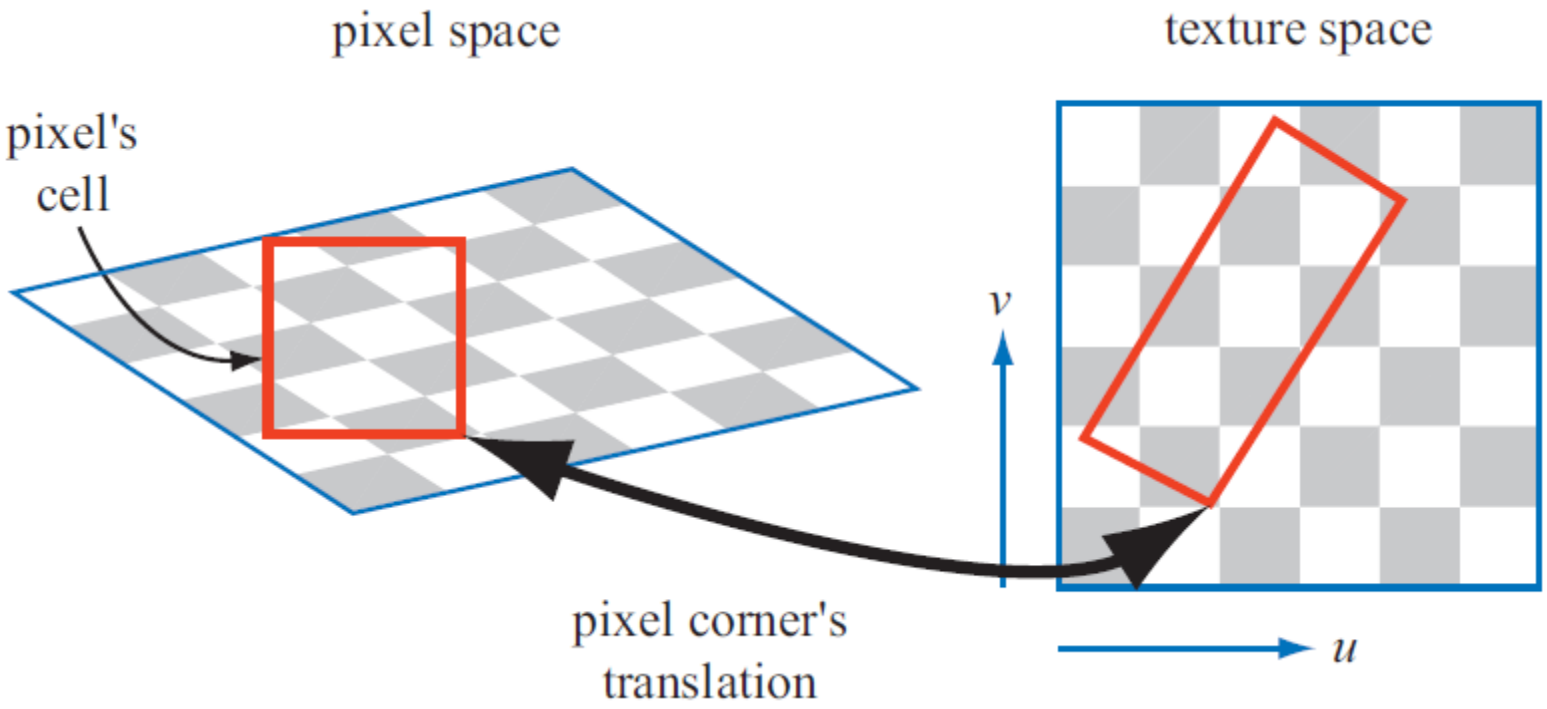
2D-to-3D mapping
Mipmap's overblurring

- **Mip-mapping results in overblurring**
  - **occurs when a pixel's pre-image covers a large number of texels in *u* direction and only a few in *v* direction. To avoid aliasing, mip-map of the lowest resolution (indicated by longer edge of quad) is selected, and but resulting in overblurring.**
- **Approaches [Real Time Rendering (2nd ed.), p 137]**
  - **Rectangular filtering (anisotropic filtering)**
    - **Ripmap [Advanced OpenGL course notes 98]**
      - **Extend mip-map to include downsampled rectangular areas as subtextures that can be accessed.**
      - **Used in high-end HP graphics accelerator in early '90.**
    - **Crow's summed-area table [Crow, Siggraph84]**
      - **Never implemented in hardware**
    - **Both are memory intensive.**
  - **Unconstrained anisotropic filtering**

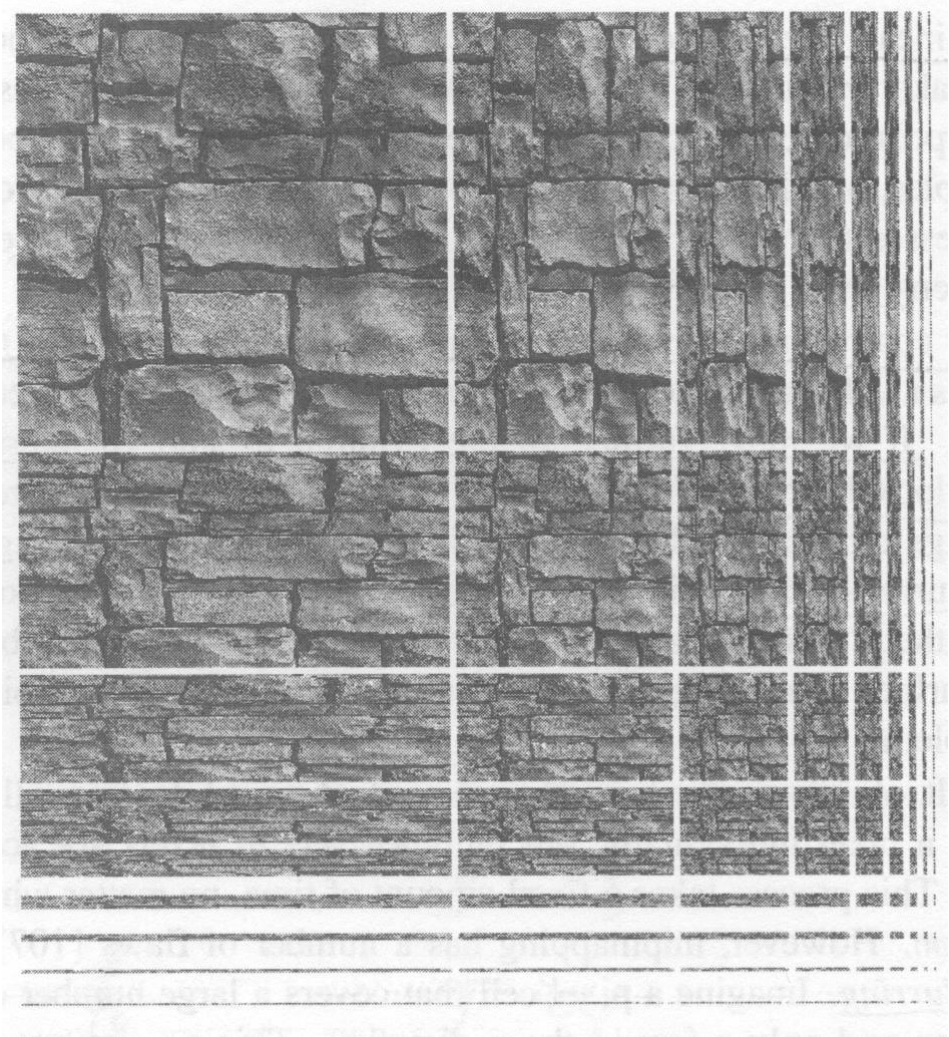
2D-to-3D mapping
Mipmap's overblurring



2D-to-3D mapping
Ripmap

- **Extends the mipmap**
  - **To include downsampled rectangular areas as subtextures that can be accessed.**
  - **Four coordinates are used to access ripmap subtexture array**
    - **Two for a location in the ripmap array**
      - **Like mipmap, the suitable level, say l, is selected according to the object-eye point distance**
      - **is computed using the pixel cell's u and v extents on the texture**
    - **Usual two (u, v) values to access each subtexture**

2D-to-3D mapping
Ripmap



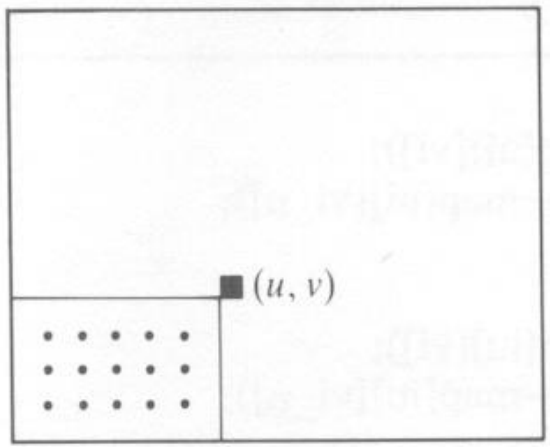
Images on diagonal are the mipmap subtextures.

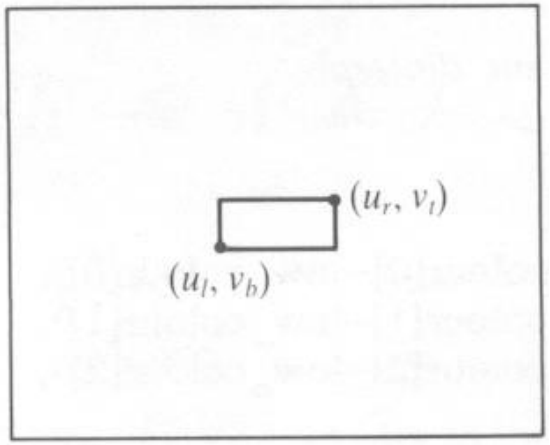
Images in offdiagonal are other ripmap subtextures

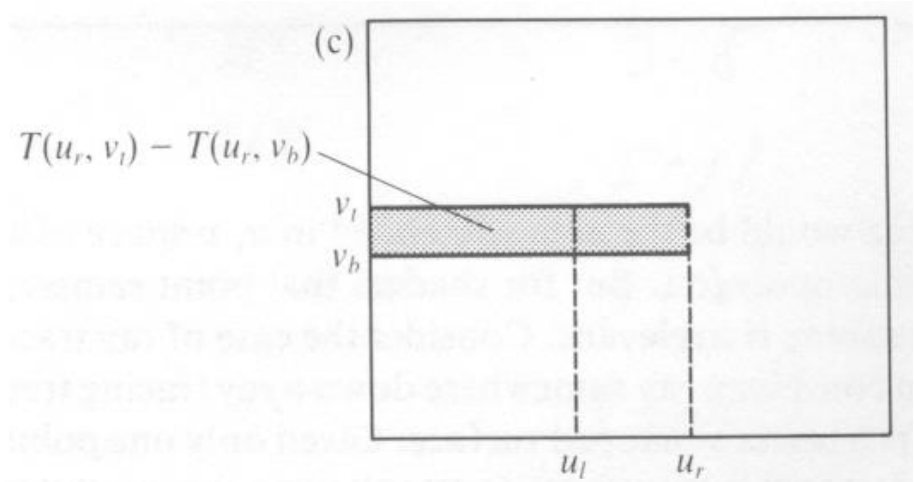
2D-to-3D mapping
Summed area table

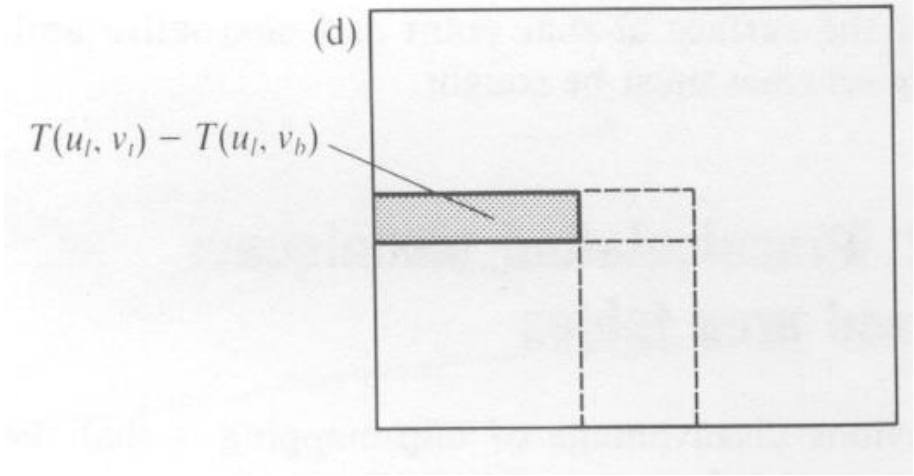
- **Pre-computed summed area table**
  - **Same size as the texture, but contains more bits of precision for the color stored.**
  - **Each value is the sum of the corresponding texture's texels in the rectangle formed by this location and texel (0,0) (the origin).**
- **During texturing, indexed by a rectangular area, whose aspect ratio can vary**
  - **Pixel's pre-image is bounded by a rectangle.**
  - **The max of the four corner of the pre-image are used to access the table.**
  - **Summed-area table is then accessed to determine the average color of this rectangle.**

2D-to-3D mapping
Summed area table

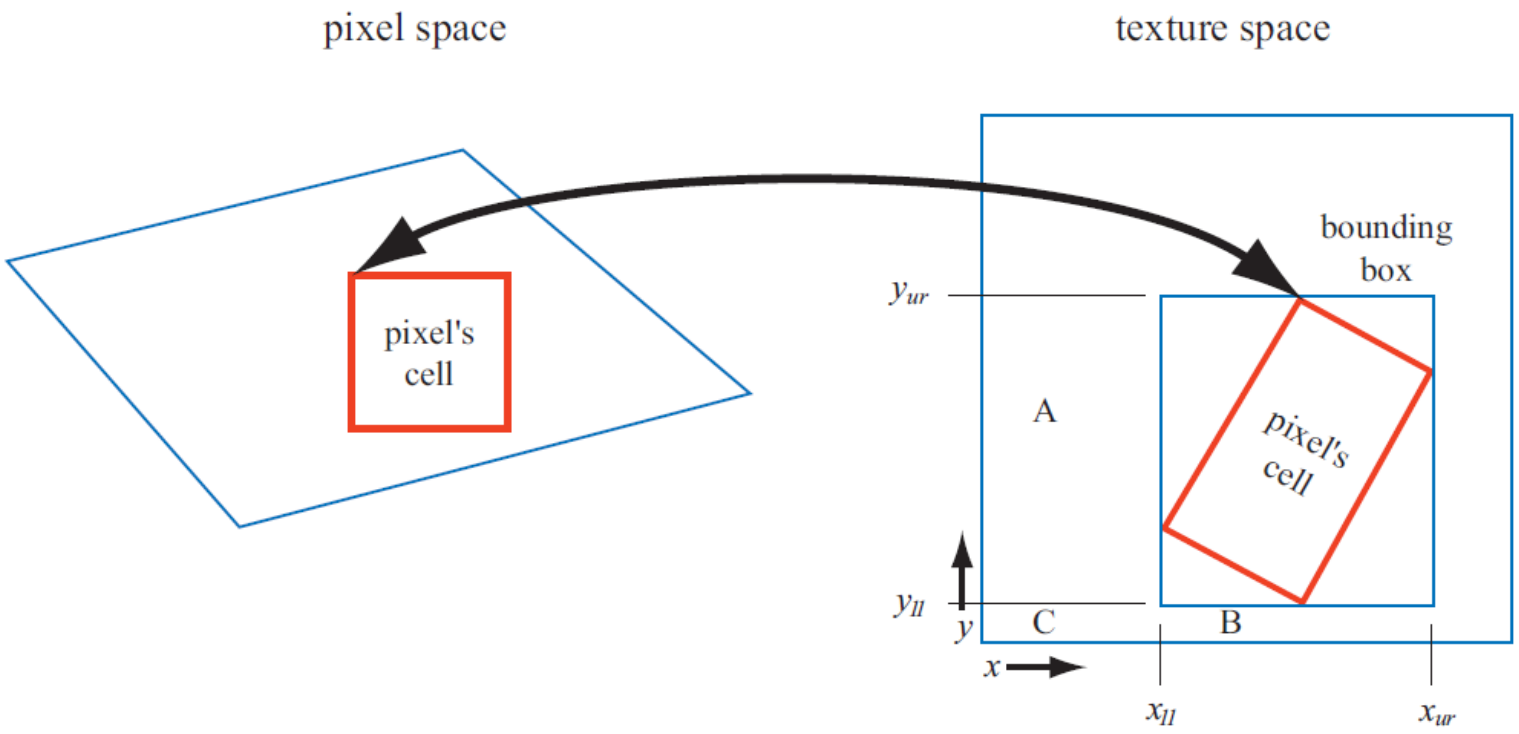






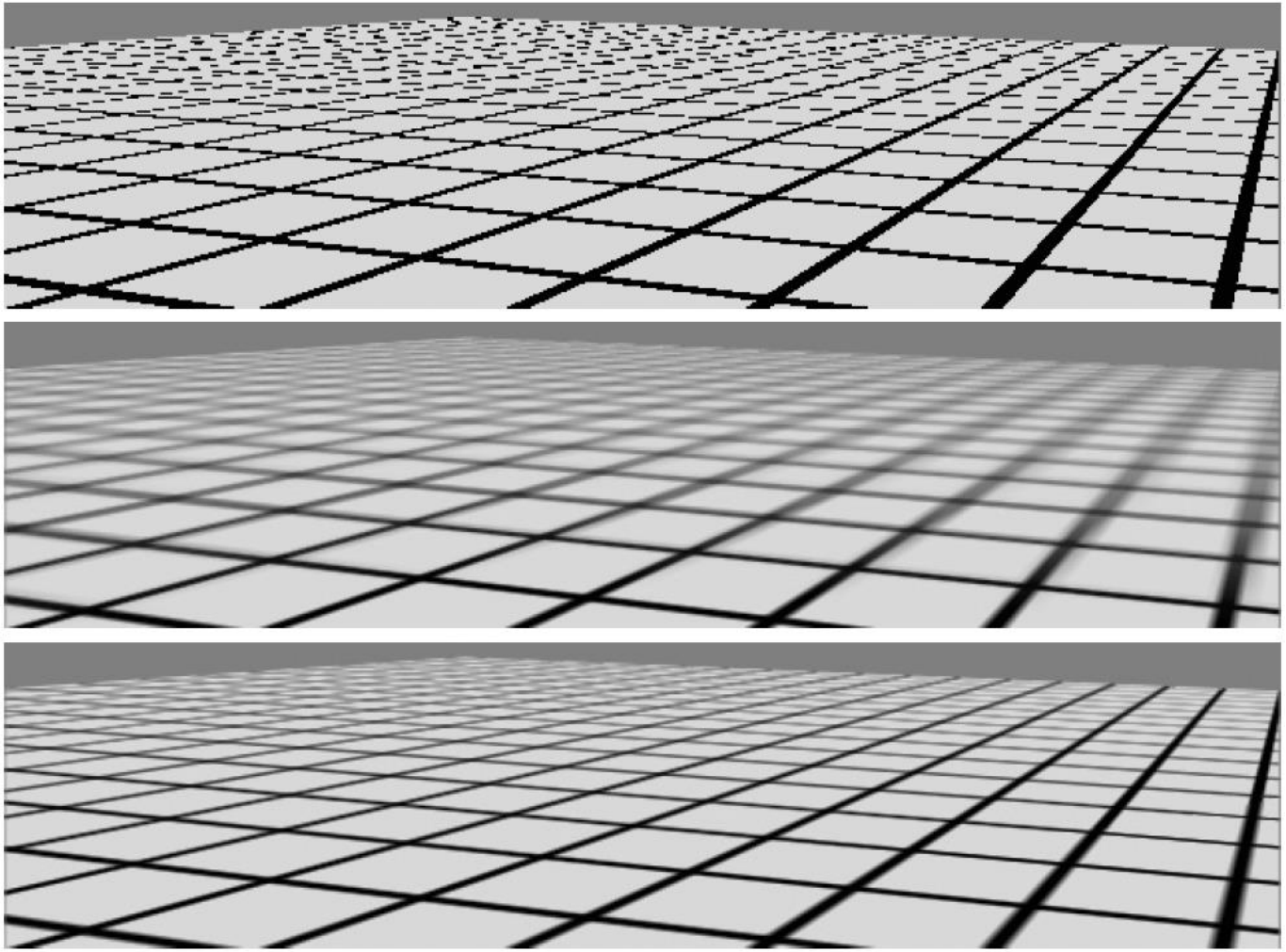


2D-to-3D mapping
Summed area table



Pixel's cell is back-projected onto the texture, bounded by a rectangle, and the four corners of the rectangle are used to access the summed area table.

2D-to-3D mapping
Antialiasing example (minification)



Point sampling with nearest neighbor

Mip-mapping (overblurring)

Summed area table

2D-to-3D mapping
Unconstrained anisotropic filtering -1

- **Mipmapping and ripmapping prefilter the texture**
  - **Match the actual spatial frequency after transform the signal onto screen**
  - **Memory intensive**
- **But the signal may be stretched in all directions, not only horizontal/vertical directions**
- **Anisotropic filtering samples texels according to the stretch direction**

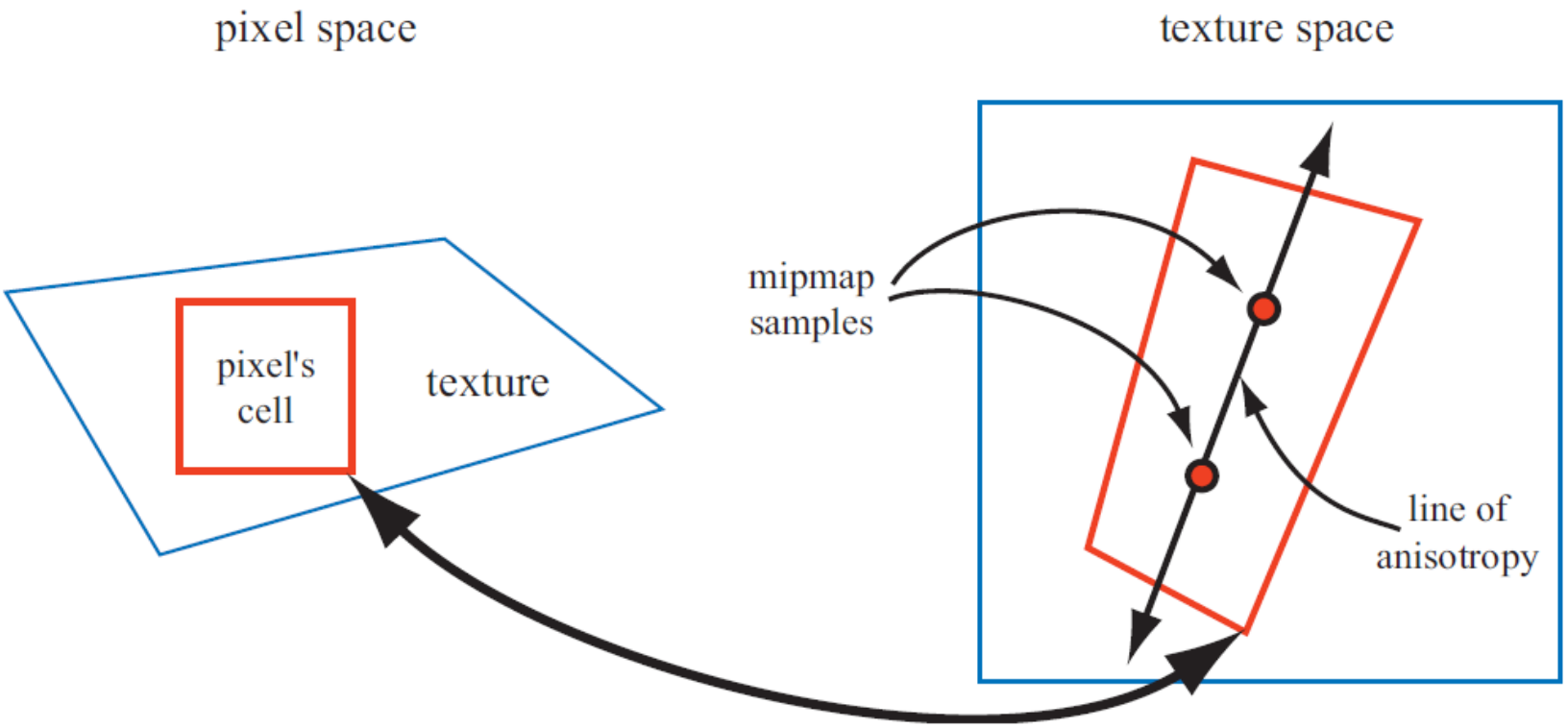
2D-to-3D mapping
Unconstrained anisotropic filtering -2

- **Reuse existing mipmap hardware**
  - **The most common method in current graphics hardware**
  - **Requires no more texture memory than mipmap does**
  - **Steps**
    - **Sample a number of mipmap squares that cover the quad derived by back-projecting the pixel cell**
    - **Combine the samples**

2D-to-3D mapping
Unconstrained anisotropic filtering -3

- **Pixel cell is back-projected to a quad on the texture.**
- **The quad is sampled by a number of mipmap squares that cover the quad**
  - **The shorter side of the quad determines *d***
  - **The longer side creates a line of anisotropy**
    - **Parallel to the longer side and through the quad's middle**
  - **Take samples along the line of anisotropy according the amount of anisotropy**
- **The samples are combined**

2D-to-3D mapping
Unconstrained anisotropic filtering -4

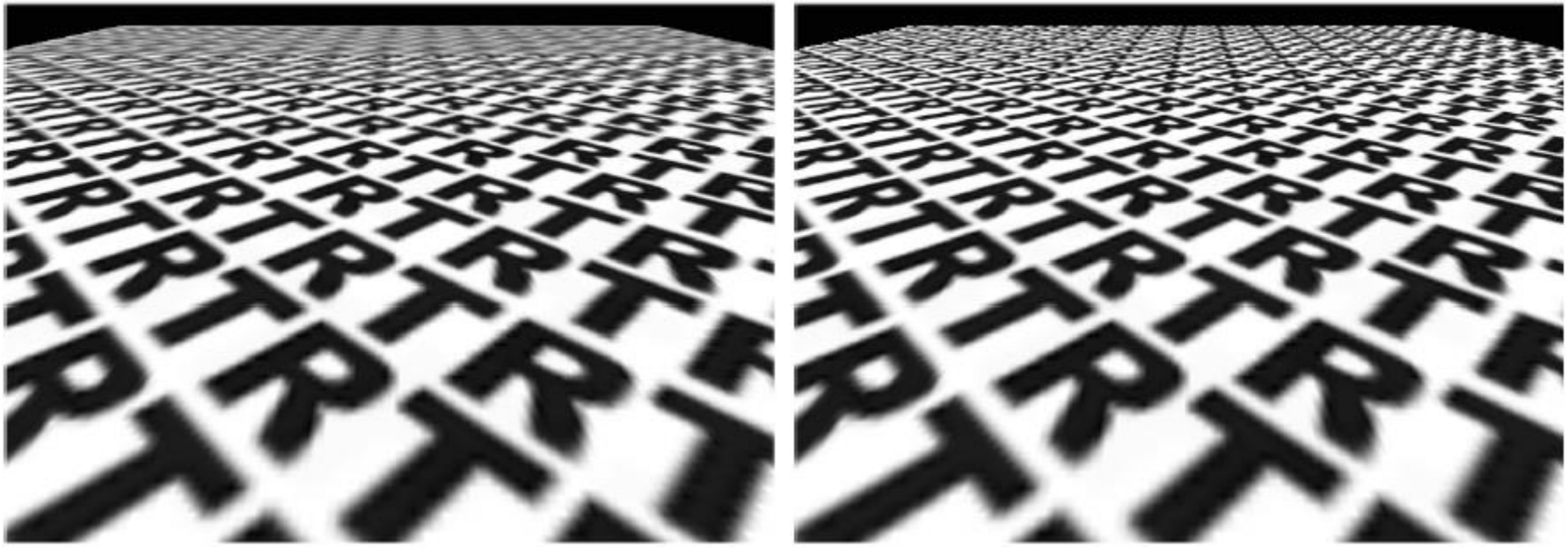


- **Back projection 4 corners of pixel onto texture**
- **Length of the shorter side determine the mipmap level used**
- **Sample the mipmapped texture along the center line of longer sides according to the ratio of longer/shorter sides**

2D-to-3D mapping
Unconstrained anisotropic filtering -5

- **Allows the line of anisotropy to run in any direction**
- **Requires no more texture memory than mipmaps do**
- **Higher anisotropic ratio can be obtained by**
  - **Having higher number of parallel pipes in chip sets, or**
  - **Having the hardware repeatedly sample**
    - **NVIDIA GeForce3 can perform 8:1 anisotropic filtering**

2D-to-3D mapping
Unconstrained anisotropic filtering -6



Trilinear mipmap versus anisotropic filtering (16:1).
Towards the horizon anisotropic filtering provides a sharper result with minimal aliasing.

2D-to-3D mapping
Unconstrained anisotropic filtering -7

[from extremtech.com]

- **Left: without AF**

Right: with AF