CS 3570

Final Project

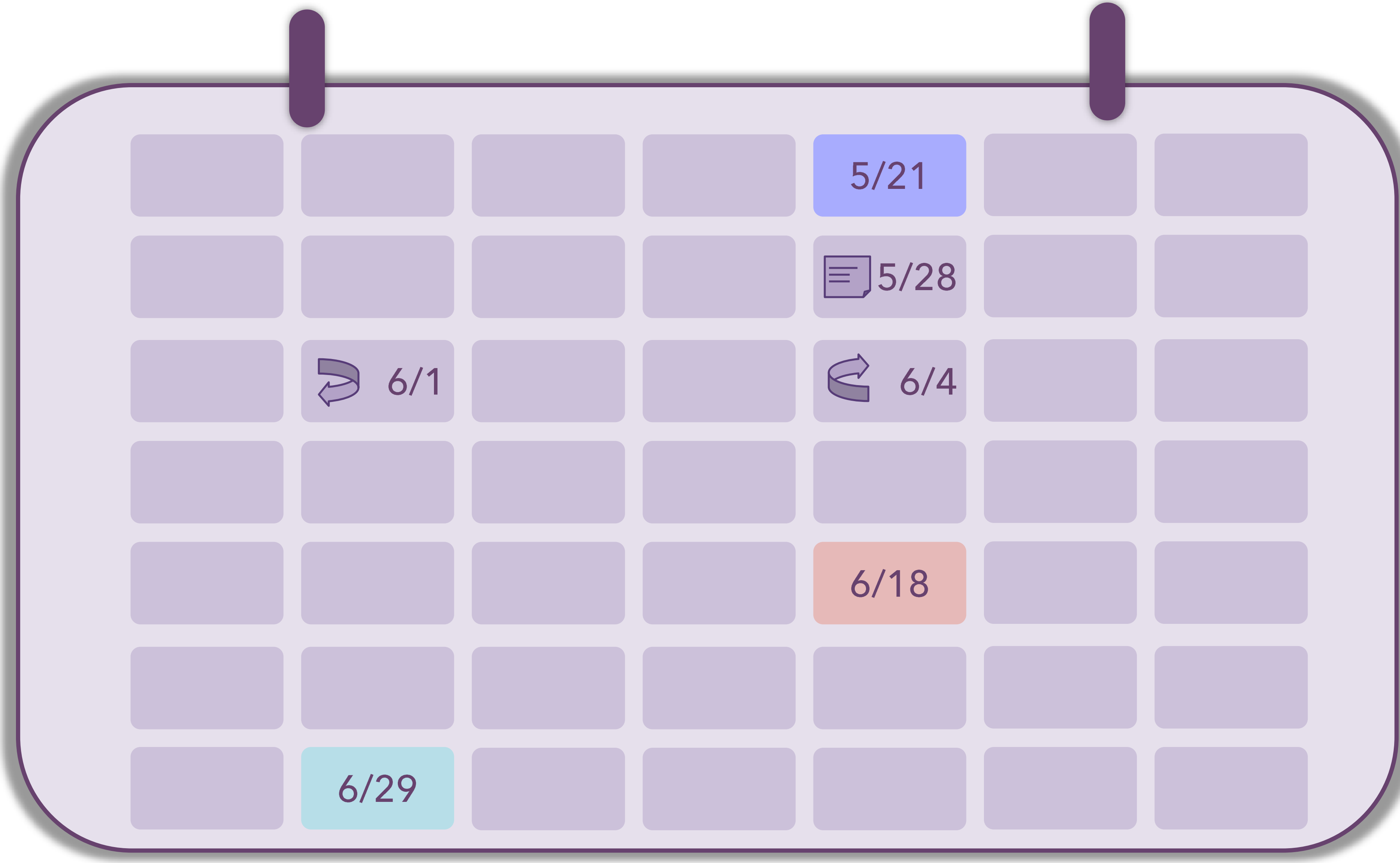
2015

Ready for challenges ?

Part 2 (5/21)

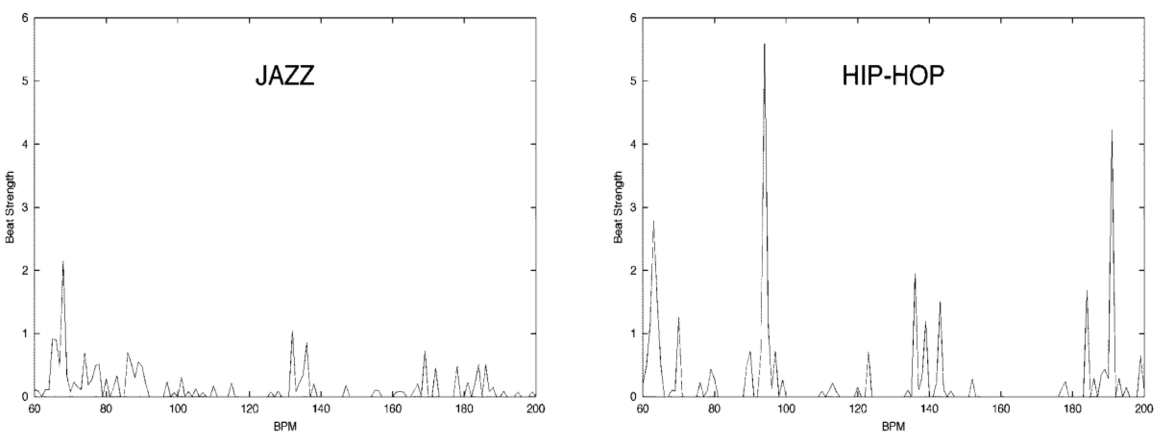
Department of Computer Science
National Tsing Hus University

Calander



Genre classification

- Musical genres are labels created and used by humans for categorizing and describing the vast universe of music.
- Apply musical genre classification of audio signals for our dataset (**GTZAN**).
- This work involve Pattern Recognition. You can use part of the musics to do **training** and the remain part to do **testing**.
- **GTZAN** has 10 group, each group has 100 songs. Each group chooses 60 songs to do training and 40 songs to do testing.

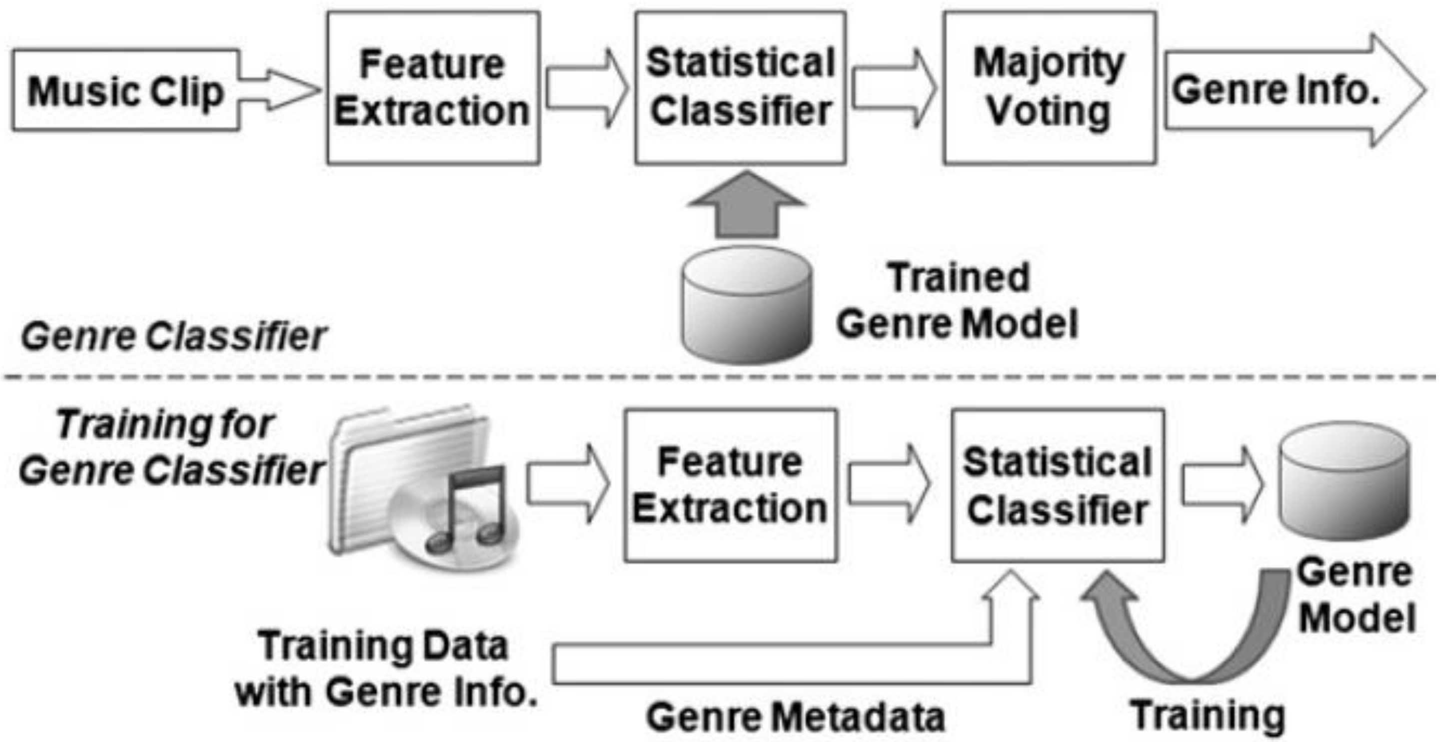


feature example (Beat histogram)

- Ref: George Tzanetakis, Student Member, Perry Cook "Musical Genre Classification of Audio Signals" IEEE 2002

Method introduction of Genre classification

- Overview



Method introduction of Genre classification

- How to do “Classification” ?

  1. Preprocessing
  2. Feature extraction
  3. Modeling
     - Description of each class in mathematical form
  4. Classification
     - The classifier divides the feature space into class regions

Method introduction of Genre classification

- Example – feature space and training sample(2D)

✓ **Audio Feature:** http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/Lecture%208%20-%20Audio%20Features2.pdf

✓ **Basic features**

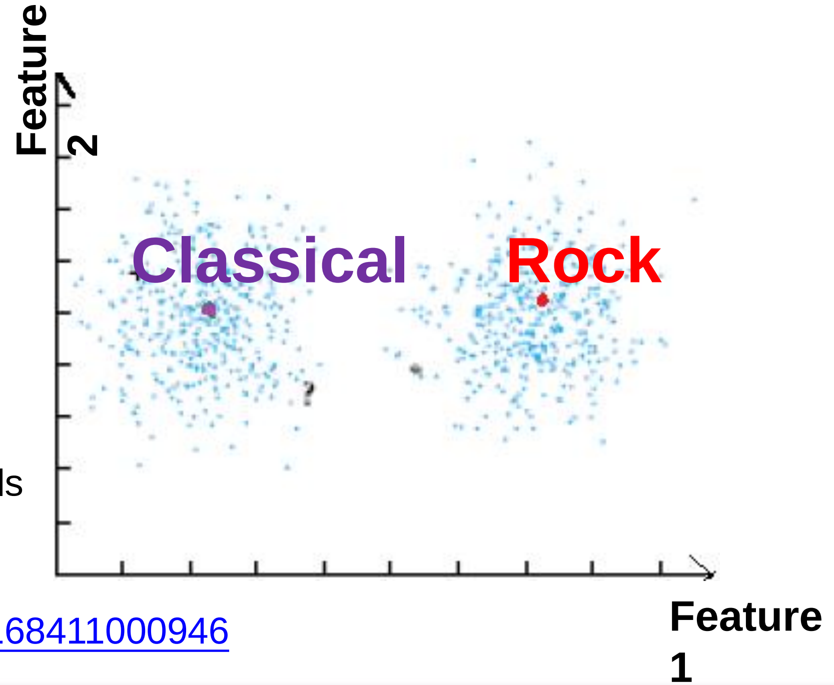
- Volume, Pitch, Timbre, Zero-crossing rate…

✓ **Advanced features**

- Centroid, Rolloff, Flux …

✓ **Others**

- Automatic Musical Genre Classification Of Audio Signals
  http://ismir2001.ismir.net/pdf/tzanetakis.pdf
- Higher-order moments for musical genre classification
  http://www.sciencedirect.com/science/article/pii/S0165168411000946



Method introduction of Genre classification

- Use training data to do modeling

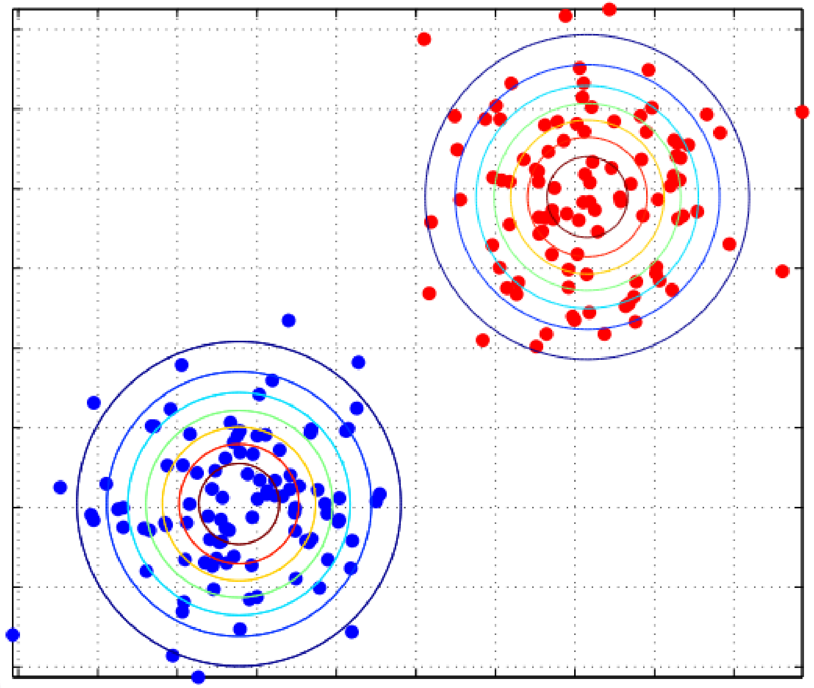
✓ **Description of each class in mathematical form** (e.g. Gaussian Distribution)

Estimate the mean:

$$\mathbf{m} = \frac{1}{N}\sum \mathbf{x}_i$$

Estimate the covariance:

$$\mathbf{C} = \frac{1}{N-1}(\mathbf{x}-\mathbf{m})^T \cdot (\mathbf{x}-\mathbf{m})$$



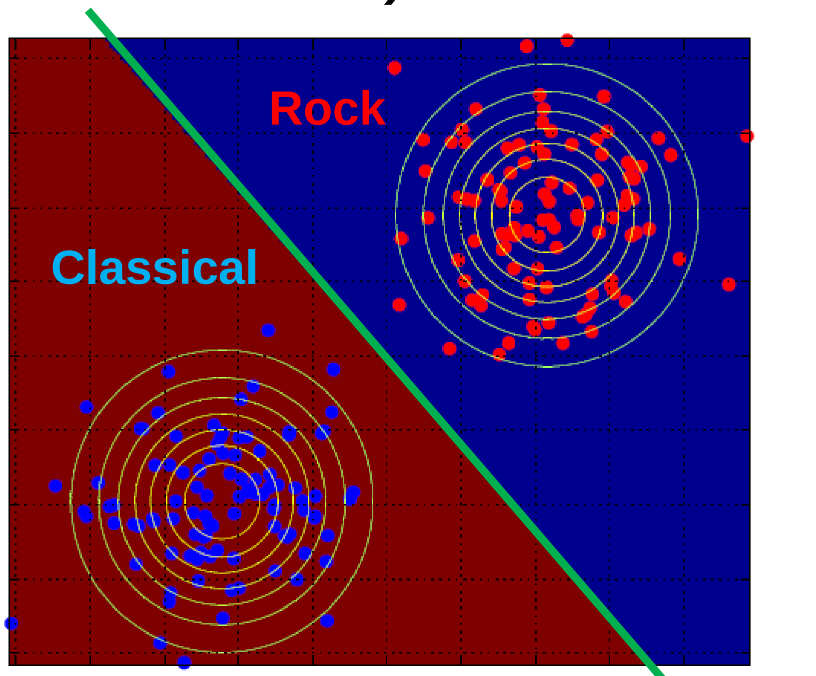
Method introduction of Genre classification

- Use testing data to do Classification
- Discriminant function (boundaries)

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

g(x) > 0 → Rock music

g(x) < 0 → Classical music



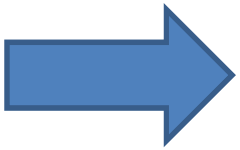
- Gaussian Classifiers

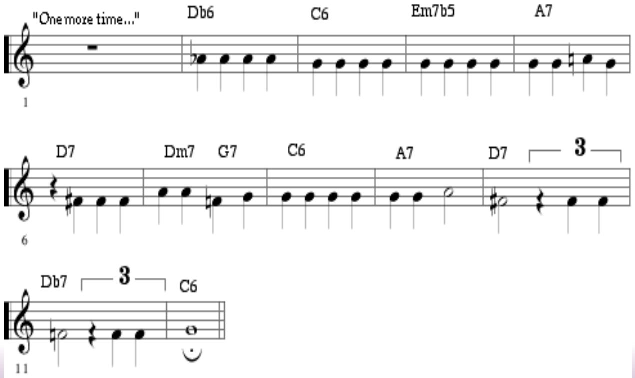
  - http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/Lecture%206%20-%20Gaussian%20Classifiers.pdf

Automatic Chord Transcription

- In music, **transcription** can mean notating a piece or a sound which was previously unnotated.
- Musicians who do not have intuitive transcription skills will search for sheet music or a chord chart, so that they may quickly learn how to play a song.
- Use **Robbie Williams Chord and Key Annotation Dataset** as ground truth, there are 5 albums and 65 songs in this dataset.

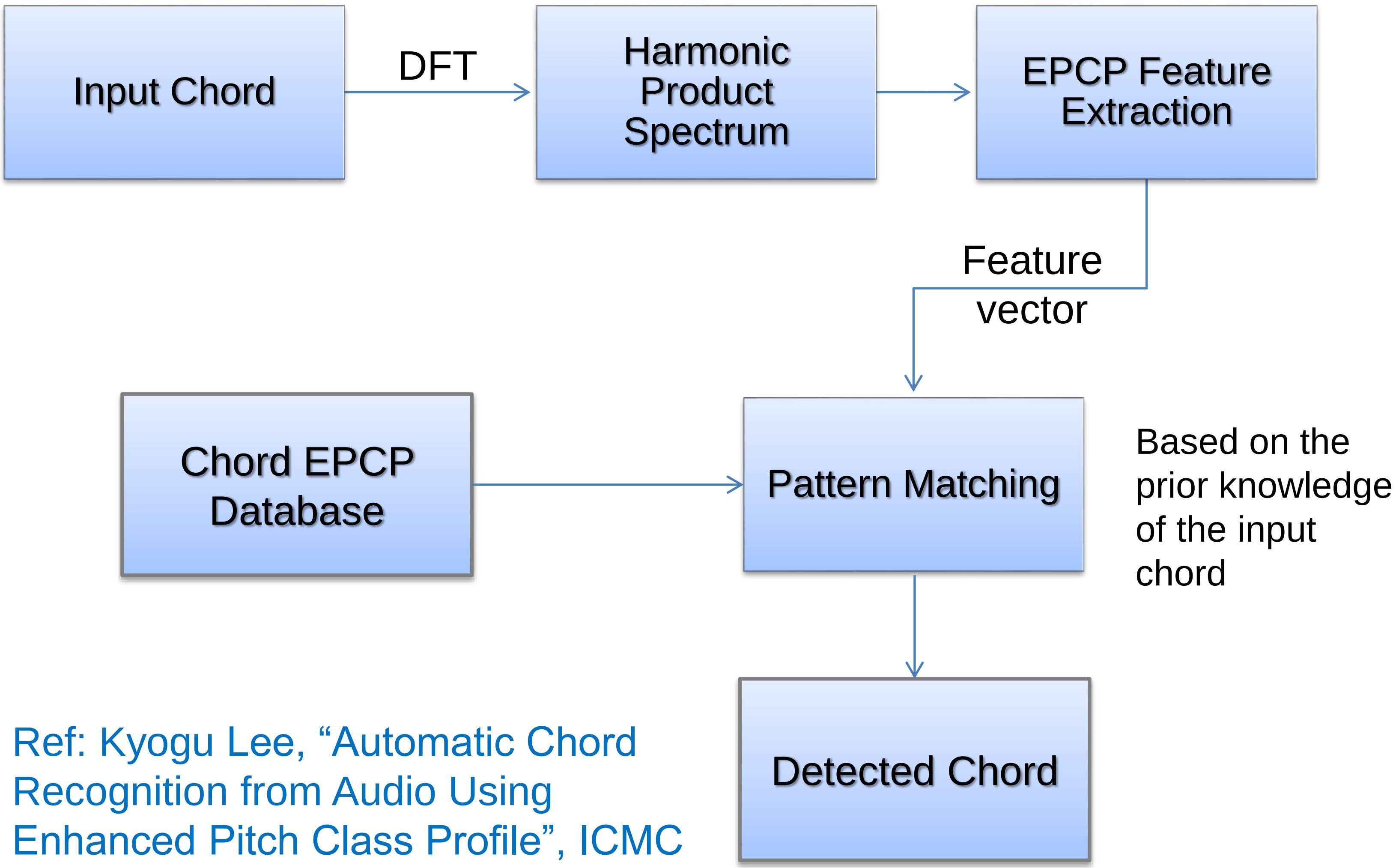
RobbieWilliamsAnnotations.zip:
http://maxzanoni.altervista.org/chord-tracking/





Department of Computer Science
National Tsing Hua University

Method introduction of Chord transcription



- Ref: Kyogu Lee, “Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile”, ICMC 2006

Method introduction of Chord transcription

- Harmonic Product Spectrum(HPS)
  - If the input signal is a musical note, then its spectrum should consist of a series of peaks, corresponding to fundamental frequency with harmonic components at integer multiples of the fundamental frequency.

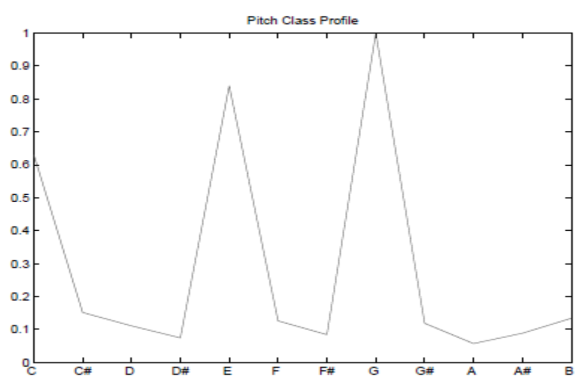
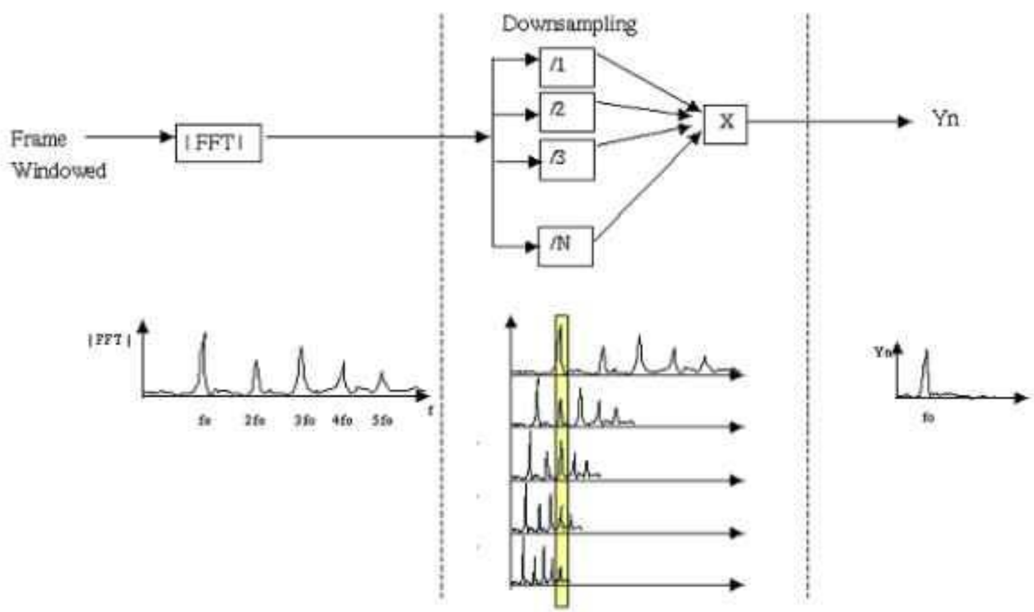


Figure 1: Chroma vector of a C major triad played by piano.

HPS Algorithm



- Ref: http://cnx.org/content/m11714/latest/

Method introduction of Chord transcription

- Enhanced Pitch Class Profile (EPCP)
  - 12-dimensional vector
  - which represents the relative intensity in each of twelve semitones in a chromatic scale.
- How to get Pitch Class Profile :
  - 1. Compute the DFT of the input signal $X(k)$
  - 2. Calculate the constant Q transform X_{CQ} from $X(k)$
  - 3. Obtain the chromagram vector CH as:

$$CH(b) = \sum_{m=0}^{M-1} |X_{CQ}(b + mB)|$$

Method introduction of Chord transcription

- Pattern matching
- Predefine the template of chord profiles
  - 12-dimensional
  - Each-bin is either 0 or 1
  - The template labeling is
    [C, C#, D, D#, E, F, F#, G, G#, A, A#, B]
  - Ex: a C major triad is
    [1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]
- Do correlation to EPCP with 24 major/minor templates

Department of Computer Science
National Tsing Hua University

Method introduction of Chord transcription

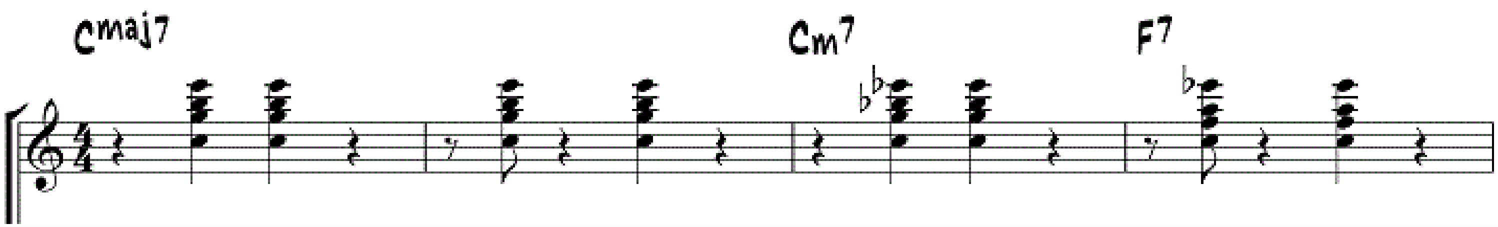
Evaluation

- **Given the time intervals of songs in dataset, you just recognize the chord each time interval.**

- $\boldsymbol{Correction\ rate} = \dfrac{\boldsymbol{the\ number\ of\ correct\ intervals}}{\boldsymbol{the\ number\ of\ all\ intervals}} \times \mathbf{100\%}$

Output Example

0.000000	1.000000	N
1.000000	5.377891	D:maj
5.377891	9.673469	G:maj
9.673469	14.100363	D:maj
14.100363	18.449524	G:maj
18.449524	22.881406	A:maj
22.881406	27.145646	G:maj
27.145646	31.545782	A:maj
31.545782	35.945578	G:maj
35.945578	40.307438	D:maj
40.307438	44.641859	G:maj
44.641859	48.993333	D:maj
48.993333	53.353379	G:maj
53.353379	57.730091	A:maj
57.730091	62.105941	G:maj
62.105941	66.466032	A:maj



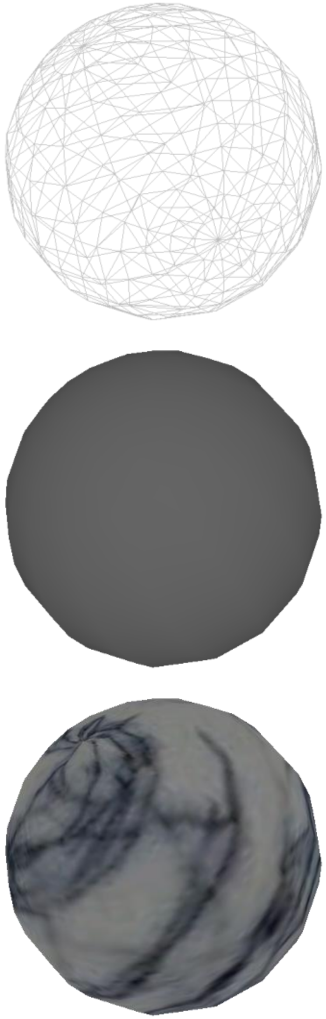
OpenGL/Direct3D based game / application

- 3D Graphics Library

Rendering some 3-D models in the world space

OpenGL/Direct3D based game / application

- 3-D model
  - Color model
  - Texture model



OpenGL/Direct3D based game / application

- Model transformation
  - Translation (moving)
  - Rotation (spinning)
  - Scaling
  - Other customized transformation using transformation matrix

OpenGL/Direct3D based game / application

- Viewing transformation
  - Camera position
  - Camera direction
  - Camera tilt

OpenGL/Direct3D based game / application

- Projection transformation
  - Focal length f and camera field of view
    - Parallel projection ($f \sim \infty$)
    - Perspective projection
    - Fish-eye effect ($f \sim 0$)

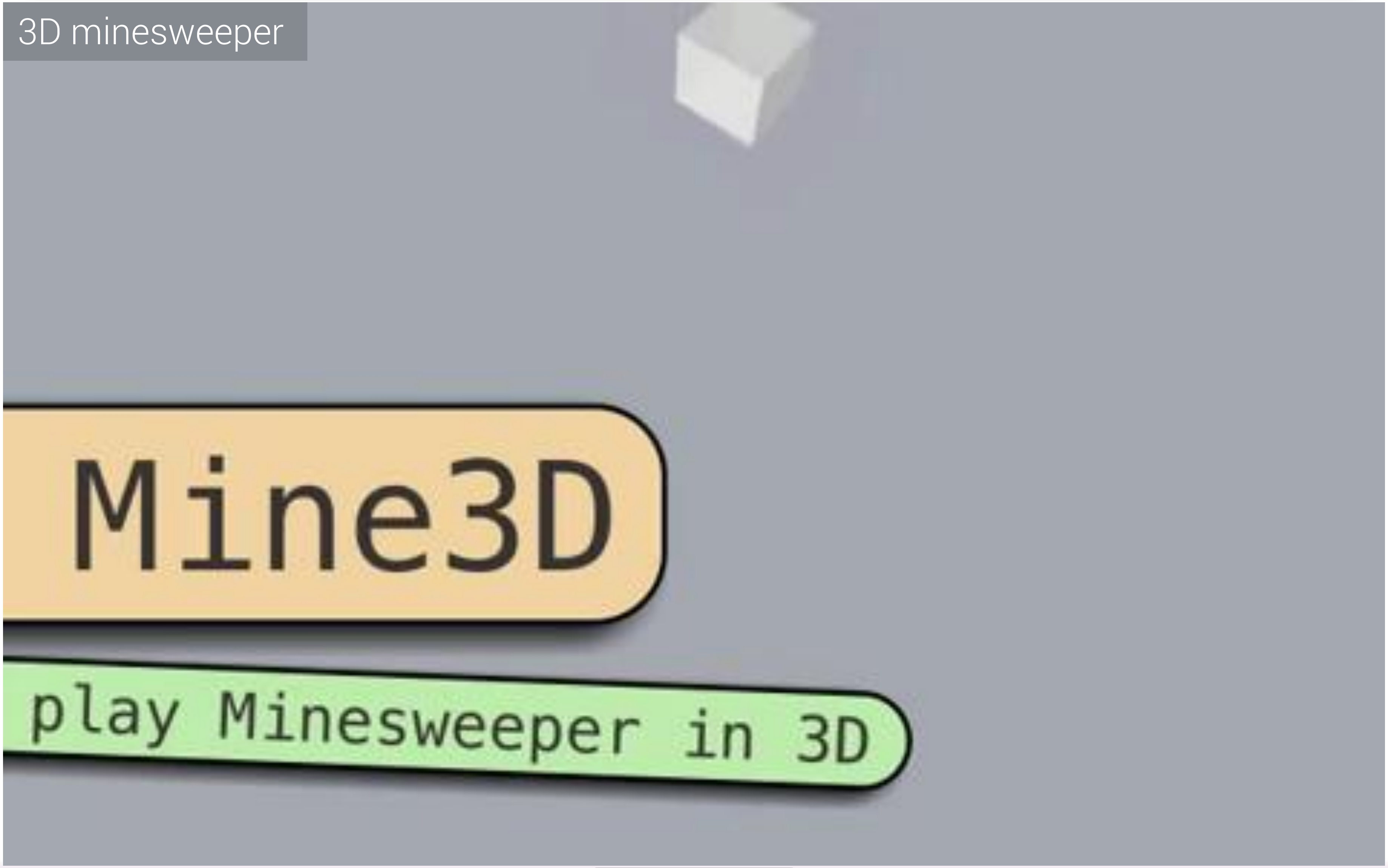
OpenGL/Direct3D based game / application

- Lighting
  - Positional (Point) Light
    - With or without spotlight effect
  - Directional (Parallel) Light
    - Positional light at infinity
  - Intensity attenuation

OpenGL/Direct3D based game / application

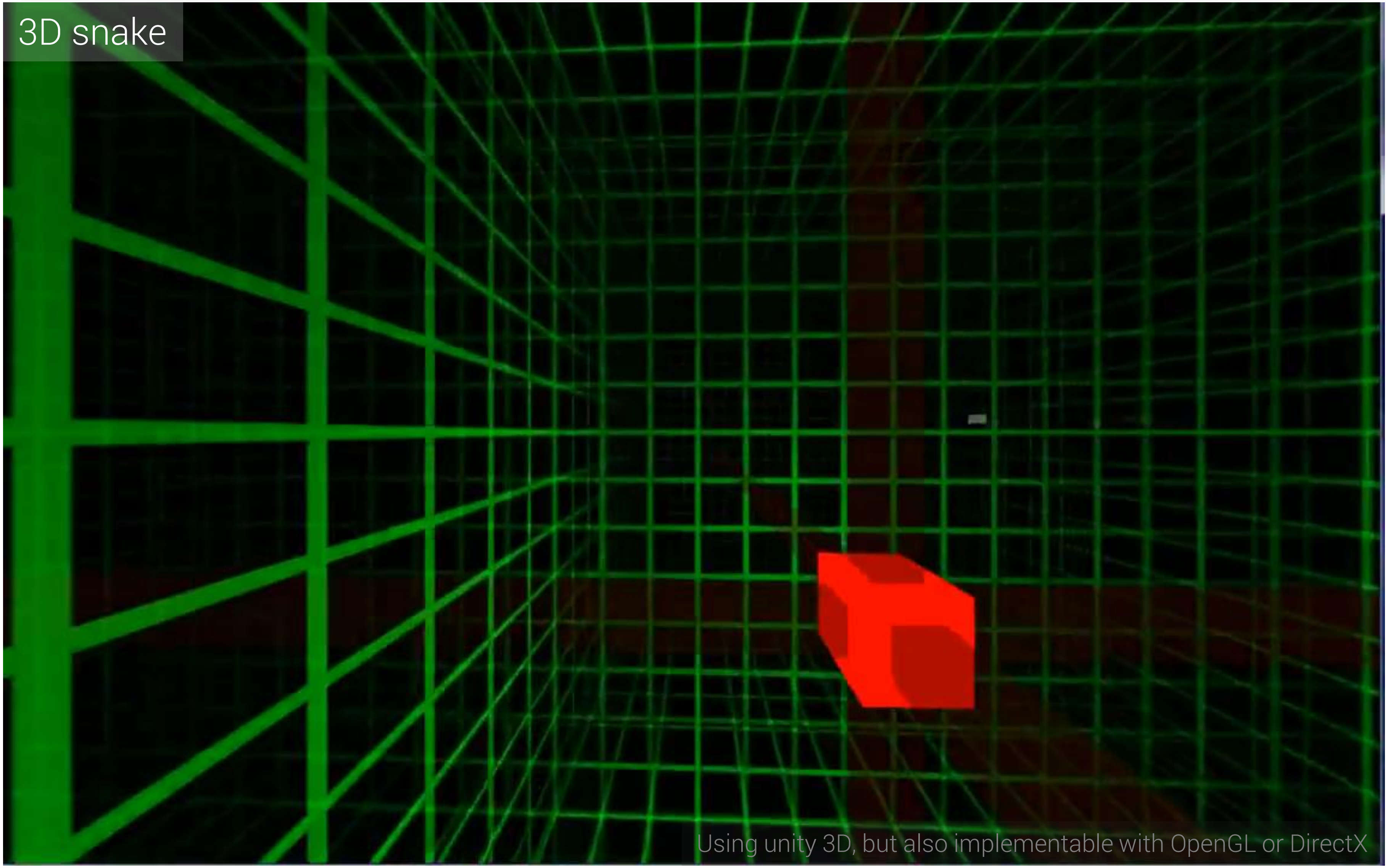
- Lighting
  - Ambient
  - Diffuse
  - Specular
  - Other light effect using shader

OpenGL/Direct3D based game / application



Youtube

OpenGL/Direct3D based game / application

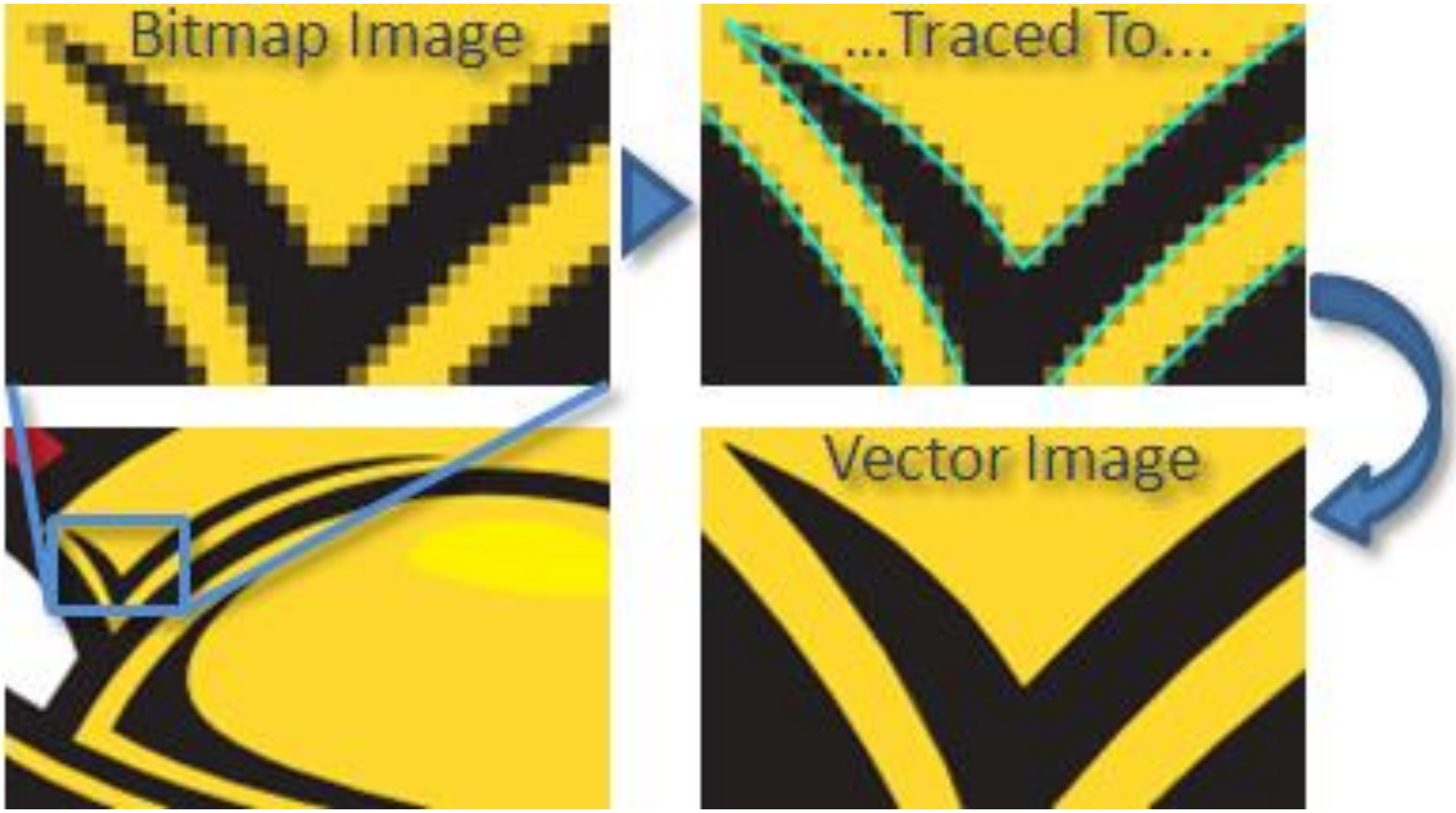


Youtube

OpenGL/Direct3D based game / application

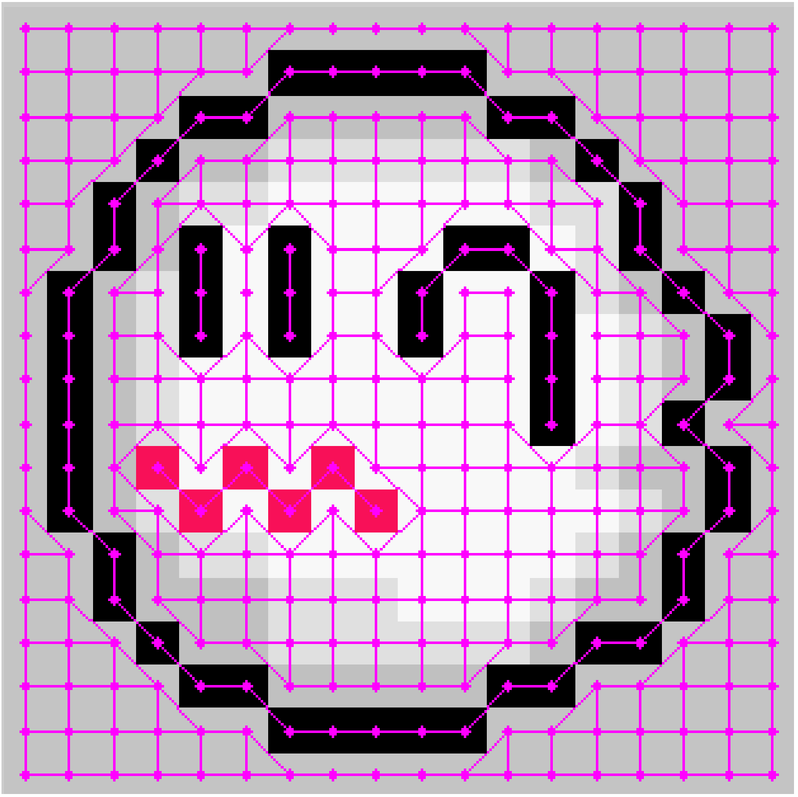
- Programming languages with OpenGL supports
- C/C++ (Freeglut)
- Java (JOGL) (LWJGL is recommended)
- Python (PyOpenGL)
- C# (CsGL)
- Objectve C (OpenGL ES)
- Java for Android (OpenGL ES)
- etc

Vectorization



Example from http://vectormagic.com/

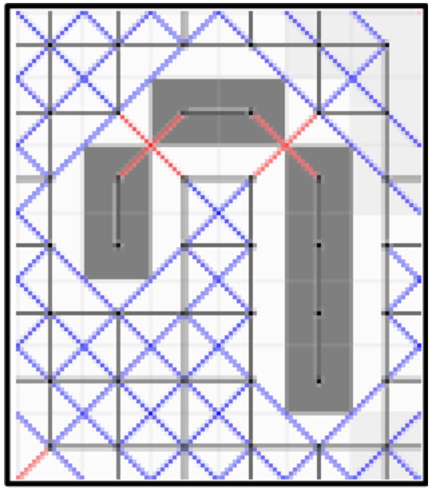
Vectorization



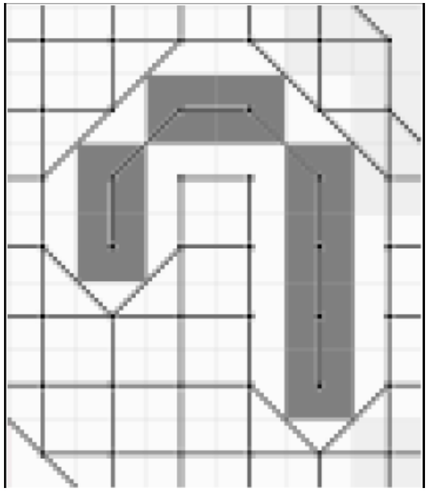
- Input a bitmap
- Extract geometry structure
- Apply curve fitting
- Output a vector image

From 蘇凱煜 and 林雯婷's masterpiece

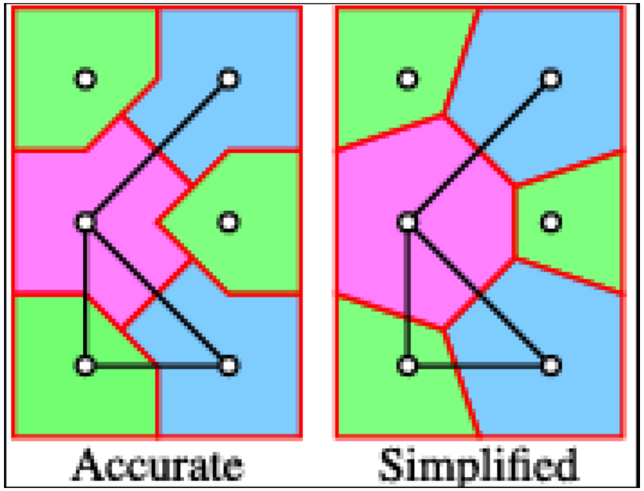
Vectorization



Construct similarity map



Remove Edge

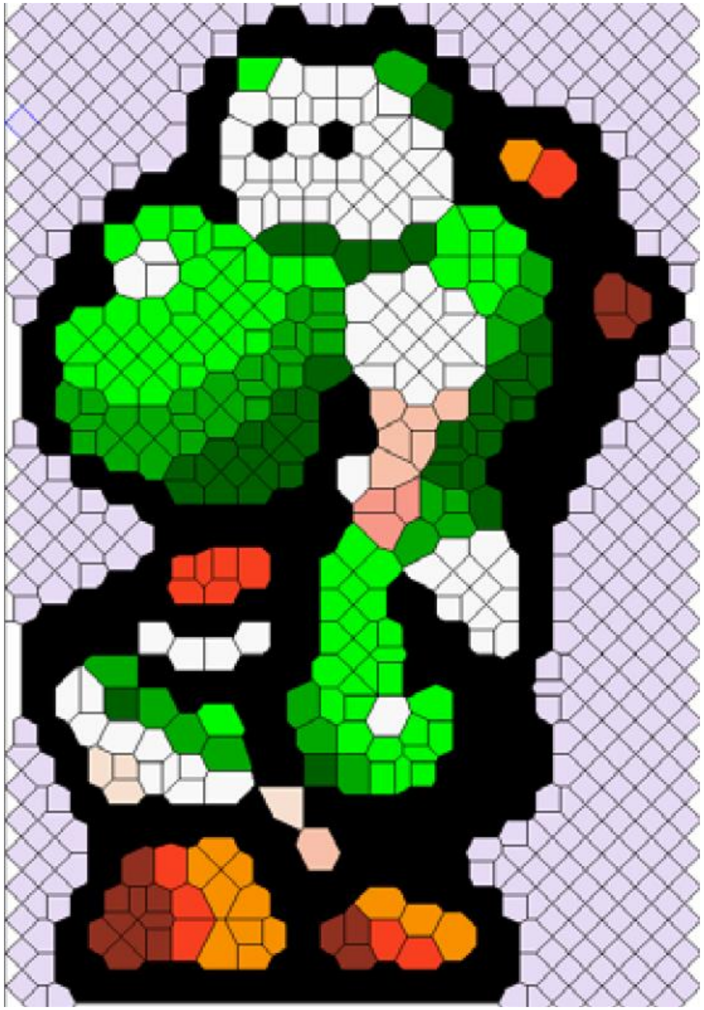


Construct Voronoi diagram

Reshape Cell

From 蘇凱煜 and 林雯婷's masterpiece

Vectorization



Referenced paper :

Depixelizing Pixel Art, Johannes Kopf, Dani Lischinski
Depixelizing Pixel Art (SIGGRAPH 2011)

Felix Kreuzer, Johannes Kopf, Michael Wimmer
Depixelizing Pixel Art in Real-Time (I3D 2015)

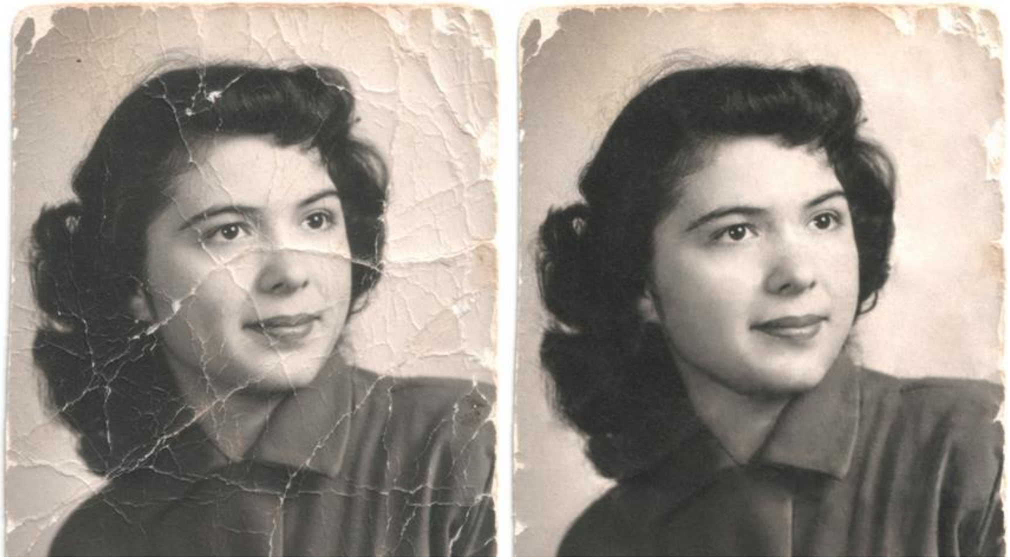
Image inpainting

- ***Inpainting*** is the process of reconstructing lost or deteriorated parts of images
- Reference:

  [Object Removal by Exemplar-Based Inpainting]

Image inpainting

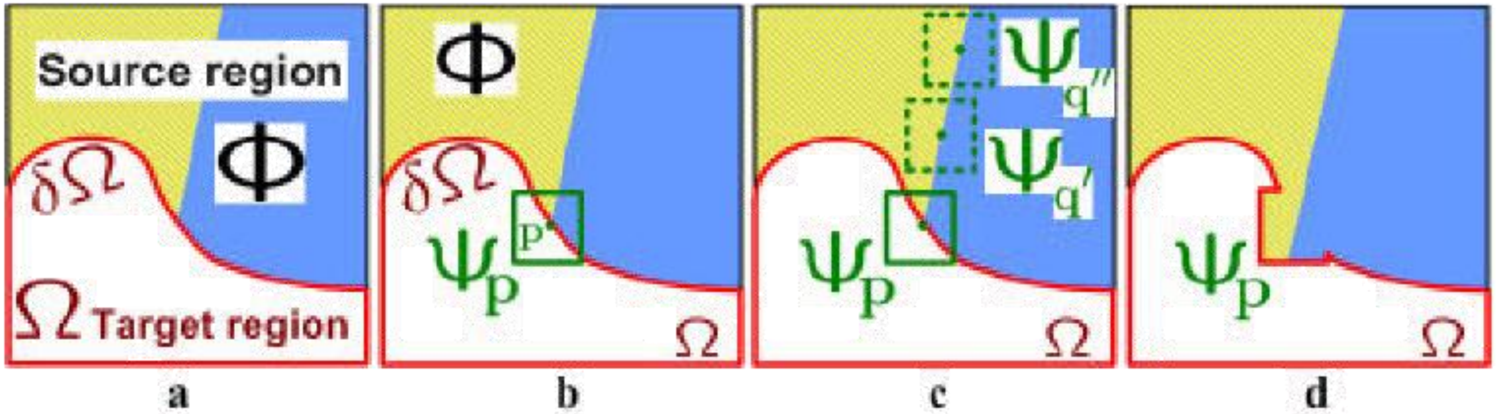
- Photo Restoration



- Object Removal

Image inpainting

- Isophote-driven image sampling process
- Find the best-match source patch



$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}) \qquad \mathbf{p} \in \delta\Omega$$

priority = confidence term * data term

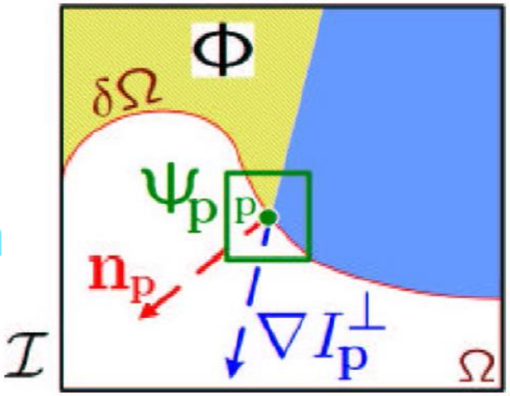


Image Matting

- In Image composition, a new image *I(x,y)* can be blended from a background image *B(x,y)* and foreground image *F(x,y)* with its alpha map α*(x,y)* $I = \alpha F + (1-\alpha)B$

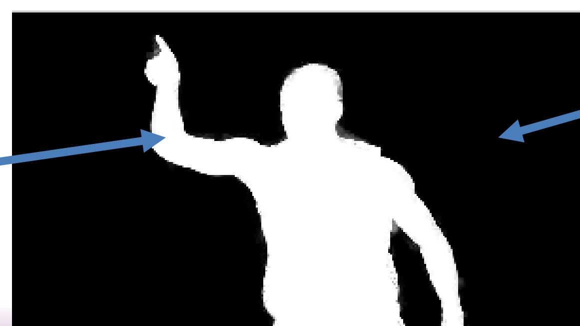
$= \alpha$

$+ (1-\alpha)$

Image foreground background

- What we need to do is to compute the "alpha map"

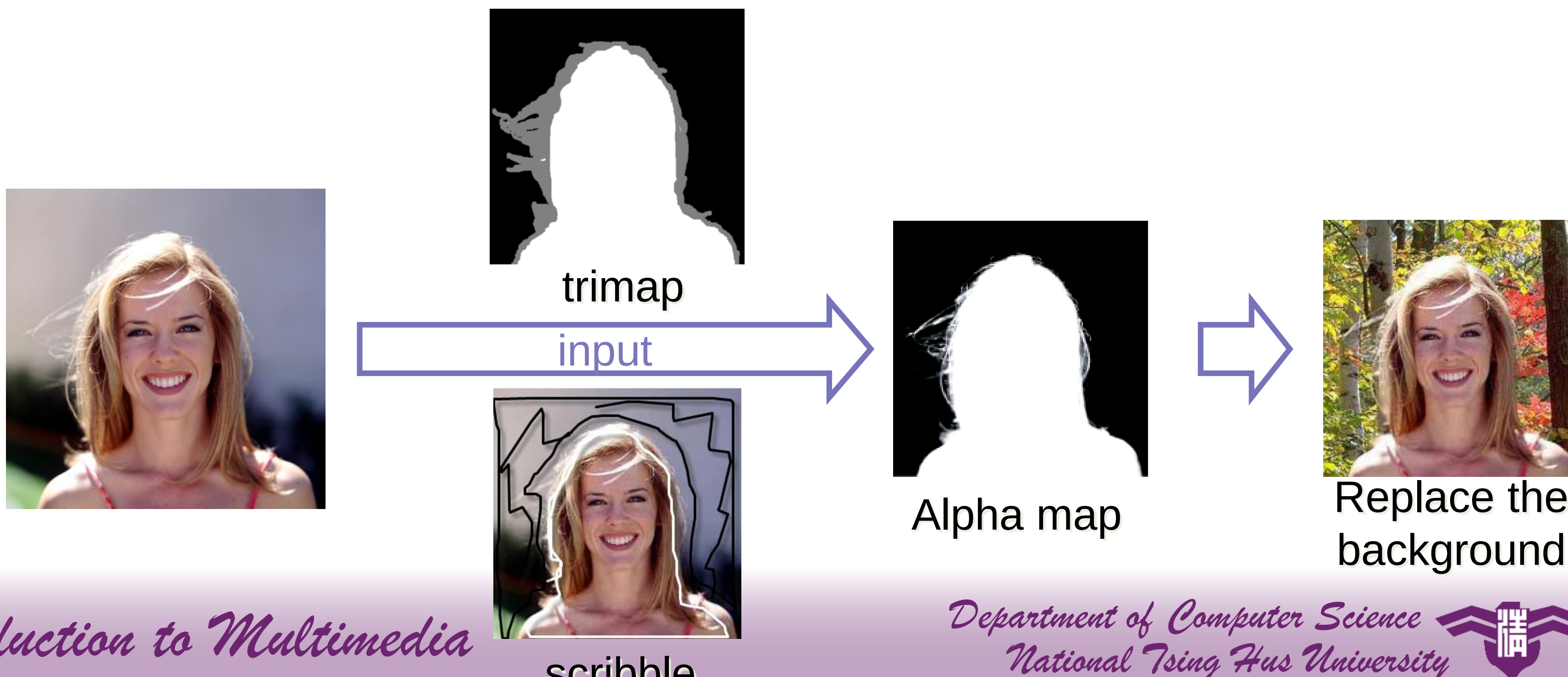
foreground



background

Method introduction of Image Matting

- Image matting often needs manual labeling(foreground, background, unknown) as input
  - e.g. trimap, scribbles
- Application: replace the background region by another background image in the input image
- Ref.
  - http://www.wisdom.weizmann.ac.il/~levina/papers/Matting-Levin-Lischinski-Weiss-CVPR06.pdf
  - http://www.wisdom.weizmann.ac.il/~levina/papers/spectral-matting-levin-etal-cvpr07.pdf



Non-local Means Image Denoising

- The goal of image denoising methods is to recover the original image from a noisy image for advanced image analysis
- A. Buades, B. Coll and J.M. Morel, “A non-local algorithm for image denoising”,IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2005.

Main Idea of NLM algorithm

- Given a discrete noisy image

  $v = \{v(i) | i \in I\}$,

  the estimated value $NL[v](i)$,

  for a pixel i, is computed as a ***weighted average of all the pixels in the image***

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j) \qquad 0 \leq w(i,j) \leq 1 \text{ and } \sum_j w(i,j) = 1$$

Weights

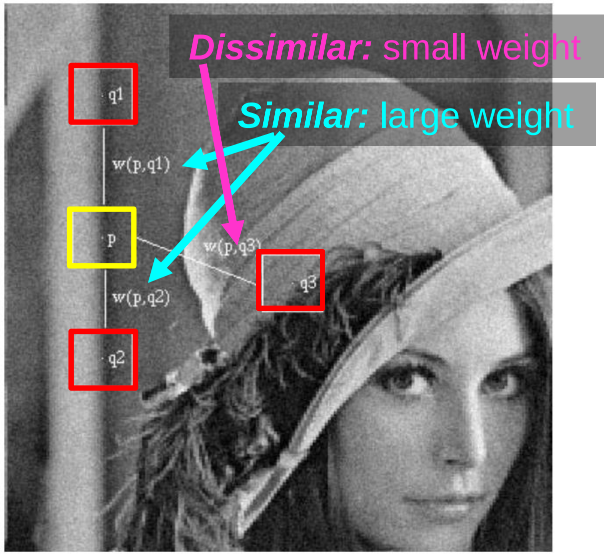
- The similarity between two pixels i and j depends on the intensity gray level vectors $v(N_i)$ and $v(N_j)$

- Weights:

Euclidean distance

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(\mathcal{N}_i)-v(\mathcal{N}_j)||_{2,a}^2}{h^2}}$$

$$Z(i) = \sum_j e^{-\frac{||v(\mathcal{N}_i)-v(\mathcal{N}_i)||_{2,a}^2}{h^2}}$$

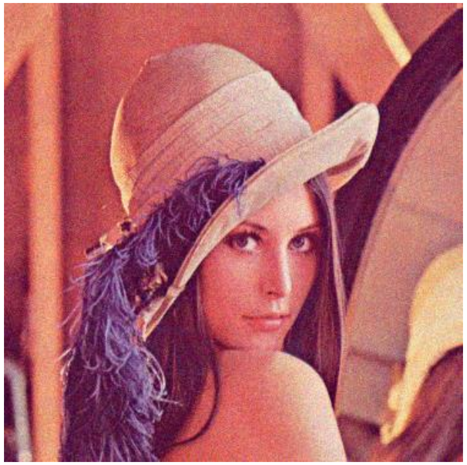


Evaluation

- Please use following two images and calculate the PSNR value with ground-truth
- Please also calculate the execution time

Ground-Truth

Basic



Challenge

Bilateral Image Filtering

Some noises on a picture

De-noising using Gaussian filtering often lacks of details

Bilateral Image Filtering

Some noises on a picture

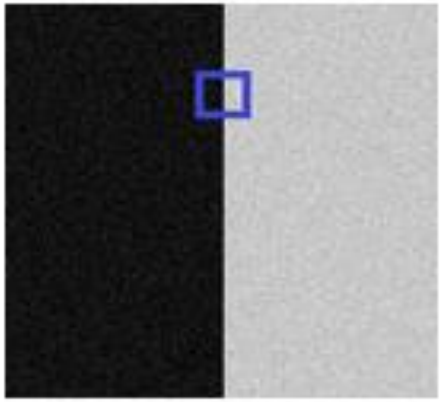
Edges and corners are smoothed, too

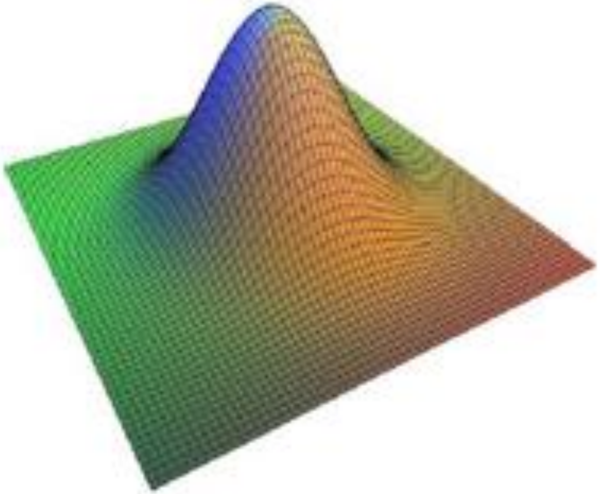
Bilateral Image Filtering

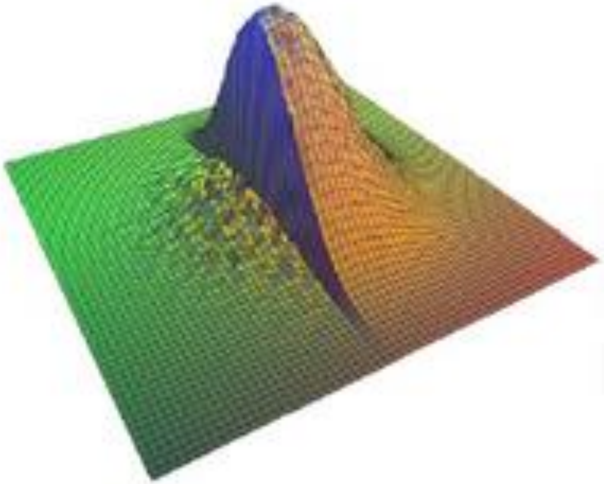
Some noises on a picture

Bilateral filtering smoothens
the surfaces but
keep the edges sharp

Bilateral Image Filtering



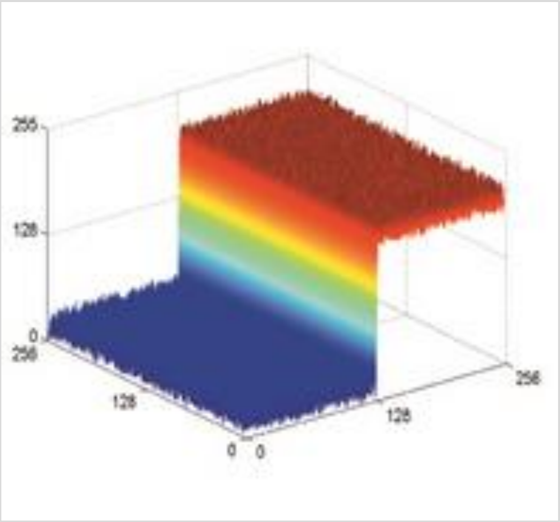


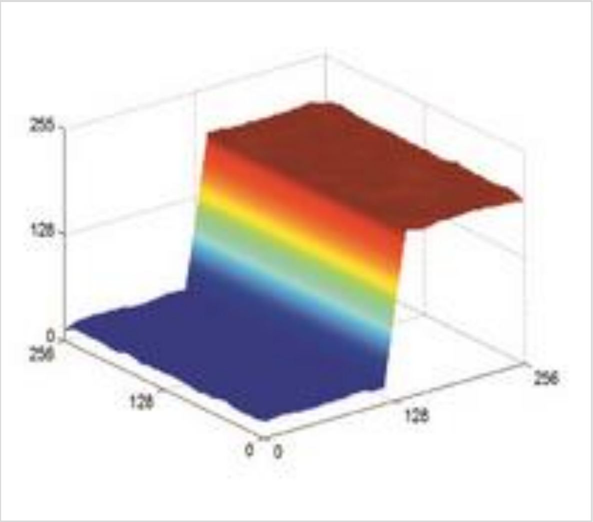


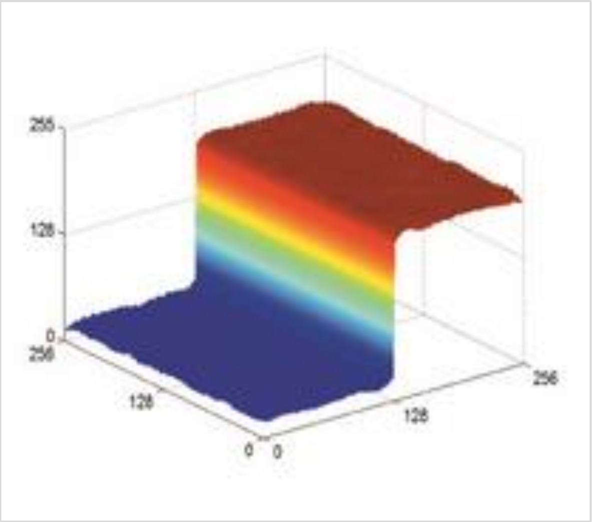
Gaussian filter samples points around the focused pixel

Bilateral filter samples points around and similar to the focused pixel

Bilateral Image Filtering







Original image

noisy, with sharp edge

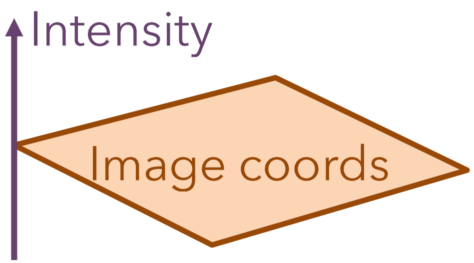
Gaussian filtered image

smooth

Bilateral filtered image

smooth, and

with sharp edge



Bilateral Image Filtering

$$g(\mathbf{x}) = \frac{1}{k(\mathbf{x})} \sum_{\xi} h(\mathbf{x},\xi)\, w(f(\xi) - f(\mathbf{x}))\, f(\xi)$$

From B. Weiss, Fast Median and Bilateral Filtering, SIGGRAPH’2006
C. Tomasi and R. Manduchi, “Bilateral filter for gray and color images”, ICCV, 1998

Poisson Blending

Seamless blending using Poisson image editing

Poisson Blending

Poisson Blending

Why there is a seam : Strong color difference

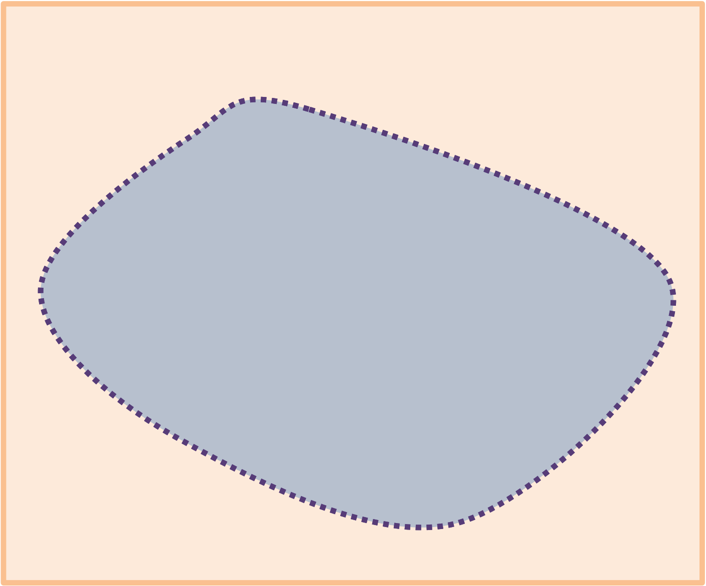


Image Plane

Even though

the border of foreground

matches

the border of background

on the image plane

Border of Background

Border of Foreground

Poisson Blending

Why there is a seam : Strong color difference

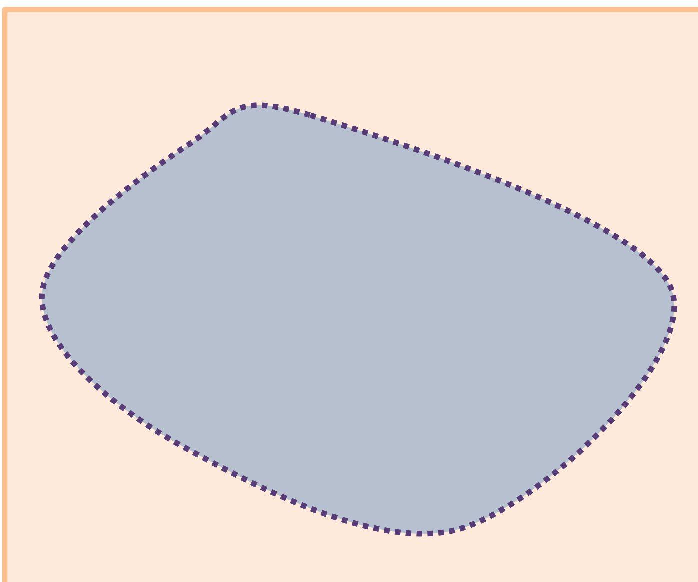
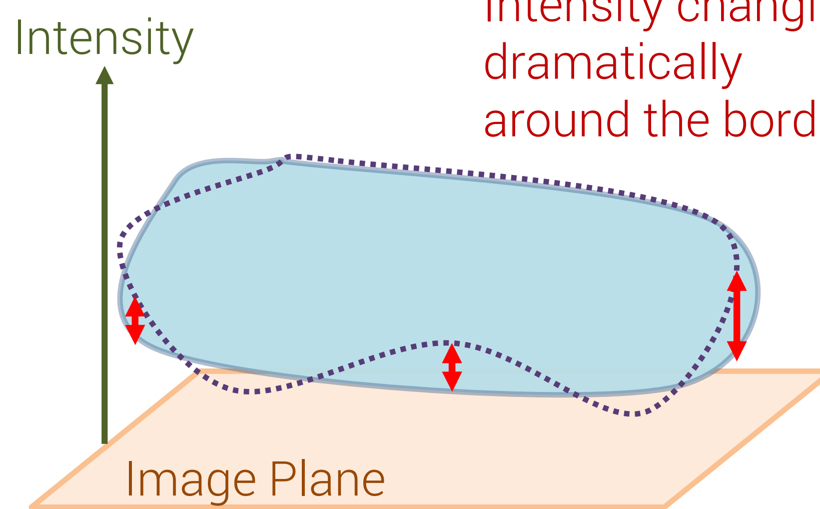


Image Plane



Intensity changing dramatically around the border

Border of Background

Border of Foreground

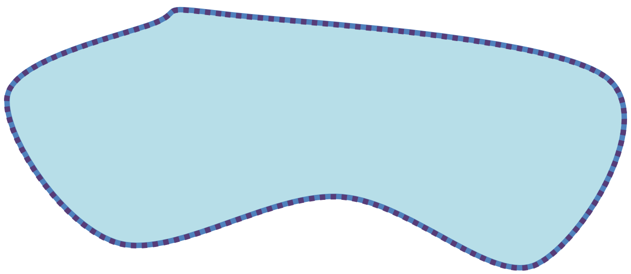
Poisson Blending

warp the border intensity

of foreground

to fit the border intensity

of background



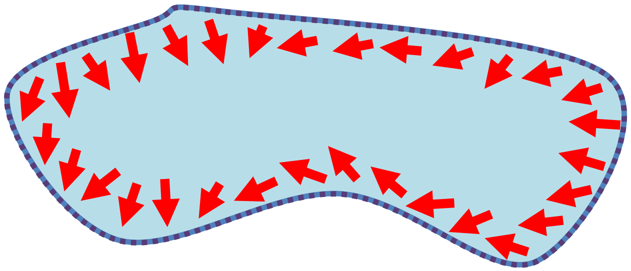
Border of Background

Border of Foreground

Poisson Blending

For the pixels of foreground but not on the border :

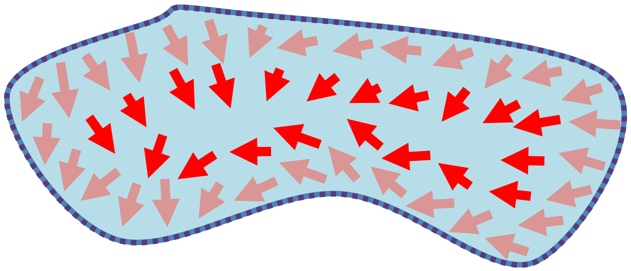
use gradient to interpolate



Poisson Blending

For the pixels of foreground but not on the border :

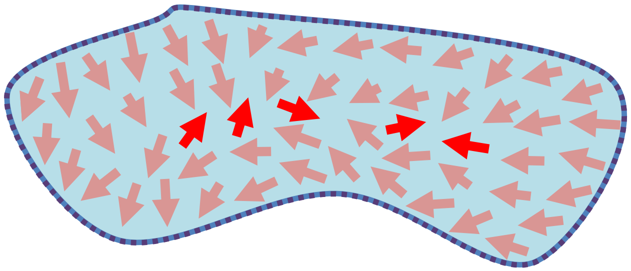
use gradient to interpolate



Poisson Blending

For the pixels of foreground but not on the border :

use gradient to interpolate



Poisson Blending

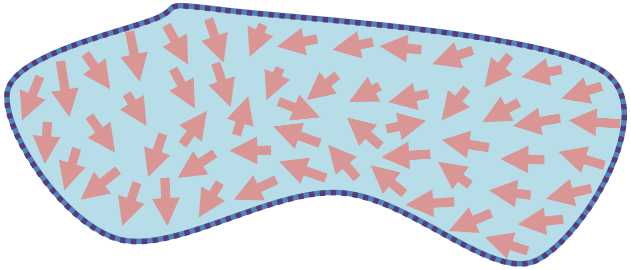


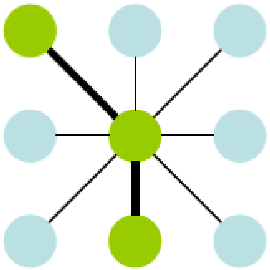
Image retargeting

- Resizing images that was aware of the actual photo's contents
- **Seam Carving** : change the size of an image by gracefully carving-out or inserting pixels by using energy function

Image retargeting

- Dynamic programming

traverse the image from the second row to the last row and compute the cumulative minimum energy M for all possible connected seams



$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1))$$

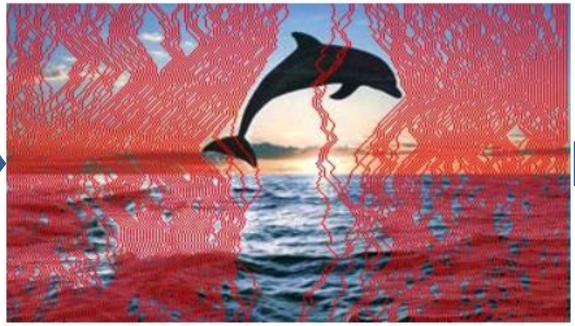
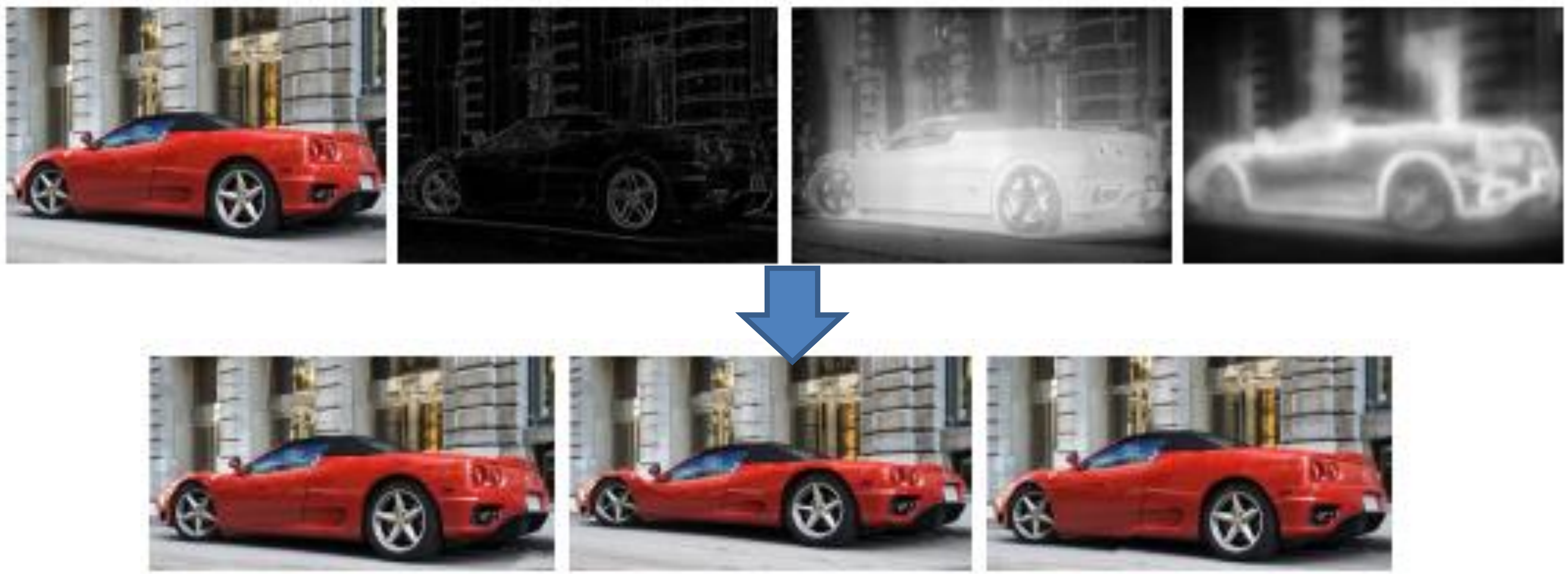


Image retargeting

- Some reference about content-aware
    1. **Context-Aware Saliency Detection**
    2. **Improved Seam Carving Using a Modified Energy Function Based on Wavelet Decomposition**



The details of the car with [1]'s saliency maps is detected more accurately, hence they are not distorted by retargeting.

DTMF decoding

- You may be familiar with “DTMF” !

Youtube

DTMF decoding

- **Dual-tone multi-frequency signaling** (**DTMF**) is used for telecommunication signaling
- This project is to decode a DTMF recording by frequency analysis

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

DTMF keypad frequencies (with sound clips)

DTMF keypad layout.

Method introduction of DTMF decoding

- DTMF stands for **D**ual **T**one - **M**ulti **F**requency and it is the basis for your telephone system.
- When you press the buttons on the keypad, a connection is made that generates two tones at the same time.

  e.g. When you press the digit 1 on the keypad, you generate the tones 1209 Hz and 697 Hz.

C#4D34**

461####333

C_93CDDCD_32_7_33_10

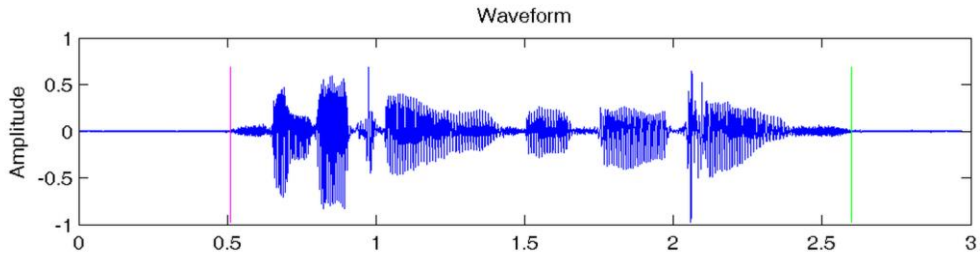
dtmf-1646347904

Method introduction of DTMF decoding

- Step 1. Read the description about **DTMF**

  http://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling

- Step 2. Use any of the end-point detection (**EPD**) methods to segment the recordings



- Step 3. Analyze frequency to decode

Method introduction of DTMF decoding

- **Reference**

✓ **DTMF Explained**

http://www.genave.com/dtmf.htm

✓ **DTMF Tone Generator**

http://www.audiocheck.net/audiocheck_dtmf.php

✓ **Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments**

http://mirlab.org/jang/books/audioSignalProcessing/paper/endPointDetection/shenHL98-endpoint.pdf