

What is an image?

- An image is a discrete array of samples representing a continuous 2D function

2

קורס גרפיקה ממוחשבת סמסטר ב' 2008

Image Processing

1 חלק מהש愧פם מudyuds משקפים של פרדי דורה, טומס פנקהוסר ודניאל קרן-אור

Sampling and Reconstruction

המראת יוזם תרשים
Sampling
Sampling rate
Reconstruction
Sampling rate

3

Converting to digital form

- Convert continuous sensed data into digital form

Sampling
Quantization
Sampling

3

Sampling Theory

- How many samples are required to represent a given signal without loss of information?
- What signals can be reconstructed without loss for a given sampling rate?

Original function
Reconstructed function

Sampling and Reconstruction

Original signal
Sampling
Sampled signal
Reconstruction
Reconstructed signal

Figure 19.9 FvDFH

Spectral Analysis

- So our image (function $f(x,y)$) describes how the signal changes over "time" (x and y axes)
- Aliasing occurs when we use too few samples (what is enough?)
- The more an image changes, the more we need to sample it.
- How do we measure how fast a signal changes?
 - Frequencies

Aliasing

- What happens when we use too few samples?
- Aliasing

אנו מושגנו פונקציית גלים בפיזיקה, אך לא מושגנו במתמטיקה. נזכיר את הטענה של ניימן: אם נמיצים נספחים סינוסיים בקצב שפחות מ加倍 מהקצב המקורי, נקבל גלים אלייזינג.

Figure 14.17 FvDFH

Fourier

- Joseph Fourier discovered in 1822 that
 - Any periodic function can be expressed as the sum of sines and/or cosines if different frequencies (Fourier Series)
 - Even functions that are not periodic can be expressed as the integral of sines and/or cosines (Fourier Transform)
 - Initial application was in heat diffusion

Spectral Analysis

- Spatial domain:
 - Function: $f(x)$
 - Filtering: convolution
- Frequency domain:
 - Function: $F(u)$
 - Filtering: multiplication

תפקידו של פורייר הוא לפרק כל גלן לסדרה של גלים סינוסיים וкосינוסיים.

Fourier Transform (1D)

- Fourier transform:
$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$
- Inverse Fourier transform:
$$f(x) = \int_{-\infty}^{\infty} F(u) e^{+i2\pi ux} du$$

Fourier Transform (1D)

המטרה של תרמו-פורייר היא לפרק גלן אחד לסדרה של גלים סינוסיים וкосינוסיים.

Figure 2.6 Wolberg

המטרה של תרמו-פורייר
היא לפרק גלן אחד לסדרה של גלים סינוסיים וкосינוסיים.

המטרה של תרמו-פורייר
היא לפרק גלן אחד לסדרה של גלים סינוסיים וкосינוסיים.



- Pixel operations
 - Add random noise
 - Add luminance
 - Add contrast
 - Add saturation
- Filtering
 - Blur
 - Detect edges
 - Sharpen
 - Emboss
 - Median
- Quantization
 - Uniform Quantization
 - Floyd-Steinberg dither
- Warping
 - Scale
 - Rotate
 - Warps
- Combining
 - Composite
 - Morph



- A signal can be reconstructed from its samples, if the original signal has no frequencies above $1/2$ the sampling frequency - Shannon
- The minimum sampling rate for bandlimited function is called "Nyquist rate"

A signal is bandlimited if its highest frequency is bounded. The frequency is called the bandwidth.

Adjusting Contrast

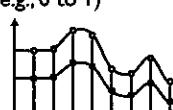
- Compute mean luminance \bar{L} for all pixels
 - luminance = $0.30r + 0.59g + 0.11b$
- Scale deviation from \bar{L} for each pixel component
 - Must clamp to range (e.g., 0 to 1)

Original More Contrast



Adjusting Brightness

- Simply scale pixel components
 - Must clamp to range (e.g., 0 to 1)



Linear Filtering (Spatial Domain)

- Convolution
 - Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



- Pixel operations
 - Add random noise
 - Add luminance
 - Add contrast
 - Add saturation
- Filtering
 - Blur
 - Detect edges
 - Sharpen
 - Emboss
 - Median
- Quantization
 - Uniform Quantization
 - Floyd-Steinberg dither
- Warping
 - Scale
 - Rotate
 - Warps
- Combining
 - Composite
 - Morph

More on blur (lowpass filters)

- We can either take a uniform kernel (mean filter)
- Or a Gaussian kernel
- A Gaussian kernel tends to provide gentler smoothing and preserve edges better

Frequency Response of the Filter

Frequency Response of the Image

Adjust Blurriness

- Convolve with a filter whose entries sum to one
- Each pixel becomes a weighted average of its neighbors

Original

Blur

Filter = $\begin{bmatrix} \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{bmatrix}$

Sharpen

- Sum detected edges with original image

Original	Sharpened

Filter = $\begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Edge Detection

- Convolve with a filter that finds differences between neighbor pixels

Original	Detect edges

Filter = $\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Emboss

- Convolve with a filter that highlights gradients in particular directions



Original



Embossed

$$\text{Filter} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Image Warping

- Move pixels of image



Source image



Destination image

Image Processing

- Pixel operations
 - Add random noise
 - Add luminance
 - Add contrast
 - Add saturation
 - Filtering
 - Blur
 - Detect edges
 - Sharpen
 - Emboss
 - Median
 - Quantization
 - Uniform Quantization
 - Floyd-Steinberg dither
 - Warping
 - Scale
 - Rotate
 - Warps
 - Combining
 - Composite
 - Morph

הנתקן
הנתקן
הנתקן
הנתקן
הנתקן
הנתקן

Example

- Image Scaling
 - $(x', y') = (sx^*, sy^*)$;
 - $I(x', y') = ?$

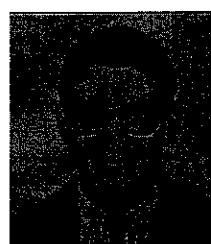
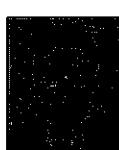


Image Warping

- Issues
 - How do we specify where every pixel goes? (mapping)
 - How do we compute colors at destination pixels? (resampling)

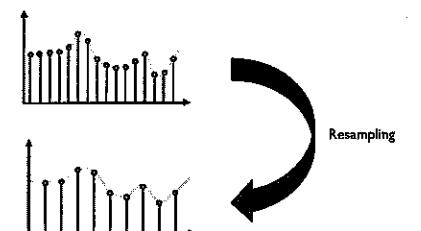


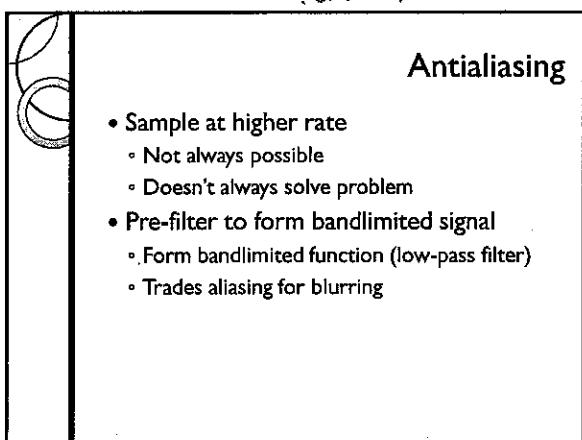
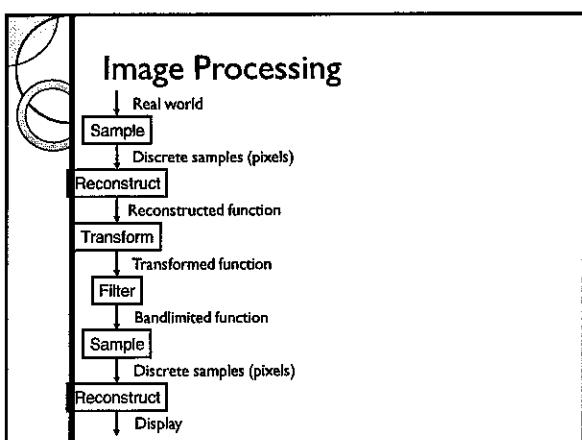
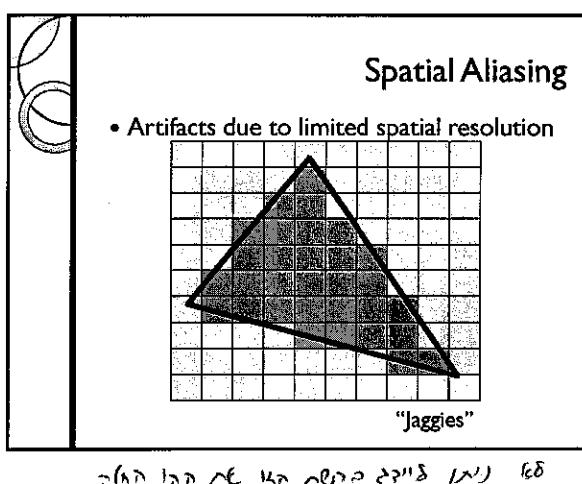
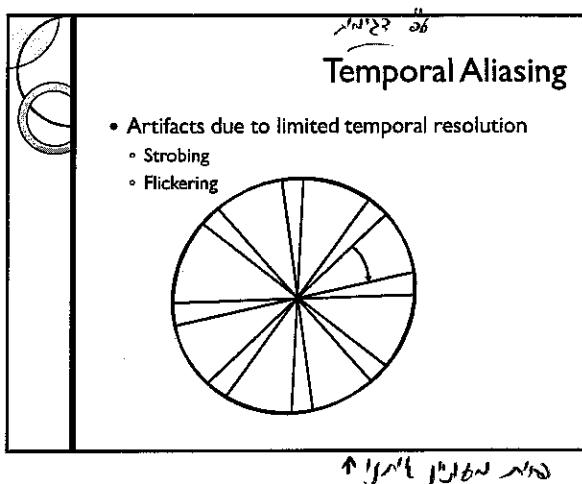
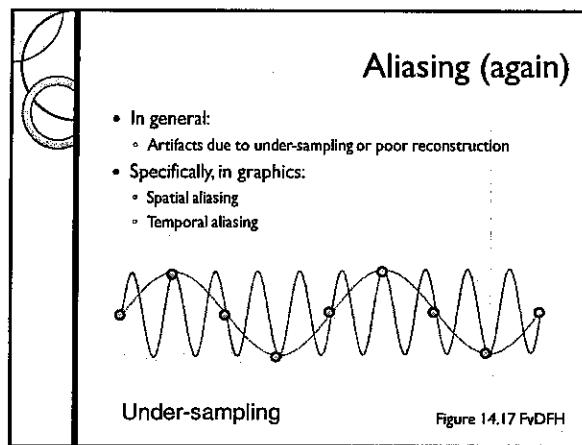
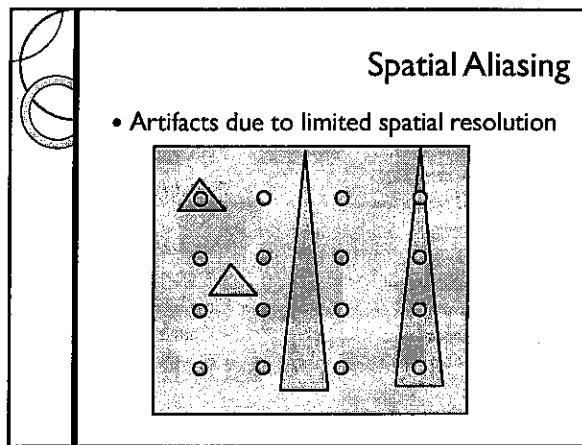
Source image

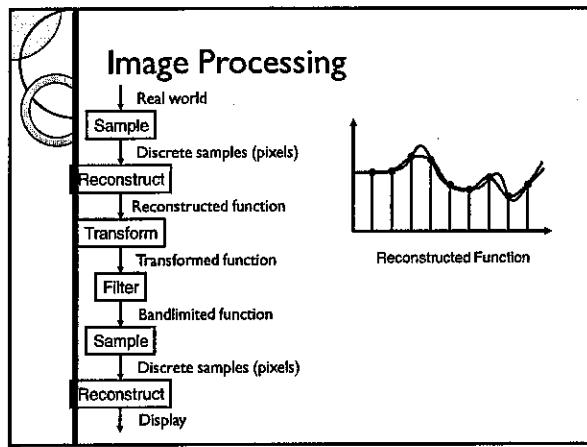
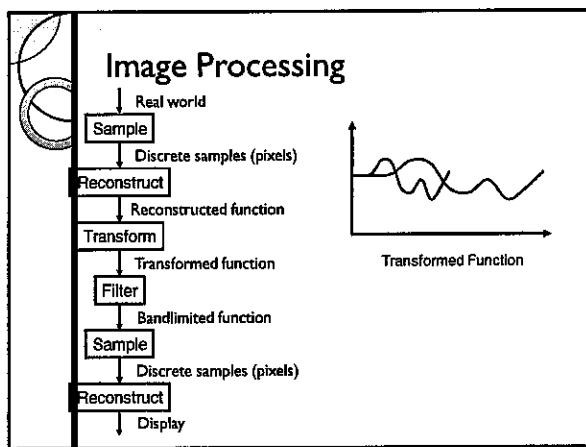
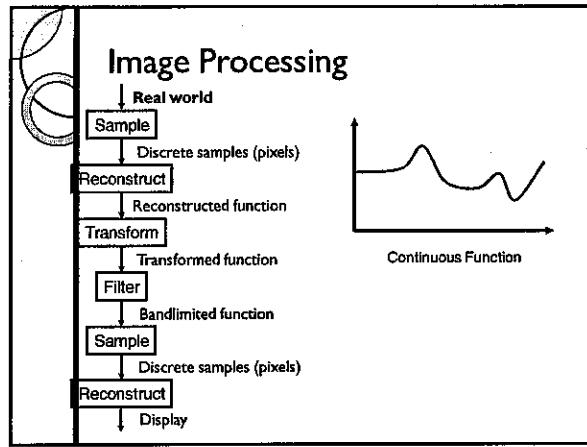
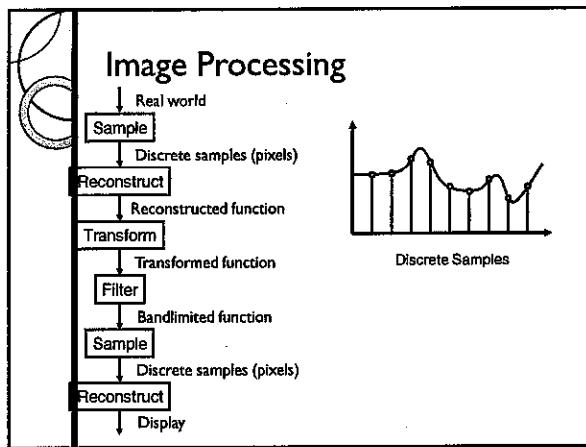


Destination image

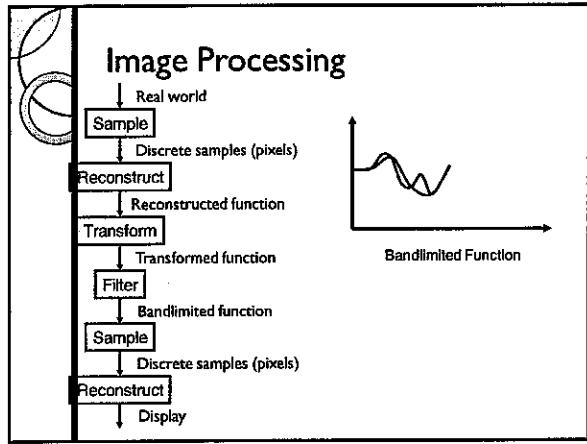
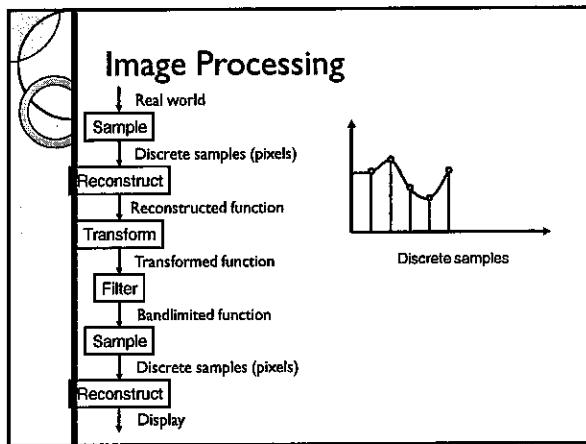
• BACK TO SAMPLING







(*) סענין זה מופיע במאמרם של אוניברסיטאות מוגדרות אלה (ד) בכרזם נטען שהם אוניברסיטאות.



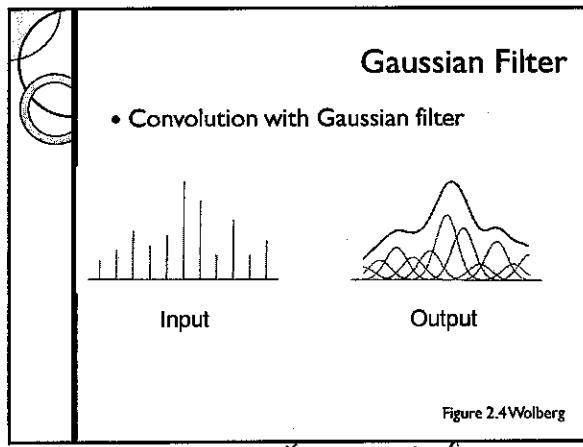
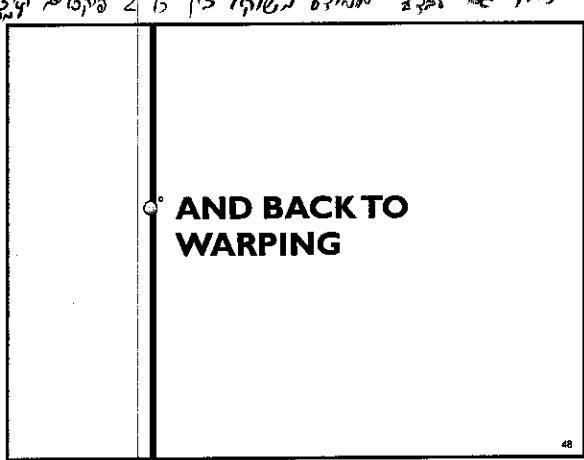
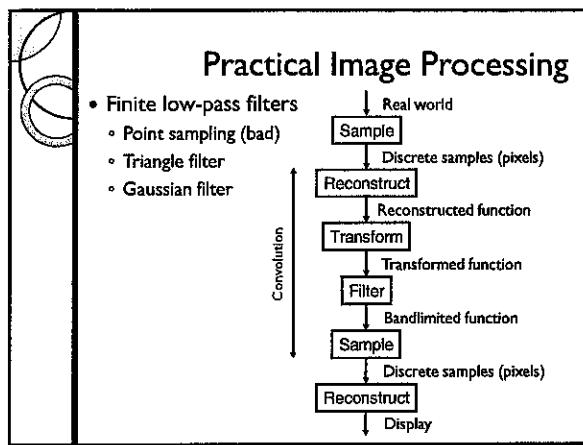
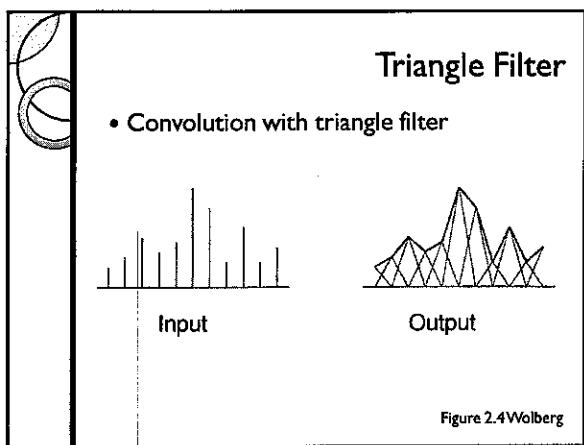
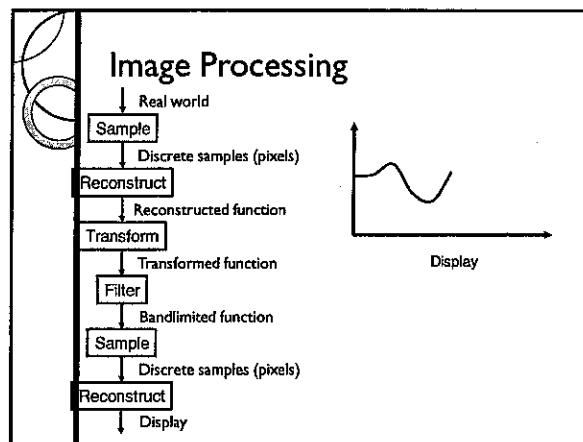
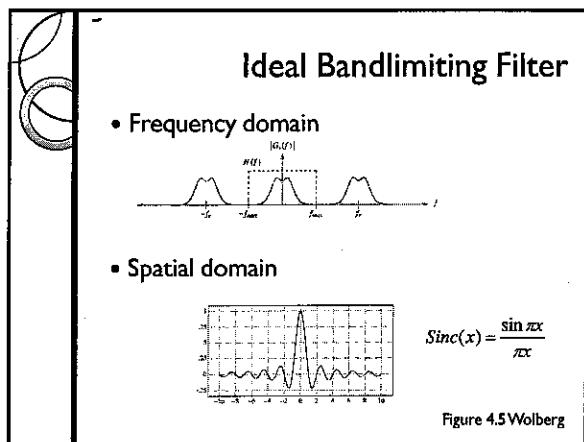


Image Resampling

- Compute weighted sum of pixel neighborhood
 - Output is weighted average

Kernel Function

```
dst(u,v)=0;
for(ix=u-w; ix<=u+w; ix++)
  for(iy=v-w; iy<=v+w; iy++)
    d=dist between (ix,iy) and (u,v)
    dst(u,v) += k(ix,iy) * src(ix,iy)
```

Image Resampling

- What if we are resampling a 2D image?

Triangle Filtering (width <= 1)

- Bilinearly interpolate four closest pixels
 - a = linear interpolation of $\text{src}(u_1, v_2)$ and $\text{src}(u_2, v_2)$
 - b = linear interpolation of $\text{src}(u_1, v_1)$ and $\text{src}(u_2, v_1)$
 - $\text{dst}(x,y) = \text{linear interpolation of } a \text{ and } b$

Image Resampling

- For isotropic Triangle and Gaussian filters, $k(ix,iy)$ is a function of d and w

Triangle filter
 $k(i,j)=1 - d/w$

Image Scale

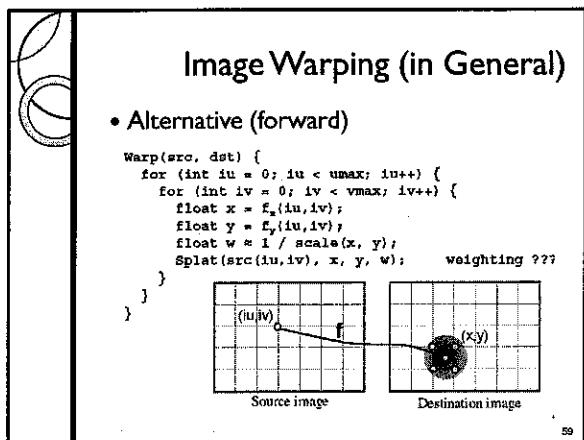
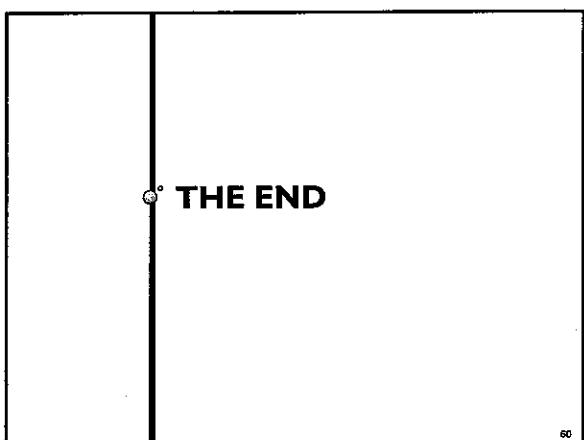
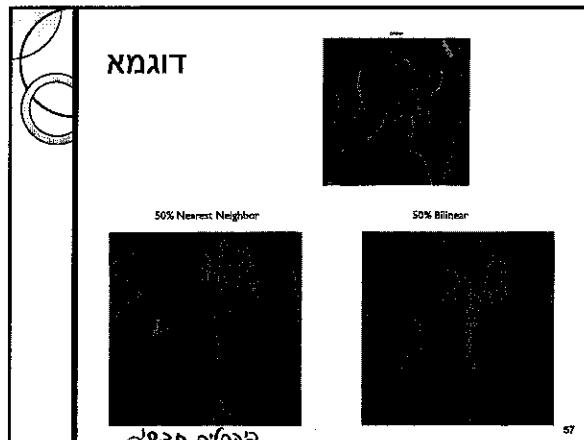
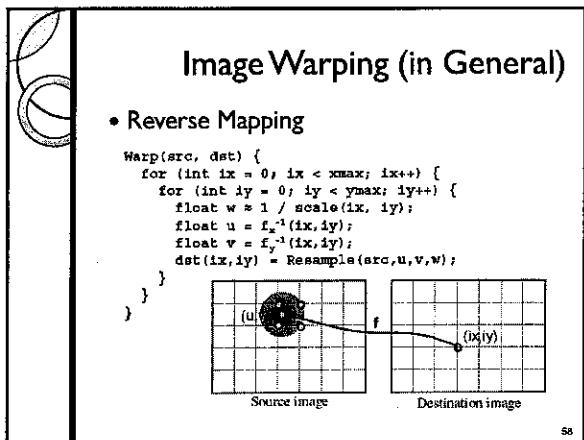
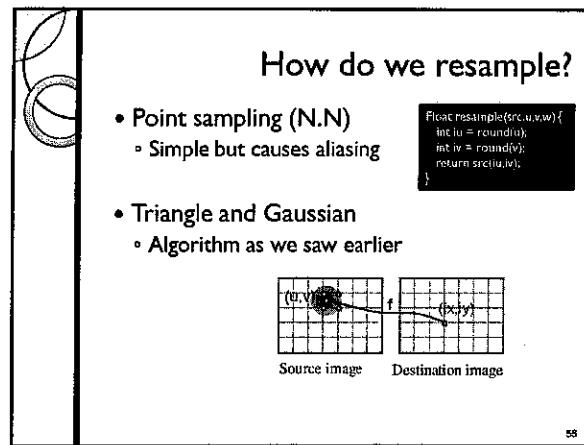
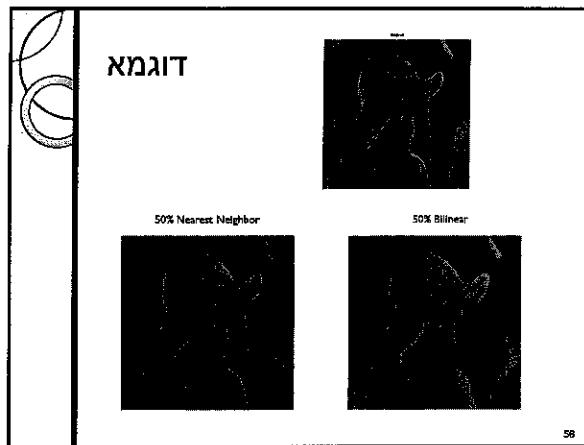
- Scale (src, dst, sx, sy):

```
w = max(1/sx, 1/sy);
for (int ix = 0; ix < xmax; ix++) {
  for (int iy = 0; iy < ymax; iy++) {
    float u = ix / sx;
    float v = iy / sy;
    dst(ix,iy) = resample(src,u,v,k,w);
  }
}
```

Gaussian Filtering

- Kernel is a Gaussian function

Gaussian Function
 $G_\sigma(d) = e^{-(d/\sigma)^2}$



Poisson Image Editing

Exercise 1
Due date: 30.03.09

General Description

The purpose of this exercise is to understand and implement a Poisson seamless cloning image editing tool.

- Part 1: Smooth image completion
- Part 2: Poisson seamless cloning

Input Images



source image



target image

Simple Cloning Result



Poisson Seamless Cloning Result



Some More Results

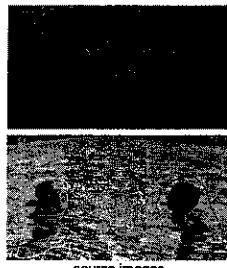


source images



target image

Some More Results



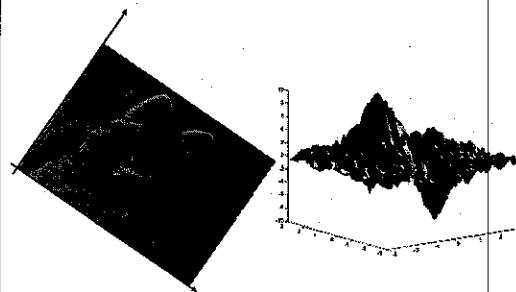
Some More Results



Part 1

Smooth Completion

Image as a 2D Function



Smooth Image Completion

What if there is a missing area ?



f^* - the known image
Scalar 2D function from (x,y) to grayscale value.
 f - the image in the unknown area
 Ω - the unknown area (domain of f)

$$\arg \min_f \iint_{\Omega} |\nabla f|^2 \text{ s.t. } f|_{\Omega} = f^*|_{\Omega}$$

Will complete the area as smoothly as possible.

Smooth Image Completion (Cont.)

Finding the minimum:

$$\arg \min_f \iint_{\Omega} |\nabla f|^2 \text{ s.t. } f|_{\Omega} = f^*|_{\Omega} \quad \xrightarrow{\text{Euler-Lagrange}}$$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{Discrete Apx: } \frac{\partial f}{\partial x} \equiv f_{x+1,y} - f_{x,y}$$

$$\Delta f(x,y) \equiv f_{x+1,y} - 2f_{x,y} + f_{x-1,y} + f_{x,y+1} - 2f_{x,y} + f_{x,y-1} = f_{x+1,y} + f_{x-1,y} + f_{x,y+1} + f_{x,y-1} - 4f_{x,y} = 0$$

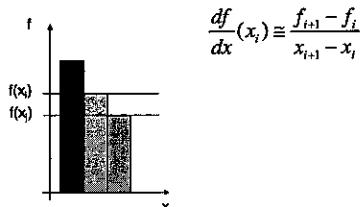
המינימום של f מתקבל כוון מינימום בז'ר

$a | b | c | d$
 $b - c$

$b - 2c - d$

Discrete Derivative in 1D

- Given a discrete function $f(x_i) = f_i$



Smooth Image Completion (Solving)

- Each $f_{x,y}$ is an unknown variable x_i , total of N variables (covering the unknown pixels)

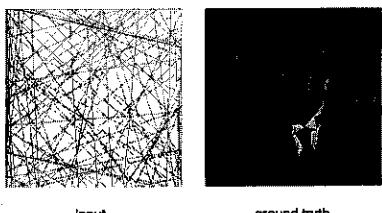
$$f_{x,y-1} + f_{x-1,y} - 4f_{x,y} + f_{x+1,y} + f_{x,y+1} = 0 \Rightarrow x_{i-w} + x_{i-1} - 4x_i + x_{i+1} + x_{i+w} = 0$$

- Reduces to the sparse algebraic system:

$$\begin{matrix} 1 & 1 & -4 & 1 & 1 \\ 1 & 1 & -4 & 1 & 1 \\ 1 & 1 & -4 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & -4 & 1 & 1 \end{matrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ b_1 \\ b_2 \\ 0 \\ 0 \\ b_3 \\ 0 \end{pmatrix}$$

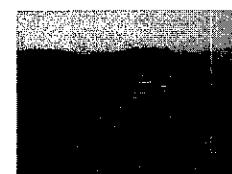
Known values of f_i contribute to the left side
 $x_{i-w} + x_{i-1} - 4x_i + x_{i+1} + x_{i+w} = f(x,y+1)$

Example



Part 2

Poisson Cloning



Poisson Cloning: "Guiding" the completion

- We can guide the completion from part 1 to fill the hole using gradients from another source image
- Reverse: Seek a function f whose gradients are closest to the gradients of the source image

Poisson Cloning

$$\arg \min \iint_{\Omega} |\nabla f - G|^2 \text{ s.t. } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$G = \nabla \text{source image}$
(forward difference)

$$\Delta f = \operatorname{div} G \text{ over } \Omega \text{ s.t. } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$$\operatorname{div} G = \frac{\partial G}{\partial x} + \frac{\partial G}{\partial y} \approx G_x(x,y) - G_x(x-1,y) + G_y(x,y) - G_y(x,y-1)$$

(backward difference)

Poisson Cloning (Solving)

- Each $f_{x,y}$ is a variable x_i as before, solving

$$\begin{aligned} f_{x,y+1} + f_{x-1,y} - 4f_{x,y} + f_{x+1,y} + f_{x,y+1} &= \text{div } G(x,y) \\ \Rightarrow x_{i,w} + x_{i,-1} - 4x_i + x_{i+1} + x_{i,w} &= \text{div } G(x,y) \end{aligned}$$
 - As before this reduces to a sparse algebraic system

Buffered Image

הנְּפָרְתָּה וְהַמִּלְחָמָה בְּעֵדָה כְּבָשָׂתָה

לפניהם נרמזו מילים ושמות של מושגים ומקומות. (4)

תְּמִימָנָה אֶתְבָּרְכָה (אַבְּדָן) בְּגַעֲמָה בְּגַעֲמָה (בְּגַעֲמָה)

knowing his (or her) (etc.)