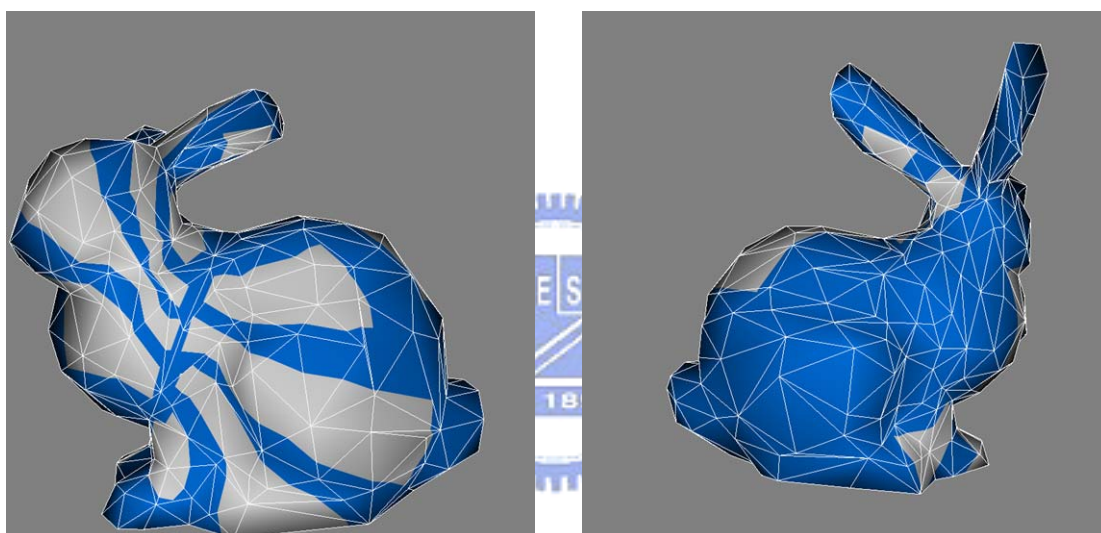


圖 4.3 在 400 個三角形的豬模型上，視覺相似度成本計算較可保留其外觀。



(a)視覺相似度化簡成本



(b)外觀保持式化簡法

圖 4.4 兔子模型原始三角形數為 10000，以(a)視覺相似度化簡成本，(b)外觀保持式化簡法，化簡至 700 個三角形

外觀保持式化簡法對於參數化比較平均的模型，也就是貼圖拉扯(texture stretch)較不嚴重的模型，可以進行非常平均的化簡。圖 4.4 的兔子模型使用非常良好的參數化，所以可以非常平均的化簡該模型。圖 4.4(a)可以觀察出我們的方法將頂點多數保留在兔子的正面，也就是貼圖內容變化較大的地方，而背面比外觀保持式化簡法保留較少的頂點。外觀保持式化簡法無法考慮貼圖內容，很有可能保留較多的頂點在會造成貼圖拉扯但又無貼圖內容變化的地方。如圖 4.5(a)，圖中的圓形在紅框框同一緯度之頂點給予貼圖拉扯較嚴重的貼圖座標，外觀保持式會因為貼圖拉扯所以保留較多那些區域的頂點，而(b)中視覺相似度成本只對於最

後呈現出來模型的外觀做為化簡依據，對於貼圖拉扯的區域，只要其上無貼圖內容的變化，進行化簡對於外觀是完全不影響的，可以得到比外觀保持式化簡法更平均地化簡。

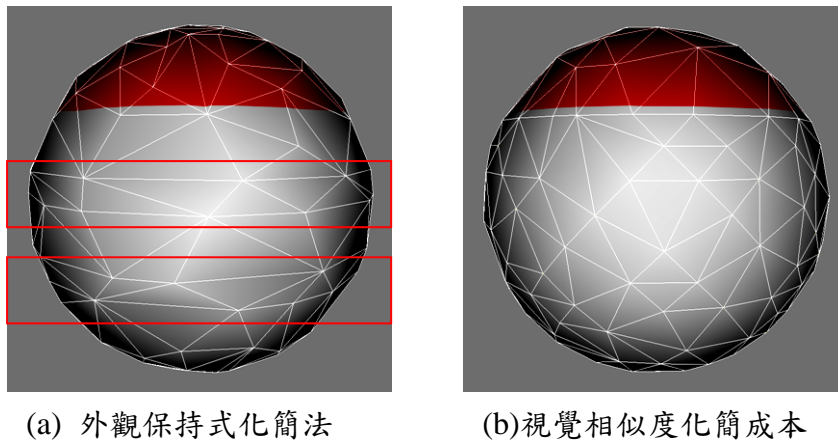


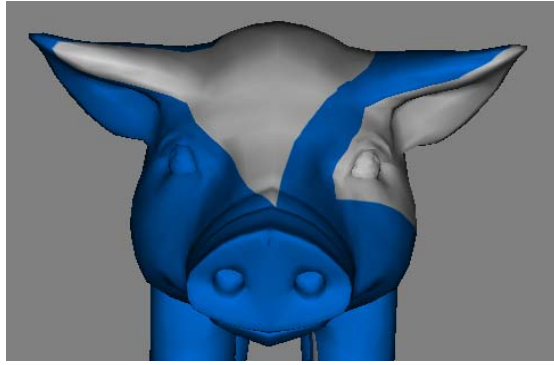
圖 4.5 外觀保持式與視覺相似度化簡成本比較，紅色箭頭代表在貼圖空間中貼圖拉扯嚴重的區域。

4.1.2 視覺相似度化簡成本與影像導向式化簡法比較

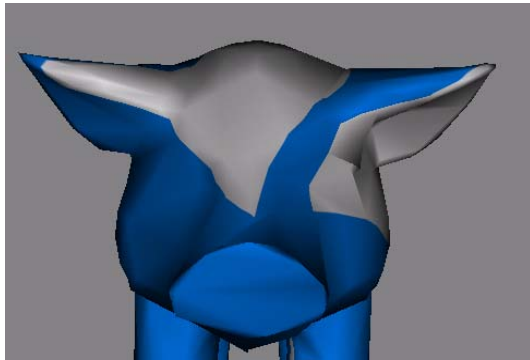


影像導向式化簡法所取樣的視點及影像的解析度決定了化簡的品質，而其視點之取樣是在整個模型的外部，故被較少的視點觀察到的區域會被急速的化簡掉，這使得模型的化簡會相當的不平均。影像導向式化簡法對於可見度較均勻物體，如類似圓球體的模型，會有較好的化簡結果，在其上的化簡區域被視點觀察到的機會是較公平的。以杯子模型為例，杯子內部的化簡區域會比杯子外部化簡區域較快被化簡，因較不會被外部的取樣視點觀察到，導致投影至畫面的像素較少使得化簡成本較小，甚至杯子內的幾何會無法維持。

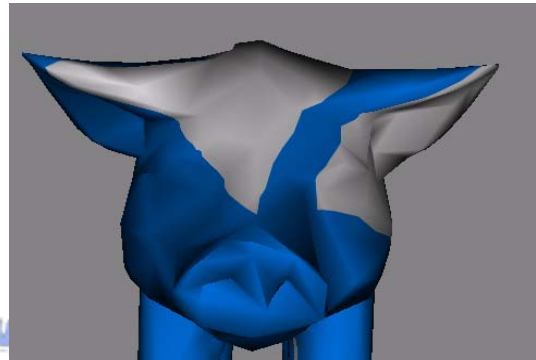
圖 4.6(a)使用影像導向式化簡法，在化簡至 2000 個三角形時，原本在鼻子上有的兩個孔已經被填平了，而圖 4.6(b)依然可以保持兩個孔的完整性。圖 4.6(c)中當化簡層次愈來愈深時，尾巴投影在各視點上影像所佔的像素數目較少，導致最後被化簡掉。



原始模型 7164 三角形



(a) 影像導向式化簡法
2000 三角形



(b) 視覺相似度化簡成本
2000 三角形



(c) 影像導向式化簡法
400 三角形



原始模型
7164 三角形



(d) 視覺相似度
化簡成本, 400
三角形

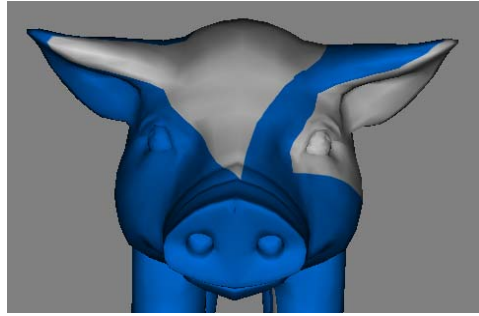
圖 4.6 影像導向式化簡法所不能保留的化簡區域，(a)(b)同為化簡至 2000 三角形的模型，(c)(d)為化簡至 400 三角形。

4.2 視覺相似度化簡成本結合全邊線摺疊之效果

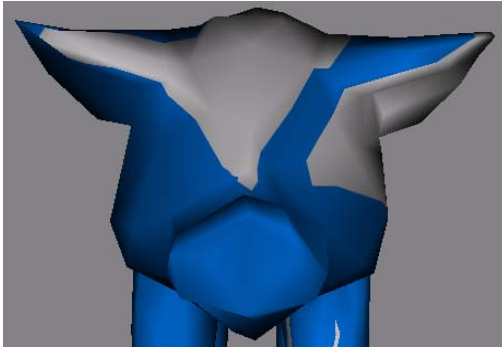
在 3.2.6 節中介紹了當我們的視覺相似度化簡成本要結合全邊線摺疊時，在摺疊邊線時要挑選一最佳的新頂點。我們選擇使用 [GH98] 所提出之方法，這個方法所挑選的新頂點基本上是幾何誤差的計算，因此可以使得模型在化簡後的幾何誤差較為改善。表 4.2 比較使用半邊線摺疊和全邊線摺疊之局部視覺相似度化簡成本計算所得到豬模型的幾何誤差。使用全邊線摺疊的結果是優於半邊線摺疊。雖然全邊線摺疊可以得到好的幾何誤差，但因視覺相似度化簡成本的特性，使得全邊線摺疊無法對模型的特徵做很好的保持。原因如圖 3.10 所示，以二維為例子，當 M_k 做全邊線摺疊後為 M_{k+1} ，而視覺相似度化簡成本演算法中取樣射線比較不容易產生只相交於一個網格的情況，如圖 3.5(b) 半邊線摺疊容易產生這類的射線，而給予這些射線處罰值，並用此值來維持在模型上的特徵。因此全邊線摺疊較無法保留模型上的細部特徵，但其幾何誤差是較好的，圖 4.8 展示使用全邊線摺疊的結果。

三角形數目	2000		1000		700		400	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
半邊線摺疊	0.07	1.01	0.142	1.190	0.180	1.334	0.278	2.134
全邊線摺疊	0.029	0.482	0.061	0.844	0.085	1.066	0.138	1.589

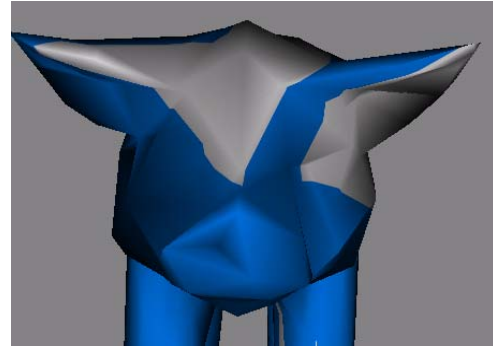
表 4.2 使用豬模型比較全邊線與半線摺疊對於視覺相似度化簡成本計算的幾何誤差。



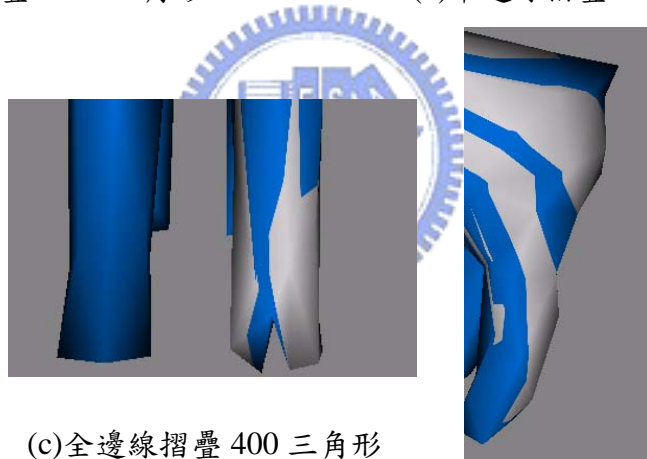
原始模型 7164 三角形



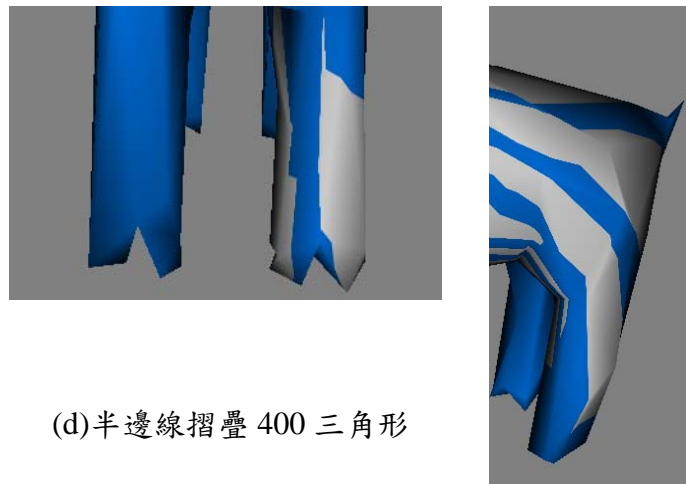
(a)全邊線摺疊 1000 三角形



(b)半邊線摺疊 1000 三角形



(c)全邊線摺疊 400 三角形



(d)半邊線摺疊 400 三角形

圖 4.7 使用全邊線摺疊之局部視覺相似度化簡成本計算。

第五章

結論與未來發展方向

5.1 結論

本論文提出一個網格邊緣化簡成本的計算機制，透過在化簡區域的周圍取樣一些視點，利用射線去觀察化簡前後網格的視覺相似度變化。經過實作與實驗發現對於含有貼圖內容的模型，化簡時不但維持模型上貼圖的可觀度，在多邊形數目少時幾何部分也可保留模型上的特徵，保留較多頂點在貼圖變化較大的區域。本論文方法是少數依貼圖內容變化而反應在化簡成本之上的方法，並且不是從整個模型外部來觀察化簡的變化而是只觀察局部區域性的變化，因此對於所有可化簡的地方給予相當平均的化簡機會，使得本論文方法比影像導向式化簡法保留更多模型上細部特徵，比外觀保持式化簡法保留更多頂點在貼圖變化較大的區域，而不是貼圖拉扯較大的區域。

5.2 未來發展方向

針對本論文所提出的視覺相似度化簡成本計算方法，未來有幾個可能的發展方向：

降低射線的取樣數目

在進行化簡之前可以對其上的貼圖內容進行分析其上的顏色對比，對比較低的區域表示貼圖的變化不明顯，因此可以不用取樣太多射線即可知道其網格的顏色變化，但需要一個機制加權回來應該有的化簡成本。

射線的處罰值

當射線只相交於一個網格而無法得到顏色差異的情況時，我們給予 0.5 的處罰值，對於不同角度的視點或許我們可以給予不同的處罰值，對幾何誤差較大的情有較大的處罰值，以期可以對於模型的幾何誤差有更好的結果。



參考資料

- [CVM+96] Cohen, J., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F., and Wright, W. "Simplification Envelopes". *Proceedings of SIGGRAPH96*. pp. 119-128. 1996.
- [CMO97] Cohen, J., Manocha, D., and Olano, M. "Simplifying Polygonal Models Using Successive Mapping". *Proceedings of IEEE Visualization '97*. pp. 395-402. 1997.
- [COM98] Cohen, J., Olano, M., and Manocha, D. "Appearance-Preserving Simplification". *Proceedings of SIGGRAPH98*. pp. 115-122. 1998.
- [CRS98] Cignoni, P., Rocchini, C., Scopigno, R. "Metro : Measuring Error on Simplified Surfaces.". *Computer Graphics Forum*. Vol. 17(2).pp.167-174.1998
- [EDD+95] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, M. "Multiresolution Analysis of Arbitrary Meshes". *Proceedings of IEEE Visualization '97*. pp. 173-182. 1995.
- [FS93] Funkhouser, T.A., and Sequin, C.H. "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments". *Proceedings of SIGGRAPH 93*. pp. 247-254. 1993.
- [GH97] Garland, M., and Heckbert, P.S. "Surface Simplification Using Quadric Error Metrics". *Proceedings of SIGGRAPH97*.pp.209-216. 1997.
- [GH98] Garland, M., and Heckbert, P.S. "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics". *Proceedings of IEEE Visualization '98*. pp. 263-270. 1998.
- [HDD+93] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. "Mesh Optimization". *Proceedings of SIGGRAPH 93*. pp. 19-26. 1993.

- [Hoppe96] Hoppe, H. "Progressive Meshes". *Proceedings of SIGGRAPH 96*. pp. 99-108. 1996.
- [Hoppe97] Hoppe, H. "View-Dependent Refinement of Progressive Meshes." *Proceedings of SIGGRAPH 97*. pp. 189-198.1997.
- [KLS96] Klein, R., Liebich, G., and Straßer, W. "Mesh reduction with error control". *Visualization 96*. ACM, November 1996.
- [LT00] Lindstrom, P., and Turk, G. "Image-Driven Simplification". *ACM Transactions on Graphics*. Vol. 19(3). pp. 204-241. 2000.
- [LE97] Luebke, D., Erikson, C. "View-dependent simplification of arbitrary polygonal environments". *SIGGRAPH97 Proc.*,1997
- [LSS+98] Lee, A., Sweldens, W., Schroder, P., Cowsar, L., Dobkin, D. "MAPS:Multiresolution Adaptive Parameterization of Surfaces." *SIGGRAPH98*, pp.95-104
- [SZL92] Schroeder, W.J., Zarge, J.A., Lorensen, W.E. "Decimation of Triangle Meshes". *Proceedings of SIGGRAPH 92*. pp. 65-70. 1992.
- [XEV97] Xia, J. , El-Sana, J. Varshney, A. "Adaptive Real-Time Level-of –Detail-based Rendering for Polygonal Models." *Proceedings of Transactions on Visualization and Computer Graphics*. Vol.3(2) pp. 171-183.1997.
- [XV96] Xia, J. , Varshney, A. "Dynamic View-Dependent Simplification for Polygonal Models." *Proceedings of IEEE Visualization '96*. pp.327-337,1996.