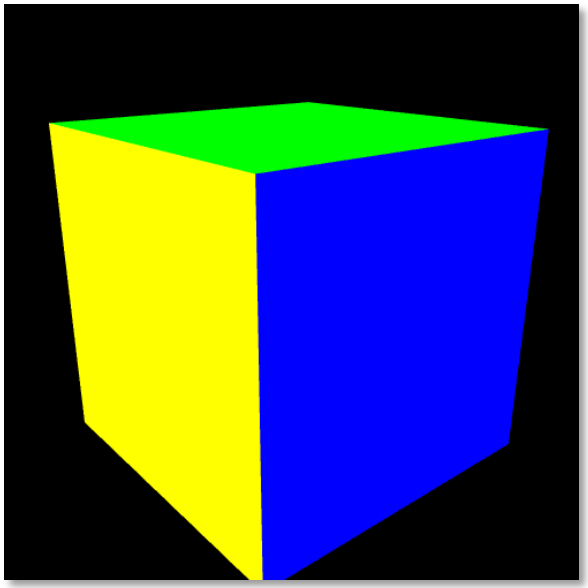
Practical Texturing (WebGL)

CS559 – Spring 2017

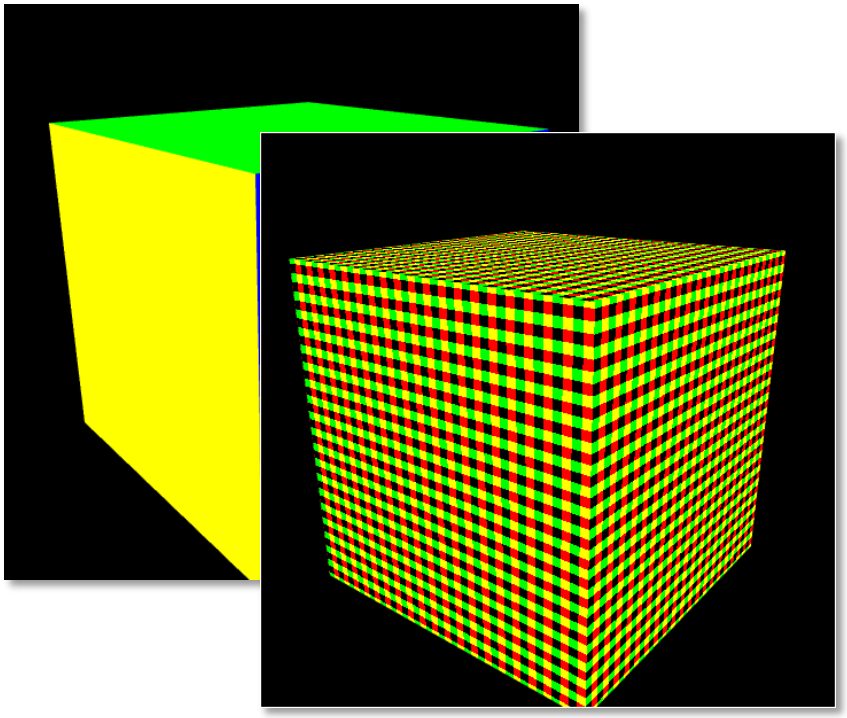
28 March 2017

In brief ...



Starting with a simple model ...

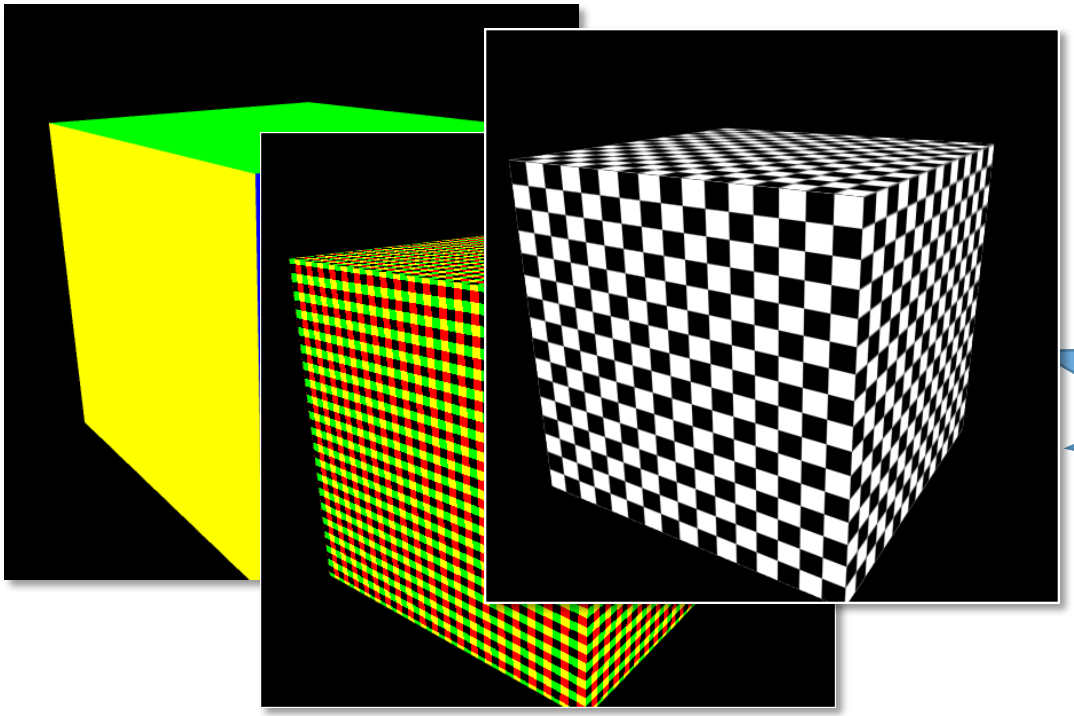
In brief ...



Caveat : Issues with sampling & aliasing

... associate texture coordinates with primitives
(can now do procedural textures)

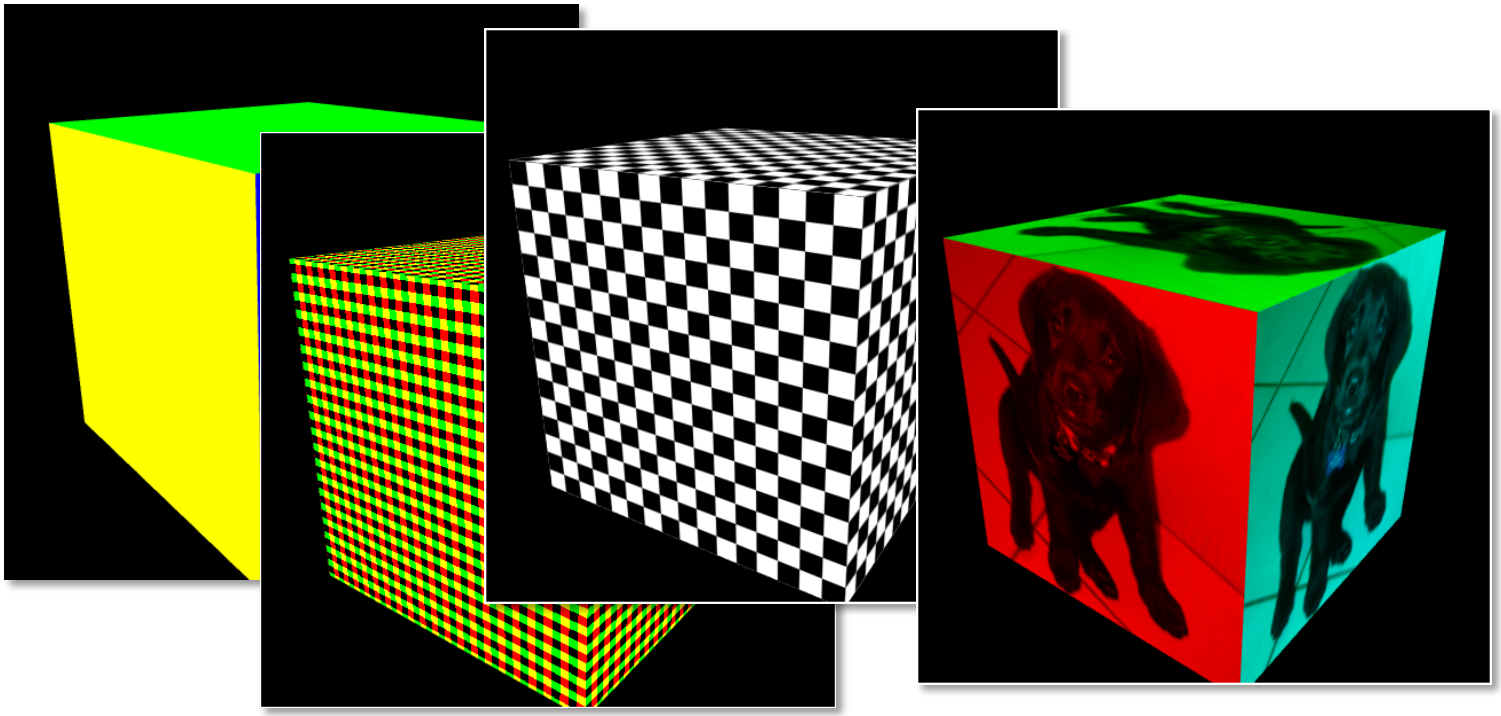
In brief ...



Caveat : Loading files, cross-origin issues

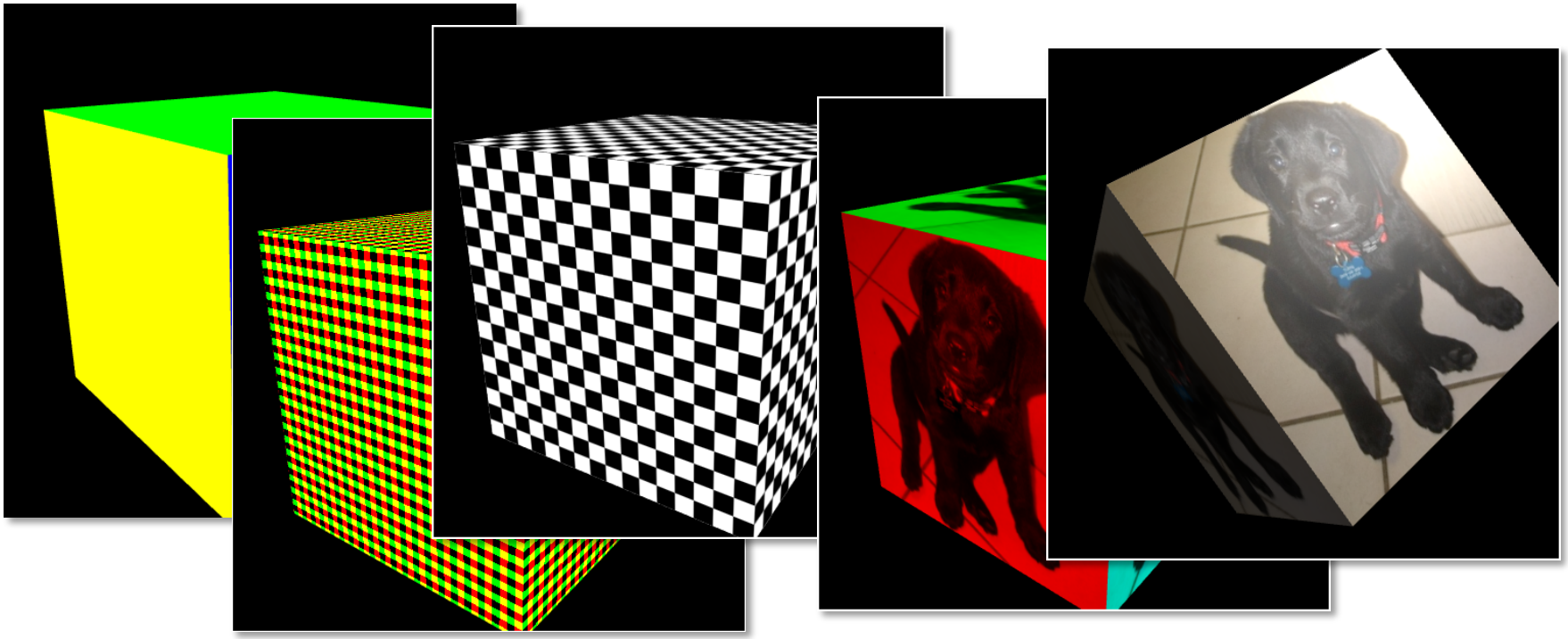
... load an actual image, use mipmap
(and figure out how to use it from shaders)

In brief ...



... then combine texture with color

In brief ...

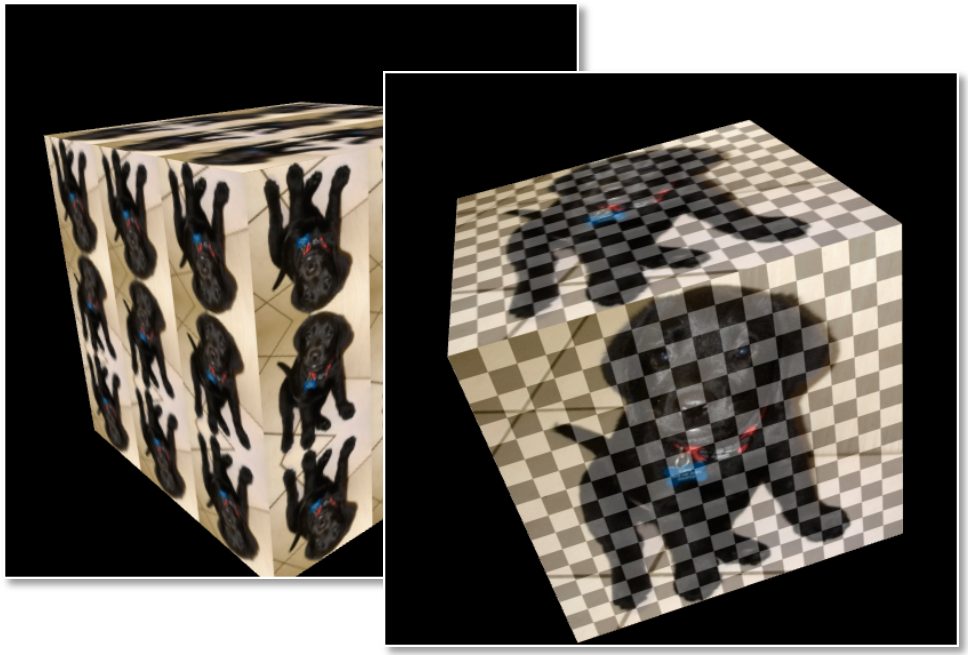


... then combine texture with color, or lighting

Add-ons ...

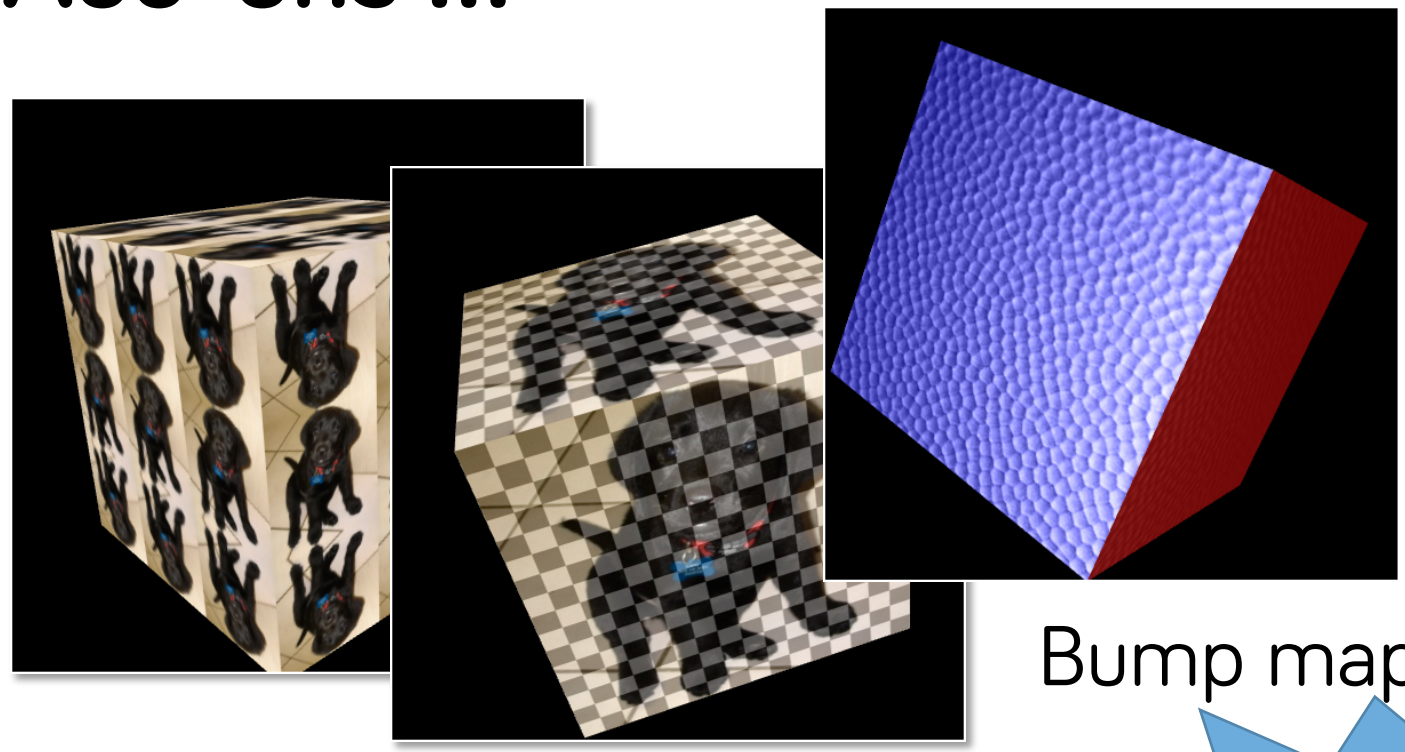
Repeating/clamping
Texture coordinates

Add-ons ...



Multiple textures

Add-ons ...

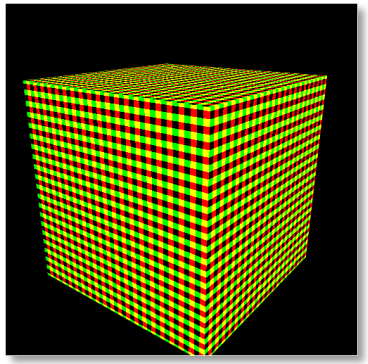


Bump mapping

More effects next week

jsbin.com/pitetahihu/edit

Texture coordinates

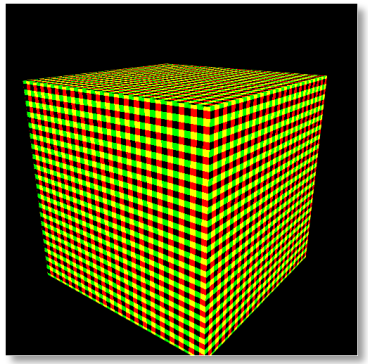


.js

```
start(){
  initShaders();
  sendData();
  draw();
}
```

jsbin.com/pitetahihu/edit

Texture coordinates



.js

```
start(){
  initShaders();
  sendData();
  draw();
}
```

```
shaderProgram.texcoordAttribute = gl.getAttribLocation(shaderProgram, "vTexCoord");
gl.enableVertexAttribArray(shaderProgram.texcoordAttribute);
```

vertexShader

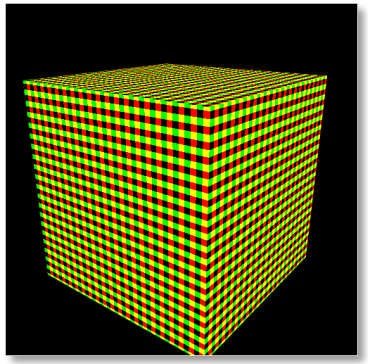
```
attribute vec2 vTexCoord;
varying vec3 fTexCoord;
void main(void) { ...
  fTexCoord = vTexCoord; }
```

fragmentShader

```
varying vec2 fTexCoord;
vec2 Stripe2D(vec2 tc){ // procedural stripe texture }
void main(void) {
   gl_FragColor = vec4(Stripe2D(fTexCoord), 0.0, 1.0); }
```

jsbin.com/pitetahihu/edit

Texture coordinates



.js

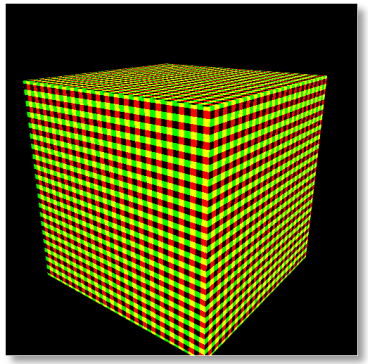
```
start(){
  initShaders();
  sendData();
  draw();
}
```

```
var vertexTextureCoords = new Float32Array(
    [  0, 0,   1, 0,   1, 1,   0, 1,
       1, 0,   1, 1,   0, 1,   0, 0,
       0, 1,   0, 0,   1, 0,   1, 1,
       0, 0,   1, 0,   1, 1,   0, 1,
       1, 1,   0, 1,   0, 0,   1, 0,
       1, 1,   0, 1,   0, 0,   1, 0 ]);
```

```
var textureBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertexTextureCoords, gl.STATIC_DRAW);
textureBuffer.itemSize = 2;
textureBuffer.numItems = 24;
```

jsbin.com/pitetahihu/edit

Texture coordinates



.js

```
start(){
  initShaders();
  sendData();
  draw();
}
```

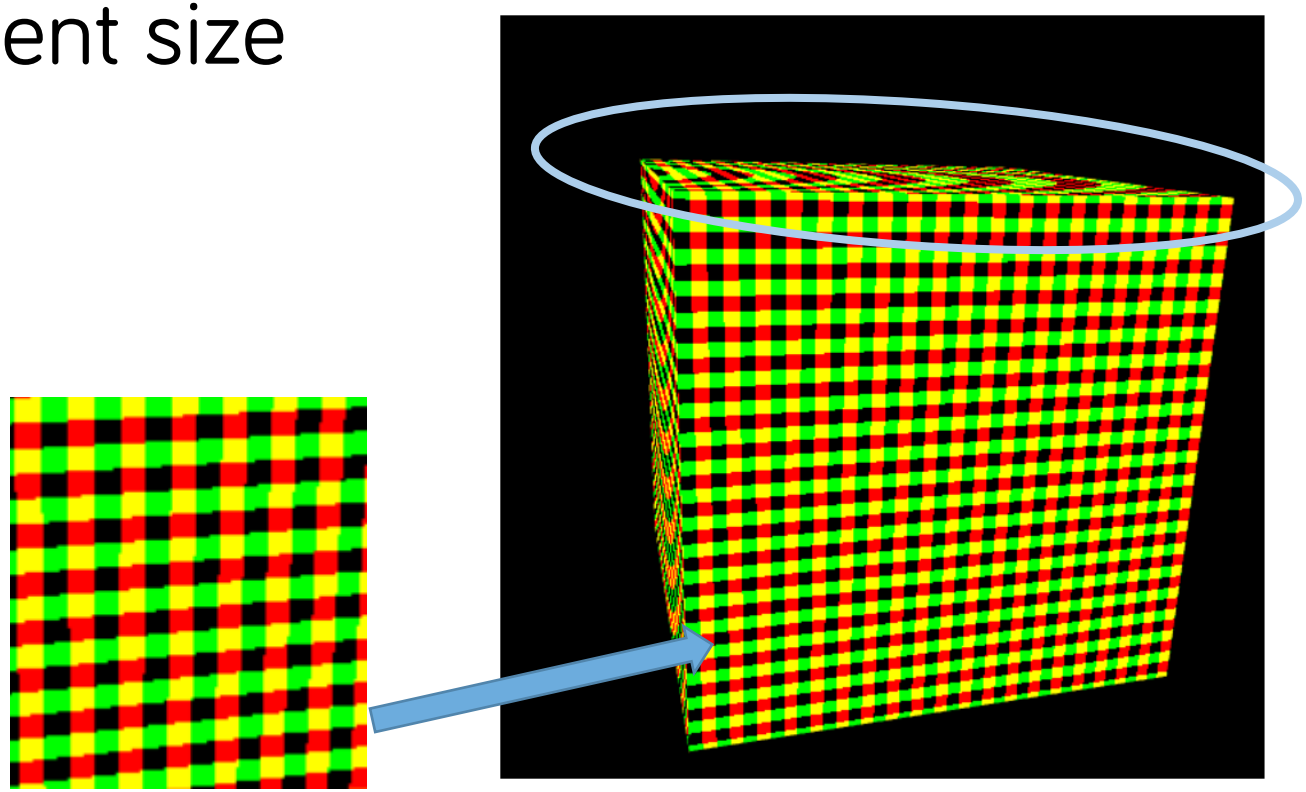
```
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.vertexAttribPointer(shaderProgram.texcoordAttribute, textureBuffer.itemSize,
  gl.FLOAT, false, 0, 0);

// Do the drawing
gl.drawElements(gl.TRIANGLES, triangleIndices.length, gl.UNSIGNED_BYTE, 0);
```

jsbin.com/pitetahihu/edit

Aliasing

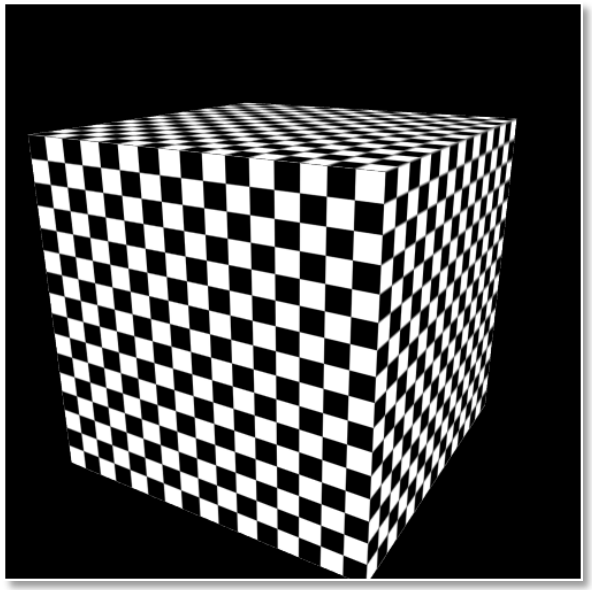
Mismatch between:

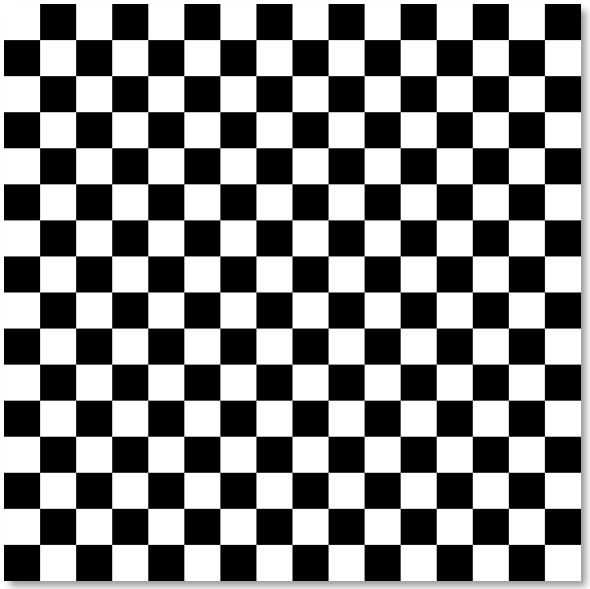
- Texture feature size (color tiles)
- Fragment size



jsbin.com/varefelatu/edit

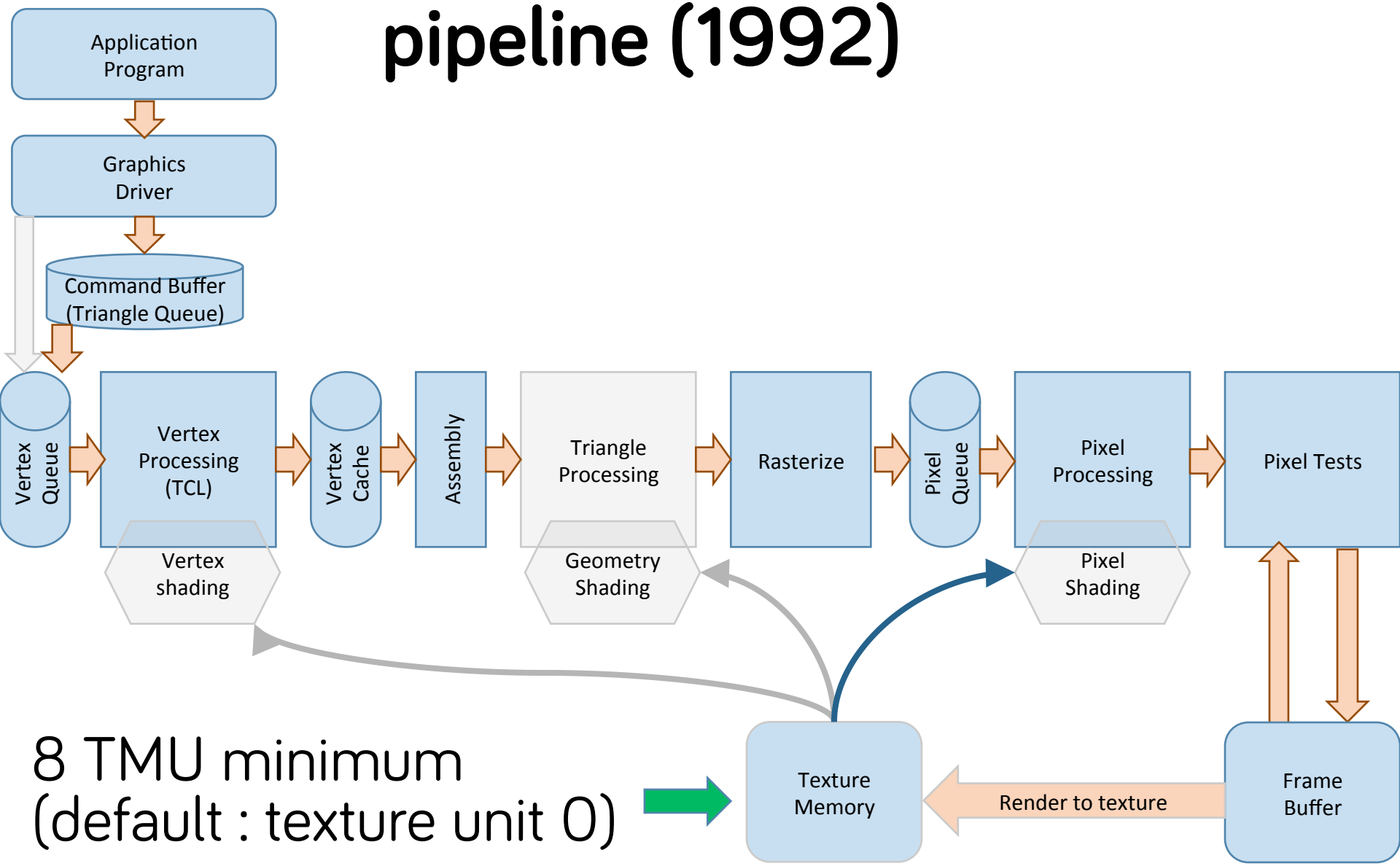
Texturing with images





(texture image)

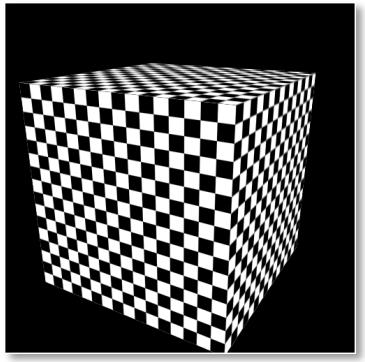
The full fixed-function pipeline (1992)



8 TMU minimum
(default : texture unit 0)

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

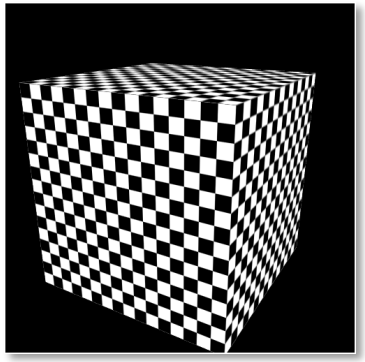
```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

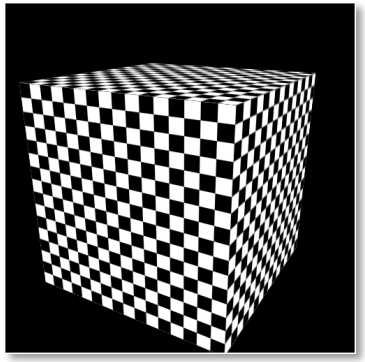
```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

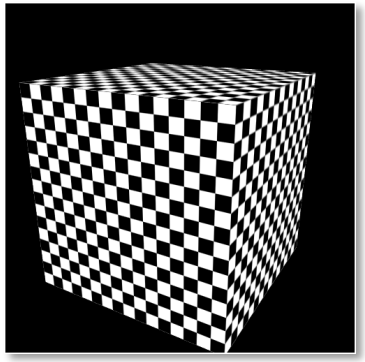
```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

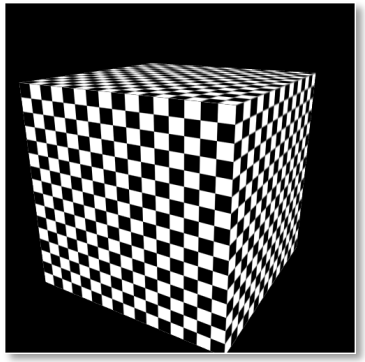
```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1,                                null);
```

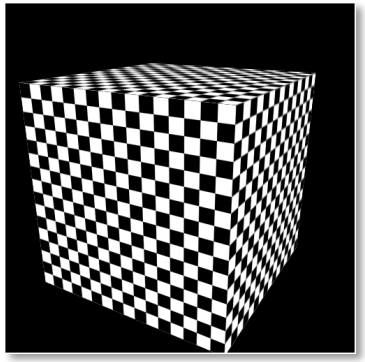
Options
gl.TEXTURE_2D
gl.TEXTURE_CUBE_MAP

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1,                                null);
```

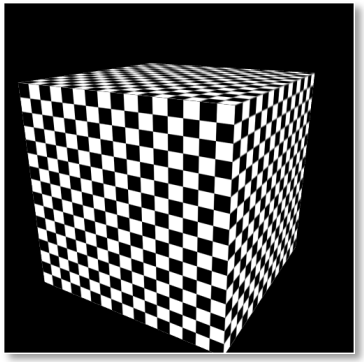
Options
gl.LINEAR_MIPMAP_LINEAR
gl.LINEAR_MIPMAP_NEAREST
gl.NEAREST_MIPMAP_LINEAR

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

jsbin.com/varefelatu/edit

Texturing with images



fragmentShader

```
varying vec2 fTexCoord;
uniform sampler2D texSampler;
void main(void) {
   vec4 texColor = texture2D(texSampler,fTexCoord);
   gl_FragColor = vec4(texColor.xyz,texColor.a); }
```

.js

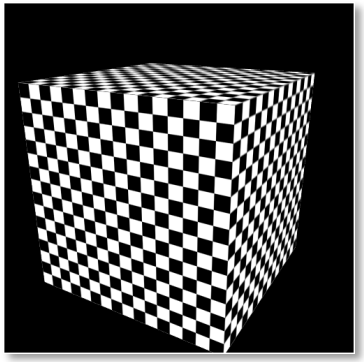
```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

attach to TMU #0

```
shaderProgram.texSampler = gl.getUniformLocation(shaderProgram, "texSampler");
gl.uniform1i(shaderProgram.texSampler, 0);
```

jsbin.com/varefelatu/edit

Texturing with images



fragmentShader

```
varying vec2 fTexCoord;
uniform sampler2D texSampler;
void main(void) {
   vec4 texColor = texture2D(texSampler,fTexCoord);
   gl_FragColor = vec4(texColor.xyz,texColor.a); }
```

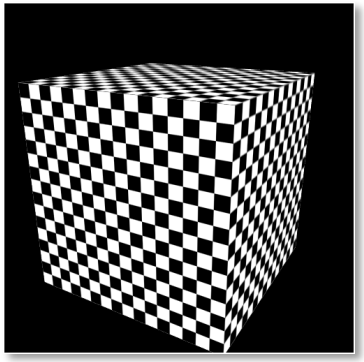
.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
shaderProgram.texSampler = gl.getUniformLocation(shaderProgram, "texSampler");
gl.uniform1i(shaderProgram.texSampler, 0);
```

jsbin.com/varefelatu/edit

Texturing with images



fragmentShader

```
varying vec2 fTexCoord;
uniform sampler2D texSampler;
void main(void) {
   vec4 texColor = texture2D(texSampler,fTexCoord);
   gl_FragColor = vec4(texColor.xyz,texColor.a); }
```

lookup returns vec4

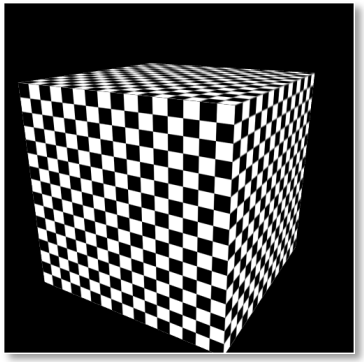
.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
shaderProgram.texSampler = gl.getUniformLocation(shaderProgram, "texSampler");
gl.uniform1i(shaderProgram.texSampler, 0);
```

Texturing with images (non-mipmap filters)

jsbin.com/varefelatu/edit



.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/ch
```

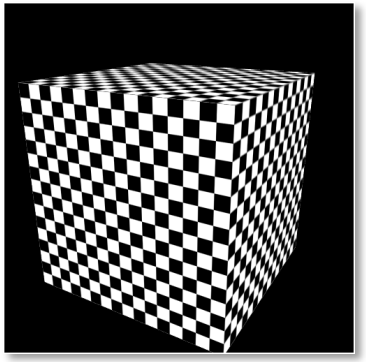
Options
gl.LINEAR
gl.NEAREST

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

```
gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR);
gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);
```

Texturing with images (asynchrony issues)

jsbin.com/qoyudupoqe/edit



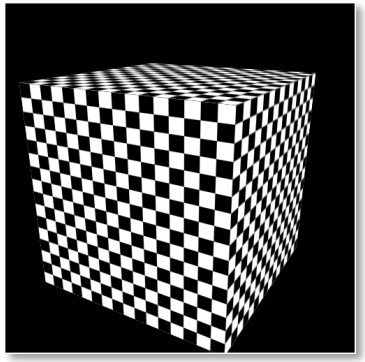
.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
var image = new Image();

function initTextureThenDraw()
{
  image.onload = loadTexture;
  image.src = "http://myurl.com/checkerboard.jpg";
  window.setTimeout(draw,200);
}

function LoadTexture()
{
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
  ...
}
```

Texturing with images (cross-origin issues)

jsbin.com/qoyudupoqe/edit



.js
start(){
initShaders();
sendData();
initTextures();
draw()
}

http://myurl.com/cube.js

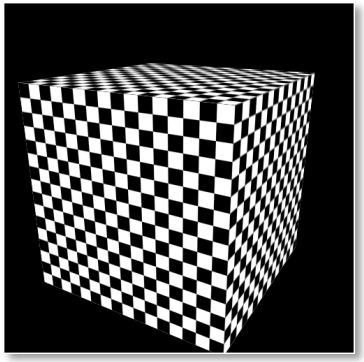
```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
```

```
var image = new Image();
image.onload = loadTexture;
image.src = "http://myurl.com/checkerboard.jpg";
```

```
function loadTexture(){
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);

  gl.generateMipmap(gl.TEXTURE_2D);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR); }
```

Texturing with images (cross-origin issues)

jsbin.com/varefelatu/edit



Cross-origin resource sharing disallowed

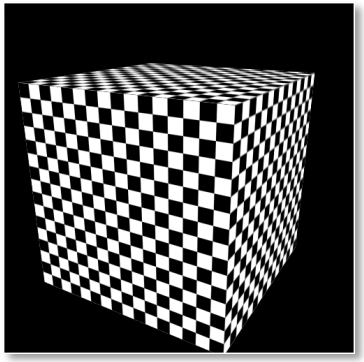
More restrictive policy for Canvas drawing than viewing in web page (img tag)

Some browsers even don't recognize "file://" as a shared origin

Texturing with images (cross-origin ~~issues~~)

workarounds

jsbin.com/qoyudupoqe/edit



```
function initTextureThenDraw()
{
  image.onload = LoadTexture;
  image.crossOrigin = "anonymous";
  image.src = "https://lh3.googleusercontent.com/-xX-m9F-ax7c/...../checkerboard.jpg";
  window.setTimeout(draw,200);
}
```

Must be CORS-approved

- cors-python, for Python web apps
- Enable CORS on static content in Google AppEngine.
- RDF::LinkedData version 0.16 and later
- cors-filter: A Java Servlet Filter implementation of server-side CORS for web containers, by eBay Software Foundation
- add-cors-to-couchdb: CLI to add CORS support to CouchDB, for use in client libraries like PouchDB.

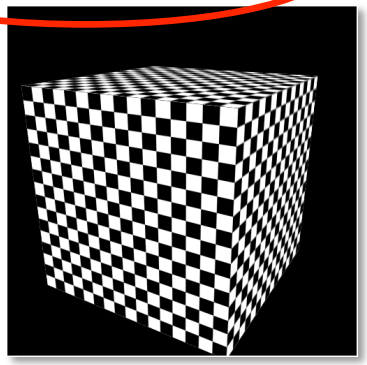
APIs that support CORS

- Amazon S3
- DBpedia Spotlight
- Dropbox API
- Facebook Graph API
- Flickr API
- FourSquare API
- Google APIs
- Google Cloud Storage
- GitHub v3 API
- MediaWiki API
- prefix.cc
- PublishMyData
- sameAs
- SoundCloud API
- Spotify Lookup API
- Sunlight Congress API
- URIBurner
- YouTube API (blog post)
- doctape API

Texturing with images (cross-origin ~~issues~~)

workarounds

jsbin.com/gekavofimu/edit



```
function initTextureThenDraw()
{
  image.onload = LoadTexture;
  image.src = ".......AAACM4AAALMAAAA//9k=";
  window.setTimeout(draw,200);
}
```

http://graphi...bundler.html

graphics.cs.wisc.edu/Courses/559-f2015/Services/imagebundler/bundler.html

Search

Select All Image Files To Bundle. ie: jpeg, png, etc. (use ctrl or shift to select multiple)

If anything breaks feel free to complain to ysohail@wisc.edu

To use the file add it as a script tag in the head. Make sure to add type="text/javascript"

The Image() object will then be accessible in the global object LoadedImageFiles[name_of_file]

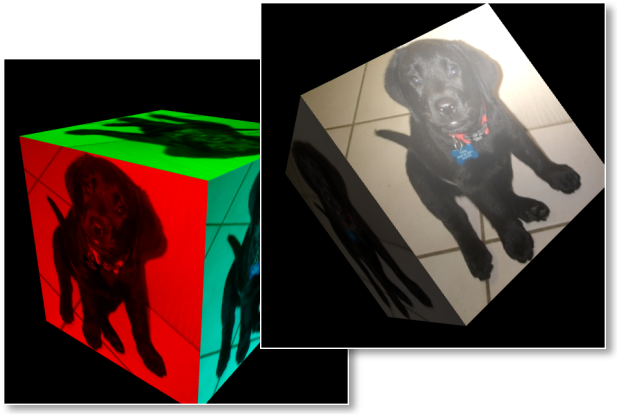
Files Selected Browse... No files selected.

No Files Selected

Please Enter A File Name | Download Selected Files

Texturing and lighting

jsbin.com/kecizocura/edit
jsbin.com/voyexukevi/edit

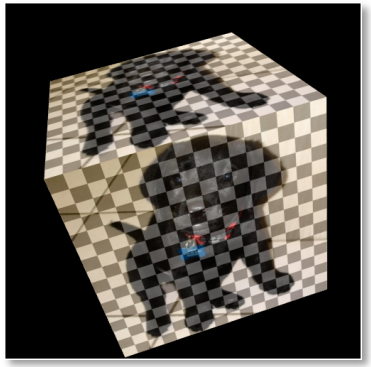


fragmentShader

```
varying vec3 fNormal;
varying vec2 fTexCoord;
uniform sampler2D texSampler;

vec3 blinnPhongDir(...)

void main(void) {
  vec3 texColor=texture2D(texSampler1,fTexCoord).xyz;
  vec3 n = (uMVn * vec4(fNormal, 0.0)).xyz;
  vec3 ColorS  = blinnPhongDir(lightV,fNormal,....specular coeffs....).y*lightCol;
  vec3 ColorAD = blinnPhongDir(lightV,fNormal,..ambient/diff coeffs..).x*texColor;
  gl_FragColor = vec4(ColorAD+ColorS,1.0); }
```

jsbin.com/popojatufi/edit

Multiple textures



fragmentShader

```
varying vec2 fTexCoord;
uniform sampler2D texSampler1;
uniform sampler2D texSampler2;

void main(void) {
   vec4 texColor1 = texture2D(texSampler1,fTexCoord);
   vec4 texColor2 = texture2D(texSampler2,fTexCoord);
   gl_FragColor = vec4(0.6*texColor1.xyz+0.4*texColor2.xyz,texColor1.a);}
```

.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

```
shaderProgram.texSampler1 = gl.getUniformLocation(shaderProgram, "texSampler1");
gl.uniform1i(shaderProgram.texSampler1, 0);
shaderProgram.texSampler2 = gl.getUniformLocation(shaderProgram, "texSampler2");
gl.uniform1i(shaderProgram.texSampler2, 1);
```

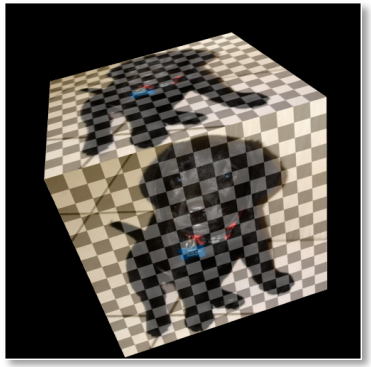
```
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture1);
gl.activeTexture(gl.TEXTURE1);
gl.bindTexture(gl.TEXTURE_2D, texture2);
gl.drawElements(gl.TRIANGLES, triangleIndices.length, gl.UNSIGNED_BYTE, 0);
```

jsbin.com/weluvebodu/edit

Multiple textures

.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```



```
var texture1 = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture1);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
var image1 = new Image();
var texture2 = gl.createTexture();
gl.activeTexture(gl.TEXTURE1);
gl.bindTexture(gl.TEXTURE_2D, texture2);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
var image2 = new Image();

image1.onload = function() { loadTexture(image1,texture1); };
image1.src = "http://myurl.com/spirit.jpg";
image2.onload = function() { loadTexture(image2,texture2); };
image2.src = "http://myurl.com/checkerboard.jpg";
```