Lesson 5 – Images

PV227 – GPU Rendering

Jiří Chmelík, Jan Čejka
Fakulta informatiky Masarykovy univerzity

27. 10. 2015

Outline

1. Introduction

2. Post-processing effects
   - Gray Scale
   - Negative
   - Thresholding
   - Blurring
   - Convolution

DevIL

- library for working with images,
- simplifies loading textures to OpenGL,
- download built SDK at `http://openil.sourceforge.net/` (`http://downloads.sourceforge.net/openil/DevIL-SDK-x86-1.7.8.zip`).

DevIL (cont.)

- update VC++ Directories (taken care of ;-)),
- pass data from DevIL to OpenGL.

Image Processing

- image effects applied to a texture,
- may be used as post-process on the framebuffer,
- gray scale,
- negative,
- thresholding,
- blurring,
- general convolution.

Texture Setup

- same way as in fixed OpenGL,
- texture unit ID passed to the sampler in the shader,
- rendered using two triangles (quad),
- camera setup so that only the quad is seen.

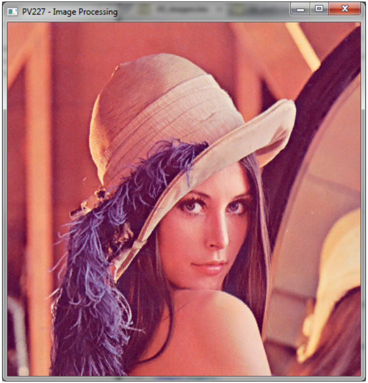


Figure: Rendered texture

Texture Read

- gvec texture(gsampler sampler, vec texCoord);
    - gvec is the texel type,
    - gsampler is the sampler type,
    - texCoord is in $[0, 1]$ range (coordinates according to texture dimension).

- ivec textureSize(gsampler sampler, int lod);
    - ivec is the integer size of the texture (coordinates according to texture dimension) ,
    - gsampler is the sampler type,
    - lod is level-of-detail (usually 0).

Outline

1. Introduction
2. Post-processing effects
   - Gray Scale
   - Negative
   - Thresholding
   - Blurring
   - Convolution

Gray Scale

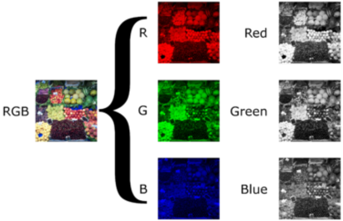


Figure: Taken from wikimedia.org

- linear combination of the RGB channels into luma (intensity),
- texel is multiplied component-wise (dot product) with the weights.

Gray scale (cont.)

- several options for choosing the weights,
- NTSC weights: 0.299, 0.587, 0.114.

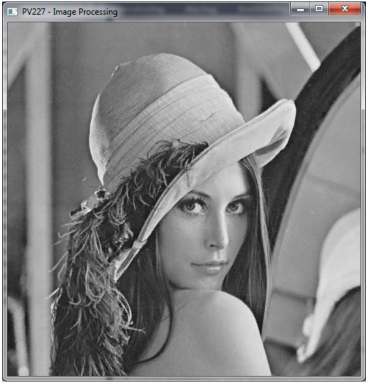


Figure: Grayscale

Outline

1 Introduction

2 Post-processing effects

- Gray Scale
- Negative
- Thresholding
- Blurring
- Convolution

Negative

- Inversion of each color channel,
- alpha channel should not be inverted.

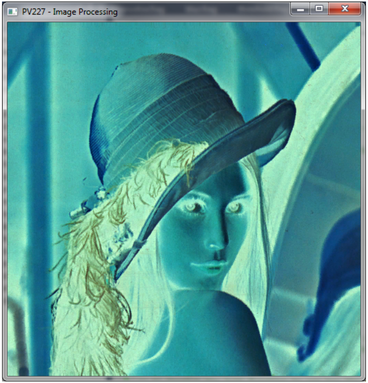


Figure: Negative

Texture Fetch

- `in vec4 gl_FragCoord;`
  - available only in fragment shader
  - contains the window-relative coordinates of the current fragment (x, y, z, 1/w).

- `gvec texelFetch(gsampler sampler, ivec texCoord, int lod);`
  - perform a lookup of a single texel within a texture
  - gvec is the texel type,
  - gsampler is the sampler type,
  - texCoord is in [0, *textureSize*] range (coordinates according to texture dimension),
  - lod is level-of-detail (usually 0).

Outline

1. Introduction

2. Post-processing effects
   - Gray Scale
   - Negative
   - Thresholding
   - Blurring
   - Convolution

Thresholding

- usually applied to gray scale images,
- assigns white to pixels above threshold, black otherwise.

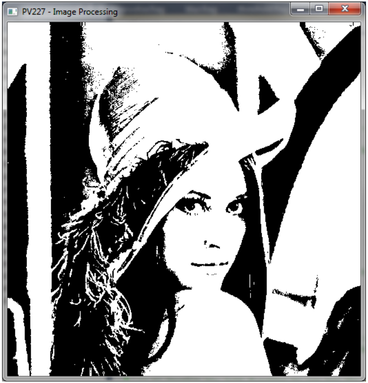


Figure: Thresholding

Outline

1. Introduction

2. Post-processing effects
   - Gray Scale
   - Negative
   - Thresholding
   - Blurring
   - Convolution

Blurring

- averaging of the image,
- the amount of blur depends on the kernel size,
- blur type is controlled by the blurring weights,
- the weights must sum to 1.

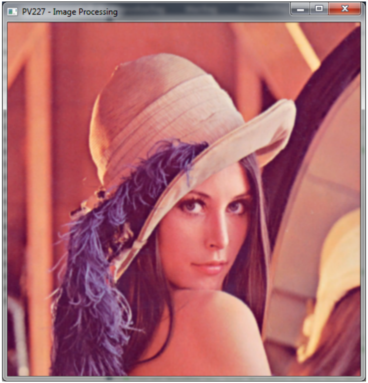


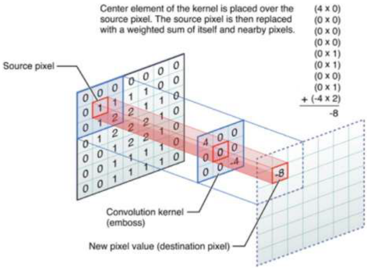
Figure: Gaussian 5x5 blur

Outline

1. Introduction

2. Post-processing effects
    - Gray Scale
    - Negative
    - Thresholding
    - Blurring
    - Convolution

Convolution



Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

Figure: Taken from illinois.edu

- used to compute any linear filter,
- $(f * g)(t) \equiv \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$,
- $(f * g)(t) \equiv \sum_{-\infty}^{\infty} f(\tau)g(t-\tau)$.

Sharpening

- inverse of blurring (subtraction of neighbourhood),
- the amount of sharpening depends on the kernel size,
- sharpen type is controlled by the convolution weights,
- the weights must sum to 1.

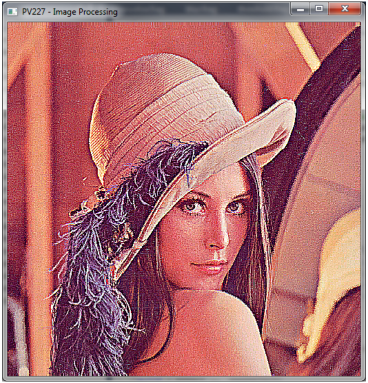


Figure: Sharpening 3x3

Edge Detection

- detects changes in intensity,
- preferably blur the image before edge detection,
- detection type is controlled by the convolution kernel,
- the weights must sum to 0.

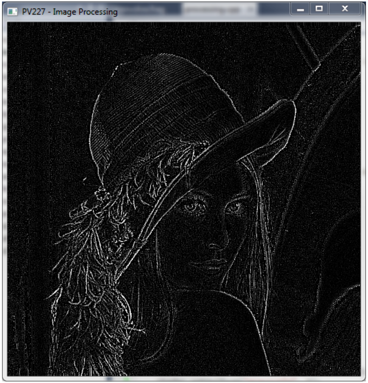


Figure: Laplacian edge detection

More

- PV131: Digital Image Processing,
- PA166: Advanced Methods of Digital Image Processing,
- PA170: Digital Geometry,
- PA171: Digital filtering,
- PA172: Image Acquisition Principles,
- PA173: Mathematical Morphology,
- …