

เอกสารประกอบการเรียน วิชา ANI 211 การขึ้นรูปสามมิติและการออกแบบแอนิเมชัน 1  
สาขาวิชาแอนิเมชัน ประจำวันที่ 4 และ 7 สิงหาคม ภาคการศึกษาที่ 1 ปีการศึกษา 2551  
วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่

---

หัวข้อ: หลักการประมวลผลภาพ  
(Rendering With Maya)

วัตถุประสงค์:

1. ให้นักศึกษาทราบถึงหลักการประมวลผลภาพในรูปแบบต่างๆ (Rendering) ข้อจำกัด ข้อเด่น และข้อด้อยที่นำไปสู่ผลลัพธ์ที่ต้องการ
2. การตั้งค่าในส่วนของ Common Tab วิธีการประมวลผลภาพทั้งแบบที่เป็นภาพนิ่ง (Still Images) และแบบภาพเคลื่อนไหว (Movie) การจัดการไฟล์ Sequence อย่างเป็นระเบียบแบบแผน
3. ทราบถึงความแตกต่างของตัว Renderers แต่ละตัวที่โปรแกรมมีให้ และสามารถเลือกใช้ได้เหมาะสม ตรงตามคุณสมบัติและโอกาสที่ใช้
4. หลักการประมวลผลภาพด้วย Maya Software Renderer การดึงประสิทธิภาพออกมาใช้ให้เกิดประโยชน์สูงสุด
5. หลักการประมวลผลภาพด้วย Maya Hardware Renderer ข้อจำกัดของ Hardware และคุณภาพที่การประมวลผลสามารถรองรับได้
6. หลักการประมวลผลภาพด้วย Mental Ray Renderer การคำนวณแสงตามธรรมชาติ และการสร้าง Photo Realistic Rendering
7. หลักการประมวลผลภาพด้วย Maya Vector Renderer การประมวลผลภาพเชิงสามมิติออกมาเป็นสองมิติ การปรับแต่งค่าคุณลักษณะต่างๆภายใต้หลักการ Vector Based Images



THE UNIVERSITY OF  
CHIANGMAI  
THAILAND

THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY  
DEPARTMENT OF ANIMATION

ARUS KUNKHET  
315, LEVEL 3, ANIMATION DEPARTMENT  
THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY  
THE UNIVERSITY OF CHIANGMAI 50200  
THAILAND

TELEPHONE +66 53 941801 (315)  
FACSIMILE +66 53 893217



## เนื้อหา:


เนื้อหาภายในบทนี้คือเรื่องของ การ Render ส่วนเรื่องของ การใช้มุกกล้องซึ่งมีความสัมพันธ์กันจะอยู่ในบทต่อไป ซึ่งเนื้อหาทั้งสองส่วนนี้มีความเกี่ยวเนื่องกันและมีความสำคัญมาก ในการแปลงงานที่เราสร้างภายในโปรแกรม ให้ออกมาในรูปแบบที่คนภายนอกสามารถรับรู้ได้

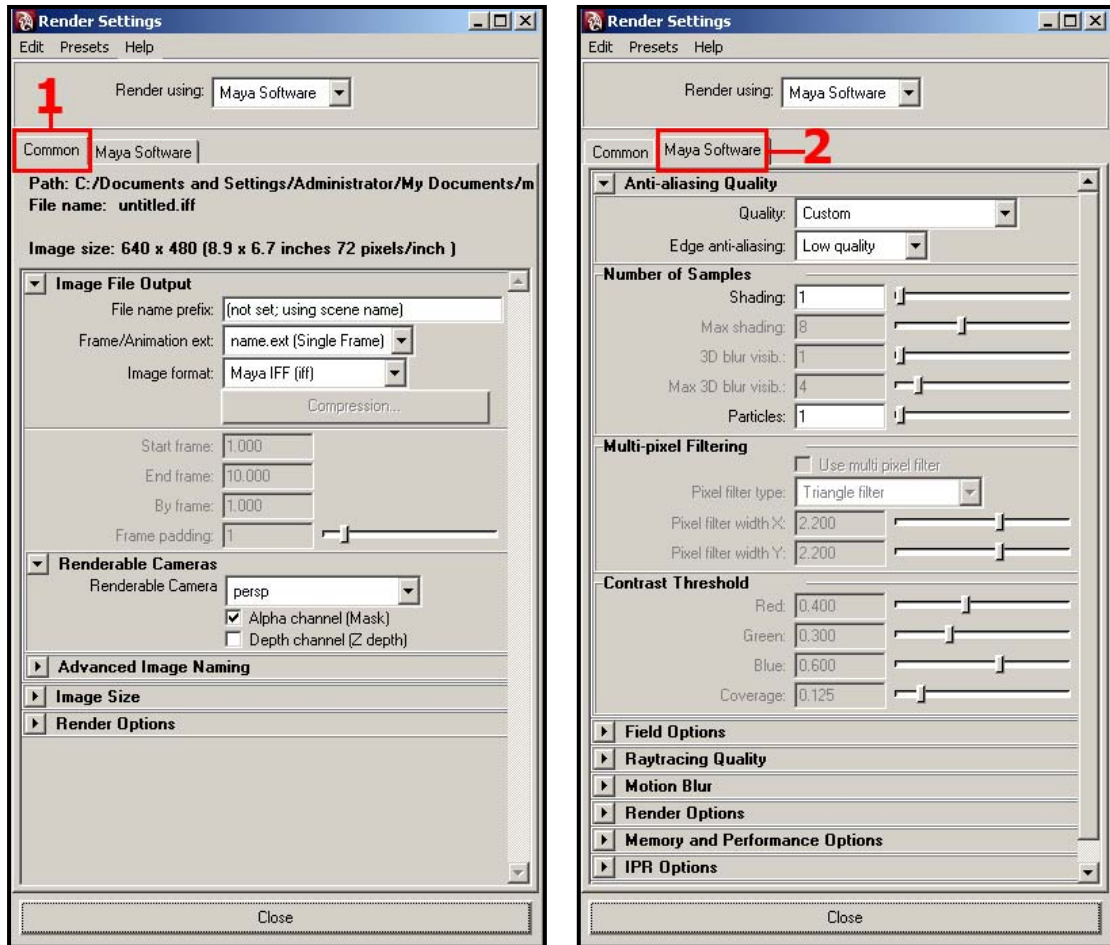
## หลักการประมวลผลภาพ

ภายในโปรแกรมมาจากการที่เราจะได้ผลลัพธ์จากสิ่งที่เราสร้างออกมาเป็นภาพนิ่ง หรือภาพเคลื่อนไหว จำเป็นต้องผ่านกระบวนการประมวลผล หรือที่เรียกว่า Rendering การประมวลผลนี้คือการที่โปรแกรมคำนวณองค์ประกอบทุกอย่างภายในฉาก ตั้งแต่รูปทรงของวัตถุ ลักษณะพื้นผิว แหล่งกำเนิดแสง ทิศทางที่แสงกระทำกับองค์ประกอบต่างๆ เป็นการคำนวณในเชิงสามมิติหรือสามแกน เพื่อประมวลผลออกมาเป็นภาพสองมิตินั่นเอง

การประมวลผลภาพด้วยมายานั้น มีตัวประมวลที่สามารถใช้ได้อยู่ทั้งหมด 4 ตัวประมวลผลด้วยกัน เริ่มจาก Maya Hardware, Maya Software, Mental Ray และ Maya Vector โดยที่แต่ละตัวประมวลจะมีคุณสมบัติและหลักในการประมวลแตกต่างกัน ขึ้นอยู่กับสถานการณ์ที่ใช้ ว่าผู้ใช้ต้องการผลลัพธ์ในรูปแบบใด และที่สำคัญการใส่ลูกเล่นหรือ Effects แต่ละชนิด จะสามารถแสดงผลได้ในรูปแบบการประมวลเฉพาะตัว (ถ้าใช้การ Render ผิดประเภท Effects นั้นจะไม่แสดงผลออกมา) ดังนั้นจึงมีความจำเป็นที่ผู้ใช้จะต้องทราบถึงรายละเอียดของการประมวลผลแต่ละตัว จึงสามารถเลือกใช้ได้อย่างเกิดประสิทธิภาพสูงสุด

## การประมวลผลภาพในรูปแบบต่างๆ

หน้าต่างที่ใช้ในการประมวลผลของมายาเรียกว่า Render Settings Window สามารถเรียกได้ที่ *Window/Rendering Editors/ Render Settings* หรือ กดไปที่ Icon รูป  จาก Status Line โปรแกรมจะทำการเปิดหน้าต่างการตั้งค่าการ Render ออกมา จากหน้าต่างนี้สามารถแบ่งออกได้เป็นสองส่วนใหญ่ๆ คือส่วนของ Common Render Settings ซึ่งเป็นการตั้งค่าพื้นฐานของการประมวลผล ไม่ว่าเราจะเลือกรูปแบบการประมวลผลอย่างไร Option ในส่วนนี้จะไม่เปลี่ยนแปลง แตกต่างกับส่วนที่สองซึ่งจะมีตัวเลือกการตั้งค่าแตกต่างกันไปตามรูปแบบการประมวลผลที่เลือก คือ Maya Software, Maya Hardware, Mental Ray และ Maya Vector โดยเราสามารถเปลี่ยนระหว่างสองส่วนนี้ได้โดยการคลิกที่ Tab บริเวณด้านบนของหน้าต่าง ดังนั้นก่อนที่เราจะเข้าไปดูในส่วนรูปแบบการประมวลผล นักศึกษามีความจำเป็นต้องเรียนรู้ส่วนของ Common Render Settings ก่อน ดูภาพประกอบที่ 6.1 แสดงหน้าต่าง Common Render Settings ในส่วนของ Common Render Setting และ Maya Software



ภาพประกอบที่ 6.1 แสดงหน้าต่าง Render Settings โดยภาพซ้ายแสดงส่วนของ Common Render Setting ภาพขวาแสดงส่วนของ Maya Software

## การตั้งค่าในส่วนของ Common Render

ในส่วนนี้เราจะสามารถตั้งค่าพื้นฐานต่างๆทั้งหมดของการประมวลผล ไม่ว่าจะเป็นขนาดภาพ ประเภทของไฟล์ มุมมองที่เลือก และรายละเอียดอื่นๆอีกมากมาย โดยสามารถแบ่งออกได้เป็น 4 ส่วนด้วยกัน ซึ่งมีรายละเอียดดังนี้

### Render Details

ส่วนแรกคือส่วนที่อยู่ด้านบนสุดได้ Common Tab (ดูภาพประกอบที่ 6.1 รูปซ้าย) จะเป็นส่วนแสดงรายละเอียดของการประมวลผล ในส่วนนี้ผู้ใช้จะไม่สามารถทำการแก้ไขรายละเอียดใดๆได้ มีไว้เพื่อใช้ตรวจสอบการตั้งค่าหลักๆของการประมวลผลเท่านั้น ในหัวข้อ *Path* จะเป็นจุดที่เมื่อเสร็จสิ้นขั้นตอนของการประมวลผลแล้ว โปรแกรมจะทำการจัดเก็บ File(s) ที่ได้จากการประมวลผลไว้ที่ใด หรือคือที่โปรแกรม Save File(s) ที่ได้จากการ Render ไว้นั่นเอง ถัดมาคือหัวข้อ *File Name* จะบอกว่าไฟล์ที่จัดเก็บจะชื่ออะไรและมี Format อะไร จากภาพตัวอย่างไฟล์ที่ประมวลผลจะมีชื่อว่า Untitled.iff และถัดมาอันที่สามหัวข้อ *Image Size* จะบอกรายละเอียดของไฟล์ภาพที่ประมวลผลออกมาว่ามีขนาดเท่าไร

## Image File Output

ส่วนที่สองนี้จะเป็นส่วนที่ผู้ใช้สามารถกำหนดรายละเอียดของไฟล์ในรูปแบบของการจัดการกับชื่อ และ Format ของไฟล์ เริ่มจากหัวข้อแรก *File Name Prefix* เราสามารถตั้งชื่อของไฟล์ที่จะทำการบันทึกได้จากหัวข้อนี้ โดยที่ถ้าผู้ใช้ไม่ได้ใส่ชื่อใดๆลงไป โปรแกรมจะตั้งชื่อโดยใช้ชื่อเดียวกับชื่อ Scene

หัวข้อที่สองคือ *Frame/ Animation Ext* ส่วนนี้มีหน้าที่อะไร ก่อนอื่นต้องขอทำความเข้าใจก่อนว่าในการประมวลผลภาพออกมาเป็นภาพเคลื่อนไหว (Animation หรือ Movie) สามารถทำได้สองวิธี วิธีแรกคือการ Render ออกมาเป็นไฟล์ภาพยนตร์โดยตรง เช่นไฟล์ในสกุลของ AVI หรือ MOV ข้อดีคือเมื่อเสร็จสิ้นขั้นตอนการ Render แล้ว ผู้ใช้สามารถนำไฟล์ที่ได้ไปใช้ได้ทันที แต่ข้อเสียคือไฟล์จะสูญเสียถ้ามีการหยุดการ Render ลงกลางคันก่อนที่จะเสร็จสมบูรณ์ ทำให้การตรวจสอบระหว่างที่ทำการ Render ทำได้ยาก วิธีที่สองคือการ Render ในรูปแบบของ Image Sequence หรือการ Render ออกมาเป็นไฟล์ภาพต่อเนื่องกันหลายๆภาพ แทนที่จะได้ผลลัพธ์เป็นไฟล์ Movie ไฟล์ใหญ่ๆไฟล์เดียว ก็จะได้ไฟล์ภาพออกมาเท่ากับจำนวนเฟรมทั้งหมดที่เราทำ ข้อดีคือเราสามารถนำไฟล์เหล่านี้ไปแก้ไข หรือทำอย่างอื่นได้อย่างสะดวก ในการประมวลผลแบบ Image Sequence นั้น เรื่องของการ Run ชื่อไฟล์มีความสำคัญ เพราะถ้าเราไม่ทราบถึงโครงสร้างที่ถูกต้อง จะเกิดปัญหาเมื่อเรา Import ไฟล์เหล่านี้เข้าโปรแกรมตัดต่อเพื่อสร้างเป็นภาพยนตร์ต่อไป เพราะโปรแกรมตัดต่อจะไม่เข้าใจว่ามันคือ Sequence และภายใต้หัวข้อ *Frame/ Animation Ext* นี้ เป็นหัวข้อที่เราสามารถจัดการกับการตั้งชื่อให้อยู่ในโครงสร้างที่ถูกต้องกัน โดยมีศัพท์ที่ใช้สื่อความหมายอยู่สามตัวคือ Name = ชื่อของไฟล์ภาพ (ในที่นี้จะป็นชื่อเดียวกับชื่อในหัวข้อ *File Name Prefix*), # = หมายเลขของเฟรม ยกตัวอย่างเช่นในการ Render มีทั้งหมด 1000 เฟรม โปรแกรมจะ Run เลขให้ตั้งแต่ 0001 ในเฟรมแรกไปจนถึง 1000 ในเฟรมสุดท้ายให้โดยอัตโนมัติ, Ext = นามสกุลของไฟล์ จะป็นไปตาม Format ที่เราเลือก เช่น JPG, TGA, TIF หรือ IFF เป็นต้น โดยภายใต้หัวข้อ *Frame/ Animation Ext* นั้น โปรแกรมจะมี Template การวางตำแหน่งของ Name, # และ Ext ให้เลือก ว่าจะเอาอะไรไว้ก่อนหลังอย่างไร ซึ่งขึ้นอยู่กับผู้ใช้ว่ามีความต้องการอย่างไร แต่มีข้อควรจำว่าเราจะเอาอะไรไว้ก่อนหลังก็ได้ เพียงแต่ควรเลือกให้ Ext หรือนามสกุลของไฟล์อยู่หลังสุด เพื่อหลีกเลี่ยงปัญหาการไม่ทราบนามสกุลจากโปรแกรมอื่นๆได้ โดยมาตรฐานสากลนิยมเรียงจากชื่อไฟล์ หมายเลขเฟรม และนามสกุลเป็นตัวสุดท้าย

หัวข้อที่สาม *Image Format* หรือประเภทของไฟล์ ในส่วนนี้สามารถแบ่งประเภทของ Format ได้เป็นสองประเภทใหญ่ คือประเภทที่เป็นไฟล์ภาพนิ่ง (Still Image) เช่น TGA, JPG และประเภทที่เป็นไฟล์ภาพเคลื่อนไหว (Movie) เช่น AVI, MOV ถ้าเรา Render ฉากที่มีมากกว่าหนึ่งเฟรมโดยเลือกประเภทของไฟล์เป็นภาพนิ่ง ผลที่ได้คือ Set ของ Image Sequence นั่นเอง

เมื่อเสร็จสิ้นจากการตั้งค่าของทั้งสามส่วนที่กล่าวมาแล้ว เราจะพบว่ามีหัวข้ออีก 4 หัวข้อที่เราสามารถแก้ไขได้ (ถ้าปล่อยให้ไม่ได้ทำการแก้ไขก็ยังสามารถ Render ได้ปกติ) ใช้สำหรับการตั้งค่าเฉพาะเจาะจงบางชนิด ประกอบด้วย *Start Frame* เป็นตัวกำหนดว่าเราจะให้โปรแกรมเริ่มต้นการ Render จากเฟรมใด, *End Frame* กำหนดว่าการ Render จะสิ้นสุดลงที่เฟรมใด, *By Frame* บอกว่าจะทำการ Render ในทุกๆเฟรม (โปรแกรมจะไม่ Render ทุกเฟรม), และสุดท้าย *Frame Padding* เป็นการบังคับให้โปรแกรมใส่เลขศูนย์ (0) นำหน้าตัวเลขลำดับที่ของเฟรม (#) เพื่อช่วยให้การจัดลำดับไฟล์ไม่ผิดพลาดเมื่อเราสั่งให้ Sort By Name ตอน Preview ไฟล์ทั้งหมดเมื่อเสร็จสิ้นการประมวลผลแล้ว

## Renderable Cameras

เราสามารถกำหนดมุมมองที่จะใช้ในการประมวลผลได้จากส่วนนี้ ภายใต้หัวข้อ *Renderable Camera* จะมี Drop Down Menu อยู่ โดยค่าที่ตั้งต้นจะอยู่ที่ Perspective View (Persp) เมื่อเราคลิกจะมีตัวเลือกมุมมองออกมาทั้ง Persp, Top, Front, Side ซึ่งเป็นมุมมองพื้นฐานที่โปรแกรมมีให้ และ Add Renderable Camera ซึ่งเป็นมุมมองพิเศษสำหรับกล้องที่ผู้ใช้สร้างขึ้นมาเอง เมื่อเราเลือกมุมมองเสร็จแล้วที่ด้านล่างจะมี Check Box อยู่สองอันเพื่อกำหนดคุณสมบัติพิเศษให้กับไฟล์ภาพคือ *Alpha Channel (Mask)* ถ้าเราเลือกที่หัวข้อนี้ โปรแกรมจะทำการคำนวณค่า Alpha Channel ให้ เราควรจะต้องเลือกที่หัวข้อนี้ทุกครั้งเพื่อประโยชน์ในการนำไฟล์ที่ได้ไปใช้กับโปรแกรมตัดต่ออื่นๆ แต่ถึงอย่างไรก็มี Format บางประเภทที่ไม่รองรับการทำงานในระบบ Alpha ยกตัวอย่างเช่น JPG, GIF หรือ Windows Bitmap ดังนั้นถ้าเราต้องการนำไฟล์ที่ได้ไปผ่านขั้นตอนการตัดต่อที่ต้องใช้ค่า Alpha อีกทีหนึ่ง ก็ควรหลีกเลี่ยง Format เหล่านี้ ส่วน Check Box อันที่สองคือ *Depth Channel (Z Depth)* แล้ว Z-Depth คืออะไร Z-Depth จะเป็นตัวเก็บค่าระยะห่างระหว่างวัตถุต่างๆภายในฉากกับกล้อง โดยออกมาในรูปแบบ Grayscale Map ซึ่งแตกต่างกับค่า Alpha Channel ซึ่งจะมีแค่ขาวกับดำ แต่ถึงอย่างไร Format ของไฟล์ภาพที่รองรับกับระบบ Grayscale Channel มีอยู่น้อยมาก เช่น IFF ซึ่งเป็นไฟล์เฉพาะของทางโปรแกรม โดยค่า Z-Depth นี้จะมีประโยชน์ในขั้นตอนของการตัดต่อ

## Advanced Image Naming

ส่วนนี้จะทำหน้าที่ในการจัดการชื่อเฉพาะของทางโปรแกรม เช่นในระหว่างเฟรมที่ 50 -100 เราต้องการให้ไฟล์มีชื่อเฉพาะแตกต่างออกไป สามารถกำหนดได้ในส่วนนี้ หรือจะกำหนด Ext พิเศษเฉพาะขึ้นมาให้กับ Sequence ที่ต้องการก็สามารถทำได้เช่นกัน

## Image Size

การจัดการเรื่องขนาดและความละเอียดของไฟล์ภาพหรือภาพเคลื่อนไหว ที่ Render ออกมาสามารถกำหนดได้ที่นี่ โดยกำหนดได้จากการเลือก Template ที่โปรแกรมมีให้ภายใต้หัวข้อ *Presets* ซึ่งมีขนาดพื้นฐานสำหรับ Footage เพื่อใช้ในโอกาสต่างๆเช่น PAL, NTSC สำหรับอุตสาหกรรมโทรทัศน์, HD 720 และ 1080 สำหรับเครื่องรับโทรทัศน์ในระบบ High-Definition, หรือจะเป็นขนาดเพื่อใช้บนสื่อ Internet ก็มีให้เช่น 1-4 Square ทั้งนี้ผู้ใช้สามารถเลือกใช้ให้เหมาะสมได้ตามความต้องการและประเภทของงาน

แต่ถึงอย่างไรเราสามารถกำหนดขนาดของไฟล์ที่ Render ได้ด้วยตนเอง โดยเลือกไปที่ Custom ในหัวข้อ *Presets* แล้วใส่ค่าที่ต้องการลงในหัวข้อ Width และ Height เช่น 1280x720 เป็นต้น หัวข้อ *Size Units* กำหนดว่าจะใช้หน่วยใดแสดงผล ส่วนหัวข้อ *Resolution* กำหนดความละเอียดของภาพ หัวข้อ *Resolution Units* กำหนดว่าจะใช้หน่วยใดในการแสดงค่า Resolution และสองหัวข้อสุดท้ายคือ *Device Aspect Ratio* และ *Pixel Aspect Ratio* ซึ่งเป็นตัวกำหนดขนาดความกว้างของภาพ เพื่อใช้สำหรับ Format บางประเภทที่ต้องการการตั้งค่าเฉพาะขั้นสูง ในการทำงานปกติเราไม่มีความจำเป็นต้องเปลี่ยนแปลงค่านี้แต่อย่างใด ข้อควรระวังคือในการ Render แบบปกติทุกครั้งภายใต้หัวข้อ *Pixel Aspect Ratio* จะต้องใส่ค่า = 1 เท่านั้น ยกเว้นการ Render ในแบบ CCIR 601/ Quantel NTSC ไม่อย่างนั้นผลลัพธ์ที่ได้ภาพอาจมีการยืดออกผิดจากอัตราส่วนที่ต้องการไป

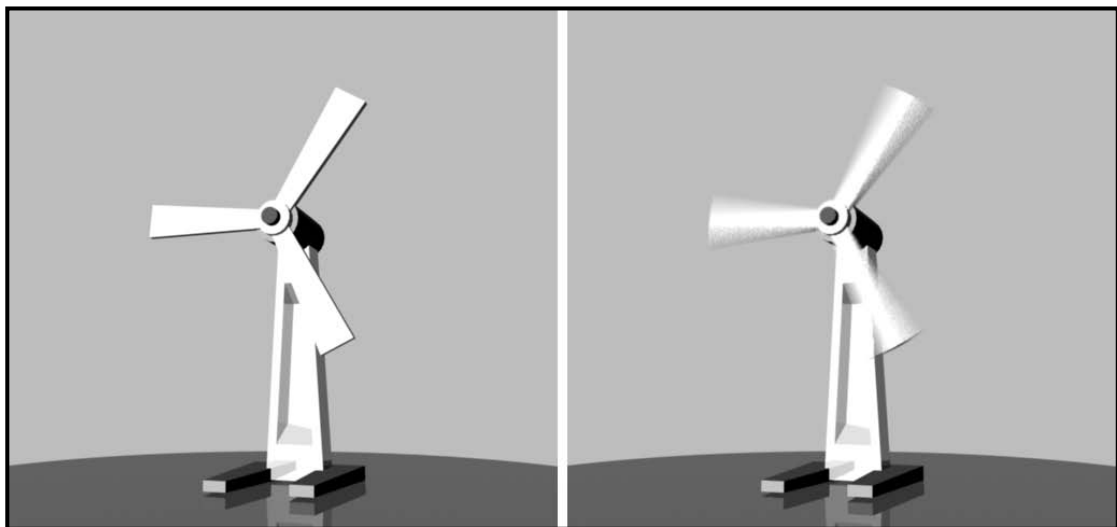
และนี่คือรายละเอียดที่มีความจำเป็นที่ควรทราบในการตั้งค่าจาก Tab Common ของหน้าต่าง Render Settings ไม่ว่าเราจะใช้การ Render ในรูปแบบใด การตั้งค่า Common Render ล้วนเป็นเช่นเดียวกัน หลังจากนั้นเราจะดูถึงข้อแตกต่างในตัวประมวลผลแต่ละตัว (Renderers) ว่ามีอะไรบ้าง

## Maya Software Render

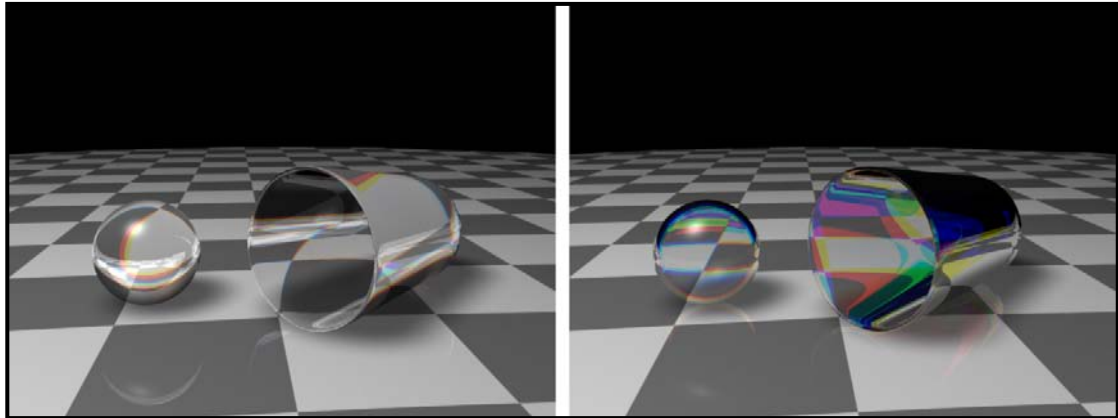
การประมวลผลด้วย Maya Software ถึงแม้จะไม่ใช่วิธีการประมวลผลที่มีความเหมือนจริงที่สุดแต่จุดเด่นอยู่ที่การแสดงผลของ Effects พิเศษต่างๆที่ผู้ใช้สามารถเพิ่มเข้าไปได้ เช่นการแสดงผลของ Paint Effects หรือ Effects ของพื้นผิวและแสง ซึ่งบางตัวไม่สามารถแสดงผลได้ภายใต้ตัวประมวลผลแบบอื่น และข้อดีอีกประการหนึ่งคือเรื่องของระยะเวลาการ Render ที่สั้นกว่าการประมวลผลด้วย Mental Ray อย่างเห็นได้ชัด

นอกจากนั้น Maya Software ยังสามารถแสดงผลของ Refraction (การหักเหของแสง) และ Reflection (การสะท้อนของแสง) ได้อีกด้วย เพียงแต่ต้องเปิดการทำงานการคำนวณแสงแบบ Raytracing ก่อน เนื่องจากการคำนวณแสงแบบปกติของ Maya Software จะไม่สามารถแสดงผลเงาสะท้อนต่างๆในลักษณะนี้ได้ การเปิดการทำงาน Raytracing ยังช่วยให้การประมวลผลภาพมีความเหมือนจริงขึ้น แต่แน่นอนว่าระยะเวลาในการ Render จะนานขึ้นด้วย แต่ถึงอย่างไรถ้าเราไม่ต้องการเปิดการทำงาน Raytracing เรายังสามารถสร้าง Reflection ให้กับภาพได้ แต่ต้องใส่เข้าไปในลักษณะของ Reflection Map (คลิกที่ Option Box ด้านหลัง Reflection เพื่อใส่ Map) เพียงแต่เงาสะท้อนที่ได้ก็จะไม่ใช่ True Reflections ที่ได้จากการคำนวณการสะท้อนของแสง หากแต่เป็นเพียงเงาสะท้อนหลอกๆที่เราสร้างขึ้นมา

จากภาพประกอบที่ 6.2 แสดงผลลัพธ์ที่ได้จากการใช้ Motion Blur Effect ภาพใบพัดที่มีการเคลื่อนไหวจะถูกคำนวณและแสดงให้เห็นได้แม้ในการ Render ภาพนี้ ส่วนภาพประกอบที่ 6.3 แสดงความแตกต่างระหว่างเงาสะท้อนของแสง (Reflection) ในภาพซ้าย และการหักเหของแสง (Refraction) ในภาพขวา



ภาพประกอบที่ 6.2 แสดงผลการประมวลผลภาพแบบธรรมดาในภาพซ้าย, ในขณะที่ภาพขวาเป็นการประมวลผลโดยใช้ Motion Blur

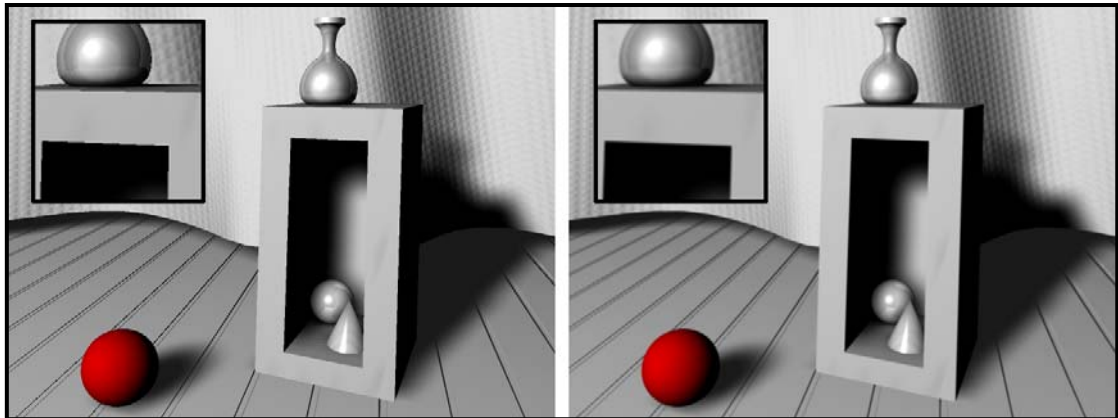


ภาพประกอบที่ 6.3 ภาพซ้ายแสดง Effect ที่ได้จากการใส่ค่า Reflection ให้กับพื้นผิว, ส่วนภาพขวาแสดง Effect จากการใส่ค่า Refraction

ในหน้าต่าง Render Setting ภายใต้ Maya Software Tab จะมีหัวข้อที่มีความจำเป็นต้องทราบอยู่ดังนี้

### Anti-Aliasing

จากที่ทราบกันว่าหน้าจอคอมพิวเตอร์สามารถแสดงผลได้จากจุด Pixels ที่เรียงต่อกัน ในการประมวลผลภาพ บางครั้งอาจเกิดปัญหาที่ Pixels ขาดความ Smooth หรือเห็นเป็นเส้นบริเวณขอบของวัตถุ *Anti-Aliasing* จะทำการเกลี่ยเส้น Pixels ใหม่ๆ เหล่านี้ให้มีความกลมกลืนมากขึ้น ดูภาพประกอบที่ 6.4 แสดงผลลัพธ์ที่ได้จากการใช้ *Edge Anti-Aliasing* ช่วยในการจัดการกับ Pixel บริเวณขอบของวัตถุ เกลี่ยให้ดูเป็นธรรมชาติมากขึ้น



ภาพประกอบที่ 6.4 ภาพซ้ายแสดงการประมวลผลแบบปกติ, ส่วนภาพขวามีการเปิดการทำงานของ Anti-Aliasing ช่วยในการจัดการกับ Pixels

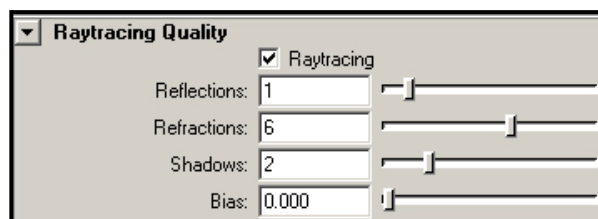
การตั้งค่าใน *Anti-Aliasing* สามารถทำได้อย่างสะดวกสบาย จาก Template ที่โปรแกรมมีให้ ภายใต้หัวข้อ *Quality* และ *Edge Anti-Aliasing* ภายใต้หัวข้อ *Quality* เป็นที่ที่เราจะกำหนดคุณภาพของภาพว่าจะเป็นอย่างไร โดยมี Template ให้เลือกตั้งแต่ *Preview Quality* ซึ่งมีความละเอียดต่ำ ไปจนถึง *3D Motion Blur Production Quality* ซึ่งมีความละเอียดสูงสุด เมื่อเราเลือก Template ใดๆในหัวข้อนี้ โปรแกรมจะทำการเลือกตัวเลือกในหัวข้อ *Edge Anti-Aliasing* ที่มีความเหมาะสมตาม *Quality* ที่ได้เลือกไป แต่ถึงอย่างไรเราสามารถเปลี่ยนแปลงได้เอง โดยเลือกจาก Template ภายใต้หัวข้อ *Edge Anti-Aliasing* ซึ่งจะมีให้เลือกตั้งแต่ *Low Quality* ไปจนถึง *Highest Quality* เมื่อเราเปลี่ยนแปลงตรงนี้ ที่หัวข้อ *Quality* จะเปลี่ยนเป็น *Custom* แทน



เมื่อได้คุณภาพการประมวลผลที่เราต้องการ จากนั้นเรายังสามารถทำการเกลี่ยของแข็งๆของวัตถุ ให้มีความกลมกลืนขึ้นได้โดยทำเครื่องหมายถูกที่ด้านหน้าคำสั่ง *Use Multi Pixel Filter* แล้วทำการปรับค่าความกลมกลืนได้จากหัวข้อ *Pixel Filter Width X* และ *Y* แต่ตัวเลือกนี้จะสามารถเลือกได้ต่อเมื่อเราเลือก *Quality* ตั้งแต่ *Intermediate Quality* และ *Edge Anti-Aliasing* ตั้งแต่ *High Quality* ขึ้นไป

### Raytracing Quality

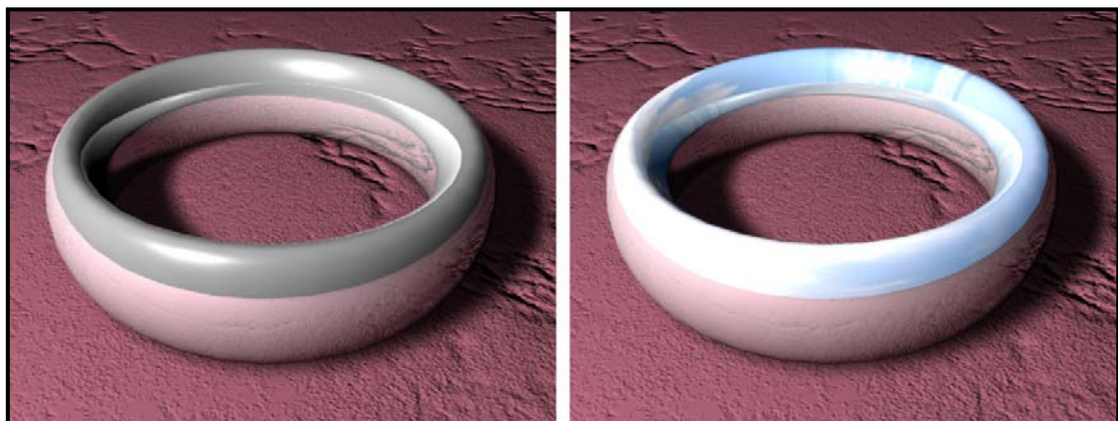
การเปิดการทำงานของ *RayTracing* เป็นการเปิดความสามารถของโปรแกรมให้คำนวณแสงในรูปแบบที่มีความเสมือนจริงเพิ่มขึ้น การคำนวณเงาสสะท้อน (Reflections) และการหักเหของแสงผ่านวัตถุ (Refractions) จะสามารถประมวลผลได้เมื่อผ่านการทำงานนี้ แต่จากปัจจัยต่างๆในการคำนวณที่เพิ่มขึ้นอย่างมาก ย่อมทำให้ระยะเวลาในการประมวลผลนานขึ้นมากเช่นกัน



ภาพประกอบที่ 6.5 แสดงหน้าต่าง Render Settings ภายใต้หัวข้อ Raytracing Quality

แล้วโปรแกรมสามารถคำนวณแสงสะท้อนเหล่านี้ได้อย่างไร *Raytracing* จะใช้การคำนวณที่มีความคล้ายคลึงกับธรรมชาติมากคือจะเริ่มคำนวณจากลำแสงจากแหล่งกำเนิดแสงวิ่งมาตกกระทบยังพื้นผิววัตถุ จากนั้นเกิดการหักเหสะท้อนเข้ามายังดวงตาของเราหรือเลนส์ของกล้อง เกิดเป็นภาพขึ้นมา ในการจัดฉากขึ้นมา บางครั้งผู้สร้างจะพบว่าเป็นเรื่องยากที่จะกำหนดสิ่งที่สะท้อนบนวัตถุหนึ่งๆได้ตรงตามความต้องการ บางครั้งสิ่งที่สะท้อนทำให้ตัววัตถุขาดความเสมือนจริงลงไปได้ ยกตัวอย่างเช่นเราต้องการสร้างฉากที่มีลูกบอลวางอยู่กลางแจ้ง (Open Area) โดยให้สะท้อนวิวทิวทัศน์ที่อยู่รอบๆ ซึ่งเป็นเรื่องยากและเสียเวลาที่จะสร้างสิ่งแวดล้อมใหญ่ๆขึ้นมาเพื่อสร้างเงาสะท้อนให้กับลูกบอลเพียงลูกเดียว

ปัญหานี้สามารถแก้ไขได้โดยการสร้าง *Reflection Map* ขึ้นมา *Reflection Map* คืออะไรและแตกต่างจากเงาสะท้อนจริงๆอย่างไร การใส่ *Reflection Map* นั้น เงาสะท้อนไม่ได้เกิดจากการหักเหของแสง แต่เกิดจากการใช้ภาพนิ่ง (Map) เข้าไปวางไว้บนพื้นผิว สร้างเป็นเงาสะท้อนหลอกขึ้นมา ดังนั้นจึงไม่มีความจำเป็นที่ผู้สร้างจะต้องสร้างองค์ประกอบทั้งหมดขึ้นมาจริงๆ (ใส่ที่ *Reflective Color* จาก Attribute ของ Material)



ภาพประกอบที่ 6.6 แสดงภาพจาก Reflection แบบปกติในภาพซ้าย, และ Reflection Map รูปท้องฟ้าในภาพขวา



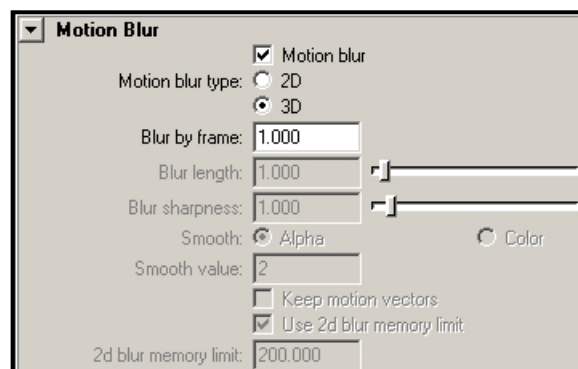
ส่วน Refractions หรือการหักเหของแสงก็สามารถแสดงผลได้เช่นกัน การหักเหของแสงต่างจากการสะท้อนแสงอย่างไร ลองนึกภาพแก้วน้ำที่ใส่น้ำเปล่าอยู่ครึ่งแก้ว เมื่อเรามองผ่านแก้วตรงส่วนที่มีน้ำกับส่วนที่ว่างเปล่าจะเห็นฉากหลังที่บิดเบี้ยวต่างกัน นั่นเนื่องจากน้ำกับแก้วมีการหักเหของแสงที่ต่าง ทั้ง Reflection และ Refraction จะแสดงผลได้ต้องอาศัยปัจจัยสองอย่างคือต้องเปิดการทำงานของ Raytracing ใน Render Settings Window และจะต้องใส่ค่าการหักเห การสะท้อนใน Attributes ของ Material ที่ใส่ให้กับวัตถุ (ภายใต้หัวข้อ Reflectivity กับ Refractions)



ภาพประกอบที่ 6.7 แสดงผลลัพธ์ที่ได้จากการใช้ Refractive Index จากซ้ายไปขวาคือ ค่า = 1.000, 1.001 และ 1.005 ตามลำดับ

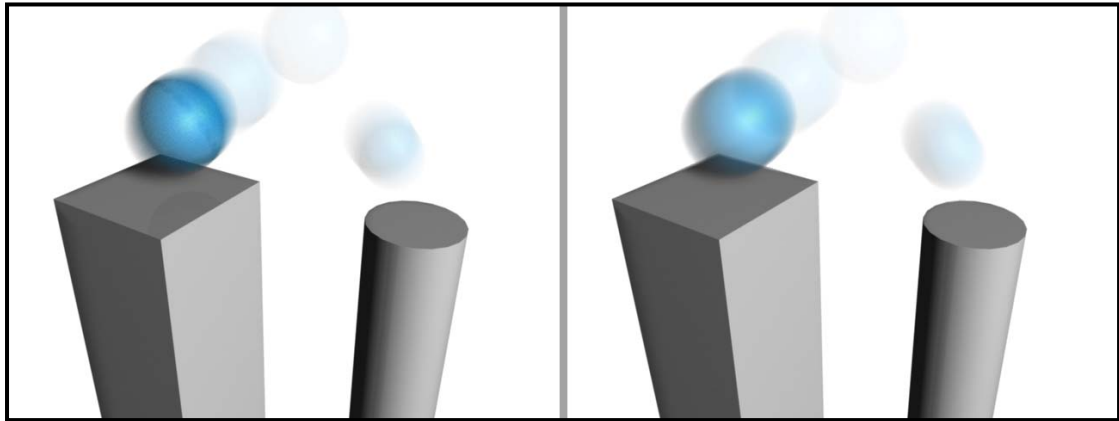
### Motion Blur

Motion Blur มีหน้าที่อะไร ลองนึกภาพแอนิเมชันที่มีการเคลื่อนไหวเร็วๆ อย่างเช่นล้อรถที่หมุน ใบพัดลม หรือ ลูกเทนนิสที่ถูกตีออกไปอย่างรุนแรง ภาพถ่ายที่ได้จากการเคลื่อนไหวเหล่านี้จะมีความ Blur เนื่องจากความไวของชัตเตอร์ไม่สามารถจับภาพที่มีความเร็วขนาดนั้นได้ แต่ในโลกสามมิติภาพจะถูกประมวลผลมาจากวัตถุโดยตรง ไม่ต้องผ่านชัตเตอร์กล้องถ่ายภาพแต่อย่างใด ดังนั้นการเบลอจึงเป็นเรื่องที่เป็นไปไม่ได้ แล้วถ้าในบางโอกาสเราต้องการได้ความรู้สึกของวัตถุที่เคลื่อนที่เร็วๆและต้องการความเบลอตรงนี้จะทำได้อย่างไร เพื่อตอบรับกับทุกสถานการณ์ที่ผู้สร้างแอนิเมชันต้องการ Motion Blur จึงเป็นทางออกของปัญหานี้



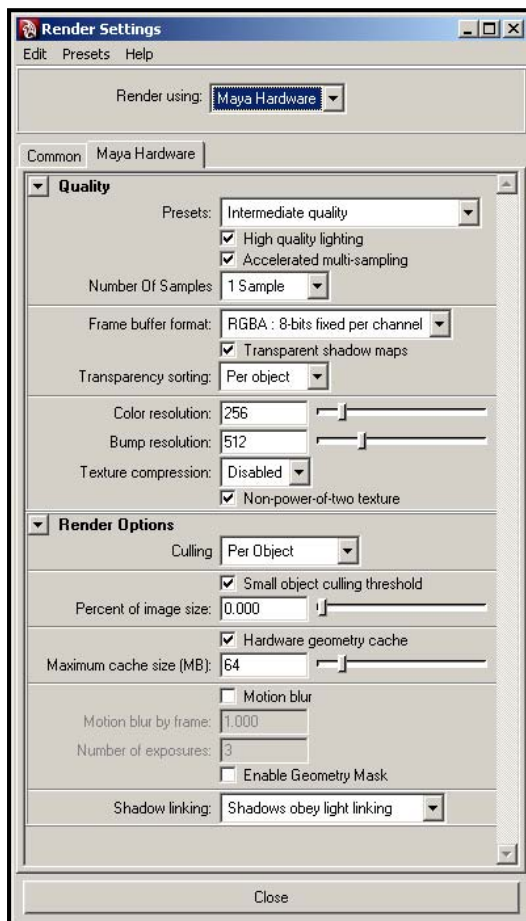
ภาพประกอบที่ 6.8 แสดงหน้าต่าง Render Settings ภายใต้หัวข้อ Motion Blur

จากภาพตัวอย่างที่ 6.8 สังเกตได้ว่า Motion Blur มีสองประเภทด้วยกัน คือ 2D และ 3D ทั้งสองประเภทมีข้อเด่นและข้อด้อยต่างกัน 2D จะประมวลผลในแนว 2 แกนเท่านั้น ทำให้ใช้ระยะเวลาในการประมวลผลสั้นกว่า ส่วน 3D จะประมวลผลโดยใช้แกนทั้งหมด 3 แกน ซึ่งจะได้ผลลัพธ์ที่มีความสมจริงมากกว่าสำหรับวัตถุที่มีการเคลื่อนที่ใน 3 แกน ดูภาพประกอบที่ 6.9 แสดงความแตกต่างระหว่าง Motion Blur แบบ 2D และแบบ 3D



ภาพประกอบที่ 6.9 เปรียบเทียบความแตกต่างของ Motion Blur ทั้งสองประเภท ภาพซ้ายคือแบบ 3D และภาพขวาคือแบบ 2D

## Maya Hardware Render



ภาพประกอบที่ 6.10 แสดงหน้าต่าง Render Setting

ภายใต้หัวข้อ Maya Hardware Render

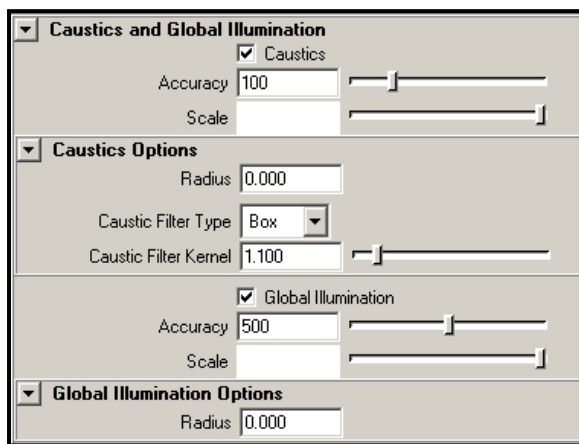
การประมวลผลด้วย Maya Hardware มีไว้สำหรับสถานการณ์ที่ผู้ใช้ต้องการ Preview ผลลัพธ์ของฉากที่สร้าง โดยที่ไม่ต้องการเสียเวลาในการ Render เท่ากับการ Render ด้วย Maya Software เพียงแต่ผลลัพธ์ที่ได้จะมีคุณภาพไม่เทียบเท่าการใช้ Maya Software ประมวลผล คุณภาพของ Maya Hardware จะเป็นคุณภาพที่อยู่ตรงกลางระหว่างคุณภาพของ Playblast กับคุณภาพของ Maya Software เนื่องจากการใช้ตัวเครื่อง (Hardware) ในการประมวลผล ดังนั้นการที่ผลลัพธ์จะออกมาดีแค่ไหน ขึ้นอยู่กับประสิทธิภาพของตัว Hardware ของเครื่องที่ใช้ทำการ Render นั่นคือคุณภาพของการ์ดจอนั่นเอง ในการ์ดจอที่มีคุณภาพสูง อาจให้ผลลัพธ์ที่มีความใกล้เคียงกับการใช้ Maya Software ประมวลผลเลยทีเดียว แต่การ์ดจอในระดับกลางถึง กลางสูง อาจไม่สามารถคำนวณผลใน Details บางส่วน เช่น Reflections, Refractions, Bump Maps หรือเงาที่ต้องการการประมวลผลสูง ได้ ดังนั้นการ Render ผลงานที่ต้องการคุณภาพสูง Maya Hardware จึงไม่ใช่คำตอบที่ดีที่สุดสำหรับสถานการณ์นั้น

## Mental Ray Renderer

โดยทั่วไปแล้ว Mental Ray จะมี Functions การตั้งค่าพื้นฐานต่างๆใกล้เคียงกับของ Maya Software เพียงแต่จะมี Option พิเศษเพิ่มเติมเข้ามาเช่น *Caustics*, *Global Illumination* และ *Final Gathering* ที่จะเข้ามาช่วยในการคำนวณค่าแสงให้มีความเสมือนจริงมากขึ้น ดังนั้นผลลัพธ์ที่ได้จากการใช้ Mental Ray คือภาพที่มีความเป็น Realistic มากกว่า Maya Software แน่นนอน เพียงแต่ก็มีข้อเสียที่ผู้ใช้ต้องยอมรับในเรื่องของระยะเวลาในการประมวลผลที่เพิ่มมากขึ้น และ Effects บางตัวที่จะแสดงผลเฉพาะใน Maya Software เท่านั้น

เนื่องจากหัวข้อในการตั้งค่าส่วนใหญ่ใกล้เคียงกับการตั้งค่า Maya Software ดังนั้นจะขอกล่าวถึงเฉพาะหัวข้อที่มีความแตกต่างออกไปนั่นคือ *Caustics and Global Illumination* กับ *Final Gathering*

### Caustics and Global Illumination



ทั้ง *Caustics* และ *Global Illumination* ล้วนทำหน้าที่ในการช่วยคำนวณแสงให้มีความเสมือนจริงเพิ่มมากขึ้น เมื่อเราเปิดการทำงานของ *Global Illumination* (ทำได้โดยการทำเครื่องหมายถูกหน้าหัวข้อ *Global Illumination*) โปรแกรมจะทำการคำนวณค่าการสะท้อนของแสง นั่นคือเมื่อแสงตกกระทบมายังพื้นผิววัตถุแล้วสะท้อนออกไป ส่องสว่างให้กับวัตถุที่อยู่ข้างเคียงได้ ซึ่งเป็นหลักการที่ใกล้เคียงกับวิธีที่แสงจากธรรมชาติทำกับวัตถุจริงๆ

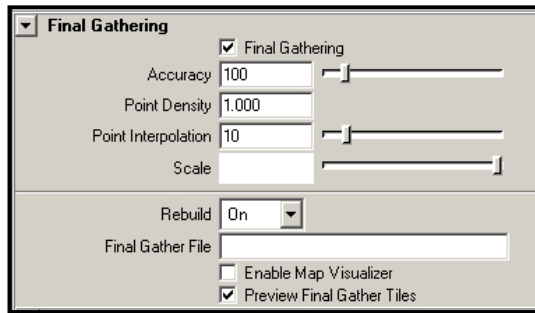
ภาพประกอบที่ 6.11 แสดงหน้าต่าง Render Settings Window ภายได้หัวข้อ Caustics and Global Illumination ใน Mental Ray

ในการคำนวณแสงแบบปกติของโปรแกรมมาายนั้น จะคำนวณเพียงแค่ต่อเดียว นั่นคือเมื่อแสงตกกระทบวัตถุแล้วมาเข้าตาเรา แต่จะไม่มีการคำนวณแสงส่วนที่เหลือที่จะวิ่งกระเด็นกระดอนไปส่องสว่างให้กับวัตถุอื่นๆ ภายในฉากซึ่งทำให้ผลลัพธ์ที่ได้ยังขาดความเสมือนจริงอยู่ จากภาพตัวอย่างที่ 6.11 ภายหลังจากที่เราทำเครื่องหมายถูกที่หน้าหัวข้อ *Global Illumination* เพื่อเปิดการทำงานของมันแล้ว เราสามารถปรับค่าการคำนวณได้จากหัวข้อ *Accuracy* ที่อยู่ด้านล่าง โดยค่าตั้งต้นที่โปรแกรมตั้งไว้คือ 500

ในส่วนของ *Caustics* จะทำหน้าที่คล้ายกับ *Global Illumination* เพียงแต่แสงที่จะนำมาคำนวณจะเป็นแสง Specular หรือแสง Highlights บนวัตถุ ดังนั้นในการทำงานที่สมบูรณ์ เราสามารถเปิดการทำงานของทั้งคู่พร้อมกัน

การทำงานของ *Caustics and Global Illumination* สามารถให้ผลลัพธ์ที่ดูเป็นธรรมชาติ เพียงแต่ต้องใช้ระยะเวลาในการ Render ที่นานขึ้นด้วย ดังนั้นในบางโอกาสผู้สร้างอาจหาทางใส่ Effects เหล่านี้ด้วยวิธีใดวิธีหนึ่งเพื่อช่วยลดเวลาการประมวลผล ยกตัวอย่างเช่นการใส่แสงไปเพิ่มเข้าไปโดยกำหนดให้มีผลเฉพาะกับวัตถุบางชิ้น เพื่อสร้างเป็นแสงสะท้อนหลอก ให้ได้ผลลัพธ์ใกล้เคียงกับการคำนวณแสงจริงได้

## Final Gathering

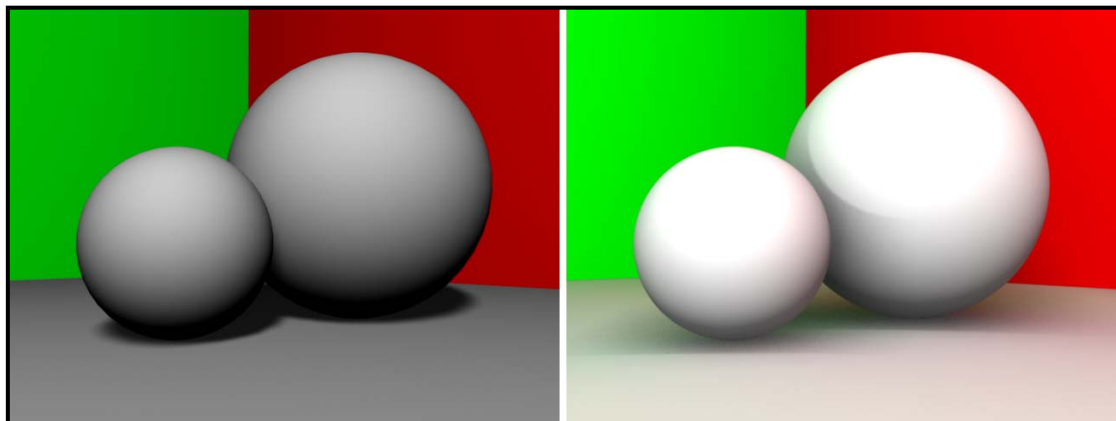


ภาพประกอบที่ 6.12 แสดงหน้าต่าง Render Settings Window

ภายใต้หัวข้อ Final Gathering ใน Mental Ray

ในการคำนวณค่าแสงแบบ *Caustics and Global Illumination* ซึ่งถือว่าได้ผลลัพธ์ตามธรรมชาติมากแล้ว แต่บางครั้งสิ่งที่สะท้อนมายังวัตถุข้างเคียงไม่ใช่แค่แสงเพียงอย่างเดียว แต่จะมีเรื่องของสีด้วย ลองจินตนาการภาพผ้าปูโต๊ะสีขาว วางอยู่ข้างผนังสีแดง แสงเมื่อตกกระทบผนังสีแดงแล้วสะท้อนมายังผ้าปูโต๊ะจะนำพาสีมาด้วย

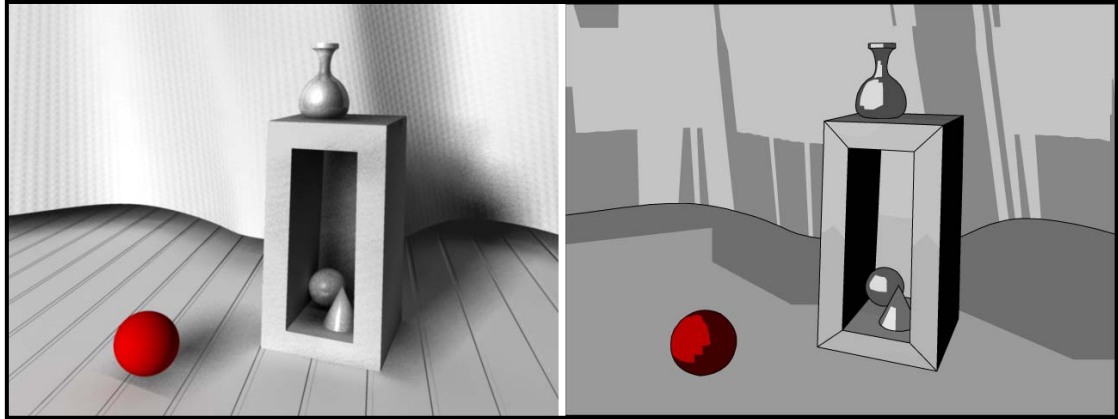
การประมวลผลโดยใช้ *Final Gathering* จะเป็นการเก็บรวบรวมแสงที่กระเด็นกระดอนอยู่ภายใต้ให้มีความสมบูรณ์มากยิ่งขึ้นและมีความเหมือนจริงมากขึ้นด้วย และการใช้ *Final Gathering* จะส่งผลให้ฉากมีความสว่างมากยิ่งขึ้นด้วย ดูภาพประกอบที่ 6.13 เปรียบเทียบผลลัพธ์จากการประมวลผลด้วย *Final Gathering* และแบบธรรมดาด้วย Maya Software สังเกตเงาสะท้อนของแสงสีเขียวจากผนังบนพื้น



ภาพประกอบที่ 6.13 เปรียบเทียบการประมวลผลด้วย Maya Software ในภาพซ้าย และ Mental Ray ที่เปิดการทำงานของ Final Gathering ในภาพขวา

## Maya Vector Renderer

โปรแกรมมาายเป็นโปรแกรมสามมิติ เน้นอนว่าภาพที่ถูกสร้างออกมาจะสวยงามจะเป็นภาพสามมิติ แต่ถึงอย่างไรการประมวลผลภาพสามมิติให้ออกมาในรูปแบบสองมิติอย่างภาพ Vector ก็ยังสามารถทำได้เช่นกัน นอกจากนั้นมายายังสามารถสร้าง Output ไฟล์ Vector เหล่านี้ให้สามารถนำไปใช้ได้กับโปรแกรมอื่นที่สามารถอ่านภาพระบบ Vector Based Images ได้อีกด้วย ไม่ว่าจะเป็น SWF สำหรับ Macromedia Flash, AI ไฟล์สำหรับ Adobe Illustrator, Encapsulated Postscript (EPS) ไฟล์ รวมทั้ง Scalable Vector Graphics (SVG) ไฟล์ ซึ่งเป็น Format ที่ได้รับความนิยมในการสร้างภาพ Vector ต่างๆ

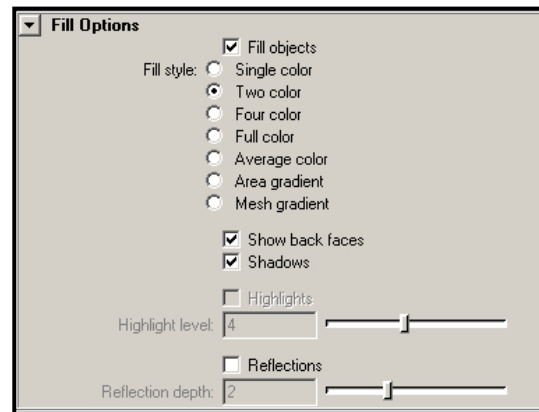


ภาพประกอบที่ 6.14 เปรียบเทียบการประมวลผลด้วย Maya Software แบบธรรมดาในภาพซ้าย, และการใช้ Maya Vector Renderer ในภาพขวา

ในการประมวลผลภาพแบบ Vector นี้มีความแตกต่างกับการประมวลผลทั้งสามแบบที่กล่าวถึงมาก่อนหน้านี้อยู่มาก ดังนั้นจึงมีวิธีการตั้งค่าที่แตกต่างกัน ยกตัวอย่างเช่นการตั้งค่าในการคำนวณค่าแสง หรือ Effects ต่างๆจะถูกตัดออกไป การใช้ Paint Effects ไม่สามารถแสดงผลได้เป็นต้น แต่จะเพิ่มการตั้งค่าในเรื่องของรูปแบบการคำนวณ Vector เข้ามาแทน เช่นลักษณะของเส้นขอบ การถมสี ความละเอียดของการไล่สีเป็นต้น ซึ่งอยู่ภายใต้หัวข้อ *Fill Options* และ *Edge Options*

### Fill Options

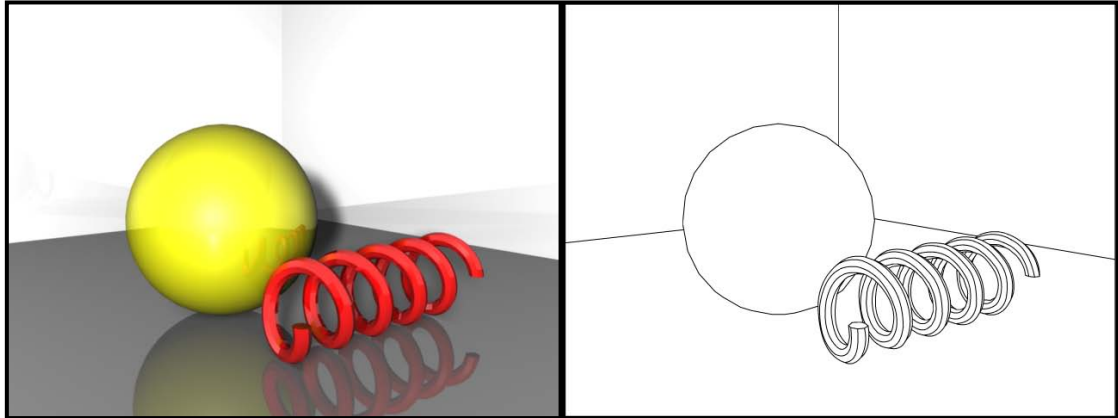
ภายใต้หัวข้อนี้ จะเป็นเรื่องของการถมสีให้กับวัตถุ เนื่องจากองค์ประกอบของภาพ Vector นั้นมีเพียงสองส่วนด้วยกัน นั่นคือเส้นขอบ และพื้นที่ภายในเส้นขอบของภาพ จากทั้งสองส่วนนี้โปรแกรมสามารถทำการคำนวณได้ว่าภาพจะออกมาเป็นเช่นไร ไม่ว่าจะเราจะย่อ/ขยายภาพอย่างไร โปรแกรมจะสามารถคำนวณออกมา เป็นภาพที่มีความละเอียดได้อย่างถูกต้อง หัวข้อ *Fill Options* จะทำงานกับพื้นที่บริเวณด้านในเส้นขอบ สามารถเปิดการทำงานได้โดยทำเครื่องหมายถูกที่ Check Box ด้านหน้าหัวข้อ *Fill Objects* (ดูภาพที่ 6.15 ประกอบ)



ภาพประกอบที่ 6.15 แสดงหน้าต่าง Render Settings Window

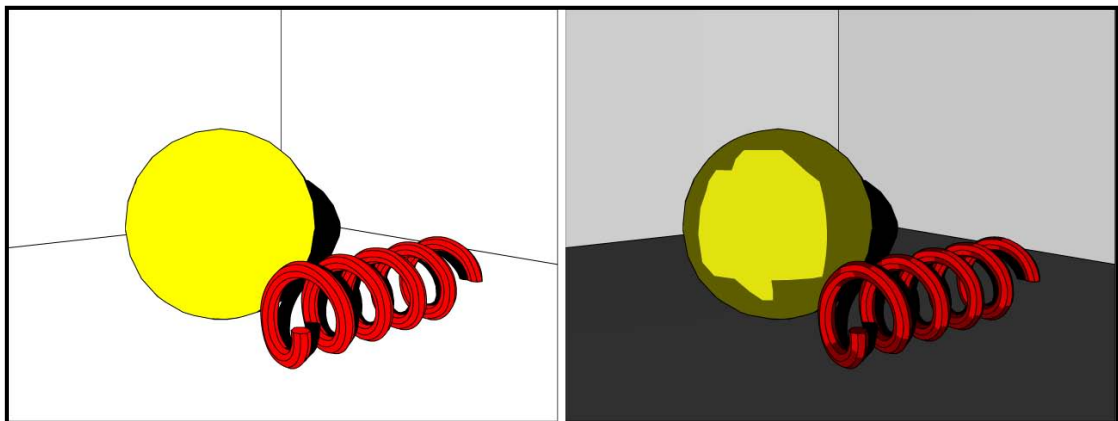
ภายใต้หัวข้อ *Fill Options* ใน Maya Vector

ถ้าเราปิดการทำงานของ *Fill Options* ผลลัพธ์ที่ได้จะเป็นภาพที่ไม่มีการถมสีใดๆ และจะไม่สามารถมองเห็นอะไรได้เลยจนกว่าเราจะเปิดการทำงานของเส้นขอบ ดังนั้นผู้ใช้ควรให้ความระมัดระวังอย่าปิดการทำงานของทั้งสองส่วนพร้อมๆกัน ไม่เช่นนั้นจะไม่สามารถมองเห็นรูปทรงของวัตถุได้เลย จากภาพประกอบที่ 6.16 แสดงผลลัพธ์ที่ได้เมื่อเราปิดการทำงานของ *Fill Objects*



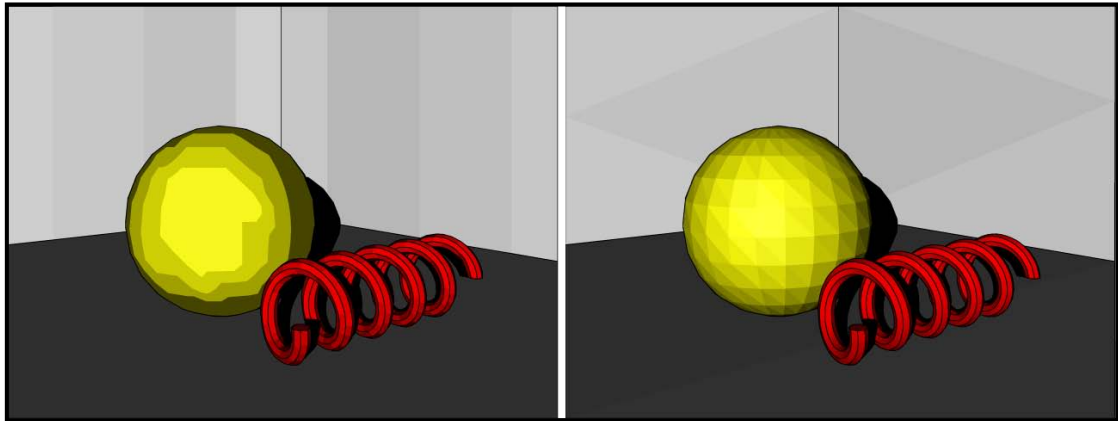
ภาพประกอบที่ 6.16 แสดงการประมวลผลด้วย Maya Software แบบปกติในรูปแบบซาย, และการใช้ Maya Vector แบบปิดการทำงานของ Fill Options ในภาพขาว ภาพจะไม่มีการถมสีแต่อย่างใด

เมื่อเราเปิดการทำงานของ *Fill Objects* แล้ว เราจะสามารถตั้งค่าเพิ่มเติมได้ในหัวข้อ *Fill Style* ที่อยู่ถัดลงมา ทำหน้าที่กำหนดว่าในการแปลงจากภาพสามมิติมาเป็นสองมิตินั้นจะมีการไล่โทนสีอย่างไร ถ้าเราเลือกที่ *Single Color* ในวัตถุหนึ่งชิ้นจะมีการใช้สีเพียงสีเดียว (ไม่มีการไล่โทนสี), ถ้าเป็น *Two* กับ *Four Color* จะมีการไล่สองและสี่สีในหนึ่งวัตถุ, *Full Color* จะมีการไล่สีอย่างละเอียดใกล้เคียงกับภาพจริง แต่บางครั้งทำให้สูญเสียความรู้สึกของภาพ Vector สองมิติไป, *Average Color* จำทำการคำนวณหาค่าสีเฉลี่ยของวัตถุนั้นๆออกมาเป็นสีๆเดียวให้กับวัตถุ, *Area Gradient* จะทำการไล่เฉดแบบ Gradient (จากสว่างไปมืดในโทนสีเดียวกัน) โดยใช้การคำนวณในลักษณะพื้นที่ เช่นรูปทรงกลมก็จะถูกคำนวณออกมาเป็นพื้นที่วงกลม, และสุดท้ายคือ *Mesh Gradient* ซึ่งมีลักษณะคล้ายกับ *Area Gradient* เพียงแต่โปรแกรมจะคำนวณจากรูปทรงสองวัตถุ ทำให้การไล่เฉดสีดูมีมิติมากขึ้น ดูภาพประกอบที่ 6.17 แสดงผลลัพธ์ที่ได้จากการตั้งค่า *Fill Style* แบบต่างๆ

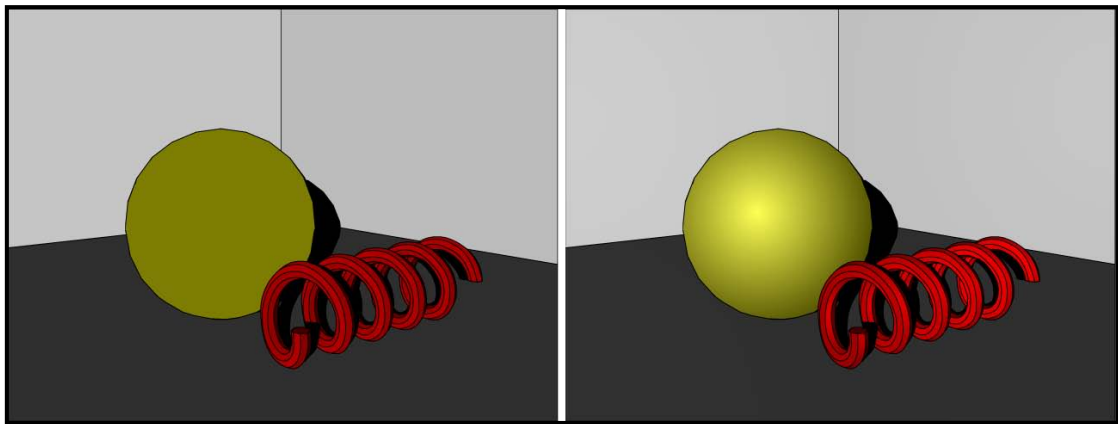


ภาพประกอบที่ 6.17.1 ภาพซ้ายคือการตั้งค่า Fill Style เป็น Single Color, ภาพขวาดังค่าเป็น Two Color

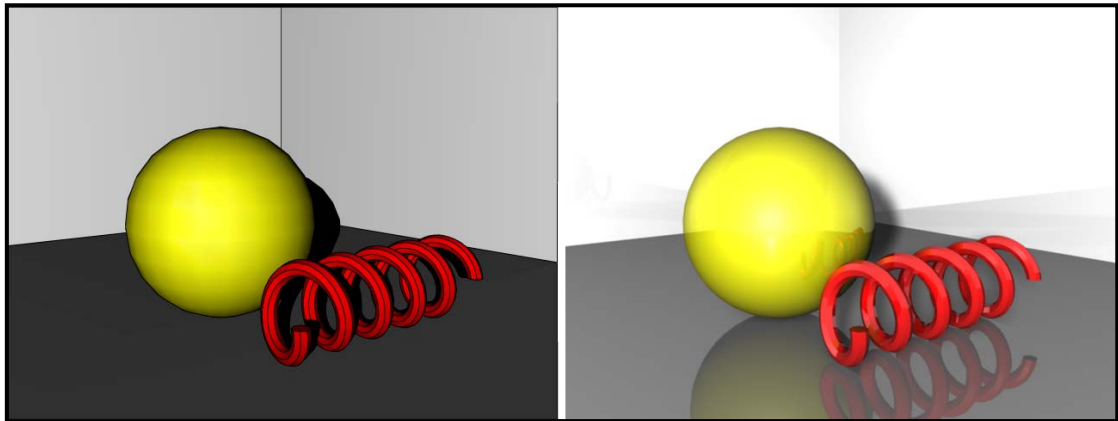




ภาพประกอบที่ 6.17.2 ภาพซ้ายคือการตั้งค่า Fill Style เป็น Four Color, ภาพขวาคือการตั้งค่าเป็น Full Color



ภาพประกอบที่ 6.17.3 ภาพซ้ายคือการตั้งค่า Fill Style เป็น Average Color, ภาพขวาคือการตั้งค่าเป็น Area Gradient

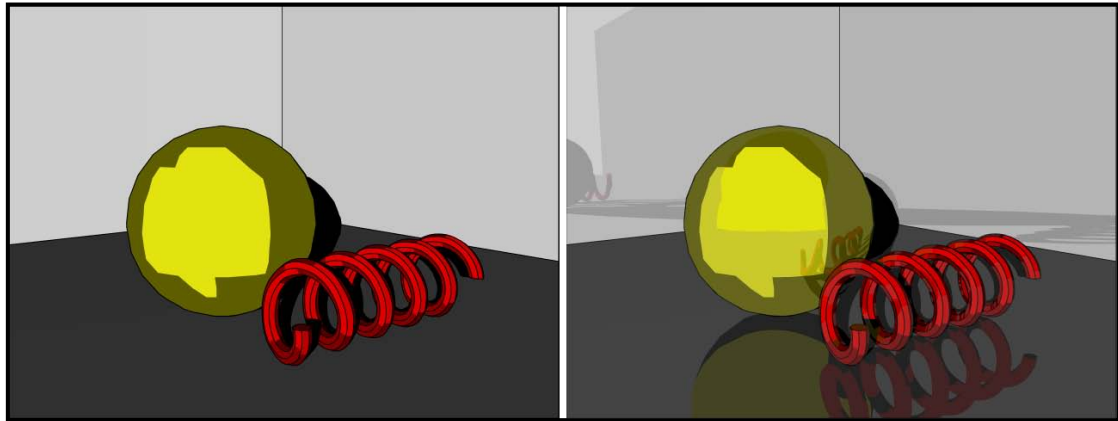


ภาพประกอบที่ 6.17.4 ภาพซ้ายคือการตั้งค่า Fill Style เป็น Mesh Gradient, ภาพขวาคือภาพที่ Render แบบธรรมดาไว้เปรียบเทียบ

จากภาพตัวอย่างจะสังเกตได้ว่าเราสามารถกำหนดคุณภาพในการถมสีได้หลายระดับ แต่ระดับจะให้ความรู้สึกแตกต่างกันไป ผู้ใช้สามารถเลือกใช้ได้ตามความเหมาะสมกับลักษณะของงานที่ต้องการ แต่ข้อควรระวังคือยิ่งมีการไล่โทนสีมากขึ้นเท่าไร การประมวลผลย่อมต้องใช้เวลาเพิ่มมากขึ้นเท่านั้น จนบางครั้งอาจจะมากกว่าการ Render ในแบบสามมิติก็เป็นได้



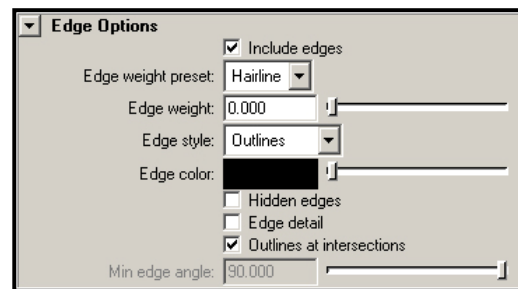
จากการกำหนด *Fill Style* แบบต่างๆ เมื่อเปรียบเทียบกับภาพที่ได้จากการประมวลผลแบบสามมิติแล้ว พบว่าสิ่งหนึ่งที่ยังขาดไปนั่นคือเรื่องของ *Reflections* ซึ่งการประมวลผลแบบ *Vector* สามารถทำได้เช่นกัน โดยทำเครื่องหมายถูกที่ *Check Box* หน้าหัวข้อ *Reflections* (ดูภาพประกอบที่ 6.15) โปรแกรมจะทำการคำนวณสร้างเงาสะท้อนให้กับภาพ แต่ข้อเสียที่ชัดเจนมากคือเรื่องของระยะเวลาประมวลผลที่เพิ่มขึ้นอีกพอสมควร



ภาพประกอบที่ 6.18 แสดงการเปิดปิดการทำงานของ *Reflection* เปรียบเทียบกัน

### Edge Options

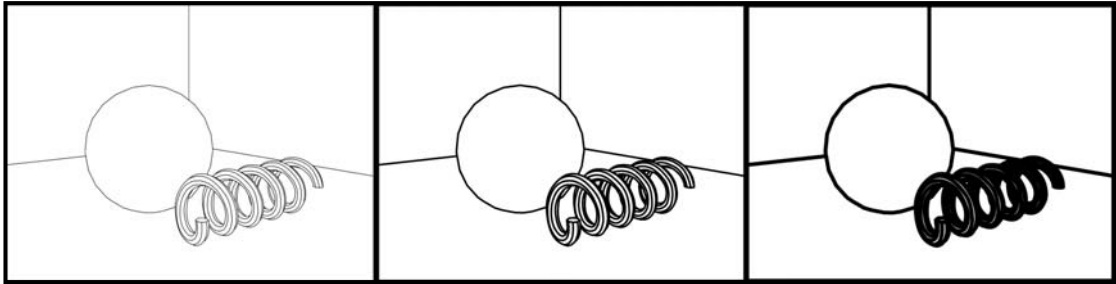
ส่วนนี้จะเป็นการทำงานกับเส้นขอบทั้งหมดของภาพ เมื่อเปิดการทำงานโดยการทำเครื่องหมายถูกหน้า *Include Edges* แล้ว เราสามารถปรับแต่งคุณลักษณะต่างๆ ของเส้นได้เช่น ขนาดของเส้น, ความกว้างของเส้น หรือลักษณะการ *Project* เส้น เป็นต้น



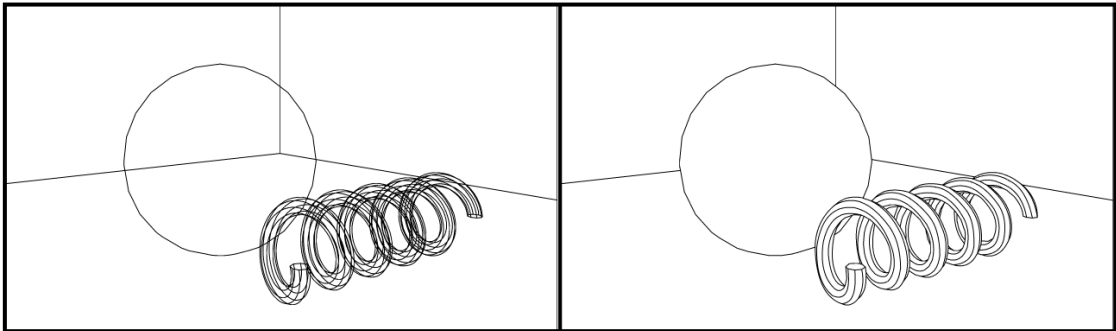
ภาพประกอบที่ 6.19 แสดงหน้าต่าง *Render Settings Window* ภายใต้หัวข้อ *Edge Options* ใน *Maya Vector*

เริ่มจากการกำหนดขนาดของเส้นสามารถทำได้ที่หัวข้อ *Edge Weight Preset* ผู้ใช้สามารถเลือกขนาดของเส้นได้จาก *Drop Down Options* หรือจะกำหนดให้เป็น *Custom* แล้วใส่ค่าที่ช่อง *Edge Weight* ก็ได้, ส่วนหัวข้อ *Edge Style* มีสองตัวเลือกคือ *Outlines* ที่โปรแกรมจะวาดเส้นขอบเฉพาะบริเวณเส้นรอบวัตถุ และ *Entire Mesh* ที่จะตัดเส้นตามเส้น *Segments* ของวัตถุ, *Edge Color* สามารถกำหนดสีของเส้นทั้งหมดได้ว่าต้องการให้เป็นสีอะไร, *Hidden Edges* จะซ่อนเส้นบริเวณที่อยู่ทับกัน ถึงแม้เราจะไม่ได้มสึใดๆ ให้กับวัตถุก็ตาม

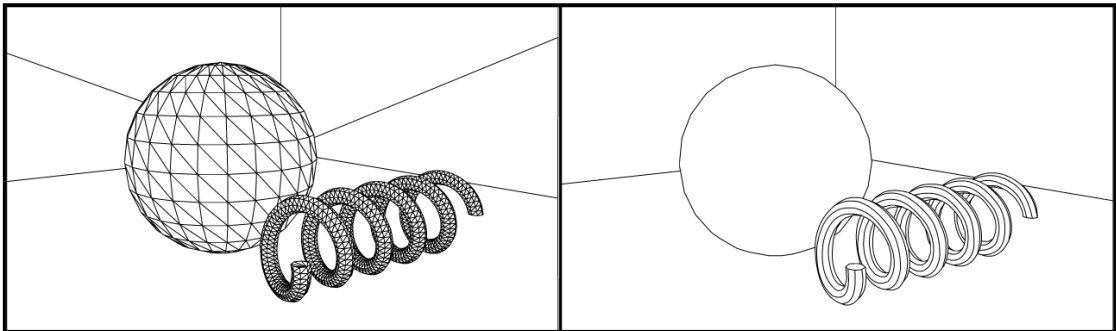
ดูภาพประกอบที่ 6.20 แสดงการตั้งค่า *Edge Options* ในรูปแบบต่างๆ



ภาพประกอบที่ 6.20.1 แสดงการตั้งค่า Edge Weight Preset = 1.0, 5.0 และ 10.0 จากซ้ายไปขวาตามลำดับ



ภาพประกอบที่ 6.20.2 แสดงการเปิดปิดการทำงานของ Hidden Edges



ภาพประกอบที่ 6.20.3 แสดงการตั้งค่า Edge Style เป็น Entire Mesh ในภาพซ้าย, และเป็น Outlines ในภาพขวา



THE UNIVERSITY OF  
CHIANGMAI  
THAILAND

THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY  
DEPARTMENT OF ANIMATION

ARUS KUNKHET  
315, LEVEL 3, ANIMATION DEPARTMENT  
THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY  
THE UNIVERSITY OF CHIANGMAI 50200  
THAILAND

TELEPHONE +66 53 941801 (315)  
FACSIMILE +66 53 893217

