Introduction to Image Processing

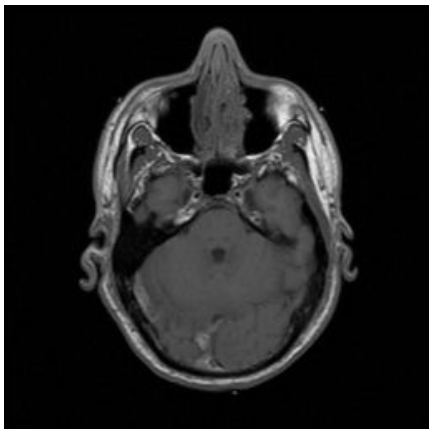
Sebastian Schüppel

Portland State
UNIVERSITY

Portland State University
Department of Electrical and Computer Engineering

01|23|2008

Intro



- Image Enhancing
- Object Recognition
- Object Tracking
- Robotic
- Medical Science
- Security
- Quality Control ...

Cross Correlation

How do we compare Signals?

- Cross-Correlation

$$r_{x,y}(\tau) = \int_{-\infty}^{\infty} x(t)\, y(\tau + t) dt \tag{1}$$

$$r_{x,y}(n) = \sum_{m=0}^{N-1} x_m\, y_{n+m} = x^*(n) * y(n) \tag{2}$$

Fourier Transform

$$X(n) = \sum_{k=0}^{N-1} x_k\, e^{-2\pi i k \frac{n}{N}} \tag{3}$$

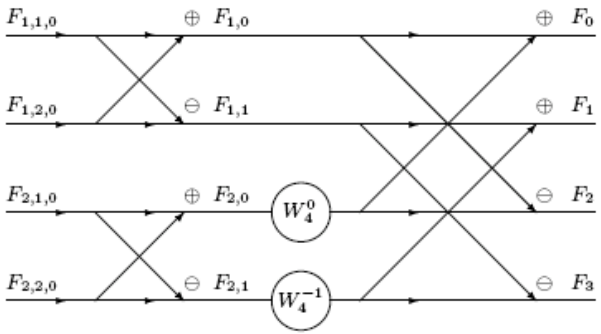
$$x^*(n) * y(n) \circ\!\!-\!\!\bullet\, X^*(k)\, Y(k) \tag{4}$$

Fast Fourier Transformation

Use of FFT algorithm

- DFT: $O(n^2)$ vs. FFT: $O(n\ log_2(n))$
- generally know through Cooley / Tukey in mid 60s
- divide and conquer - use $N = 2^p$

$$X(n) = \sum_{k=0}^{N/2-1} x_{2k}\ e^{-2\pi i(2k)\frac{n}{N}} + \sum_{k=0}^{N/2-1} x_{2k+1}\ e^{-2\pi i(2k+1)\frac{n}{N}} \quad (5)$$

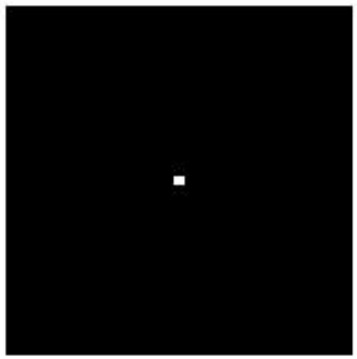


2-D Fourier Transformation

Fourier Transformation for Images

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_{x,y}\, e^{-2\pi i \frac{xu+vy}{N}} \tag{6}$$

$$F(u,v) = \sum_{x=0}^{N-1} F_{x,v}\, e^{-2\pi i \frac{xu}{N}} \quad \textit{and} \quad F(x,v) = \sum_{x=0}^{N-1} f_{x,y}\, e^{-2\pi i \frac{vy}{N}}$$



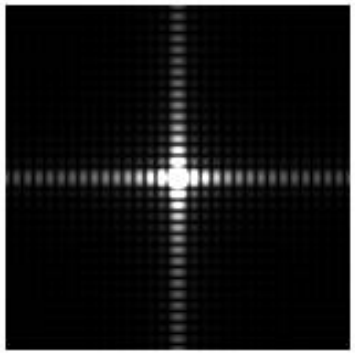
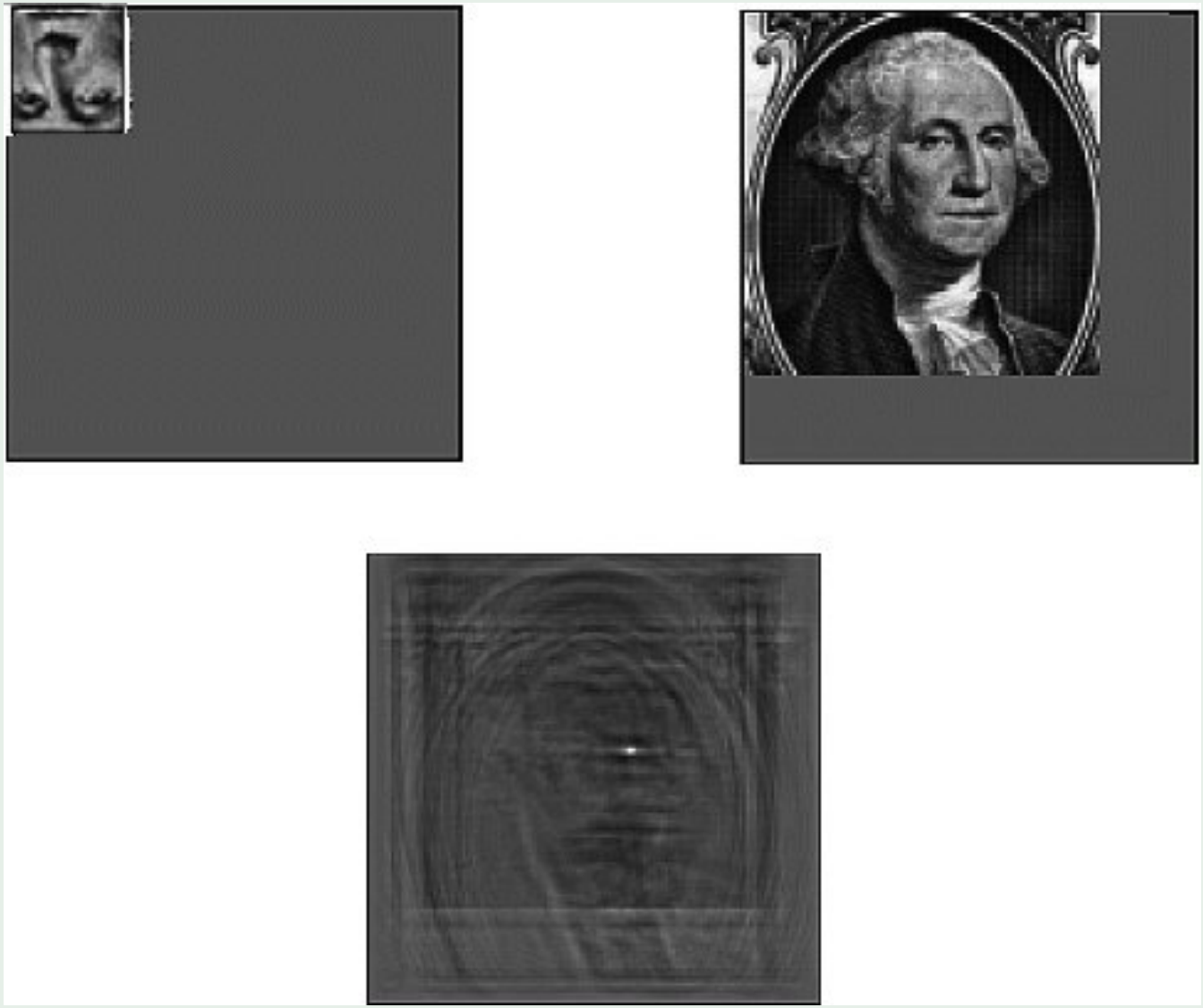


Image Matching

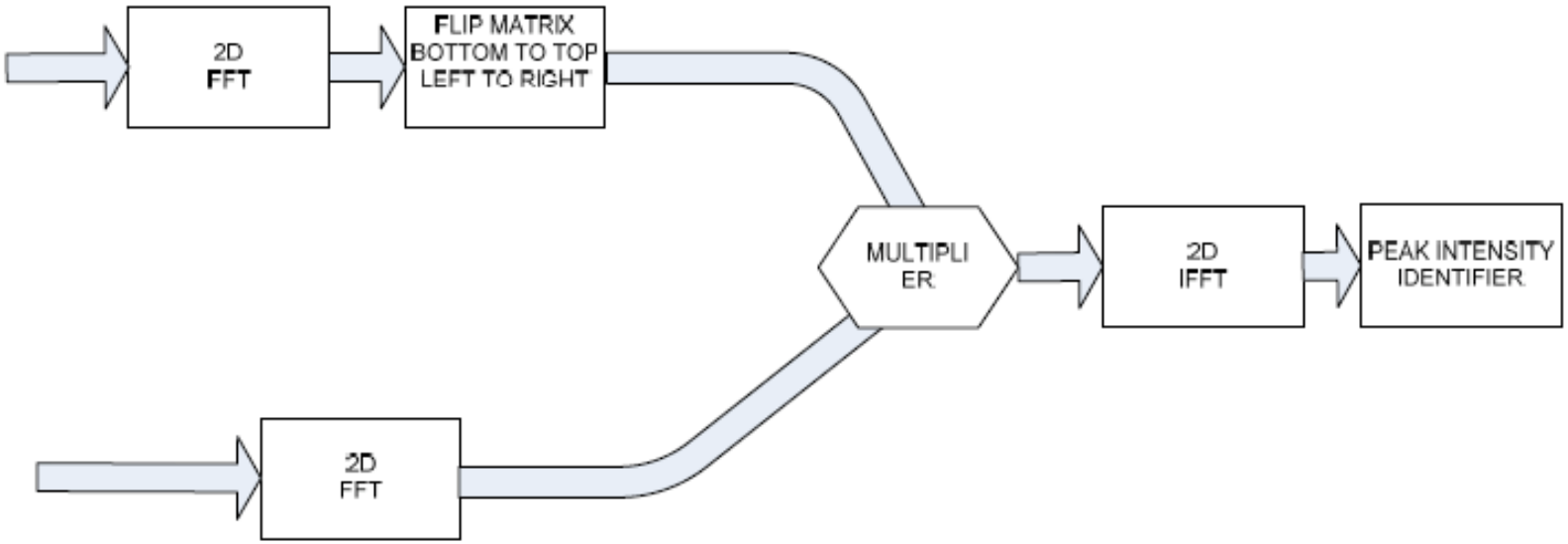
Correlation with images



FFT IP-Cores

Why using IP-Cores?

- Easy to Implement
- Free (Altera + Xilinx)
- Optimized in Speed and Area
- Customize



Summary

- optimize logic
- check FFT-core options
- use of external RAM modules
- different algorithm