

國立暨南國際大學資訊工程學系
碩士論文

**以 Lattice Boltzmann Method 為基礎的 NPR
水彩繪畫模擬**

**Simulating NPR Watercolor Painting based on the
Lattice Boltzmann Method**

指導教授：陳履恆博士

研究生：李宜靜

中華民國 九十七 年 六 月

國立暨南國際大學碩士論文考試審定書

_____資訊工程_____學系(研究所)

研究生李宜靜_____所提之論文

以 Lattice Boltzmann Method 為基礎的 NPR 水彩繪畫模擬

Simulating NPR Watercolor Painting based on the Lattice Boltzmann Method

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

吳曉芝

委員兼召集人

蔡孟峰

委員

陳復恒

委員

中華民國 97 年 6 月 16 日

博碩士論文電子檔案上網授權書

本授權書所授權之論文為授權人在暨南國際大學資訊工程學系

96 學年度第二學期取得碩士學位之論文。

論文題目：以 Lattice Boltzmann Method 為基礎的 NPR 水彩繪畫模擬

指導教授：陳履恆

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館及本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

- 讀者基非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：李宜靜

簽名：李宜靜

中華民國 97 年 06 月 20 日

致謝

本篇論文能夠順利完成，首先，我要感謝我的指導教授 陳履恆博士。陳老師在
我有研究上的困惑時，會給予提示，常讓我擁有獨立思考的機會，並在我需要幫助時，
給了我最好的指導與鼓勵，讓我學會了很多思考問題的方式與解決問題的方法。此
外，我要感謝口試老師 吳曉光博士與蔡孟峰博士給予我論文上的指導與建議，使我
的論文更加的完整與嚴謹。

我也非常感謝研究室學長：宗志、岐智與禹翔學長，給予我研究上的協助，還有
竣蘭與鎮宇同學在這兩年來的陪伴與互相鼓勵，也感謝學弟妹的支援與關心。再來，
我要特別感謝我的室友慧敏與玉玲，這兩年來在生活上的互相照顧與細心陪伴。我還
要感謝我的大學老師 張志標老師與朋友純雅、怡涵，他們給了我剛到暨大時心靈上
的陪伴與鼓勵。

最後，我要感謝我的家人，他們是我的避風港，對我有永遠的關心與支持，他們
永無止盡的愛讓我勇敢的追尋我的目標，並也完成了這篇論文。

論文名稱：以 Lattice Boltzmann Method 為基礎的 NPR 水彩繪畫模擬

校院系：國立暨南國際大學資訊工程學系

頁數：56

畢業時間：97/6

學位別：碩士

研究生：李宜靜

指導教授：陳履恆 博士

摘要

Non-Photorealistic Rendering(NPR)在圖學的領域中是一個重要的課題，此一特殊的成像方式不以物品的擬真度為優先考量，而是著重筆觸、繪畫風格等的呈現。對於 NPR 水彩研究而言，模擬水在畫紙上的流動與擴散現象，是一個重要的研究項目。

在本篇論文中，我們實做了一個 NPR 水彩系統。首先，以 3D model 來建構出場景，並藉由探討水彩濃度與不同顏色顏料的互相混色來產生出具有水彩繪畫風格的視覺效果

為了達到以上目的與模擬渲染的效果，我們利用流體力學中的 Lattice Boltzmann Method(LBM)物理模型，來模擬水在 2D 畫布上的流動，然後藉由抽取三維模型自身的特性如輪廓以及骨架結構等，來模擬出多種水彩筆觸。

我們的筆觸由多個 pixel 組成，並利用 Phong's reflection model 來計算出每一個 pixel 的顏色。因此，沿著 pixel 的移動，筆觸上的顏色會漸漸的轉變。利用這個方法我們可以得到自然而生動的筆觸。換言之，我們提出的方法是根據場景資訊自動產生出具有明暗效果的筆觸。

由實驗結果，我們可以得到具有自然又有趣的色彩、明暗與筆觸的 NPR 水彩動畫影片。

關鍵字：電腦圖學(CG)、Non-Photorealistic Rendering(NPR)、Lattice Boltzmann Method(LBM)、水彩、渲染、顏色混色、明暗、動畫

Title of Thesis: Simulating NPR Watercolor Painting based on the Lattice Boltzmann Method

Name of Institute: National Chi Nan University, Dept. of CSIE

Pages: 56

Graduation Time: 6/2008

Degree Conferred: Master

Student Name: Yi-Jing Lee

Advisor Name: Dr. Lieu-Hen Chen

Abstract

Non-Photorealistic Rendering (NPR) has been an important research topic in the field of Computer Graphics. Instead of highlighting the realism of CG synthesized images, NPR focus on the representation of the stroke and painter's style. For the research of NPR watercolor, simulating water diffusion and flow in paper is an important task.

In this paper, we implement a NPR watercolor system. Our scene is composed with 3D models. The special visual effects of watercolor paintings are created by simulating the color mixing and the different concentration of watercolor.

To achieve the above goal and the "Brush-Washing" effect, the water flow on 2D paper is simulated based on the Lattice Boltzmann Method (LBM). Moreover, the strokes of watercolor are simulated by utilizing the auxiliary bone structures and the shapes of three-dimensional models.

A stroke consists of multiple pixels. The color on each pixel is calculated by Phong's reflection model. Therefore, the color on a stroke gradually transits from pixel to pixel. This approach creates nature and vivid appearances of strokes. In another word, our proposed method creates the tone of strokes automatically based on the 3D information of scene.

As shown in the experiment results, our system synthesizes watercolor NPR animation with interesting and nature color, tone and strokes.

Keyword: CG 、 Non-Photorealistic Rendering(NPR) 、 Lattice Boltzmann Method (LBM) 、 watercolor 、 washing 、 color mixing 、 tone 、 animation

目錄

摘要.....	II
Abstract.....	III
目錄.....	IV
圖目錄.....	VI
第一章 序論.....	1
1.1 研究動機與目的.....	1
1.2 論文編排模式.....	2
1.3 本論文的貢獻.....	3
第二章 研究背景.....	4
2.1 NPR 概述.....	4
2.2 水彩概述.....	6
2.2.1 水彩工具與材料.....	6
2.2.2 色彩.....	7
2.2.3 水彩技法.....	8
2.3 水彩畫法的分析與量化.....	9
2.4 相關研究.....	11
第三章 Lattice Boltzmann Method.....	13
3.1 簡介.....	13
3.2 LBE 的兩個運作步驟.....	14
第四章 系統概述.....	16
4.1 系統架構及運作流程.....	16
4.1.1 系統架構.....	17
4.1.2 運作流程.....	18
4.2 資料讀取及虛擬畫布.....	19
4.2.1 資料讀取及整理.....	20
4.2.2 虛擬畫布：Modified Z-buffer.....	22
第五章 Shading 與光.....	26
5.1 shading.....	26
5.2 光.....	27
第六章 水彩模擬.....	28
6.1 Simulation Overview.....	28
6.2 水彩模擬流程圖.....	28
6.3 color blending.....	34
第七章 筆觸模擬與動畫生成.....	36
7.1 筆觸模擬.....	36

7.1.1 2D shape-based Stroke.....	36
7.1.2 3D model-based Stroke.....	38
7.2 描繪過程.....	40
7.3 動畫生成.....	41
第八章 研究成果.....	42
第九章 結論以及未來工作.....	53
9.1 結論.....	53
9.2 未來工作.....	54
參考文獻.....	56

圖目錄

[圖 1] NPR 中的各種繪畫風格.....	5
[圖 2]各種水彩畫筆與其筆觸形狀.....	7
[圖 3](a)色相環[12]，(b)(c)水彩的三原色與二次色[13].....	8
[圖 4]各種水彩技法[10].....	9
[圖 5]水彩繪畫中用高光處來表現物體受光面.....	11
[圖 6] Curtis[1]等人提出模擬水彩在紙上成色的三層流體模型.....	12
[圖 7] Lattice Boltzmann Method 晶格模型.....	13
[圖 8] stream step.....	14
[圖 9]collide step.....	14
[圖 10]筆觸的細節.....	16
[圖 11] System architecture.....	17
[圖 12]系統流程.....	19
[圖 13] Adapter 資料匯入示意圖.....	20
[圖 14]判斷 X-file 中有 Bone 與否.....	21
[圖 15]兩個 modified Z-buffer 的主要資料結構.....	22
[圖 16]綜合三種線段產生出來的 Output.....	23
[圖 17] Adapt Depth Test 流程圖.....	24
[圖 18] Depth-Test 前後結果圖.....	25
[圖 19]執行完 Hidden Line Elimination 後的結果圖.....	25
[圖 20]使用不同 shading 方式所產生的影像.....	26
[圖 21]具有 specular 效果的影像.....	27
[圖 22]水彩模擬流程圖.....	29
[圖 23] stream step simulation 前後圖.....	31
[圖 24]不同沉澱比例所造成的筆觸樣式.....	31
[圖 25]一滴水在虛擬畫布上的擴散過程.....	33
[圖 26]一筆水彩筆觸的繪畫過程模.....	33
[圖 27]RGB 色彩系統.....	34
[圖 28]CMYK 色彩系統.....	34
[圖 29]CMYK 三原色兩兩混色圖.....	35
[圖 30] 2D shape-based stroke vector 存取示意圖.....	37
[圖 31]柱狀體上的筆觸方向.....	38
[圖 32]手掌的骨架階層構造.....	39
[圖 33]3D model-based stroke vector 存取示意圖.....	39
[圖 34]手掌上，Thumb 部分的三維筆觸分布.....	39
[圖 35]水彩描繪過程示意圖.....	40

[圖 36]投影平面為基礎的筆觸連續示意圖	41
[圖 37]系統模擬畫面.....	42
[圖 38]連續的 Teapot 影格	43
[圖 39] Teapot-描繪過程:3D model-based Stroke.....	44
[圖 40] [圖 39]之第 1 個圖放大	45
[圖 41] [圖 39]之第 2 個圖放大	45
[圖 42] [圖 39]之第 3 個圖放大	46
[圖 43] [圖 39]之第 4 個圖放大	46
[圖 44] [圖 39]之第 5 個圖放大	47
[圖 45]Teapot:2Dshape-based Stroke	47
[圖 46] Scene-描繪過程:2D shape-based Stroke.....	48
[圖 47] [圖 46]之第 1 個圖放大	49
[圖 48] [圖 46]之第 2 個圖放大	49
[圖 49] [圖 46]之第 3 個圖放大	50
[圖 50] [圖 46]之第 4 個圖放大	50
[圖 51] [圖 46]之第 5 個圖放大	51
[圖 52]不同角度的 Scene:2D shape-based	51
[圖 53] Flower:2D shape-based Stroke.....	52

第一章 序論

1.1 研究動機與目的

在電腦圖學的領域中，對於筆觸的模擬及研究由來已久，其中對此著墨甚多的，就屬 Non-Photorealistic Render(NPR)這一塊領域了。

然而，在瀏覽了幾種近年來較具指標意義的 NPR 水彩模擬演算法之後，仍無法找到離我們的目標較接近的成果。利用電腦模擬人繪，就彷彿是無窮的逼近般，並沒有一個完美的模擬方法。而目前的 NPR 水彩模擬生成，在[2]中採用 Navier-Stokes equations 去模擬流體在 surface 上的流動，而在[4]中採用即時座標算法來模擬水彩，而[5、6]採用計算 texture 的方式來模擬水彩的表現，也有探討畫紙為主的 NPR 水彩模擬[8]。在成像方面，有些方法採用 Image procession 的方法，將照片生成具有 NPR 水彩風格的結果；有些雖然使用 3D-model 作為參考資料，但仍需要使用者互動式的指定筆觸的走向與筆觸色彩；再不就是採用局部的觀點生成筆觸方向，結果造成筆觸的不連續。這些模擬模式，無論從畫者或觀者的角度來看，都仍有相當大的進步空間。

因此，我們提出這篇研究論文，希望能夠採用近幾年計算機用來模擬流體的一個主要方法，Lattice Boltzmann Method 來模擬流體，並參考 3D-model 的資訊，並且觀察畫家在繪畫時採用的幾種常用的技法，藉由這些資訊來模擬出更逼近人類手繪繪畫的結果。同時，我們更希望能夠藉由這些結果，整合 3D 建模本身的優勢，生成連續的水彩繪畫過程影像動畫。

1.2 論文編排模式

為了能夠生成「高可信度」的模擬結果，我們嘗試從水彩的色彩、技法與畫法來分析，抽取出可供量化的參數。並且，為了能充分模擬此種繪畫的感覺，我們將探討繪線的幾種模式。此一部分將於稍後的第二章來做探討。

在第三章，我們將簡述近幾年計算機用來模擬流體的一個主要方法，Lattice Boltzmann Method，也是本篇論文用來模擬流體的方法。

第四章將介紹我們早期系統的一些基本架構，包括系統架構與資料讀取時所處理的前置工作的介紹，並講解系統自訂的虛擬畫布：Modified Z-buffer。此一 Z-buffer 亦可視為本系統在運作上的核心。

第五章則大致說明在實作時所採用的 Shading method，以及 specular 效果的相關運算。在上述的工作準備完成後，我們在第六章講解我們水彩模擬的過程。包括了模擬流程解說與 color blending 的部分。

第七章主要講解筆觸的生成與動畫的製作。包括 2D shape-based Stroke 與 3D model-based Stroke 的製作，與如何使筆觸生成連續的方法

最後，我們在第八章展示我們的研究成果，以及第九章的結論說明及未來的工作。

1.3 本論文的貢獻

對於學術研究、國家發展及其他應用方面預期之貢獻。

學術部分：

- a. 根據水彩的特殊性質，我們可以建構出具有強烈水彩風格，而又不失 object 之特徵性與整體性的輸出影像與動畫，由於現有的 NPR 技術在影片方面主要的應用仍侷限於 tone-rendering 的各種改進版本，即賽路風格的卡通成像，或是偏向擬真的真實鋼體模擬成像，少見有完全以水彩筆觸構成的圖片及影像。因此本研究若能有理想的成果，相信可以在技術部份作出許多的突破與貢獻。
- b. 提供了一個 3D 的場景，object 會因為場景中的各項資訊而呈現出在水彩畫中應有的效果，一般看到的水彩畫作都是平面的，很少有 3D 的水彩畫作，因此若本研究能夠有理想的結果，相信可以在數位藝術方面做出許多的貢獻與突破。

其他應用：動畫製作、靜態電腦繪圖

- a. 本研究成果提供自動化的水彩筆觸影像生成，因此可以對傳統的手繪動畫在製作上提出自動化的改進，進而加速影片的製作速度以及品質的提升。
- b. 由於水彩筆觸風格強烈，藉由各種 plug-in 的開發，我們可以將此系統導入 3D-成像引擎中，給這些工具員先的使用者另一種完全不同以往的成像風格。

第二章 研究背景

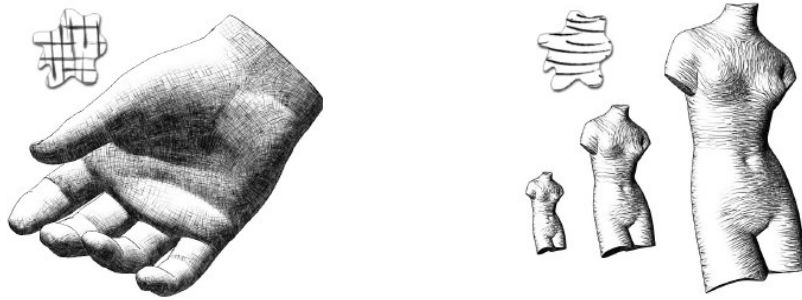
2.1 NPR 概述

Non-Photorealistic Rendering(NPR)在圖學的領域中由來已久，此一特殊的成像方式不以 object 的擬真度為優先考量，而是著重於筆觸(stroke)、繪畫風格(paint style)等的呈現，而其成像多以誇張的手法表現欲突顯的部分，而不注重其呈現出的正確精確性，使產生的影像呈現出像是鉛筆[7]、水墨畫[3]、油畫或是水彩畫[2]等類似用手工繪畫出來的風格。

在 NPR 的領域中，根據資料來源劃分，大致上可以分為兩類：2D image-based method 以及 3D model-based method。2D image-based 的 NPR 根據取得的資訊生成各種形式的筆觸，其原始資訊樣本為單張的 image 或是連續張數的 video frame。所使用的處理手法其相關演算法亦較偏向於 image-processing 的領域。

另一方面，3D model-based 的 NPR 則是以 3D-model 以及場景為資料來源，由於這些模型定義都是長久以來在 CG 領域中所廣為使用，因此有許多相關領域上的研究成果(如 3D-transformation, lighting, shading, etc.)都可以直接帶入以提供不同的參考及貢獻。對於具有 3D-model 的場景，欲產生自由且連續的 3D animation 這個課題而言，仍然有著許多有趣並且值得探討的研究空間。

電腦圖學的領域中，大多數的研究皆為針對靜態的圖片為樣本作成像，其所得的成品亦是靜態的圖片。而我們所要研究的是針對在 3D-model 上做出具有動畫效果的水彩畫模擬。



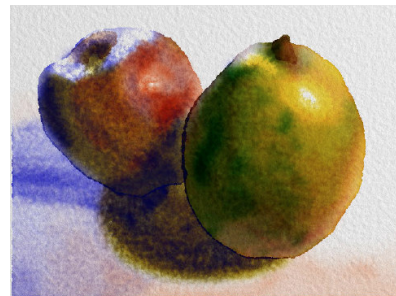
(a)鉛筆畫中的細線法[7]



(b)MoXi 系統所繪的水墨畫風[3]



(c)油畫畫風



(d) 2D image-based 的水彩畫風[2]

[圖 1] NPR 中的各種繪畫風格

2.2 水彩概述

「水彩畫」顧名思義，就是以水為媒介調和顏料作畫的表現方式。因此，水彩畫的範疇可以推展到古埃及人的畫卷，波斯人富有異國情調的細密畫等，隨著時間的推演，歷經十八、十九世紀歐洲水彩畫的興起，尤其是英國多位水彩畫家們的努力，到十九世紀末，水彩畫已經發展出完整的獨立體系。從本質上來說，水彩藝術就是色彩、綜合與光。

2.2.1 水彩工具與材料

水彩畫的工具與材料，大致上有筆、紙與顏料。

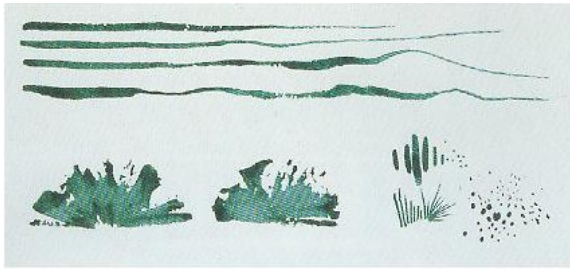
水彩筆一般可以分成線筆、扁筆、圓筆、排筆以及特殊用途的扇形筆等[13]。圓筆是水彩畫的主要用筆，筆腹圓、筆頭圓而微尖，可以描繪任何物體。



(a)圓筆：畫大部份的圓形和薄塗顏料時，必須使用圓頭筆



(b)扁筆：用來畫乾筆法的乾硬線條，可以形成重複排列、長度相同的線條



(c)線筆：要畫細線時會用到這種筆

[圖 2] 各種水彩畫筆與其筆觸形狀

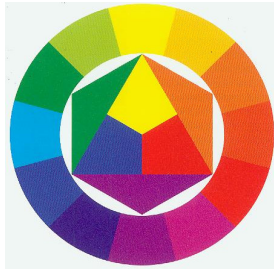
一張好的水彩紙可以讓畫者獲得信心，因為所有的技法都可以因而發揮出來。畫紙成分可分成碎布紙與木漿紙兩種。除了成分外，畫紙還有重量與開數的區別。以重量來說，畫家最常用的為重磅紙，開數則有全開、對開與四開等的區別。

水彩顏料是由色料、阿拉伯膠及甘油，還有稀釋液等構成，通常由礦物以及動植物中提煉出來。水彩顏料也有透明與不透明之分，而常用的水彩顏色約有三十八種。

2.2.2 色彩

色彩一般有三個附屬的特性，稱之為「色彩的三屬性」，分別為色相、明度與彩度。「色相」其實就是色彩的種類，如紅、黃、藍紫、黃綠等，和色彩的明暗、強弱沒有關係，僅區別這一色與另一色的一種名稱而已。「明度」就是色彩的明暗度，如色相本身的深淺與受光與不受光所造成的明暗。「彩度」又稱色度、純度或飽和度，感覺上彩度高的顏色耀眼一些，彩度低的顏色稀薄或混濁一些。

紅、黃、藍三色是水彩中的三原色，表示這三種顏色是無法由調色取得的，卻可以利用他們調配出所有的顏色。



(a) 伊登十二色相環，中間三角形即三原色，每一邊上為二次色，最外環為三次色



(b) 三原色：紅、黃、藍



(c) 以二個原色調出二次色 (濕重疊)：黃和紅調出橙色，黃和藍調出綠色，而紅和藍則調出紫色

[圖 3](a)色相環[12]，(b)(c)水彩的三原色與二次色[13]

2.2.3 水彩技法

在所有的繪畫技法之中，水彩畫技法變化最大也最多樣化，究其原因，流動不羈的「水」是一個相當重要的因素。

「平塗法」是最基本的水彩技法。平塗法是將畫面分成幾個區域，每個區域平塗一種顏色，不做明暗與筆觸轉折的變化，它的美完全靠色塊與色塊間的關係表達出來。

「重疊法」奠基於平塗法，卻靠著一次次的疊筆，以整張畫同時進行，由面而線而點，且由淺而深疊色，做出明暗、立體空間與筆觸之美。

「縫合法」就是將兩個並置的色塊像縫衣般的接合在一起。

「乾筆法」又名乾擦法，它是將畫筆裡的水份減到最少，然後沾上顏料在粗糙的紙上乾擦筆觸。

「渲染法」又名溼中溼畫法，必須先將紙打溼或者浸泡在水中，然後再上顏色，一直到最後，整張畫都是在潮濕狀態下完成的。渲染法完成的作品並不講求外型的精確，重要的是傳達一份浪漫情趣。渲染法由於都是朦朧的輪廓組成，所以如果能在幾乎完成的畫乾後，薄施小部分重疊筆，渲染法將更完美。

「複合技法」就是綜合運用上述的技法來完成作品，如此也能夠使作品更有變化。



(a) 重疊法



(b) 縫合法



(c) 乾筆法



(d) 渲染法

[圖 4] 各種水彩技法[10]

2.3 水彩畫法的分析與量化

要完成一幅水彩作品，必須先構圖，之後再加上色彩。在構圖方面，沿用了之前的研究結果，在研究中將 3D-model 投影在 2D 的虛擬畫布上，所以我們可以在虛擬畫布上得到整個 3D 場景中物體的輪廓線。所以隨後的加入色彩部分即是我們本節所要分析的重點。

比起其他媒材，水彩的技法最豐富，變化也最多，所以當一個畫家在作畫時，往往會依照畫面的需要而選擇技法，所以在一幅畫中畫家常會採用複合技法的方式來完成。而在用色方面，畫家會依據當下眼睛所看到的顏色來上色，同一靜物或景色會因為所處的空間與時間不同，在畫家筆下也會有不同的表現。依照以上所述，這些無疑的都增加了我們對水彩分析與量化的困難度。

大體而言，每種物體都有其相異的質感，如光滑、粗糙、笨重、細膩等，善用描繪方式的不同，可以利用明暗的變化及筆觸的風格來達到這些題材的特色。以下大略的分出兩大類水彩筆觸。

(A) 以模型形體為主的筆觸。筆觸在這個環節所扮演的角色，往往是用來強調物件的三維特性，讓觀者能由 2D 的繪畫在腦海中重建出物件原始的 3D 型態。在本論文中我們用物體本身的 skeleton 特性來計算此一筆觸。skeleton 是 3D-model 一項很重要的形狀特徵，此資訊容許我們根據 skeleton 重建 3D-model 的形體，也常應用在許多的不同的領域。

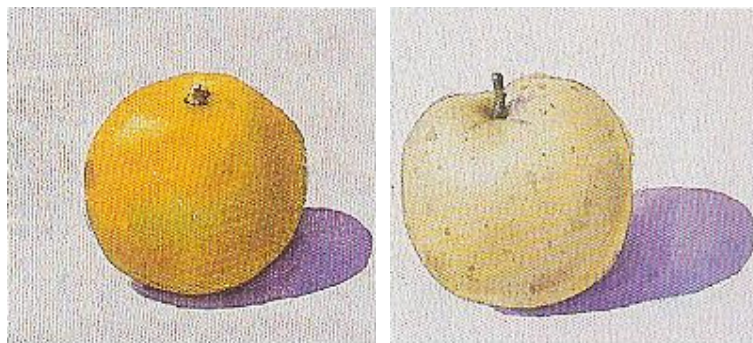
(B) 根據 3Dmodel 投影後的 2D-image shape 的特性所產生的筆觸。在繪圖的時候，有時我們需要額外的線條來強調物體本身 2D 的形狀。這種繪畫手法尤其使用在當物件本身具有不規律起伏的表面時。大致來說，此種繪線常會隨著物件本身輪廓線起伏、生成。

另一類為：

(a) 參考物體的特徵所產生的筆觸。一個物體的特徵往往是用來辨別物體種類的方法，例如：skeleton、edge 或 color 都屬於物體特徵。

(b) 依照物體 texture 所產生的筆觸。物體的 texture 也就是物體的顏色與紋理。

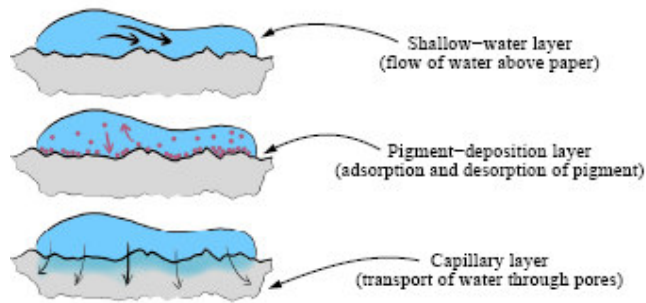
除了筆觸的分析外，明暗度也是水彩繪畫上一個重要的環節。在繪圖時常需要考慮到物體上的明暗層次，以便做出物體的立體感。所以常會在物體最亮的部份留白或是以清水洗潤亮面的高光處。



[圖 5]水彩繪畫中用高光處來表現物體受光面

2.4 相關研究

在 NPR 研究中，以水彩為主題的研究已相當多且具有不錯的成果。例如，由 Curtis 等人[2]提出的水彩模擬模型。他們模擬水彩與紙的交互作用，並將水彩在紙上的流動作用分為如[圖 6]的三層，採用 Navier-Stokes equations[11]模擬流體在 surface 上的流動，利用 KM-model 模擬光對顏料的影響。在成像方面，以 Image procession 與 3D-model 為原始資料，來產生水彩風格的 NPR Image，但每次都以大片面積的繪圖，再一層層的疊上去，最後再產生 Image。與我們研究的差別，我們採用近幾年計算機用來模擬流體的一個主要方法，Lattice Boltzmann Method 來模擬流體，並將畫布分為流動層與沉澱層，最後以一筆、一筆筆觸繪畫的方式呈現，並能產生出連續的水彩動畫。



[圖 6] Curtis[1]等人提出模擬水彩在紙上成色的三層流體模型：第一層模擬水在紙上的流動，第二層模擬顏料的沉澱，第三層模擬毛細現象

在利用 Lattice Boltzmann Method 來實做繪畫方面，有香港科技大學 Nelson Chu 博士所發表用來模擬中國水墨畫風的 MoXi 系統[3]。他們模擬墨水與吸收性良好的紙之間的相互作用，他們將 paper-model 分成三層，分別為 surface、flow 與 fixture layer，主要是利用 Lattice Boltzmann Method 模擬墨水在 flow layer 的流動與擴散現象。在成像方面，主要是建立了一個跟使用者互動的介面，使用者使用他們所提供的介面再利用手寫板便能在螢幕上呈現出自行繪畫且具有水墨畫風的影像。與我們研究最大的差別，除了模擬的對象不同之外，還有成像方面，我們的系統主要是能夠讀入 3D-model，然後根據 model 上的資訊，便能夠自動化的產生出具有水彩筆觸風格的影像，而讀入 3D-model 的優點，我們的系統可以生成出水彩繪畫風格的動畫。

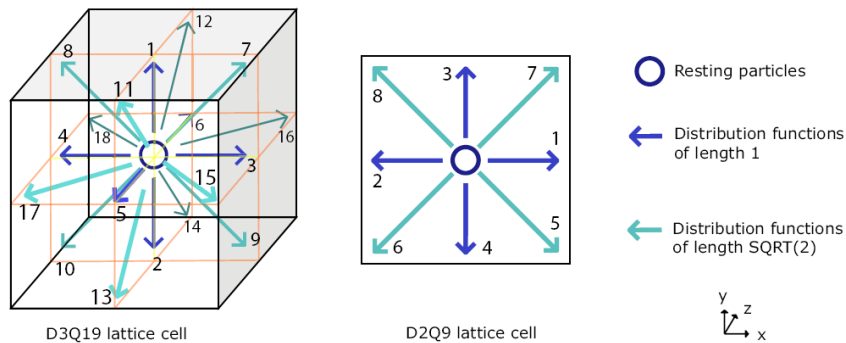
第三章 Lattice Boltzmann Method

本篇論文利用 Lattice Boltzmann Method 來模擬流體。Lattice Boltzmann Method 的優點[9]是不需要像 Navier-Stokes equations 處理複雜的偏微分方程式，而且只需要做簡單且局部的運作便能模擬流體的運動。

3.1 簡介

Lattice Boltzmann Method(LBM) 物理模型，是近幾年計算機用來模擬流體的一個主要方法。顧名思義它是以晶格為基礎而運作的模型。也就是說 LBM 是利用精簡的粒子運動模型來模擬流體。LBM 也可以被描述成是細胞自動機的一種類型，所有的細胞在每一個時間 t 中會依簡單的規則更新，而且每一個細胞的狀態都是由在上一個時間中($t-1$)包圍它的細胞的狀態來決定的。

LBM 模型是模擬不可壓縮流體，允許粒子只沿著晶格向量移動，常見的標準晶格模型為 D3Q19 與 D2Q9。在本研究中我們使用模擬水在 2D 平面上流動的 D2Q9 晶格模型，其中的 2 用來表示維度，而 9 則是用來表示有 9 個不同的晶格速度向量。如[圖 7] 左圖為 D3Q19 的晶格模型，右圖為 D2Q9 的晶格模型

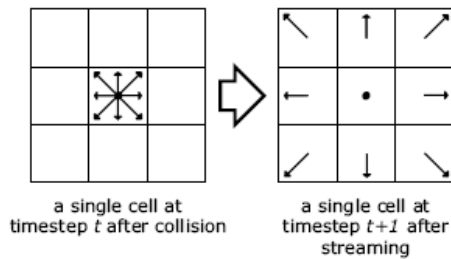


[圖 7] Lattice Boltzmann Method 晶格模型

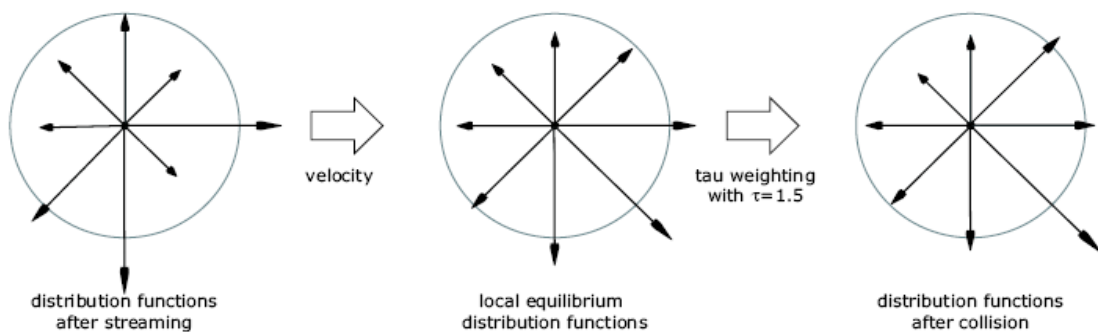
3.2 LBE 的兩個運作步驟

由[1]可知，在LBE模擬的過程當中包含了兩個步驟，而且在每一個時間 t 中都會重複這兩個步驟。第一個步驟為 stream step，第二個步驟為 collide step。stream step 描述晶格上粒子的移動；collide step 表示粒子在移動間所發生的碰撞。

我們由[圖 8]得知，stream step 只是將每一個晶格中，所有的 distribution functions 複製到相鄰晶格的相對向量上，其中的向量 0 是不會移動的，而在每一次的 stream step 當中，所有的晶格必須紀錄沿著每一個向量所移動的粒子個數，稱 particle distribution functions，通常用 f_i 來表示。在真實的流體中，粒子是不停的互相碰撞且互相影響的，所以 collide step 用來改變所有 particle distribution functions 的粒子分配，讓整個流體達到碰撞後的平衡。



[圖 8]stream step。左圖說明了在 D2Q9 中單一晶格分別在 streaming 前後的粒子分



[圖 9]collide step。在 collide step 期間，利用 stream step 所產生的 distribution functions(左圖)來計算 velocity 與 density，再利用 velocity 與 density 計算得到 local equilibrium distribution functions(中圖)。之後，兩個 functions 經由權重計算相加，便可以得到在 collide step 之後所以產生新的 distribution functions(右圖)。

在整個模擬過程中的兩個步驟，stream、collide step 在數學上可由以下兩個式子表示。 f_i' 表示為新的粒子分配函數， $f_i^{(0)}$ 為 collide 的平衡分配函數。

$$f_i' = (1 - \omega)f_i + \omega f_i^{(0)} \quad (1)$$

(1) 中的 ω 為一個 relaxation parameter，這個參數會影響流體的黏性， ω 越小越黏稠、越大越接近水的性質。

$$f_i^{(0)} = w_i \left\{ \rho + \rho_0 \left[3(\bar{e}_i \cdot \bar{u}) + \frac{9}{2}(\bar{e}_i \cdot \bar{u})^2 - \frac{3}{2}(\bar{u} \cdot \bar{u}) \right] \right\} \quad (2)$$

(2) 中的 w_i 表示每一個向量的 weight，當 $i=0$ ， w_i 為 4/9； $i=1,2,3,4$ ， w_i 為 1/9；

$i=5,6,7,8$ ， w_i 為 1/36。 \bar{e}_i 表示 9 個不同方向的晶格向量。 $\rho_0=1$ 。 ρ 表示為每一個晶格的密度(density)(3)。 \bar{u} 為速度(velocity)(4)。

$$\rho = \sum_{i=0}^8 f_i \quad (3)$$

$$\bar{u} = \frac{1}{\rho_0} \sum_{i=0}^8 e_i f_i \quad \text{for } \rho_0 = 1 \quad (4)$$

第四章 系統概述

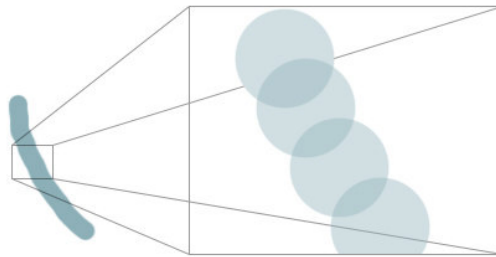
本篇論文，主要是建構在我們早期所開發的 NPR System 之上[10]，所以本章將介紹早期 NPR System 的系統架構、資料讀取與虛擬畫布的開發。

4.1 系統架構及運作流程

本節我們說明早期系統的構想、系統架構以及運作流程圖。

在早期構想中，考慮最原始的筆觸構成。假設單點的筆觸圖案為一實心同色圓，則沿某一軌跡連續貼出此圖案即可模擬出最簡單的筆觸。而根據此一基本想法加以擴充，便產生了構成不同風格的筆觸所需的各種參數。

我們還需要一個能夠模擬畫布的資料結構。這張模擬畫布的每一個 pixel，必須可以自由的參考 3Dmodel 以及矩陣運算過後的 2D-shape，或者整個世界場景中的相關資料如光影、方位等。基於這個構想，我們嘗試改良原本繪圖卡所提供的 Z-buffer 這樣的繪圖機制。藉由重新定義 Z-buffer 上的儲存資料結構來達到我們需要的結果。

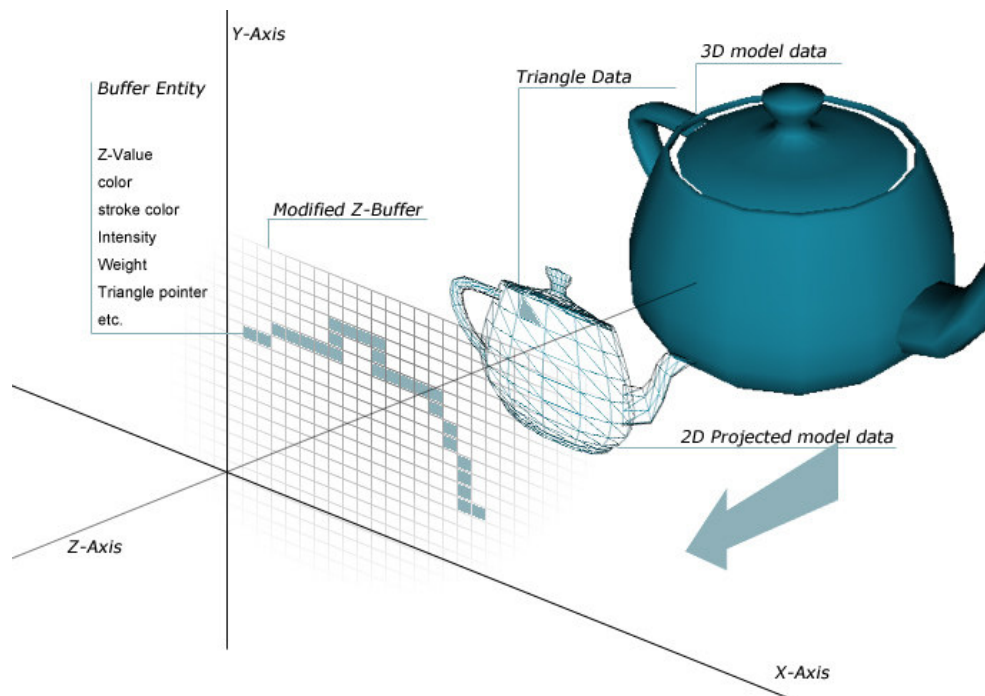


[圖 10]筆觸的細節

4.1.1 系統架構

為了可以處理畫布上每一條筆觸的細節，我們建構出一塊由 Buffer 組成的虛擬畫布。此一畫布除了傳統的 pixel - Z information 之外，還必須能夠參考到原始模型的資訊，以及平面化後的 shape 資訊，以便於計算出合適的筆觸資料。

如[圖 11]所示，此為 NPR-System 的基本架構。3D model data 即為原始的模型資料；Modified Z-Buffer 即代表「虛擬畫布」。每一個 Buffer 上的像素必須儲存一個 Triangle data 的連結，以方便讓 Modified Z-Buffer 上的每一塊像素可以便利的參考 3D 原始資料，以及套用過所有的轉置矩陣(模型座標矩陣、世界座標矩陣、投影矩陣)後的 2D projected model data。其次，為了整理出模型 Triangle-Triangle 之間的相互關係，以及便利找尋模型上的輪廓線，我們必須重新定義原始模型中三角形的 Index 資料，重新儲存為 Triangle data。



[圖 11] System architecture 列舉幾個系統中重要元件的關係

4.1.2 運作流程

由我們早期的 NPR-System 概括來看，可以分為三個主要的部分，如[圖 12]。

(A) 3D model loader – Pre Processing part

我們將由 X-file 讀入的資料，重新拆解並整理成我們系統需要的資料結構。模型與骨架的階層關係、模型內部的表面關聯性都在此作前置處理。

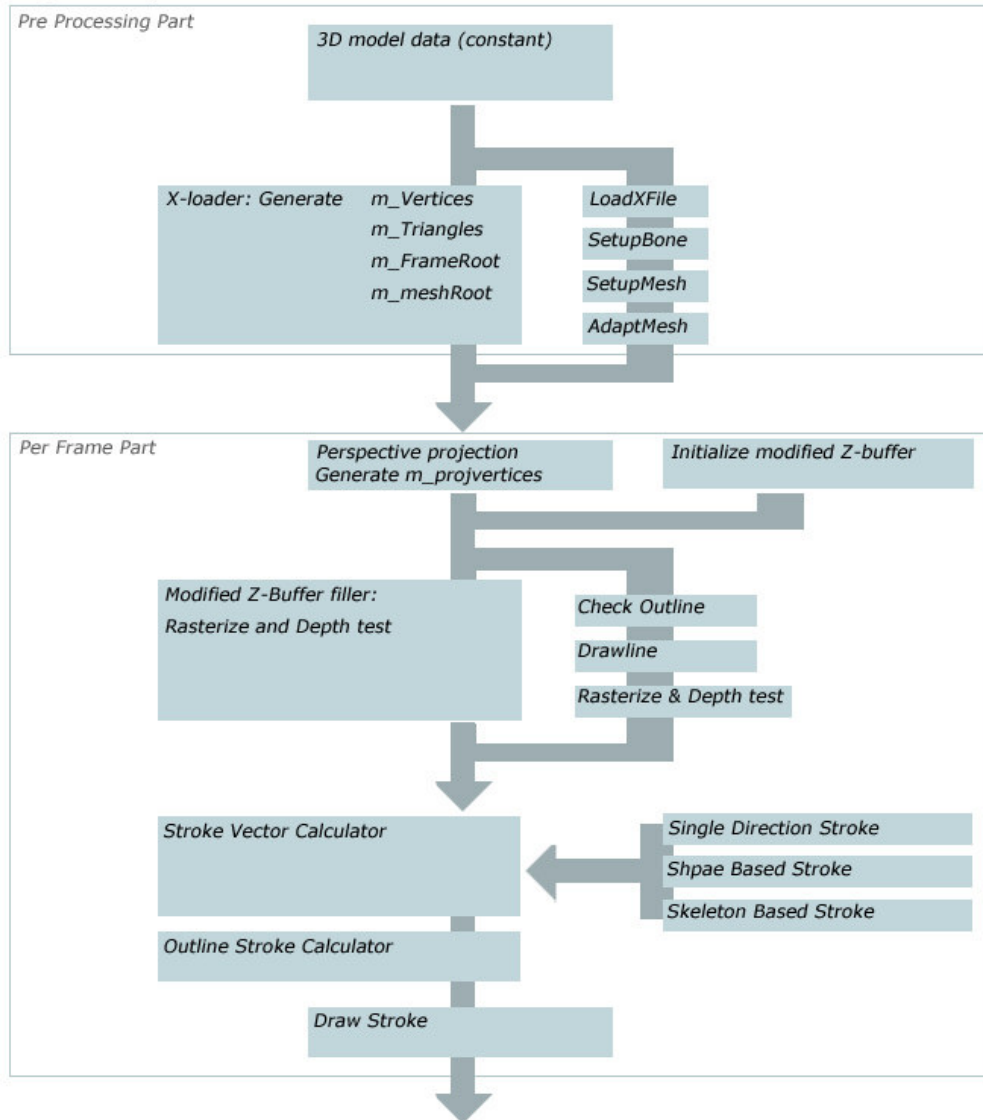
(B) Modified Z-Buffer: Rasterizing & Depth test – Per Frame part

1. 判別線段資訊。填入需要繪製的線段
2. 初始化 modified Z-buffer，並將 3D-model 的浮點資訊光柵化並填入 Z-buffer。
3. 深度測試，消除隱藏線。

(C) Stroke Calculator & Stroke Drawer

1. 選擇不同的筆觸生成方式，生成筆觸的走向。
2. 給定外框線段的權重。
3. 繪出筆觸效果。

Single frame procedure



[圖 12] 系統流程。包含前置處理部份以及單次影格處理程序。

4.2 資料讀取及虛擬畫布

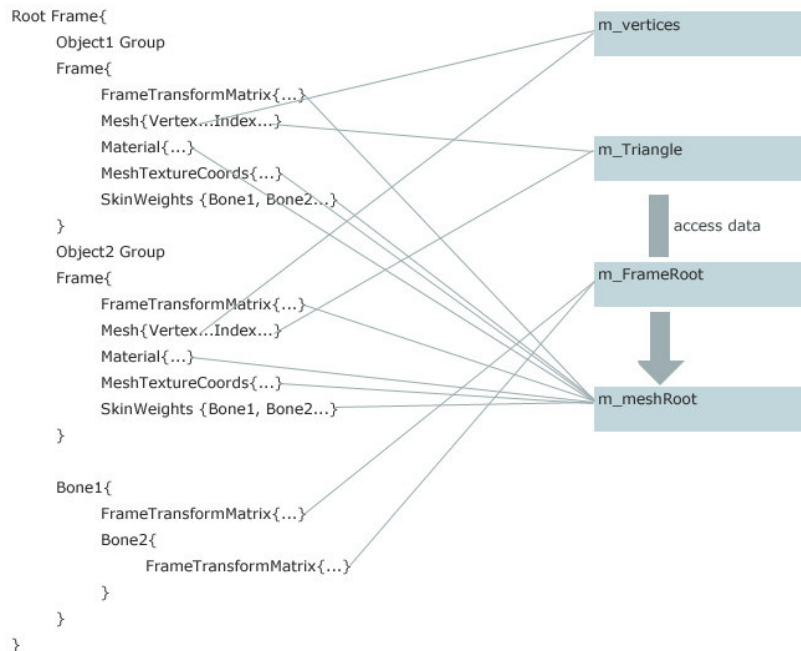
在本系統中，我們選擇使用 Direct3D 的 X-file 作為資料的儲存格式。另外，由於著名建模軟體如 Maya、3DsMax 皆有支援 X-file 的檔案格式，更便利於日後的相關開發。在虛擬畫布方面，我們使用自定的格式來實作一個 Z-buffer，即先前所提的虛擬

畫布。使所有的 Stroke 計算以及生成均可以利用此畫布達到需求的效果。

4.2.1 資料讀取及整理

我們讀取進來的資料格式為 X-file 的資料格式，其優點為具有階層特性的物件關係。由於 X-file 中的頂點資訊為“non-unique”的特性，若沒有經過整理與剔除多餘的頂點的步驟而直接使用，將會大量消耗不必要的系統運算資源。

因此，為了可以讓資料更適於我們的系統使用，所以在前置處理的部分亦增加一個 Data Adapter，用來將 X-file 的檔案資料轉換成我們的系統所需要使用的資料結構。在系統裡，我們將場景中的所有 vertex 全部匯入整理成一份 Vertex List，而將群組關係及 Triangle 特性交由 Triangle List 管理。包含 Index 資訊，群組資訊，骨架資訊等。因此必須將 Input 的 X-file 重新拆解、整理。



[圖 13] Adapter 資料匯入示意圖

(A)Vertices Data :

由各 Object 之 Vertex 資料取得。由於 Maya / 3DsMax 轉換的 X-file 其頂點資訊皆不具有唯一性，也就是說一個簡單的正四面體，卻會出現 $3 \times 4 = 12$ 個頂點資訊。因此，必須先做過一次資料篩選以及建立 Replace Table 更改其 Index 參照資訊之後，才能將具有唯一性的頂點資訊填入 Vertices List。

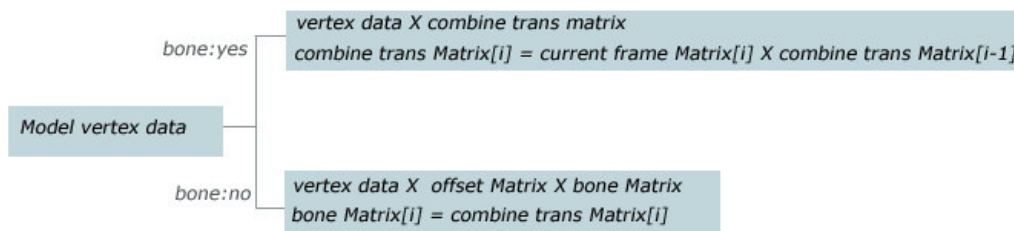
(B)Triangle Data :

根據上面的 Replace Table 以及物件間的 Offset Table 來更新資料資後，輸入到 Index List。

(C)FrameRoot :

儲存 Bone 的階層關係以及 Bone 的轉置矩陣、頭尾資訊等。

此外，當讀取的 X-file 僅有基本的模型資料時，此 X-file Adapter 也負責轉換此種類型的模型資料，模型中 Bone 的有無，將決定 Vertex 所需參照的 Matrix。



[圖 14] 判斷 X-file 中有 Bone 與否

4.2.2 虛擬畫布：Modified Z-buffer

我們使用自定的格式來實作一個 Z-buffer，即先前所提的虛擬畫布。使所有的 stroke 計算以及生成均可以利用此畫布達到需求的效果。

如[圖 15]所示，右側的表為 Buffer Entity，代表每一個螢幕上出現的像素。其中除了最基本的 Z value，也包含了關於此 Pixel 的權重(weight)、經過 3D 場景運算後的 color、以及各種筆觸的走向(vector[i])及相對應的筆觸顏色(strokeColor[i])等。其中的 tPtr 則是用來連結所屬的三角形。透過此一指標，我們可以輕易的取得物體原本在 3D-model 所保留的資訊。

```
class mZbuff {
public:
    mZentry buf[HEIGHT][WIDTH];

    int getSize();
    void clear();

    int pointNum;

    mZbuff();
    ~mZbuff();
};

class mZentry {
public:
    D3DXVECTOR3    vector;
    D3DXVECTOR3    strokeVector[3];

    DWORD          color;
    DWORD          strokeColor[3];

    float          intensity;

    TRIANGLE       *tPtr;
    LPSTR          meshName;

    bool          exist;
    int           type;
    float         Z;
    float         weight;

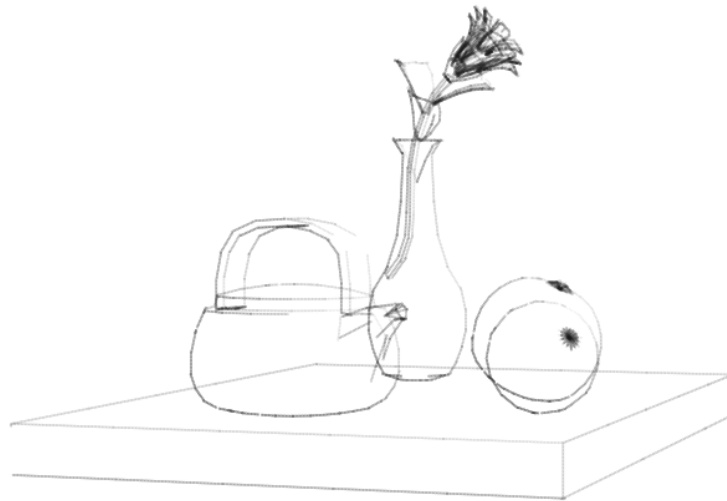
    mZentry();
    ~mZentry();
};
```

[圖 15] 兩個 modified Z-buffer 的主要資料結構

由於我們的參考物件為實體 3D 模型，因此整個物件的外框線(Outline)將根據生成方式區分為三種：Silhouettes, Boundaries, Discontinuities。

- (A) Silhouettes：當線段的相鄰兩三角形其中一個三角形法向量與視角夾角小於 90 度，另外一個大於 90 度，即代表這條線處於一個可視的三角形與一個不可視(culled)的三角形中間。此種線段稱為 Silhouettes。
- (B) Boundaries：當線段只有一個相鄰三角形時，即代表這條線在模型的邊緣，此種線段稱為 Boundaries。
- (C) Discontinuities：當線段的相鄰兩三角形法向量夾角大於某個固定角度時，即判斷此線段需要繪出，稱為 Discontinuities。

然而，藉由上述三項線段結合所產生的輪廓線，會發生從被遮蔽的物體後方透出來的情形[圖 16]，因此需要將這些「應該被前物遮蔽的線條」從線段中剔除，稱為 Hidden-Line Elimination。

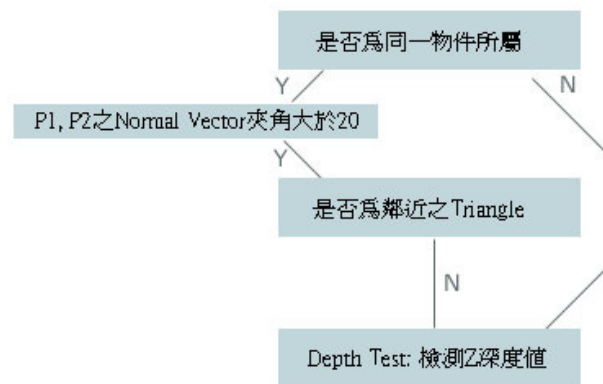


[圖 16] 綜合三種線段產生出來的 Output

為了消除這些線段，我們採用較為普遍的 Z-Depth Test 實作。對於場景中所有物件的 Triangle，利用 Rasterization，我們依序將其填入虛擬畫布(Modified Z-buffer)中，同時在每一個 Pixel 填入時，藉由比對 Z-buffer 中每一個填色物件的深度(Z value)，來決定最後此 Buffer 將歸於哪一個物件。

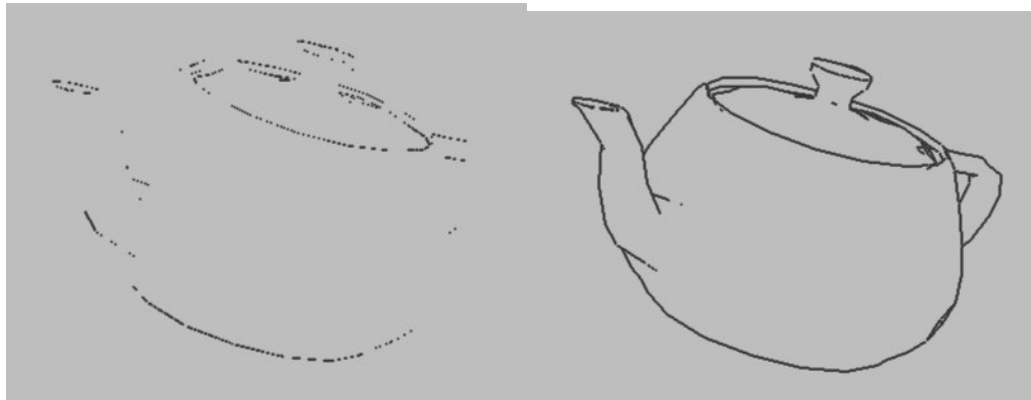
然而此一方法在實作時，對於我們的系統卻出現一個問題。我們將 modified Z-Buffer 單元分為三類:Empty、Texel、Edge。Empty 代表沒有被任何物件填色的 Pixel；Texel 代表內圖色塊；Edge 則代表物件的 Outline。傳統 Z-Buffer 在使用時，考慮到的乃單純物件 pixel 的深度，並不需要考慮內圖色塊(Texel)與邊緣線條(Edge)的分別。因此，當物件為“高複雜度、圓弧狀表面”時，便會出現問題。

當物件的平滑表面變成輪廓線時，由於過於平滑，用來判斷是否為 Outline 的 3D-model 邊緣會被自己的內圖色塊遮蔽(Z-Value 判斷後，texel 會遮蔽 Outline)。導致部份的 Outline 消失。為了解決這個問題。我們改進了 Depth-Test 的檢測方法。基本檢測流程如下：

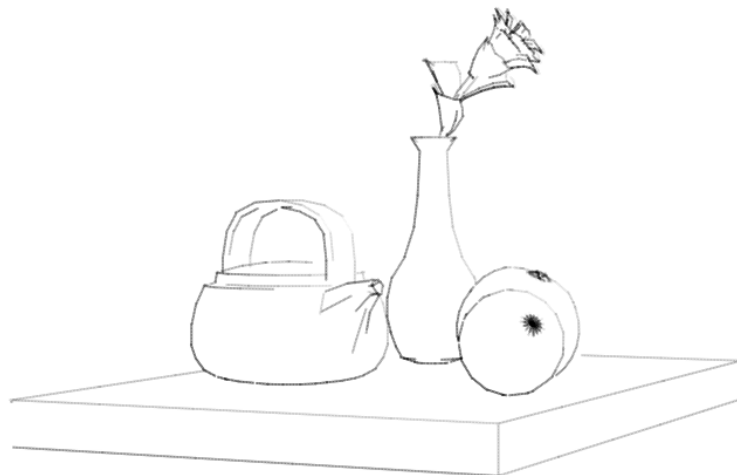


[圖 17] Adapt Depth Test 流程圖

- (A) 若 P1, P2 分別屬於不同的物件，則直接作深度測試。
- (B) 反之，則判斷 P1, P2 點上的 NormalVector(由 3Dmodel 取得)之夾角，若超過一預設之門檻(20 度)則繼續判定條件(C)
- (C) P1, P2 若彼此為非相鄰之 Triangle，則作深度測試。



[圖 18] Depth-Test 前後結果圖。左圖為套用傳統的 Z-Depth 的結果，Outline 被刪除的情形相當嚴重。右圖為改進 Depth-Test 後的結果，Outline 正常顯示。



[圖 19] 執行完 Hidden Line Elimination 後的結果圖

第五章 Shading 與光

5.1 shading

在 NPR 的影像生成中，所有的像素顏色僅僅擔任原始的畫面參考色，由這些參考色經過混色以及筆觸生成之後才生成最終的輸出影像。即是說，對 model 上顏色精細度的要求較低。但是，為了避免在大面積的水彩渲染塗色上造成面與面之間有接縫的問題出現，所以我們改善了早期的 shading 方式，將早期使用的 Flat Shading 改成 Gouraud Shading。

Flat Shading 是根據每個面的法向量計算顏色，然後均一的塗在整個面上；Gouraud Shading 會去計算每個頂點的法向量，然後計算每個頂點應塗的顏色，再使用沿線內插顏色的方式，得到面上每一個 pixel 的顏色。雖然 Gouraud Shading 比 Flat Shading 較花時間，但是形成的影像較為真實而且也改善了面與面之間接縫的問題。



[圖 20]使用不同 shading 方式所產生的影像。上圖為使用 Flat Shading 的茶壺。
下圖為使用 Gouraud Shading 的茶壺。

5.2 光

明暗度是水彩繪畫上一個重要的環節。在繪圖時常需要考慮物體上的明暗層次，以便做出物體的立體感。所以，為了使我們的水彩模擬更加豐富，我們參考了場景中給定的照明條件，在物體上產生 specular 的效果。Phong Shading 是將 Gouraud Shading 加以改進，是一種可以在物體上顯示出 specular 效果的 shading 方式，但是由於 Phong Shading 需要較長的運算時間，所以我們將 Phong Shading 的觀念帶入到 Gouraud Shading 中，最後，利用 Phong's reflection model 來計算出每一個 pixel 的顏色。

Phong Shading 是利用三角形頂點上的法向量去內插計算出表面上每一個 pixel 的方向向量，來得到每一個 pixel 的亮度。而在我們原本的 shading 中，我們增加對於三角形三個頂點上的亮度計算，再去做 Gouraud Shading 的計算，如此我們可以利用較少的時間得到類似 Phong Shading 的 specular 效果。以下[圖 21]即我們得到具有 specular 效果的影像：



[圖 21] 具有 specular 效果的影像

第六章 水彩模擬

6.1 Simulation Overview

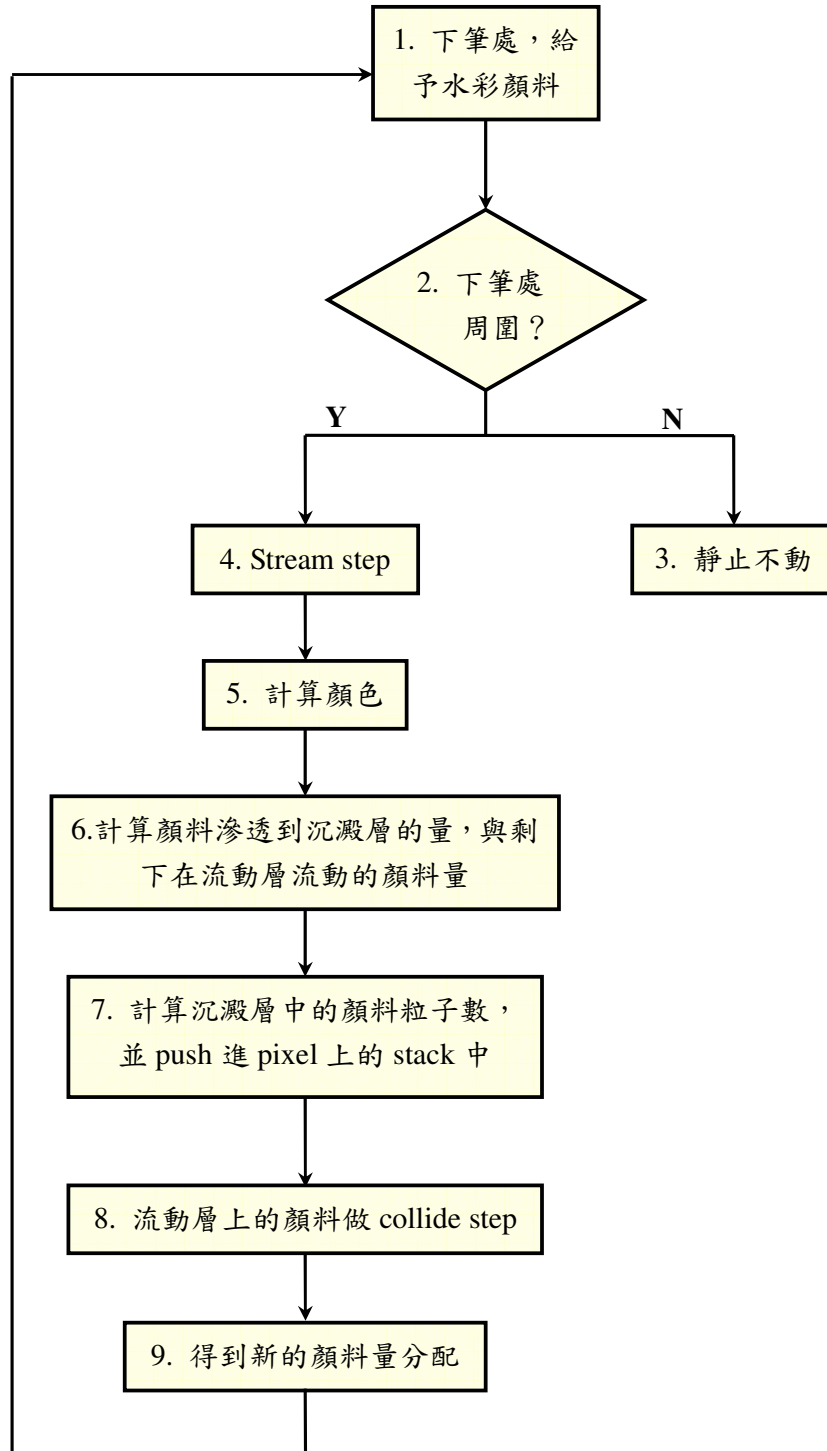
本系統主要是利用我們早期所發展的 3D model-based NPR-System 為基礎，藉由流體力學中 Lattice Boltzmann Method(LBM)的導入，建構出水的模型，用來模擬水在畫紙上流動、擴散與其他的一些物理現象。然後，以這個水模型為基礎依照不同的水彩濃度，來產生出水彩多樣的色彩色調與渲染效果，進而模擬出水彩的繪畫風格。

在研究中，我們將虛擬畫布分成兩層，分別為「流動層」與「沉澱層」。一開始，水彩畫在虛擬畫布上，然後因為水彩中水的特性，水彩會在流動層中流動與擴散，漸漸的，水彩粒子會慢慢的在沉澱層中沉澱與累積。

所以，在我們的研究主要有兩個模擬部份，分別為水的擴散與顏料粒子的沉澱。

6.2 水彩模擬流程圖

水的擴散，是隨著時間的遞增向外擴散的，所以每一個時間(t)的運動都必須參考上一時間($t-1$)的擴散結果再來做向外擴散的運動，之後再依此運作不斷擴散出去，而顏料粒子是溶解在水中的，所以顏料粒子會隨著水在畫紙上流動並沉澱。[圖 22]為本研究的水彩模擬流程圖：



[圖 22] 水彩模擬流程圖

以下將說明水彩模擬流程圖的各個步驟：

1. 下筆處，給予水彩顏料

每一次都去判斷，筆觸落筆會落在哪些 pixel 上，並給予落筆處的 pixel 水彩顏料來源。

2. 下筆處周圍？

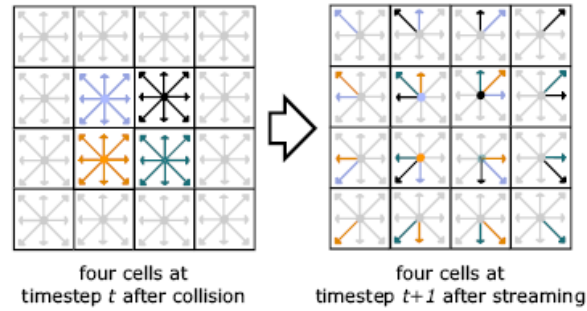
用來判斷 pixel 是否在下筆的筆觸周圍。因為我們假定當水彩剛畫在畫布上時，並不會馬上滲透進畫布中，水彩中水的流動特性，會對周圍的 pixel 造成某個程度上的影響。例如：以下筆處為中心，給一個 10*10 的區塊，則在這 10*10 區塊上若 pixel 上有水彩顏料便會作水的擴散模擬。

3. 靜止不動

若該 pixel 不在下筆的筆觸周圍，則此 pixel 不受影響，pixel 不會得到任何顏料且 pixel 上的顏料靜止不動。

4. Stream step

若屬於剛下筆的 pixel 或其周圍的 pixel，此時 pixel 會做 LBM 的 stream step simulation。將 pixel 中的 distribution functions 複製到相鄰 pixel 的相對向量上，再將 pixel 上每一向量上的數量加總起來，便會得到這個時間點上此 pixel 所有的顏料數了。



[圖 23] stream step simulation 前後圖。右圖為做完 stream step simulation 後的分配圖，再將每一 pixel 向量上的數量加總起來，便是此次的模擬所得到的顏料量了

5. 計算顏色

做完 stream step，pixel 需要計算從四面八方得到的顏料所混合出來的顏色。更詳細的顏色計算將於下一節 6.3 小節做介紹。

6. 計算顏料滲透到沉澱層的量，與剩下在流動層流動的顏料量

將虛擬畫布分為流動層與沉澱層，根據經驗法則，當顏料粒子越多時則沉澱越多，反之，則沉澱越少。當粒子沉澱在畫布上的越少，留在流動層上的粒子越多，顏料粒子會隨著水的擴散被帶的越遠，形成的筆觸顏色較淺且較不明顯；反之則顏色較深且筆觸明顯。目前在我們的系統中，給定一個使用者參數用來表示粒子沉澱的比例。



[圖 24]不同沉澱比例所造成的筆觸樣式。(a)1/100 (b)1/50 (c)1/10

7. 計算沉澱層中的顏料粒子數，並 push 進 pixel 上的 stack 中

對於每一次顏料的沉澱，我們認為顏料在畫布上是一次、一次由下而上累積的，所以我們提供虛擬畫布上的每一個 pixel 一個 stack 資料結構，用來存放每一次沉澱的顏料粒子。而 stack 「先進後出」的特性也符合了最先畫的顏料會在最下層，而最後畫的顏料在最上層的性质。

8. 流動層上的顏料做 collide step

在 collide step 會使用流動層上所剩的顏料來做計算，利用 pixel 上的 distribution functions 計算出 ρ 與 \bar{u} ，並代以下二個式子

$$f_i^{(0)} = w_i \left\{ \rho + \rho_0 \left[3(\bar{e}_i \cdot \bar{u}) + \frac{9}{2}(\bar{e}_i \cdot \bar{u})^2 - \frac{3}{2}(\bar{u} \cdot \bar{u}) \right] \right\}$$

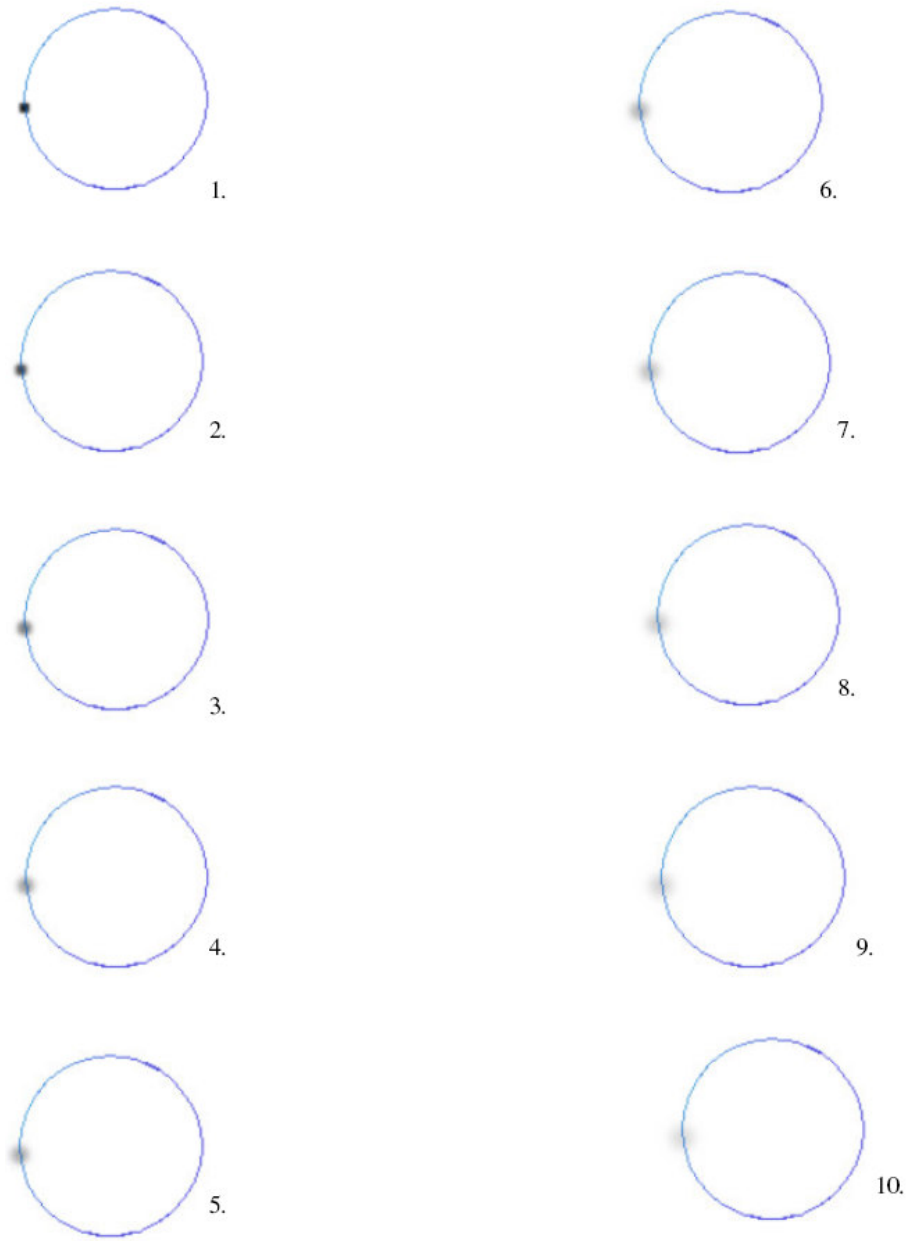
$$f_i' = (1 - \omega)f_i + \omega f_i^{(0)}$$

便可以得到新的顏料粒子分配。

9. 得到新的顏料量分配

得到了新的顏料粒分配後，就儲存下來，以便下一時間的擴散使用。

之後，每當有水彩顏料畫在畫布上，就執行這個水彩模擬流程，直到最後一筆筆觸畫完。



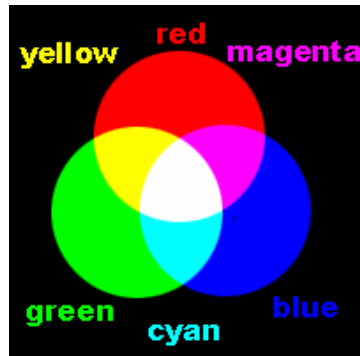
[圖 25]一滴水在虛擬畫布上的擴散過程



[圖 26]一筆水彩筆觸的繪畫過程模

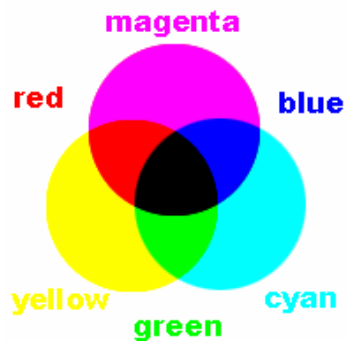
6.3 color blending

電腦裡與螢幕都是使用光的三原色(RGB)來標示顏色，也就是紅(red)、綠(green)、藍(blue)。RGB 使用的是「加色法」，三種顏色等量相加可以得到白色，兩兩等量相加可以得到中間色：黃(yellow)、青(cyan)、洋紅(magenta)。



[圖 27]RGB 色彩系統

但是在水彩的色彩系統中是採用「減色法」，三原色相加會得到黑色，這與 RGB 所採用的加色法不同，於是我們的系統使用了印刷上的 CMYK 系統，也就是青、洋紅、黃、黑。事實上除了黑色外，用顏料三原色無法混和出許多暗色系的顏色，為了彌補這個缺點，因此實際印刷的時候會額外加入黑色的顏料，以解決無法產生黑色的問題。因此就有所謂 CMYK 的色彩模式，K 表示黑色。



[圖 28]CMYK 色彩系統

因為顏料與 RGB 所採的色彩模式不同，所以我們在系統運作中必須要做色彩轉換的步驟。

首先，將 3D-model 載入到我們的系統中，得到的像素色是採用 RGB 計算的，但是因為水彩的顏料色彩模式與 RGB 不同，所以我們必須把得到的 RGB 值轉換成 CMYK 值，之後，在我們的系統中所有的色彩計算，都採用 CMYK 模式來做運算，等到最後要輸出到螢幕時，再將最終得到的 CMYK 值轉換成 RGB 作輸出，如此便可以在螢幕上得到我們水彩模擬的成果了。以下[圖 29]為利用 CMYK 三原色(青、洋紅、黃)兩兩相混所得到的二次色：



[圖 29]CMYK 三原色兩兩混色圖。(a)洋紅加青色得到藍色(b)青色加黃色得到綠色(c)洋紅加黃色得到紅色

而在筆觸的色彩上，因為我們改善了之前研究所用的 shading 方式，所以我們的 model 都有依照場景光源位置而反映出明暗效果，因為筆觸色彩是參考 model 上的顏色得來的，而且每一條筆觸是由多個 pixel 組成，所以在筆觸的色彩生成上，可以產生具有明暗效果的筆觸。這個方法可以不需要使用者的指定輸入，我們就可以自動化的產生具有明暗效果的筆觸。

第七章 筆觸模擬與動畫生成

7.1 筆觸模擬

為了能使生成的影像更接近於普通的水彩繪畫，我們歸納出兩大類重要的筆觸，這些筆觸各自有著不同的走向特性、不同的粗細特性等。

(A) 2D shape-based Stroke: edge

(B) 3D model-based Stroke: skeleton

(a) Feature- based Stroke

(b) Texture- based Stroke

因為本篇論文的重點不在討論筆觸的種類上，而且我們早期的研究在 2D shape-based Stroke 與 3D model-based Stroke 的開發上已有不錯的結果，所以我們將直接套用早期的研究結果。而在未來工作中，我們將把 Feature- based Stroke 與 Texture-based Stroke 整合到本篇論文中。

7.1.1 2D shape-based Stroke

在繪圖的時候，有時我們需要額外的線條來強調物體本身 2D 的形狀。大致來說，此種繪線常會隨著物件本身的 Outline 起伏、生成。為了能夠模擬出具有此種特性的繪線。首先我們必須決定存在於虛擬畫布上的每一個 pixel，分別受到哪些 outline pixel 的影響。

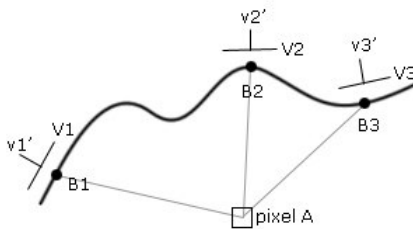
在處理這個步驟上，我們首先將所有的 outline pixel 以各自所屬的物件分類之，以加速影響筆觸的搜尋。同時，這個分類的步驟也可以避免不同物件間，不必要的筆觸相互影響。故，對於虛擬畫布上的每一個存在的 pixel，我們反覆執行以下的步驟：

(A) 搜尋所有影響到該點的 outline pixel。

(B) 給定一 threshold，判斷每一個 outline pixel 到該 pixel 的長度。若小於 threshold 則予以保留為 intensity pixel。

(C) 根據篩選出來的 intensity pixel，作最後加總平均的計算。

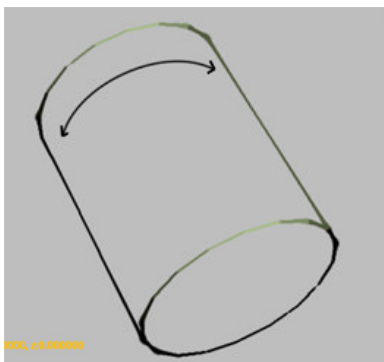
所有的 outline vector 都是雙向的。若直接將所有找出的 intensity pixel 直接加總，則無法統一所有筆觸的方向。因此，相較於加總所有 intensity pixel 上的方向向量，我們給定一個向量位於 intensity pixel 且垂直於該 pixel 上的方向向量，然後加總這個向量。如[圖 30]所示。



[圖 30] 2D shape-based stroke vector 存取示意圖。Pixel A 為運算中的向量；B1, B2, B3 為範圍內的 Intensity Pixel；相較於加總 V_1, V_2, V_3 ，我們加總 V_1', V_2', V_3' 三個可以確定其方向的向量。因為 V' 的方向可以根據作用中的 pixel 與其之間的夾角決定之。

7.1.2 3D model-based Stroke

當物件形體具有強烈的形狀特徵時，我們會依據模型本身的三維結構來給定其筆觸方向。

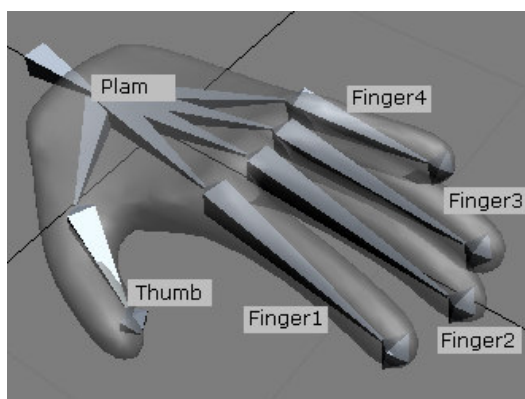


[圖 31] 柱狀體上的筆觸方向，為建模軟體中的基本型體之一。通常我們使用如圖上所示的方向筆觸來加強觀者對物件型體的認知。

在傳統的繪畫中，物件的形體表現一直是重要的一環。而筆觸在這個環節所扮演的角色，往往是用來強調出物件的三維特性，讓觀者能由 2D 的繪畫在腦海中重建出物件原始的 3D 型態。即是說，筆觸往往能夠幫助觀者更正確的認知一幅畫中的物件。

為了模擬出這類型的筆觸，我們採用模型本身具有的骨架結構。從模型中抽取出的模型建構法，由目前市面上較著名的建模軟體，如 3DS-MAX、MAYA 等所提供的骨架結構抽取之。

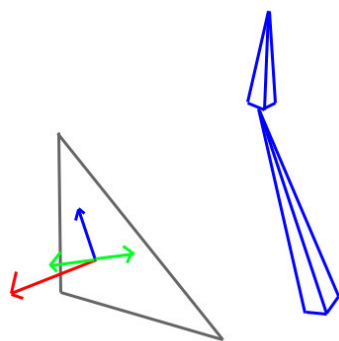
基本的骨架架構，由 Parent/Child 的階層結構組成。如[圖 32]所示，palm 為 thumb 以及其餘四個 finger 的父階層。而每一個 Bone 會有其影響的範圍、權重以及轉置矩陣，骨架依據這些可取得的資訊來控制模型的動作。



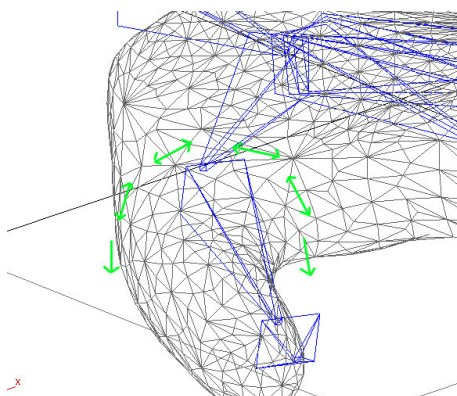
[圖 32] 手掌的骨架階層構造

我們從中取得骨架的相關資訊，即其個別的階層結構、範圍、權重以及轉置矩陣。以這些資訊來運算系統中的筆觸走向。

至此，根據每個 Bone 的影響範圍，我們可以讓每一個 Triangle 結構獲得一個 Bone 可供參考方向。外積此 Bone 方向向量與 Triangle 的法向量之後，可得到一個三維雙向性的筆觸方向，有了三維的方向向量之後，我們再對其作一次平面投影的矩陣運算後便可以獲得我們在 Buffer 上需要的二維方向向量。如[圖 33]、[圖 34]所示。



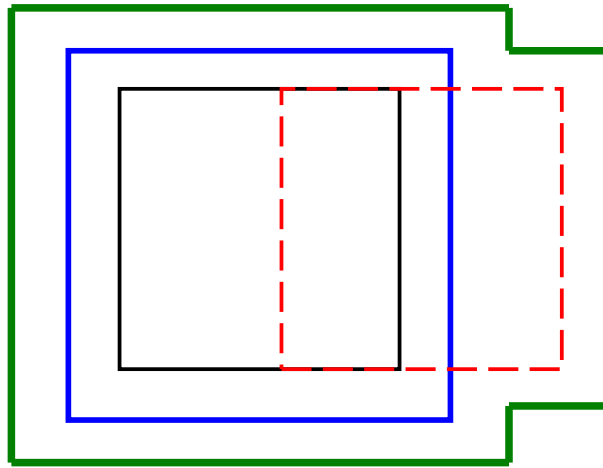
[圖 33] 3D model-based stroke vector 存取示意圖。藍色箭頭代表骨架方向；紅色箭頭為 Triangle 法向量。綠色為兩者外積後的三維筆觸向量。



[圖 34] 手掌上，Thumb 部分的三維筆觸分布。

7.2 描繪過程

如第六章所提，筆觸的生成是在每個時間 t 中會有一個新的筆觸位置下筆，而且同一筆筆觸的下筆都是具有連續性與相關性的。所以，必須要考慮到下筆處周圍 pixel 的擴散現象。例如：當筆觸的第一個下筆處下筆會擴散一次，隨著時間遞增，當第二個下筆處下筆，第一個下筆處應該要擴散第二次，而第二個下筆處擴散第一次，此後，依此類推。如[圖 35]所示。

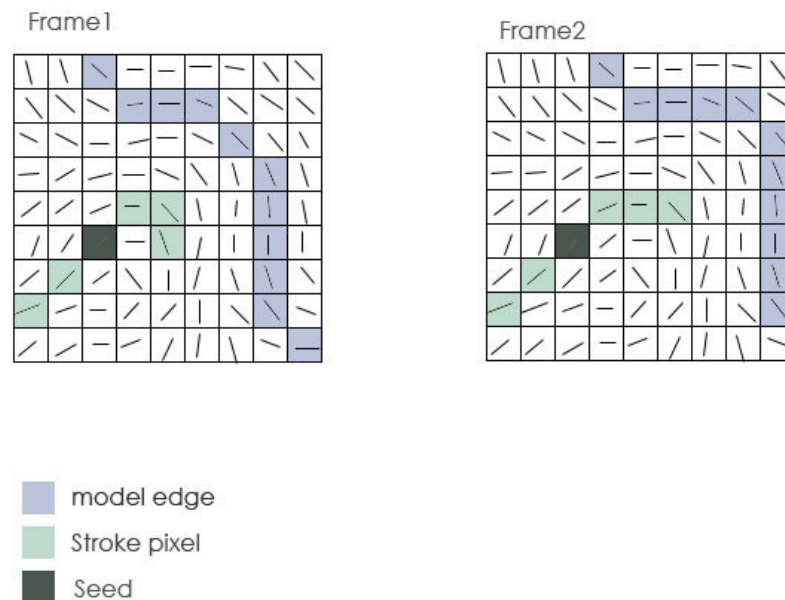


[圖 35]水彩描繪過程示意圖。用黑色框來表示第一個下筆處，第一個下筆處會擴散成為藍色框。隨後，用紅色框來表示第二個下筆處加入，而藍色框會與紅色框一起擴散成為綠色框。

7.3 動畫生成

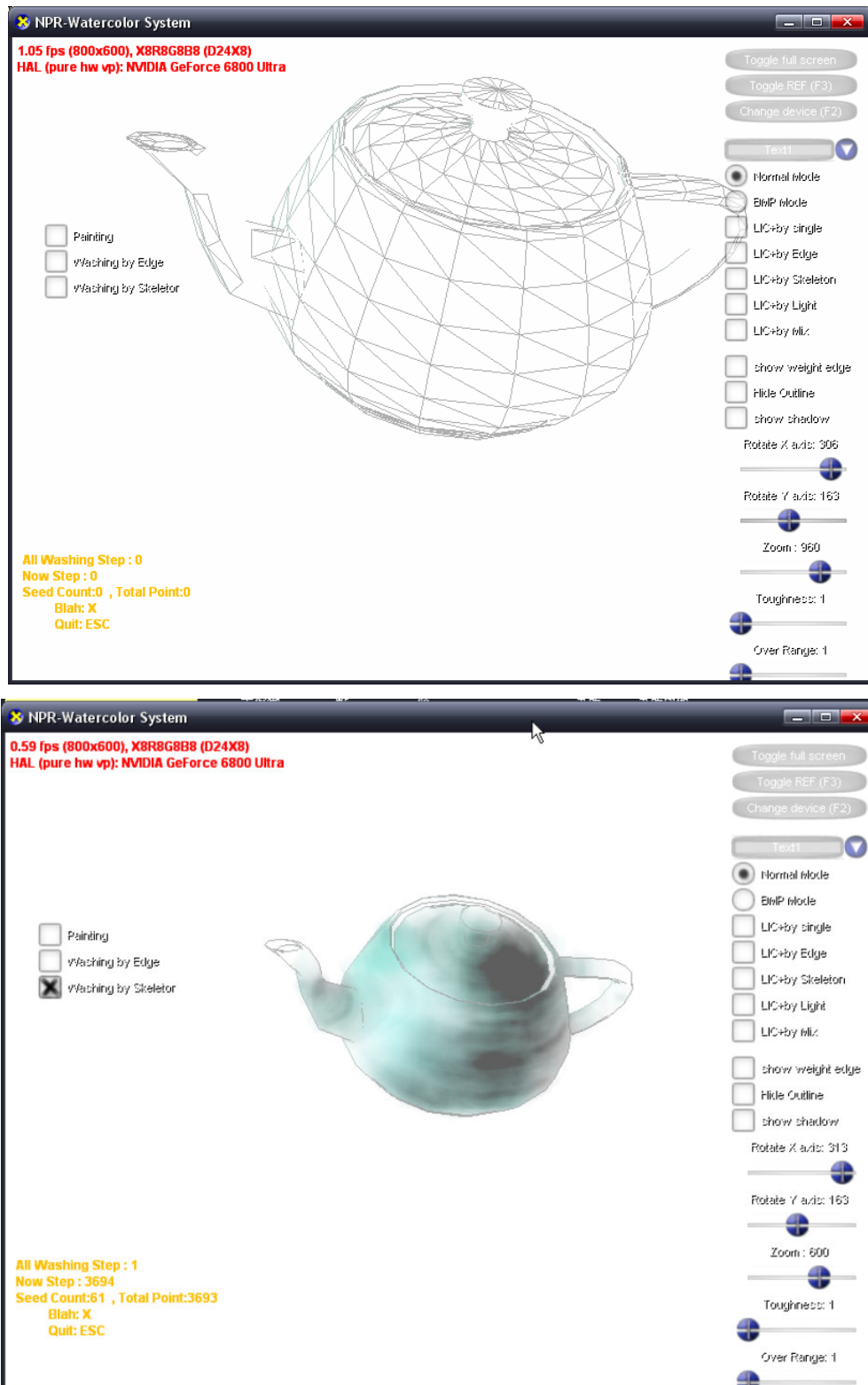
簡單的動畫由連續的影像產生。動畫中的物件，通常一定有物件是會移動以及形變的。如同現今的卡通動畫一樣，分為前景及背景。由於在每張影格，前景都會有所變動，故前景的部分通常由簡單的色塊構成，以減少動畫師的負擔。而在我們的研究中，產生動畫最重要的課題在於如何使筆觸生成連續。

本研究採用「以投影平面為基礎的筆觸連續」：視窗平面上的每一條筆觸，在 Frame 與 Frame 之間固定其生成位置。為了讓筆觸能夠連續，我們使一開始灑出的 Seed 固定，之後在下一個 Frame 以這些已經固定的 Seed 為優先生長對象，而因為物件的位移而超出筆觸生長範圍的 Seed 則刪除之。同理，新增加的區域則另外灑上新的 Seed 並紀錄其位置。



[圖 36] 投影平面為基礎的筆觸連續示意圖。從 frame1 到 frame2，model edge 向右橫移了一個 pixel，但是筆觸的 Seed 位置不變，pixel 上的斜線為我們運算出來的筆觸向量。

第八章 研究成果



[圖 37]系統模擬畫面。上圖為系統讀入的 3D-model，下圖為經過本水彩模擬系統計算後的水彩影像。



[圖 38]連續的 Teapot 影格



[圖 39] Teapot-描繪過程:3D model-based Stroke



[圖 40] [圖 39]之第 1 個圖放大



[圖 41] [圖 39]之第 2 個圖放大



[圖 42] [圖 39]之第 3 個圖放大



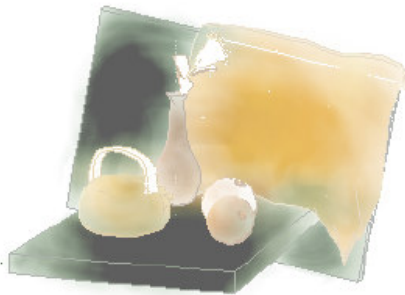
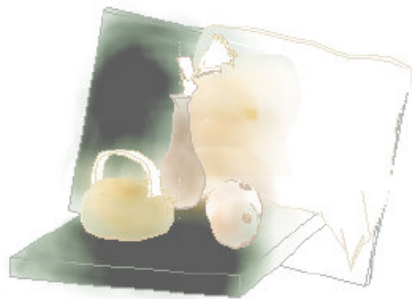
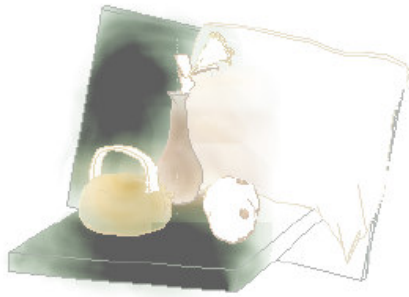
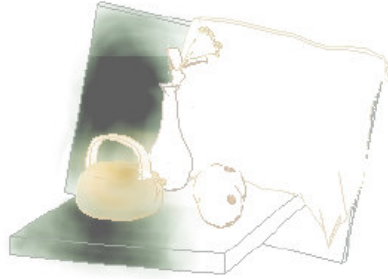
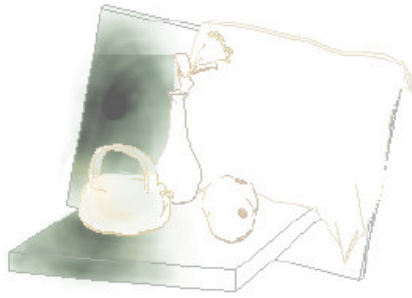
[圖 43] [圖 39]之第 4 個圖放大



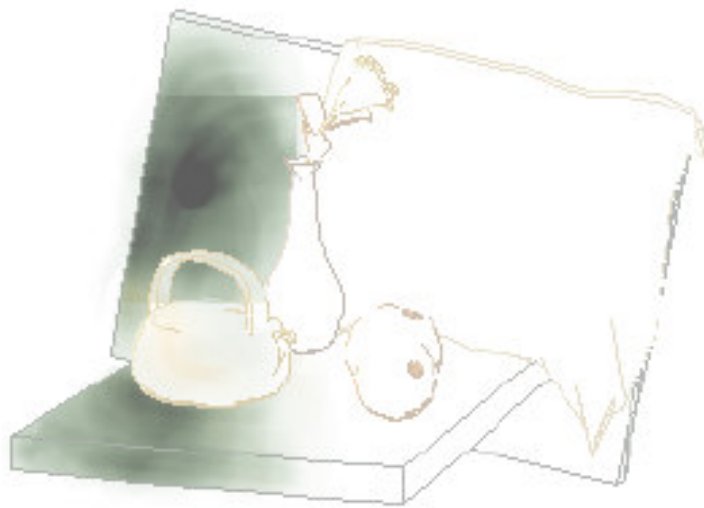
[圖 44] [圖 39]之第 5 個圖放大



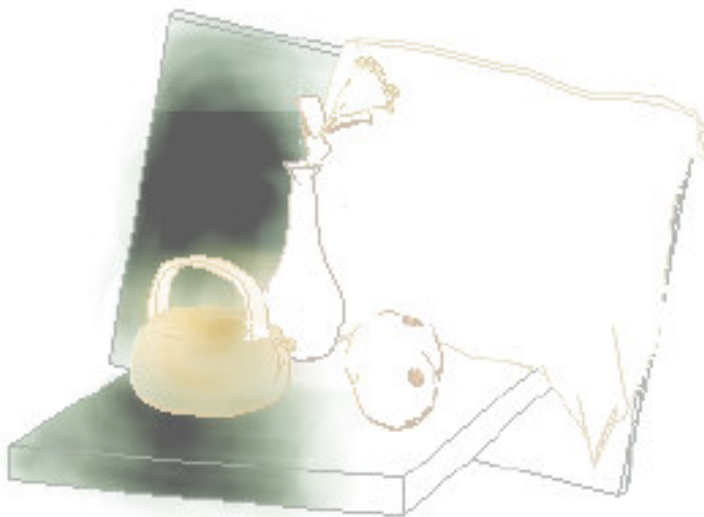
[圖 45] Teapot: 2D shape-based Stroke



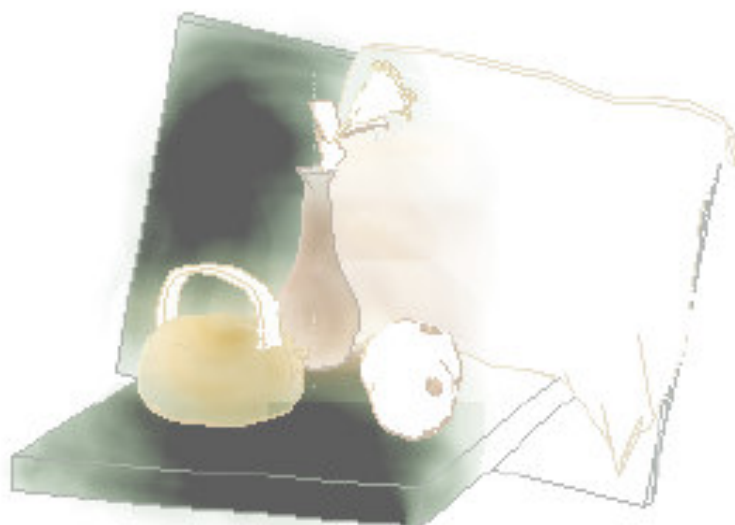
[圖 46]Scene-描繪過程:2D shape-based Stroke



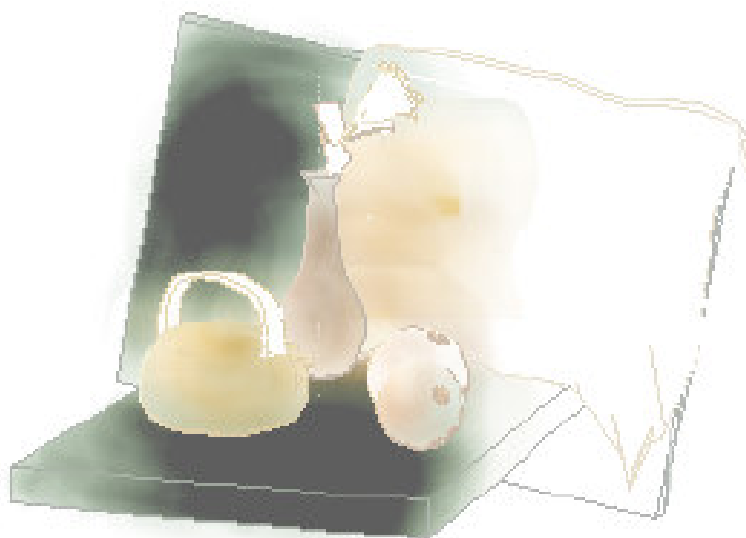
[圖 47] [圖 46]之第 1 個圖放大



[圖 48] [圖 46]之第 2 個圖放大



[圖 49] [圖 46]之第 3 個圖放大



[圖 50] [圖 46]之第 4 個圖放大



[圖 51] [圖 46]之第 5 個圖放大



[圖 52] 不同角度的 Scene:2D shape-based



[圖 53] Flower:2D shape-based Stroke

更多的水彩風格影像與動畫，請參考實驗室網站

<http://cgda.csie.ncnu.edu.tw/gallery.htm>

第九章 結論以及未來工作

9.1 結論

在本篇論文中，我們實做了一個 NPR 水彩系統。我們載入 3D-model 當作原始資訊，並根據 model 與場景資料，能夠自動化的產生具有水彩繪畫風格的影像或動畫。

首先，從水彩的特性與技法來做分析與量化，希望能夠由水彩的本質：「色彩」、「綜合」與「光」為基礎，模擬出水彩繪畫風格。應用 Lattice Boltzmann Method 模擬流體，並探討水彩濃度與不同顏色顏料混合所造成色彩上的差異，在水彩技法方面，本篇論文模擬出了水彩的渲染效果。

在本篇論文中，提出了一個能夠自動產生具有明暗度筆觸的方法。此方法根據場景資訊並利用 Phong's reflection model 來計算出 model 的顏色，我們的筆觸顏色是參考 model 上的顏色得來的，而每一條筆觸由多個 pixel 構成，隨著 pixel 的移動，在同一筆筆觸上就有了明暗度的變化。本研究還探討了二種水彩筆觸繪畫手法，我們也從早期的研究中得到這兩種筆觸，分別為 2D shape-based Stroke 與 3D model-based Stroke。

由我們的研究，或許無法造出可以取代傳統藝術畫作的成品，然而若能將此近似的水彩成像手法以電腦演算，並以連續的繪圖以動畫呈現，將可以在動畫成像的技術上，作出一種自動化且突破性的進展。

9.2 未來工作

本篇研究藉由Lattice Boltzmann Method來模擬流體在虛擬畫布上的流動，並由Modified Z-buffer與3D-model based的許多概念進行整合。目前只有針對兩種筆觸的走向以及特性做處理，在畫布上個別位置的溼度差異與光所造成的陰影效果，都尚欠缺考慮，另外，對於跟使用者互動的介面也可以加入系統中。

底下列出可在未來改進此系統的幾個方向：

1. 更多筆觸的開發

目前系統只考慮了 2D shape-based Stroke 與 3D model-based Stroke，還有已分析的 Feature-based Stroke 與 Texture-based Stroke 尚未加入本系統，我們相信還有許多筆觸特性是還沒考慮到的，希望藉由本系統，未來模擬出更多的水彩筆觸效果。

2. 畫家作畫流程的再現

目前系統筆觸的 seed 是以亂數灑點的方式生成，而畫家在作畫時，通常有一定的作畫流程，所以未來可以考慮將畫家實際作畫的流程整合進本系統的水彩模擬上。

3. 畫布上多種濕度的筆觸表現

我們的系統目前尚未考慮到畫布上的溼度，但是，在實際繪畫時，畫布不同位置的溼度是有差異的，所以，若能夠在我們系統上加入一個讓使用者自行決定溼度的介面，與一個能夠即時反應畫布上溼度的相對應視窗，讓使用者能夠掌握畫布上的溼度資訊，我們相信模擬出來的水彩風格影像會更有趣。

4. 陰影效果

我們的系統已經模擬了光對於水彩色彩的影響，但在於光所造成的陰影效果還欠缺考慮，所以，我們的系統未來考慮使用如：Stencil Buffer，來解決陰影的問題，包括畫布上的陰影、物體上的陰影與陰影重疊的問題，希望加入陰影後，會使我們的系統更加完整。

5. 使用者界面的開發

我們希望提供一個 3D 的場景，這個 3D 場景可以讓使用者自由的配置那些經由與系統的互動所產生出來的水彩 3D-model，而這些 3D-model 也會因為在場景中擺放位置的不同，而呈現出不一樣的效果，並且這些效果都是遵從水彩畫裡 object 配置位置所應有的表現方式。我們期望能夠提供一個空間讓 user 發展自我的風格。

ACKNOWLEDGMENTS

This research project is partially supported by NSC-94-2213-E260-021

參考文獻

1. A single-phase free-surface lattice-Boltzmann method. , THUERREY N. ,2003
2. Computer-generated watercolor., C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin. In *SIGGRAPH'97*, 1997.
3. MoXi: Real-Time Ink Dispersion in Absorbent Paper, Nelson S.-H. Chu, Chiew-Lan Tai
4. Interactive Watercolor Animations, ThomasLuft, Oliver Deussen
5. Non-Photorealistic Rendering using Watercolor Inspired Textures and Illumination, Eric B. Lum, Kwan-Liu Ma
6. Real-Time Watercolor Illustrations of Plants Using a Blurred Depth Test, Thomas Luft, Oliver Deussen
7. Real-Time Hatching, Emil Praun, Hugues Hoppe, Matthew Webb, Adam Finkelstein
8. Real-time Watercolor Painting on a Distributed Paper Model, Tom Van Laerhoven, Jori Liesenborgs, Frank Van Reeth
9. Viscous flow computations with the method of lattice Boltzmann equation, Yu D., Mei R., Luo L-S, Shyy W., 2003, Progress in Aerospace Sciences 39,329-367
10. Shape-Oriented Brush Stroke Synthesis in Non-Photorealistic Rendering, 蘭心皓,2006
11. “工程流體力學” page164-169，朱佳仁 著，成陽出版社，2001初版
12. “水彩畫法的奧秘”，謝明錫 著，雄獅圖書股份有限公司，1996初版
13. “繪畫大師Q&A·水彩篇”，David Norman 著，羅若蘋 譯，視傳文化事業有限公司，2005初版