**10-17 DOKU İZ DÜŞÜMLEME**

**Doku Redüksiyon (Küçültme) Paternleri**

Animasyon ve diğer uygulamalarda, cismin büyüklüğü (size) çoğunlukla değişir. Doku paternleri ile birlikte görüntülenen cisimler için o zaman doku iz düşümleme prosedürlerini cismin değiştirilmiş boyutlarına (dimensions) uygulamamız gerekir. Dokulu cismin büyüklüğü değiştiği zaman, doku paterni daha küçük bölgeye uygulanır, ve bu da doku distorsiyonlarına yol açabilir. Bundan kaçınmak için, cismin daha küçük ölçeklerde görüntüleneceği zamanlarda kullanılmak üzere, bir dizi **redüksiyon paternleri** oluşturabiliriz.

Genelde her redüksiyon paterni, önceki paternin yarı büyüklüğündedir. Örneğin, eğer iki-boyutlu 16\*16’lık paternimiz varsa, o zaman 8\*8, 4\*4, 2\*2, ve 1\*1 redüksiyon yapılmış büyüklüklerinde dört ilave patern kurabiliriz (set up). Bir cismin herhangi bir görünümü için distorsiyonları minimize etmek üzere uygun bir redüksiyon paternini kullanabiliriz. Bu redüksiyon paternlerine çoğu kez **MIP maplar** veya **mip maplar** denir. Buradaki mip terimi Latince olan *multum in parvo* deyiminden gelmektedir. Bunun anlamı “much on a small object” yani “küçük cisim üzerinde daha fazla” demektir.

**10-21 OpenGL DOKU FONKSİYONLARI**

OpenGL’de çok büyük doku fonksiyon takımı vardır. Çizgiye, yüzeye, uzaysal bölgenin iç hacmine ilişkin bir patern belirleyebiliriz veya bir paterni başka doku paterninin içine sokulacak bir alt-patern olarak belirleyebiliriz. Ve doku paternlerini çeşitli şekillerde uygulayabilir ve maniple edebiliriz. Ayrıca doku paternleri, ortam iz düşümlerini simülasyon yapmak için kullanılabilir. Bazı parametreler bir renk-tablo indeksi kullanılarak setlenebildikleri halde OpenGL doku rutinleri yalnız RGB (RGBA) renk modunda kullanılabilir.

OpenGL Çizgi-Dokusu Fonksiyonları

Tek-subscript renk arrayinde tanımlanan bir boyutlu RGBA doku paterninin parametreleri örneğin aşağıdaki gibi olsun.

***glTexImage1D (GL\_TEXTURE\_1D, 0, GL\_RGBA, nTexColors, 0, dataFormat, dataType, lineTexArray);***

***glEnable (GL\_TEXTURE\_1D);***

***glTexImage1D***fonksiyonundaki birinci argümanı bir boyutlu çizgi (line) cisim için bir doku arrayi tanımladığımızı göstermek için ***GL\_TEXTURE\_1D*** OpenGL sembolik sabitine setledik. Eğer belirlenen parametrelere sahip doku paternini bu sistemin destekleyeceğinden emin değilseniz, ***glTexImage1D*** nin birinci argümanı için ***GL\_PROXY\_TEXTURE\_1D*** sembolik sabitini kullanırız. Doku arrayinin elemanlarını tanımlamadan önce sistemi ilk sorgulamamıza izin verir, ve bir sonraki kısımda sorgulama prosedürlerini açıklayacağız. Bu durum, doku arrayının elemanlarını belirlemeden önce ilk olarak sistemi sorgulamamıza izin verir, ve ilerdeki kısımda bu sorgulama prosedürleri incelenecektir.

Bu örnek fonksiyonun ikinci ve beşinci argümanları için, **0** (sıfır) değerini kullanırız. Birinci **0** değeri (ikinci argüman) bu arrayin, başka bir doku arrayinin indirgemesi olmadığını gösterir. Beşinci argümanın **0** değerli olması, doku etrafında bir sınır (border) istemediğimizi gösterir. Eğer bu argümana **1** değeri atansaydı (bu olabilen başka tek olasılıktır) doku paterni kendi etrafında bir-piksellik sınır (border) olacak şekilde görüntülenecekti (display), bu sınır komşu doku paternleri ile bu paterni birleştirmek için kullanılır. Üçüncü argümandaki ***GL\_RGBA***değeri, doku paterninin her bir renginin dört RGBA değerleri ile belirlendiğini gösterir. Yalnız üç RGB renk değerlerini kullanabilirdik, ama RGBA değerleri bazen daha verimli olarak proses edilebilir, çünkü o renkler işlemci bellek sınırları ile uyuşurlar (align with). Tek şiddet (intensity) veya lüminansi içeren (parıltı), sayısız diğer renk spesifikasyonları mümkündür. ***nTexColors*** parametresine yani dördüncü argümana lineer doku paternindeki renklerin sayısını gösteren pozitif bir tamsayı atanmalıdır. Beşinci argüman için (sınır parametresi) **0** değeri listelendiğinden doku paternindeki renklerin sayısı 2’nin kuvveti olmalıdır. Eğer beşinci argümana **1** değeri atanmış olsaydı, o zaman doku paternindeki renklerin sayısı 2’nin kuvvetinin 2 fazlası olmak zorunda olacaktı. Bu iki sınır rengi, komşu paternlerle renk kaynaşması (blending=harmanlama) sağlamak için kullanılır. Bir\_subscript doku paternini 64+2 renge kadar renkle belirleyebiliriz (specify), ve bazı OpenGL gerçeklemeler daha büyük doku paternlerine izin verir. Doku renklerini ve sınır renklerini tanımlayan parametreler ***lineTexArray***’de saklanır (strored). Bu örnekte sınır (border) yoktur ve arraydeki dört elemanın her ardışık grubu doku paterninin bir renk bileşenini temsil eder. Bundan dolayı, ***lineTexArray***’deki elemanların sayısı ***4\*nTexColors*** değerindedir. Özel bir örnek olarak, 8 renkli bir doku paterni tanımlamak (define) istersek, doku arrayi 4\*8=32 eleman içermelidir.

***dataFormat*** *ve* ***dataType*** parametreleri, ***glDrawPixels*** ve ***glReadPixels*** fonksiyonlarındaki argümanlara benzer (Kısım 3-19’a bakınız). Doku arrayinde renk değerlerinin nasıl tanımlanacağını (specify) göstermek için bir OpenGL sembolik sabitini ***dataFormat***’a atarız. Örneğin, ***GL\_BGRA*** sembolik sabitini, renk bileşenlerini mavi, yeşil, kırmızı, alfa sırasında verebilmek için kullanabilirdik. BGRA veya RGBA data türünü (type) göstermek için ***dataType*** parametresine ***GL\_UNSIGNED\_BYTE*** OpenGL sabit değerini atayabiliriz. Seçtiğimiz data formatına bağlı olarak ***dataType*** parametresine atanabilecek diğer olası değerler ***GL\_INT ve GL\_FLOAT*** değerleridir.

Sahnedeki (scene) cisme bir dokunun çok sayıda kopyasını veya bu doku renklerinin herhangi bir sürekli (contiguous) alt-kümesini iz düşürebiliriz. Bir grup doku elemanı bir veya daha fazla piksel alanlarına iz düşürüldüğü zaman, doku elemanlarının sınırları genellikle piksel sınırlarının konumları ile hizalanmaz (align). Bir piksel alanı tek bir RGB (veya RGBA) doku elemanının sınırları içinde kapsanabilir veya bu alan birkaç doku elemanı ile örtüşebilir. Doku iz düşürmedeki hesaplamaları indirgemek için her piksele en yakın doku elemanının rengini vermek için aşağıdaki fonksiyonları kullanırız.

***glTexParametri (*** ***GL\_TEXTURE\_1D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST);***

***glTexParametri (*** ***GL\_TEXTURE\_1D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST);***

Birinci fonksiyon, sahnede belirtilen koordinat alanına sığdırmak için bir doku patern parçası genişletileceği zaman dokulama rutinleri tarafından kullanılır, ve ikinci fonksiyon ise doku paterni küçültüleceği zaman kullanılır. (OpenGL’deki bu iki doku operasyonu büyütme, MAG; ve küçültme, MIN, olarak adlandırılır.) Bir piksele en-yakın rengin atanması çok hızlı gerçekleştirilmesine rağmen, aliasing etkilerine yol açar. Örtüşen doku renklerinin lineer kombinasyonu olarak piksel rengini hesaplamak için ***GL\_NEAREST*** sembolik sabiti yerine ***GL\_LINEAR***’i kullanırız. Birkaç başka parametre değeri ***glTexParameter*** fonksiyonu ile setlenebilir, ve bir sonraki kısımda bu opsiyonlara göz atacağız.

Bir sahne için OpenGL doku paternlerinin belirlenmesi (specify) bir dereceye kadar yüzey-normal vektörlerini, RGB renklerini, veya diğer atributleri belirlemeye benzer. Paterni cisim ile ilişkili kılmamız gerekir, ama tek (single) renk setlemesinden farklı olarak, şimdi renk değerlerinin koleksiyonuna sahip oluruz. Bir-boyutlu doku uzayı için, renk değerleri, doku uzayı üzerinde 0.0-1.0 arasında değişen tek s koordinatı ile referanslanır (Kısım 10-16’ya bakınız). Bu yüzden doku paterni sahnedeki cisimlere, cisim konumlarına doku koordinat değerleri atanarak uygulanır. Bir-boyutlu doku uzayındaki muayyen bir s-koordinatı aşağıdaki komut ile seçilir.

***glTexCoord1\* (sCoord);***

Bu fonksiyona ilişkin izin verilen suffix kodları, ***sCoord*** doku koordinat parametresine ilişkin data türüne bağlı olarak, ***b*** (byte), ***s*** (short), ***i*** (integer), ***f*** (float), ve ***d*** (double) türündedir. Ve s-koordinat değeri bir arrayde veriliyorsa o zaman ***v*** suffixini kullanabiliriz. Renk ve diğer benzeri parametrelerde olduğu gibi, ***s*** koordinatı bir durum parametresidir ve sonradan tanımlı tüm dünya-koordinat konumlarına uygulanır. s koordinatın default değeri 0.0 dır.

Dünya-koordinat sahnesindeki konumlara bir lineer doku paternini iz düşürmek için, bir doğru parçasının uç-noktalarına s koordinatlarını atarız. Daha sonra doku renkleri çeşitli yollarla cisme uygulanabilir; ve OpenGL’in default yöntemi, cismin her piksel renk değerini doku paternindeki mukabil renk değeriyle çarpmaktır. Eğer bu doğrunun rengi beyaz (1.0, 1.0, 1.0, 1.0) ise, bu renk değeri bir sahnedeki cisimler için default renktir, o zaman bu çizgi (line) doku renkleriyle gösterilecektir.

Aşağıdaki örnekte, alternatif olarak yeşil ve kırmızı renklerini alan dört-elemanlı lineer doku paterni oluşturacağız. 0.0 dan 1.0 kadar değişen tüm (entire) doku paterni o zaman düzgün-doğru parçasına (straight-line segment) atanır. Bu doğru beyaz olduğundan, default olarak doku renkleriyle görüntülenecektir.

***GLint k;***

***GLubyte texLine [16]; //16-elemanlı doku arrayi.***

***/\* Doku paterni için iki yeşil eleman tanımla \*/***

***/\* Her doku rengi dört array konumunda belirtilir \*/***

***for (k=0; k<=2; k+=2)***

***{***

***texLine [4\*k] =0;***

***texLine [4\*k+1] =255;***

***texLine [4\*k+2] =0;***

***texLine [4\*k+3] =255;***

***}***

***/\* Doku paterni için iki kırmızı eleman tanımlayınız. \*/***

***for (k=1; k<=3; k+=2)***

***{***

***texLine [4\*k] =255;***

***texLine [4\*k+1] =0;***

***texLine [4\*k+2] =0;***

***texLine [4\*k+3] =255;***

***}***

***glTexParametri (*** ***GL\_TEXTURE\_1D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST);***

***glTexParametri (*** ***GL\_TEXTURE\_1D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST);***

***glTexImage1D (GL\_TEXTURE\_1D, 0, GL\_RGBA, 4, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, texLine);***

***glEnable (GL\_TEXTURE\_1D);***

***/\* Tüm doku renk alanının bir doğru parçasına atanması \*/***

***glBegin (GL\_LINES);***

***glTexCoord1f (0.0);***

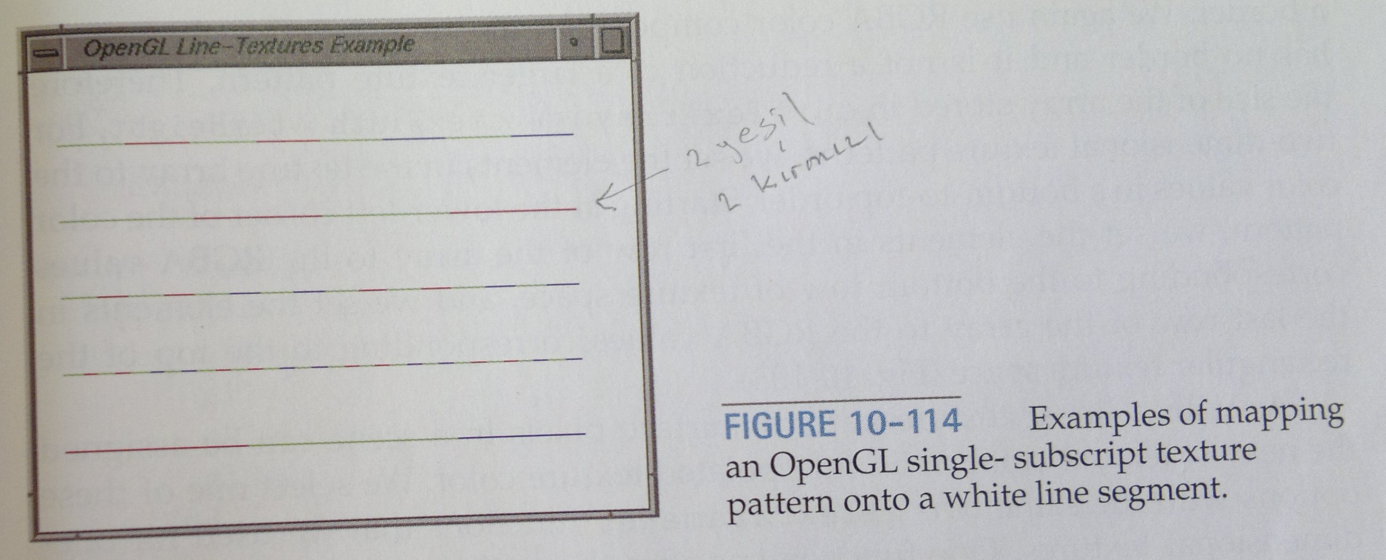
***glVertex3fv (endPt1);***

***glTexCoord1f (1.0);***

***glVertex3fv (endPt2);***

***glEnd ( );***

***glDisable (GL\_TEXTURE\_1D);***

****

**Şekil 10-14** OpenGL tek-subscriptdoku paterninin beyaz doğru üzerine iz düşürülmesi örneği.

Bu doğru parçası, doğru boyunca uzanan yeşil ve kırmızı kısımlar halinde görüntülenir. s koordinatlarına herhangi bir değer atayabiliriz. Örneğin, doku paterninin ortadaki kırmızı ve yeşil renkleri aşağıdaki deyimlerle (statements) bu doğru üzerine iz düşürülebilir.

***glBegin (GL\_LINES);***

***glTexCoord1f (0.25);***

***glVertex3fv (wcPt1);***

***glTexCoord1f (0.75);***

***glVertex3fv (wcPt2);***

***glEnd ( );***

Bu yüzden, bu doğrunun birinci yarısı kırmızı ve ikinci yarısı da yeşildir. s değerlerini 0.0-1.0 alanının dışında da seçebiliriz. Eğer s=-2.0 değerini doğrunun bir uç noktasına ve s=2.0 değerini de diğer uç noktasına atamış olsaydık, bu doku paterni bu doğru üzerine dört defa iz düşürülmüş olacaktı. O zaman bu doğru 16 yeşil kısım ve 16 kırmızı kısım halinde görüntülenmiş olacaktı. Birim aralığın dışındaki s koordinat değerleri için, s değerleri 0 veya 1.0 değerine kenetlenmesi gerektiği belirtilmedikçe (clamped) tamsayı kısımları göz ardı edilir. Bazı olası doğru paternleri, iki yeşil renk ve iki kırmızı renge ilişkin RGB değerlerini içeren array kullanılarak, Şekil 10-114’te gösterilmiştir.

OpenGL doku paterni ile çok sayıda parametre ve opsiyon mevcuttur. Ama, OpenGL doku rutinlerinin bu özelliklerine girmeden önce, ilk olarak iki-boyutlu ve üç-boyutlu doku paternlerini üretmek için gerekli temel fonksiyonları inceleyeceğiz.

OpenGL Yüzey-Dokusu Fonksiyonları

Bizim bir-boyutlu doku patern örneğimizdeki fonksiyonlara benzer fonksiyonlar kullanarak iki-boyutlu RGBA doku uzayı için parametreleri setleyebiliriz.

***glTexImage2D (GL\_TEXTURE\_2D, 0, GL\_RGBA, texWidth, texHeight, 0, dataFormat, dataType, surfTexArray);***

***glEnable (GL\_TEXTURE\_2D);***

Buradaki tek fark, üç-subscript doku arrayi için hem genişliği (width=number of coums) hem de yüksekliği (number of rows) belirlememizin gerekmesidir. Hem genişlik hem de yükseklik sınır (border) hariç 2’nin kuvveti veya sınır ile birlikte 2’nin kuvvetinin iki fazlası olmalıdır. Yine RGBA renk değerlerini kullanırız, ve bu paternin sınırının (border) olmadığını ve daha büyük doku paterninin bir redüksiyonu olmadığını belirtiyoruz. Bundan dolayı, ***surfTexArray’***desaklanmış arrayin boyutu (size) ***4\*texWidth\*texHeight*** değerindedir. İki-boyutlu doku paternleri için, doku arrayindeki elemanları bottom-to-top sırasında renk değerlerine setleriz. Renk paterninin alt-sol köşesinden başlayarak, arrayin birinci satırındaki elemanları doku uzayının taban (bottom) satırına karşı düşen RGBA değerlerine setleriz, ve arrayin son satırındaki elemanları da dikdörtgen doku uzayının tepesine (top) karşı düşen RGBA değerlerine setleriz (Şekil 10-103).

Lineer doku paterninde olduğu gibi, sahnedeki yüzey piksellerine en yakın doku rengi veya interpole edilmiş doku rengi atanabilir. Bu opsiyonlardan birini, bir-boyutlu dokular için kullandığımız iki ***glTexParameter*** fonksiyonu ile seçeriz. Bir fonksiyon, koordinat alanına bir doku paterni sığdırılmak üzere büyütüleceği zaman kullanılacak opsiyonu belirler, ve diğer fonksiyon ise patern redüksiyonu ile kullanılması zorunlu opsiyonu belirtir. Ve iki-boyutlu doku paterni bir doğrultuda uzatılabilir (stretch) ve diğer doğrultuda sıkıştırılabilir (compress). Örneğin aşağıdaki deyimler (statements) en yakın doku rengini kullanarak projeksiyonlu yüzey konumlarını görüntülemek için dokulama rutinlerine emreder.

***glTexParametri (*** ***GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST);***

***glTexParametri (*** ***GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST);***

Yüzey piksellerine interpolasyonlu doku rengini atamak için, ***GL\_NEAREST*** yerine ***GL\_LINEAR*** sembolik sabitini kullanırız.

İki-boyutlu doku uzayındaki bir koordinat konumu aşağıdaki ile seçilir.

***glTexCoord2\* (sCoord, tCoord);***

Doku uzayı, 0.0-1.0 arsında değişen koordinat değerleri ile patern referanslanarak normalize edilir. Ama, yüzey üzerinde bir paternin replikasını (kopyasını) yapmak için herhangi bir doku koordinat değerini kullanabiliriz. Doku koordinatları çeşitli formatlarda belirlenebilir, ve b, s, i, f, veya d suffix kodu ile bir data formatı gösterilebilir. Eğer doku koordinatları bir dizi halinde verilirse o zaman v suffixini de ekleriz.

İki-boyutlu doku uzayına ilişkin OpenGL fonksiyonlarını göstermek için, aşağıdaki kod dilimi 32\*32’lik bir patern kurar ve onu dörtken (quadrilateral) yüzey üzerine iz düşürür. Her bir doku rengi dört RGBA bileşeni ile belirlenir, ve bu paternin sınırı (border) yoktur.

***GLubyte texArray [32][32][4];***

***/\*Next: assign the texture color components to texArray. \*/***

***/\* Select nearest-color option. \*/***

***glTexParametri (*** ***GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST);***

***glTexParametri (*** ***GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST);***

***glTexImage2D (GL\_TEXTURE\_2D, 0, GL\_RGBA, 32, 32, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, texArray);***

***glEnable (GL\_TEXTURE\_2D);***

***/\* Assign the full range of texture colors to a quadrileteral. \*/***

***glBegin (GL\_QUADS);***

***glTexCoord2f (0.0, 0.0); glVertex3fv (vertex1);***

***glTexCoord2f (1.0, 0.0); glVertex3fv (vertex2);***

***glTexCoord2f (1.0, 1.0); glVertex3fv (vertex3);***

***glTexCoord2f (0.0, 1.0); glVertex3fv (vertex4);***

***glEnd ( );***

***glDisable (GL\_TEXTURE\_2D);***

OpenGL Hacim-Dokusu Fonksiyonları

Üç-boyutlu doku uzayına ilişkin fonksiyonlar, iki-boyutlu doku uzaylarına ilişkin fonksiyonların basit uzantılarıdır. Örneğin sınırı (border) olmayan dört-subscript RGBA doku arrayi aşağıdaki fonksiyonlarla setlenebilir.

***glTexImage3D (GL\_TEXTURE\_3D, 0, GL\_RGBA, texWidth, texHeight, texDepth, 0, dataFormat, dataType, volTexArray);***

***glEnable (GL\_TEXTURE\_3D);***

RGBA doku renkleri, ***4\* texWidth\* texHeight\* texDepth*** eleman içeren ***volTexArray***’de saklanır. Ve Arrayin genişliği, yüksekliği, ve derinliği ya 2’nin kuvveti olmalı veya 2’nin kuvvetinin 2 fazlası olmalıdır.

Aşağıdaki deyimler yardımıyla en yakın (nearest) doku rengini kullanarak pikselleri görüntüleriz.

***glTexParametri (*** ***GL\_TEXTURE\_3D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST);***

***glTexParametri (*** ***GL\_TEXTURE\_3D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST);***

Lineer interpolasyonlu doku renkleri için ***GL\_NEAREST*** yerine ***GL\_LINEAR*** değerini koyarız.

Üç-boyutlu doku koordinatları aşağıdaki fonksiyonla seçilir.

***glTexCoord3\* (sCoord, tCoord, rCoord);***

Doku uzayında seçilmiş her konum, o zaman dünya-koordinat sahnesindeki uzaysal koordinat konumu ile ilişkilendirilmiş olur.

Doku Paternleri için OpenGL Renk Opsiyonları

Doku uzayına ilişkin elemanlar çok farklı şekillerde belirtilebilir. ***glTexImage1D, glTexImage2D,*** ve ***glTexImage3D*** fonksiyonlarındaki üçüncü argüman, paternin her elemana ilişkin renk bileşenlerinin genel formatını ve sayısını belirlemek için kullanılır. Bu belirleme (specification) için yaklaşık 40 sembolik sabit mevcuttur. Örneğin her doku elemanı RGBA değerlerinin bir seti, RGB değerlerinin bir seti, tek bir alfa değeri, tek bir kırmızı yoğunluk (intensity) değeri, tek bir parlaklık (luminance) değeri, veya alfa değeri ile çift yapılmış bir parlaklık (luminance) değeri olabilir. Ayrıca, bazı sabitler de bit boyunu belirler. Örneğin 3 biti kırmızı bileşene, 3 biti yeşil bileşene, ve 2 biti de mavi bileşene olmak üzere GL\_R3\_G3\_B2 OpenGL sabiti, bir baytlık (8-bit) RGB rengini belirtir.

Doku fonksiyonlarındaki ***dataFormat*** parametresi, doku elemanlarına ilişkin spesifik formatı belirtmek için kullanılır. Bu durum her doku elemanını renk tablosunda bir indeks olarak, tek bir alfa değeri olarak, tek bir parlaklık (luminance) değeri olarak, parlaklık-alfa değer çifti olarak, RGB bileşenlerinin biri için tek bir şiddet değeri olarak, üç RGB bileşenleri olarak, veya BGRA sırasında RGBA spesifikasyonunun dört bileşenini belirlememize izin verir. Ve ***dataType*** parametresine **GL\_BYTE, GL\_INT, GL\_FLOAT** gibi bir değer, veya hem data türünü hem de bit boyutunu (size) belirleyen bir sembolik sabit atanır. Data-tür parametresi için 20 sembolik sabit takımından bir değer seçebiliriz.

OpenGL Doku-İz Düşüm Opsiyonları

Doku değerleri halihazırdaki cismin renk bileşenleriyle birleştirilerek (combine) veya doku değerleri cisim renginin yerine koyularak (replace) doku elemanları bir cisme uygulanabilir. Aşağıdaki fonksiyon ile bir doku iz düşürme yöntemini seçebiliriz.

***glTexEnvi (GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, applicationMethod);***

Eğer ***applicationMethod*** parametresine **GL\_REPLACE** değeri atanırsa, o zaman doku rengi, parlaklık, renk şiddeti, veya alfa değeri mukabil cisim değerinin yerini alır. Örneğin alfa değerlerinden oluşan bir doku paterni cismin alfa değerlerinin yerini alır. Benzer yerine koyma opersyonları, tek lüminans veya şiddet (intensity) değeri ile belirlenen doku paterni ile de kullanılır. Yeşil şiddet değerlerinden oluşan bir patern, cisim renginin yeşil bileşenlerinin yerini alır.

***applicationMethod*** parametresine **GL\_MODULATE** değerinin atanması, cisim renk değerlerinin “modülasyonu” ile sonuçlanır. Yani, halihazırdaki cisim değerleri, doku değerleri ile çarpılır. Spesifik sonuçlar doku paternindeki elemanların formatına bağlıdır, öyle ki örneğin alfa değerleri alfa değerlerini ve şiddet değerleri de şiddet değerlerini modüle eder. Doku paternine ilişkin default uygulama yöntemi ***GL\_MODULATE*** dir. Eğer bir cismin rengi beyaz ise (cismin default rengidir) modülasyon operasyonu yerine koyma operasyonu ile aynı sonucu üretir, bu da doku patern elemanlarının nasıl belirlendiğine bağlıdır.

Doku iz düşürme operasyonları için ***GL\_DECAL*** sembolik sabitini de kullanabiliriz, bu da daha sonra saydamlık katsayıları olarak RGBA alfa değerlerini kullanır. Bu durumda, cisim sanki zemindeki (background) doku rengi ile saydammış gibi muamele edilir. Eğer doku paterni yalnız RGB değerlerini içerirse, alfa bileşeni yoksa, doku rengi cisim renginin yerini alır. Ayrıca, doku paterninin yalnız alfa değerlerini içermesi gibi bazı durumlarda, decal operasyon tanımlanmaz.

***applicationMethod*** parametresine ***GL\_BLEND*** sabitini atadığımız zaman, doku rutinleri aşağıdaki fonksiyon ile belirlenmiş rengi kullanarak renk harmanlaması (blending) gerçekler.

***glTexEnv\* (GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_COLOR, blendingColor);***

Harmanlama-renginin data türüne göre i veya f suffixini ekleriz. Ve eğer harmanlama rengi bir arrayde verilirse, v suffixi de eklenebilir.

OpenGL Doku Çoğullama (wrapping)

Doku uzayındaki koordinat değerleri 0.0-1.0 alanının dışında olduğu zaman, aşağıdaki komutu kullanarak doku arrayinde tanımlanan paterlerinin çoğullaması (replicate) yapılabilir.

***glTexParameter\* (texSpace, texWrapCoord, GL\_REPEAT);***

Doku-uzayı koordinat değerinin yalnız kesirli (fractional) kısmını kullanarak paternler çoğullanır (replicate). ***texSpace*** parametresine ***GL\_TEXTURE\_1D***, ***GL\_TEXTURE\_2D,***  veya ***GL\_TEXTURE\_3D*** sembolik değerlerinden biriatanır, ve ***texWrapCoord*** parametresi ***GL\_TEXTURE\_WRAP\_S***, ***GL\_TEXTURE\_WRAP\_T***, veya ***GL\_TEXTURE\_WRAP\_R***’den birini kullanarak bir doku uzayı koordinatı belirler.

Birim aralığa bir doku koordinatını kenetlemek (clamp) için, ***GL\_REPEAT*** yerine ***GL\_CLAMP*** sembolik sabitini kullanırız. Eğer bir kenetlenmiş doku koordinatı 1.0’dan daha büyük değere sahipse, ona 1.0 değeri atanır. Benzer şekilde, 0.0’dan daha küçük bir kenetlenmiş doku koordinatına 0.0 değeri atanır. Muayyen bir doku uzayındaki koordinatlar için tekrarlama (repeating) ve kenetlemenin (clamping) herhangi bir kombinasyonunu belirleyebiliriz. Tüm koordinatlar için default değer ***GL\_REPEAT*** dir.

OpenGL Doku Paternlerini Frame Bufferdan Kopyalama

Bir orijinal patern veya subpatern, frame bufferda saklanmış değerlerden elde edilebilir. Aşağıdaki fonksiyon, bir RGBA piksel değerler bloğunu kullanarak halihazırdaki doku durumuna ilişkin iki-boyutlu bir patern kurar.

***glCopyTexImage2D (GL\_TEXTURE\_2D, 0, GL\_RGBA, x0, y0, texWidth, texHeight, 0,);***

Argüman listesindeki iki 0 değeri bu paternin bir redüksiyon olmadığını ve bir sınıra (border) sahip olmadığını gösterir. Bufferın sol-alt köşesine göre tanımlanmış (x0, y0) frame buffer konumu, ***texWidth\* texHeight*** boyutlu piksel renkleri bloğunun sol-alt köşesini referanslar.

Doku subpaterni olarak bir piksel renkleri bloğu elde etmek için benzer bir fonksiyon vardır.

***glCopyTexSubImage2D (GL\_TEXTURE\_2D, 0, xTexElement, yTexElement, x0, y0, texSubWidth, texSubHeight);***

Bu piksel değerleri bloğu, (***xTexElement, yTexElement)*** doku-elemanı konumundaki current paterne koyulur. ***texSubWidth*** ve ***texSubHeight*** parametreleri piksel bloğunun boyutunu (size) verir. Bu bloğun sol-alt köşesi, frame bufferın (x0, y0) konumunu gösterir.

OpenGL Doku-Koordinat Arrayleri

Renk datasında, yüzey-normal vektörlerinde, ve poligon kenar bayraklarında olduğu gibi; doku koordinatlarını liste halinde belirleyebiliriz, ve köşe arrayleri ile birleştirebilir veya ilişkili kılabiliriz (Kısım 3-17 ve 4-3’e bakınız).

***glEnableClientState (GL\_TEXTURE\_COORD\_ARRAY);***

***glTexCoordPointer (nCoords, dataType, offset, texCoordArray);***

***nCoords*** parametresine 1, 2, 3, veya 4 değeri atanır, bu da doku paterninin boyutunu (dimensionality) gösterir. Default değer 4, homojen koordinat formunda doku uzayını referanslamak için kullanılır, çünkü doku-uzayındaki konum ilk üç koordinat değeri dördüncüye bölünerek hesaplanır. Bu biçim örneğin doku paterni bir perspektif fotoğraf ise yararlıdır. ***dataType*** parametresine ***GL\_SHORT, GL\_INT, GL\_FLOAT*** (default değerdir)veya ***GL\_DOUBLE*** sabit değeri atanır. ***texCoordArray*** arrayindeki koordinat konumları arasındaki byte ofseti, ***offset*** parametresinde belirlenir. Bunun default değeri 0 dır.

OpenGL Doku Paternlerinin Adlandırılması

Çoğu kez bir uygulamada birkaç doku paterni kullanmak faydalıdır. Bu yüzden OpenGL, çok sayıda adlandırılmış doku paternlerinin oluşturulmasına izin verir. Daha sonra, herhangi bir anda hangi adlı dokunun uygulanacağını belirleriz. Bu durum her seferinde ***glTexImage*** fonksiyonunu çağırmaktan daha verimli bir yöntemdir. Çünkü ***glTexImage*** fonksiyonunu her çağırma, muhtemelen bir data dosyasındaki renk değerlerinden, o paternin yeniden oluşturulmasını gerektirir. Bir doku paternini isimlendirmek için, o patern tanımlanmadan (define) önce bir pozitif (işaretsiz=unsigned)tamsayı seçeriz. Örnek olarak aşağıdaki deyimler, bizim daha önceki örneğimizden gelen yeşil ve kırmızı doğru paternini duku (texture) 3 olarak adlandırır ve daha sonra da aktif yapar.

***glBindTexture (GL\_TEXTURE\_1D, 3);***

***glTexImage1D (GL\_TEXTURE\_1D, 0, GL\_RGBA, 4, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, texLine);***

***glBindTexture (GL\_TEXTURE\_1D, 3);***

Birinci ***glBindTexture*** deyimi paterni adlandırır, ve ***glBindTexture***’e ikinci çağırma **mevcut doku durumu** (current texture state) olarak o paterni belirler. Eğer çoklu doku paternleri oluşturmuş isek, sahnedeki bir nesneye uygulama yapmak için o dokuyu aktiflemek üzere ***glBindTexture***’i başka bir patern adıyla tekrar çağırabilirdik. İki-boyutlu veya üç-boyutlu bir patern için ***glBindTexture*** fonksiyonunun birinci argümanını ***GL\_TEXTURE\_2D*** veya ***GL\_TEXTURE\_3D*** yaparız. Bir doku adı ilk çağrıldığı zaman, patern parametrelerine ilişkin default değerler kullanılarak bir doku paterni oluşturulur.

Mevcut bir veya daha fazla doku paterni aşağıdaki komut ile silinir (delete).

***glDeleteTextures (nTextures, texNamesArray);***

***nTextures***parametresi, ***texNamesArray*** arrayinde listelenen patern adlarının sayısını belirtir.

Kullanmış olduğumuz adları izlemek zorunda kalmamamız için paternin adını OpenGL’in seçmesine de izin verebiliriz. Örneğin

***static GLuint texName;***

***glGenTextures (i, texName);***

***glBindTexture (GL\_TEXTURE\_2D, texName);***

Örnek olarak, aşağıdaki kod altı kullanılmamış doku adlarının bir listesini elde eder ve bir patern üretmek için onlardan birini kullanır.

***static GLuint texNamesArray [6];***

***glGenTextures (1, texNamesArray [3]);***

***glBindTexture (GL\_TEXTURE\_2D, texNamesArray [3]);***

Var olan bir pattern için bir doku adının kullanılmakta olup olmadığını araştırmak için OpenGL’de bir sorgulama komutu vardır:

***glIsTexture (texName);***

***texName***, mevcut paternin adı ise bu fonksiyon, ***GL\_TRUE*** değerini geri döndürür, aksi halde ***GL\_FALSE*** değerini geri döndürür. Eğer bir hata meydana gelirse, yine ***GL\_FALSE*** değeri geri döndürülür.

OpenGL Doku Subpatternleri

Doku paterni bir kere tanımlanınca (define), orijinal paternin herhangi bir parçasını veya tümünü modifiye etmek için subpatern adı verilen başka bir patern oluşturabiliriz. Subpaterndeki doku değerleri, orijinal paterndeki belirlenmiş değerlerin yerini alır. Bu subpatern genellikle yeni elemanları olan bir dokuyu yeniden oluşturmaktan daha verimli bir süreçtir. Örneğin aşağıdaki fonksiyon, sınırı (border) olmayan ve daha büyük bir paternin redüksiyonu olmayan, iki-boyutlu dokunun bir bölümünün yerine koyulacak RGBA renk değerleri setini belirler.

***glTexSubImage2D (GL\_TEXTURE\_2D, 0, xTexElement, yTexElement, GL\_RGBA, texSubWidth,***

***texSubHeight, 0, dataFormat, dataType, subSurfTexArray);***

***xTexElement*** ve ***yTexElement*** parametreleri, orijinal patern içinde bir doku elemanının tamsayı koordinat konumunu seçmek için kullanılır. Burada (0, 0) konumu, paternin sol-alt köşesindeki doku elemanını referanslar. Bu subpatern, (***xTexElement, yTexElement)*** konumunda sol-alt köşesi olan orijinal paterne yapıştırılır (paste). ***texSubWidth*** ve ***texSubHeight*** parametreleri subpaternin boyutunu (size) verir. Bir RGBA doku paternine ilişkin ***subSurfTexArray*** arrayindeki renk elemanlarının sayısı ***4\*texSubWidth\*texSubHeight*** dır. Diğer parametreler ***glTexImage*** fonksiyonundaki ile aynıdır, ve benzer subpaternler bir-boyutlu veya üç-boyutlu dokular için kurulabilir.

OpenGL Doku Redüksiyon Paternleri

Küçültülmüş cisim boyutları (sizes) için, **mip map**lar olarak adlandırılan bir dizi redüksiyon paternleri oluşturmak üzere OpenGL rutinlerini kullanabiliriz (Kısım 10-17’ye bakınız). Küçültülmüş paternler oluşturmanın bir yolu, ***glTexImage*** fonksiyonunu, fonksiyondaki ikinci argüman (seviye sayısı=level number) için daha büyük tamsayı değerleri kullanarak tekrar tekrar çağırmaktır. Orijinal patern, 0 redüksiyon-seviye numarası ile referanslanır (adlandırılır). Orijinal paternin yarı boyutuna sahip redüksiyon paternine 1-seviye numarası, ikinci yarı boyutlu redüksiyon paternine 2-seviye numarası, vesaire atanır. Keza seviye numarasını 1 veya daha büyük değere setlediğimiz zaman ***copyTexImage*** fonksiyonu bir redüksiyon paterni doğurur.

Alternatif olarak, OpenGL’e redüksiyon paternlerini otomatik olarak ürettirebiliriz. Örneğin RGBA redüksiyon paternleri, aşağıdaki GLU fonksiyonu kullanılarak, 16\*16 yüzey dokusu için elde edilebilir.

***gluBuild2DMipmaps (GL\_TEXTURE\_2D, GL\_RGBA, 16, 16, GL\_RGBA, GL\_UNSIGNED\_BYTE,***

***surfTexArray);***

8\*8, 4\*4, 2\*2, ve 1\*1 küçültülmüş boyutlardadört patern seti bu fonksiyon ile üretilir. Aşağıdaki fonksiyon kullanılarak da seçilmiş redüksiyonlar üretilebilir.

***gluBuild2DMipmapLevels (GL\_TEXTURE\_2D, GL\_RGBA, 16, 16, GL\_RGBA, GL\_UNSIGNED\_BYTE, 0,***

***minLevel, maxLevel, surfTexArray);***

Bu fonksiyon ***minLevel*** ve ***maxLevel*** parametreleri ile belirlenen bir seviye numaraları alanı için redüksiyon paternleri üretir. Her bir durumda, mip-maplar, 0 seviye numarasında belirlenmiş (specified) mevcut doku patern için kurulurlar.

***glTexParameter*** fonksiyonunu ve ***GL\_TEXTURE\_MIN\_FILTER*** sembolik sabitini kullanarak redüksiyon paternlerinden piksel renklerini belirleyen bir yöntem seçeriz. Örnek olarak, aşağıdaki fonksiyon iki-boyutlu doku paterni için iz düşüm prosedürünü belirler.

***glTexParameter (*** ***GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST\_MIPMAP\_NEAREST);***

Bu fonksiyon belirtir ki, piksel boyunu (size) en çok eşleyen (***MIPMAP\_NEAREST)*** redüksiyon paternini doku rutinleri kullanmalıdır. Daha sonra o redüksiyon paternindeki en yakın (***GL\_NEAREST*** ) doku elemanının rengi bir piksele atanır. ***GL\_LINEAR\_MIPMAP\_NEAREST*** sembolik sabiti ile, en yakın redüksiyon paterninden gelen doku renklerinin linear kombinasyonunu belirleriz. ***GL\_NEAREST\_MIPMAP\_ LINEAR*** (default değerdir) sembolik sabiti ile de, piksel boyuna (size) en yakın redüksiyon paterlerinin her birindeki en yakın doku elemanlarından hesaplanmış bir ortalama renk belirleriz. Ve en yakın boyutlu (closest-size) redüksiyon paternlerinden gelen doku renklerinin lineer bir kombinasyonunu kullanarak, ***GL\_LINEAR\_MIPMAP\_ LINEAR*** sembolik sabiti bir pikselin rengini hesaplar.

OpenGL Doku Sınırları (Borders)

Çoklu dokular veya tek dokunun çoklu kopyaları bir cisme uygulandığı zaman, doku renklerinin lineer interpolasyonu ile piksel renkleri hesaplanırsa, komşu paternlerin kenarlarında aliasing etkileri görülebilir. Bundan her doku paternine bir sınır (border) katarak kurtulabilinir. Sınır renkleri, doku kenar renklerini komşu paternde eşler.

Doku sınır rengi birkaç şekilde belirlenebilir. Komşu paterndeki renk değeri, ***glTexSubImage*** fonksiyonu kullanılarak diğer paternin sınırına kopyalanır, veya sınır renkleri ***glTexImage*** fonksiyonu ile belirtilen doku arrayine doğrudan atanabilir. Başka bir opsiyon ***glTexParameter*** rutinini kullanarak sınır rengini setlemektir. Örneğin aşağıdaki fonksiyon ile iki-boyutlu patern için bir sınır rengi atayabiliriz.

***glTexParameterfv (GL\_TEXTURE\_2D, GL\_TEXTURE\_BORDER\_COLOR, borderColor);***

Burada ***borderColor*** parametresine RGBA renk bileşenlerinin dört-elemanlı seti atanır. Default sınır (border) rengi (0.0, 0.0, 0.0, 0.0) siyah rengidir.

OpenGL Vekil (Proxy) Dokuları

***glTexImage*** fonksiyonlarının herhangi birinde, sembolik sabiti doku vekiline (proxy) setleyebiliriz. Bu sabitin amacı, bu paterni halledecek (handle) yeterli kaynağın olup olmadığı anlaşılıncaya kadar doku paterninin tanımını (definition) tutmaktır. İki-boyutlu patern için vekil sabiti ***GL\_PROXY\_TEXTURE\_2D*** dir, ve benzer sabitler lineer ve volümetrik paternler için de mevcuttur. Doku vekilini bir kere kurduğumuzda, spesifik parametre değerlerini alıp alamayacağını (accommodate) belirlemek için ***glGetTexLevelFunction*** fonksiyonunu kullanırız.

Doku vekilini kullanan bir örnek olarak, aşağıdaki deyimler iki-boyutlu patern için belirlenmiş yüksekliğin (height) kullanılıp kullanılamayacağını tayin etmek için sistemi sorgular.

***GLint texHeight;***

***glTexImage2D (GL\_PROXY\_TEXTURE\_2D, 0, GL\_RGBA12, 16, 16, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE,***

***NULL);***

***glGetTexLevelParameteriv (GL\_PROXY\_TEXTURE\_2D, 0, GL\_RGBA12, GL\_TEXTURE\_HEIGHT,***

***&texHeight);***

Eğer sistem gerekli patern yüksekliğini (bu durumda 16 dır) sindiremezse (accommodate), ***texHeight*** parametresinde 0 değerini geri döndürür. Aksi halde, geri döndürülen değer istenilen değerdir. Diğer patern parametreleri de benzer şekilde ***GL\_TEXTURE\_WIDTH***, ***GL\_TEXTURE\_DEPTH, GL\_TEXTURE\_BORDER,*** ve ***GL\_TEXTURE\_BLUE\_SIZE*** sembolik sabitleri kullanılarak sorgulanabilir. Her bir durumda geri döndürülen 0 değeri, ***glTexImage*** fonksiyonundaki sindiremeyen istek yapılmış parametre değerini gösterir. Kayan-noktalı data değerleri için i sufix kodunun yerine f kodu koyulur.

Önerilen doku için olumlu (affirmative) cevap almış olsak da, o paterni belleğe yine de saklayamamış olabiliriz. Bu durum, mevcut belleği işgal eden başka bir patern olduğu zaman ortaya çıkar.

Dörtgen Yüzeylerin Otomatik Dokulanması

Belirli uygulamalarda otomatik olarak doku koordinatlarını üretmek için OpenGL’de rutinler mevcuttur. Bu özellik çoğunlukla bir cismin yüzey koordinatlarını doğrudan belirlemenin (determine) zor olduğu durumlarda faydalıdır, ve bu rutinleri dörtgen yüzeylere uygulamak için GLU fonksiyonu vardır.

Dörtgen yüzeye bir doku paternini iz düşürmek için ilk olarak doku yüzeyine ilişkin parametreleri setleriz. Daha sonra, aşağıdaki fonksiyonu çağırırız ve Kısım 8-6’da açıklandığı gibi, dörtgen cisim tanımlanır (define).

***gluQuadricTexture (quadSurfObj, GL\_TRUE)***

Bu fonksiyondaki ***quadSurfObj*** parametresi dörtgen cismin adıdır. Eğer dörtgen yüzeyin dokulanmasını aktif olmaktan çıkarmak istersek, ***GL\_TRUE*** sembolik sabitini ***GL\_FALSE*** değerine dönüştürürüz.

Homojen Doku Koordinatları

Dört-koordinatlı doku-uzayı konumu aşağıdaki fonksiyon ile belirlenir (specify).

***glTexCoord4\* (sCoord, tCoord, rCoord, htexCoord);***

Doku koordinatları, sahne koordinatları gibi 4\*4 boyutlu bir matris kullanılarak transform edilir. Her koordinat, homojen parametreye bölünür (Kısım 5-2). Bu yüzden yukardaki fonksiyondaki s,t,r doku koordinatlarına ilişkin değerler, gerçek doku-uzayı konumunu üretmek için *hte*x homojen koordinat bileşenine bölünür.

Çoklu perspektif efektler bir ekranda birleştirileceği zaman, doku uzayındaki homojen koordinatlar faydalıdır. Örneğin bir cismin perspektif görünümü, farklı perspektif-projeksiyon transformasyonu ile üretilmiş bir doku içerebilir. Bu doku paterni, daha sonra doku perspektifini ayarlamak (adjust) için homojen doku koordinatları kullanılarak modifiye edilebilir. Doku iz düşümünü maniple etmek için homojen doku koordinatlarını kullanan başka çok efektler mümkündür.