Final Project

CS 357000, 2017

General Guideline

2~3 students form a team

Select a topic from our suggestion list,
or think a new one yourself

Submit 1-page proposal on iLMS before **5/16**, including team name, members, project name, brief description and goal. And also hand in a copy of your proposal to TAs in class

If your proposal is not approved, you will know before **5/19**

Please hand in another one before **5/23**

General Guideline

Team with more members are recommended to take a challenging topic

Hand in your report before presentation

Every team member needs to participate the presentation

Project Requirement

- Realize the algorithms
- Implementation

  Partially using open source code and toolbox are permitted
- Comparison and analysis
- Improvement (Quality, accuracy, speed, or others)
- System for demonstration

Presentation in class

6/9(Fri), 6/13(Tue)

13:20 - 15:30 (approximately 2 hours each day)
5 mins for each group

We are all judges

Best 3 teams get bonus points

Calendar

Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
		5/16			5/19	
		5/23				
					6/9	
		6/13				

Project Categories

1	Image Processing
2	Audio Processing
3	Graphics
4	Video Processing

Image Processing

Image inpainting

Image stitching

Seam carving

Special effects camera app

Image matting

Rectangle rectification for document capture

Image blending

Image inpainting

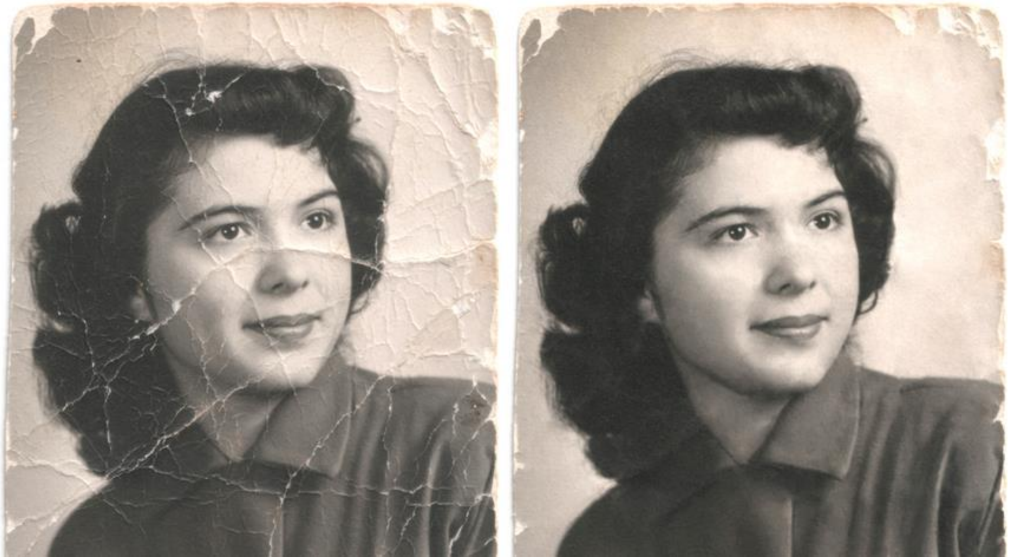
- ***Inpainting*** is the process of reconstructing lost or deteriorated parts of images
- Reference:

[Object Removal by Exemplar-Based Inpainting]

[Exemplar-Based Image Inpainting Technique using Image Partitioning]

Image inpainting

- Photo Restoration



- Object Removal

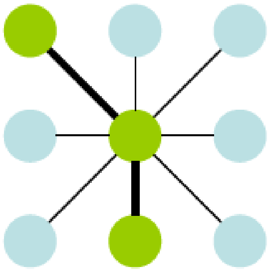
Image retargeting

- 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



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

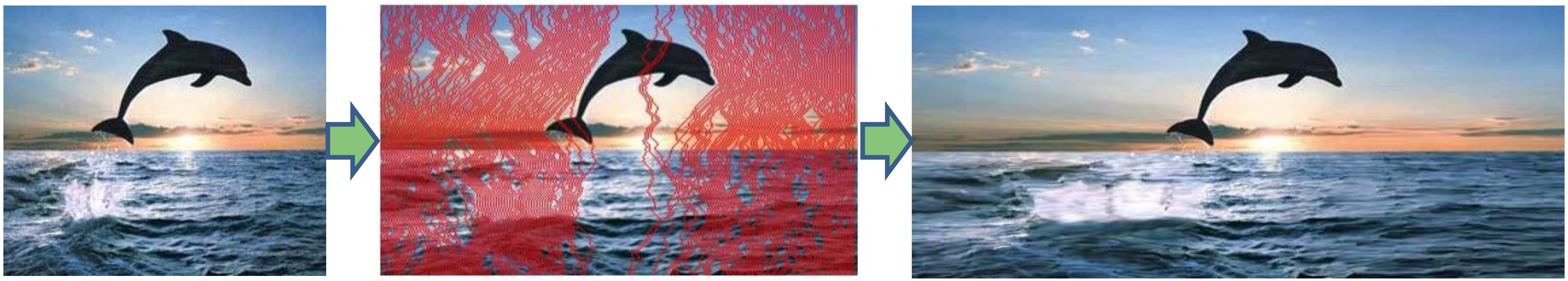
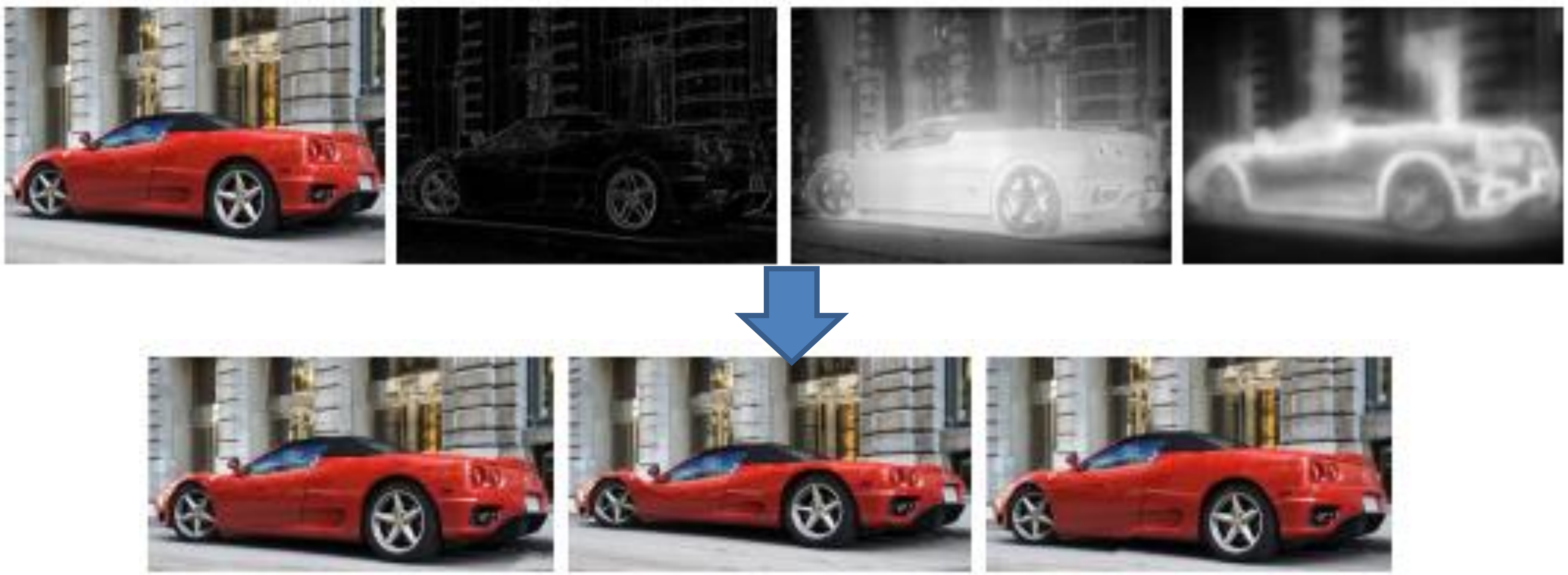


Image retargeting

- 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.

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)* $I = \alpha F + (1-\alpha)B$

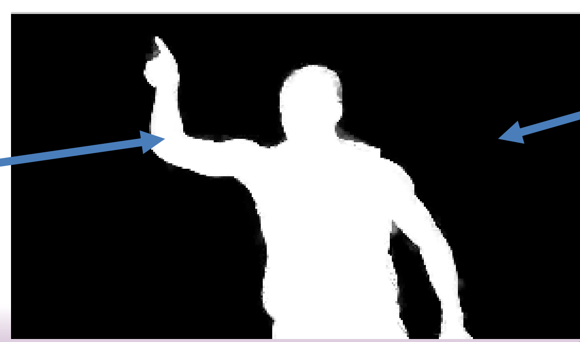
$= \alpha$

$+ \; (1 - \alpha)$

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

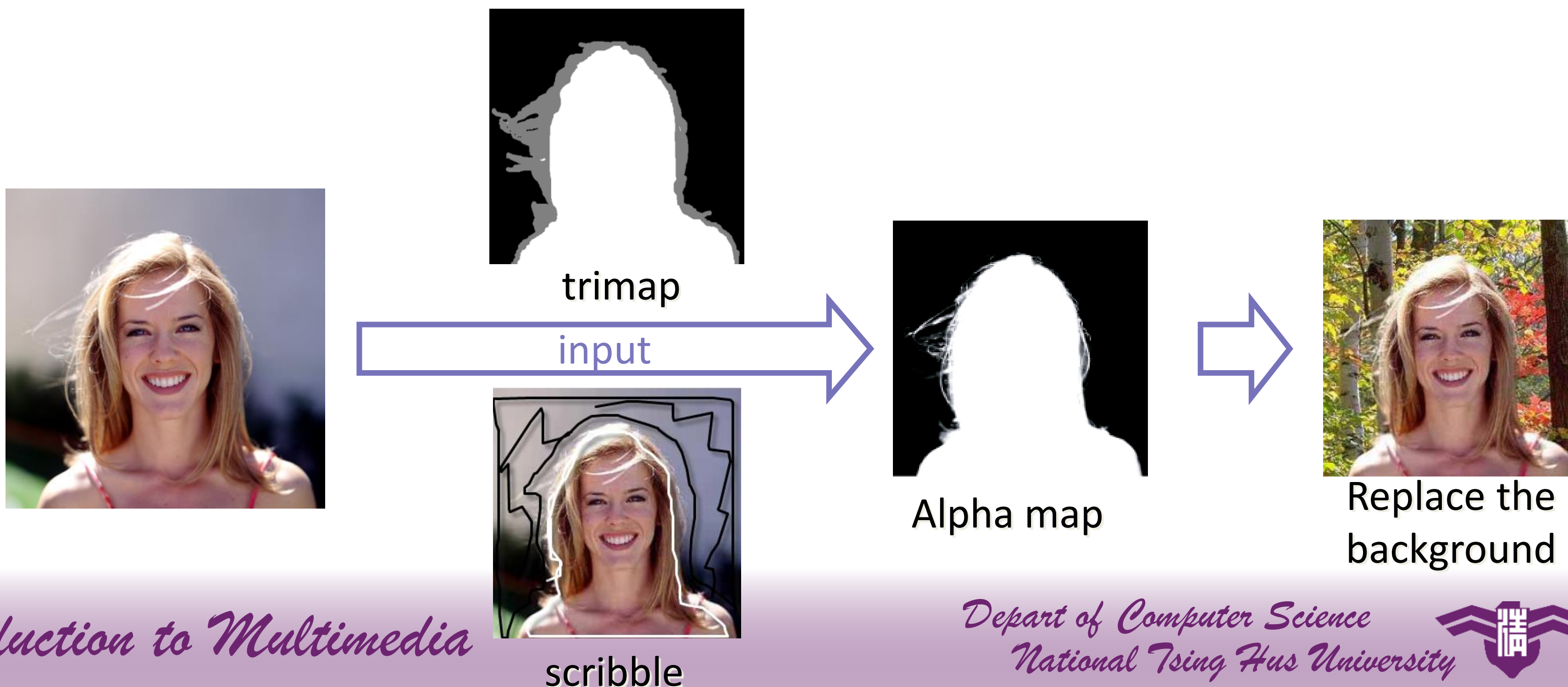


Image Stitching

- The process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image.

Calibration → Registration → Blending → Projective layouts

You may need:
- Homography with RANSAC
- Image warping
- Image blending

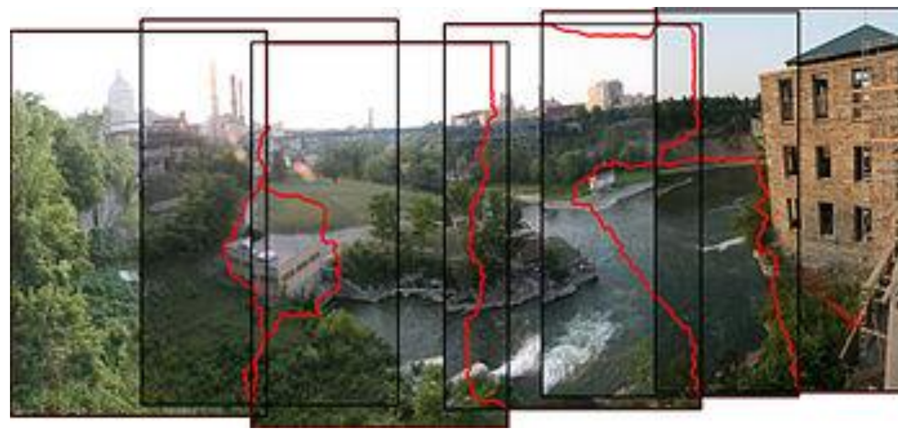


Image Stitching

- **Projective layouts**: "Rectilinear" give distorted image suitable for creating a true panorama (might view with Quicktime VR). "Equirectangular" will give you something similar to a very large photograph. "Cylindrical" will be somewhere in between.

Project
the result

Equirectangular

Cylindrical

Rectilinear

Spherical stitching

- Stitches multiple images from different viewpoints and build a 360° scene.

"Little Planet" effect
With Stereographic projection

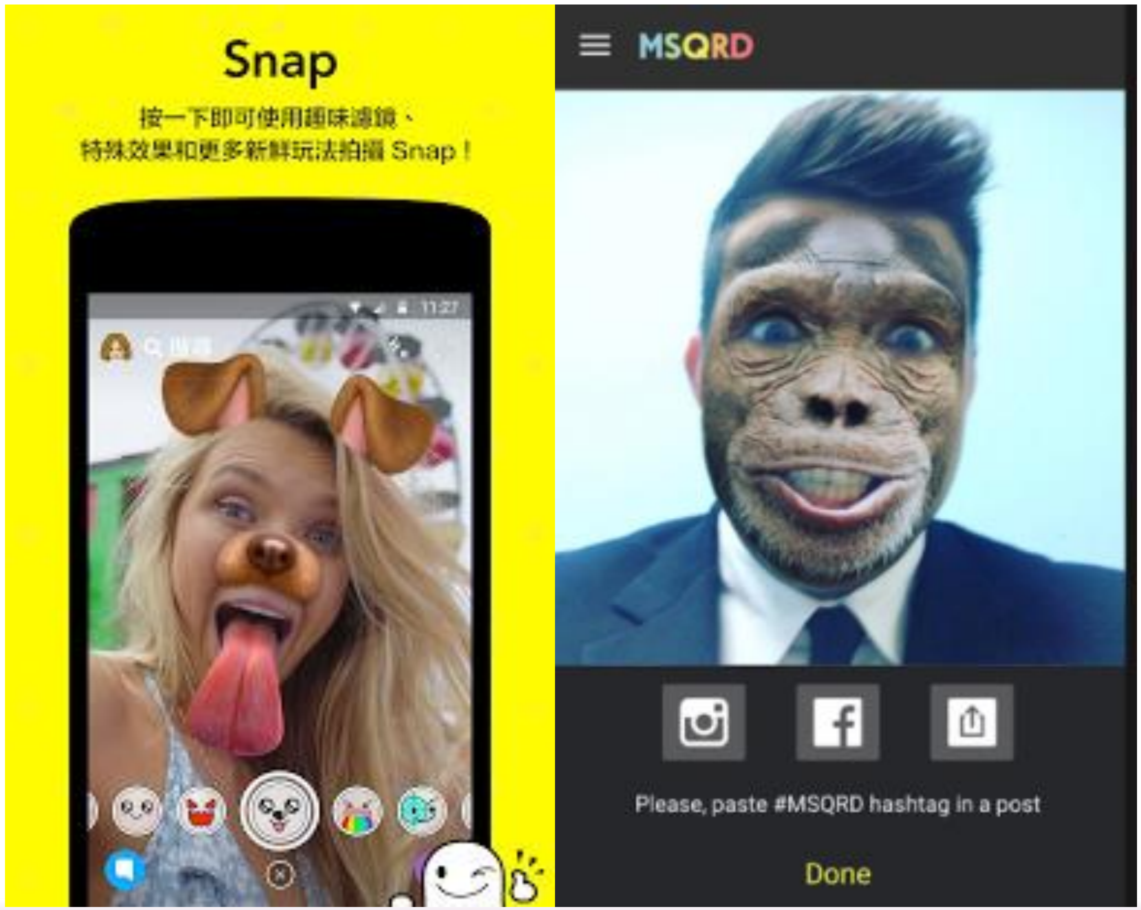
360 scene with Equirectangular projection

Image Blending

Seamless blending using Poisson image editing

Special effects camera app

Seek existing special-effect camera for inspiration, e.g. face change with [MSQRD](), [Snapchat]()

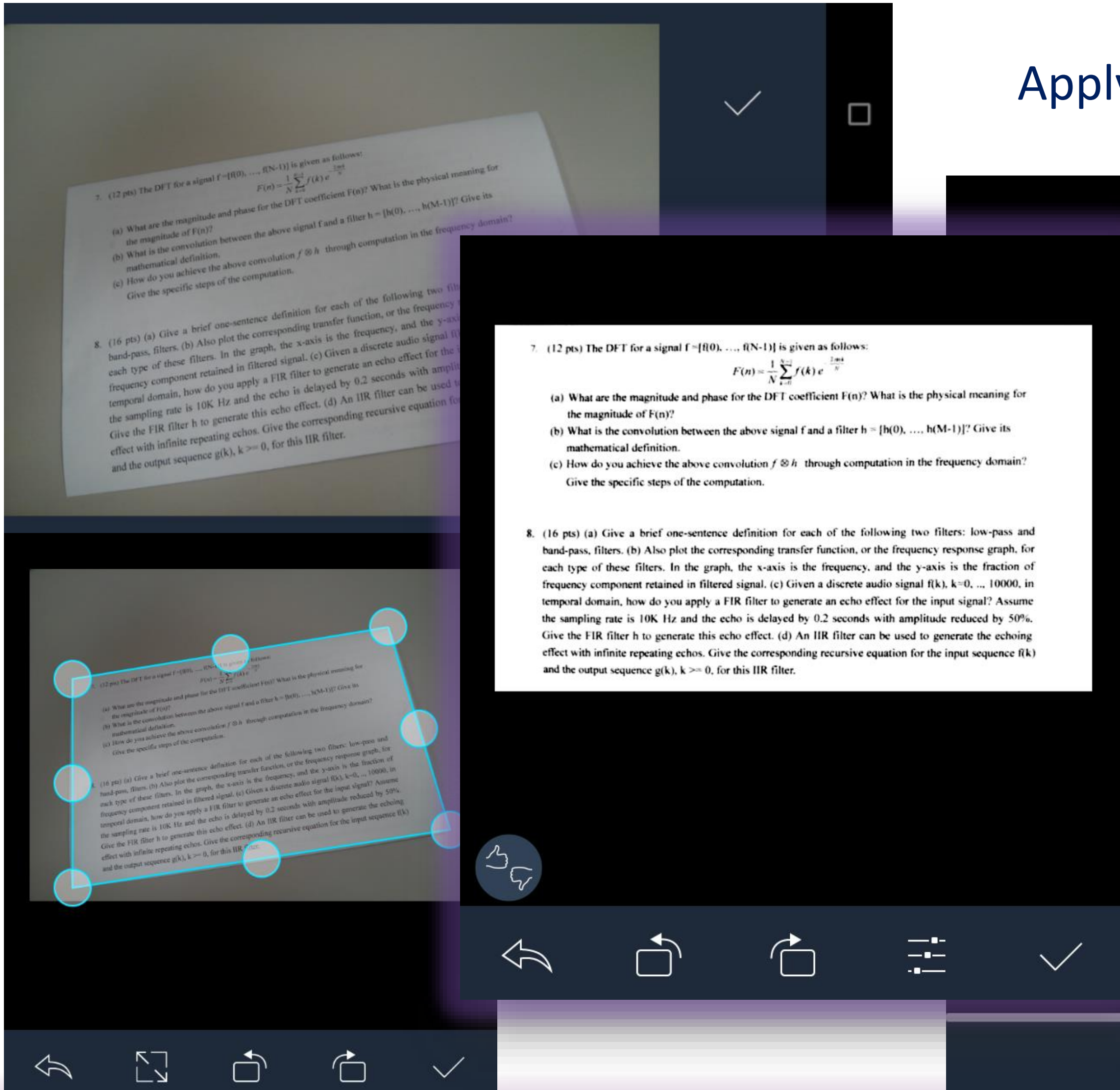


Special effects camera app

Seek existing special-effect camera for inspiration, e.g. photo style transfer, special photo effect, HDR with tone mapping

Rectangle rectification for document capture

Apply Homography for rectification



7. (12 pts) The DFT for a signal f = [f(0), …, f(N-1)] is given as follows:

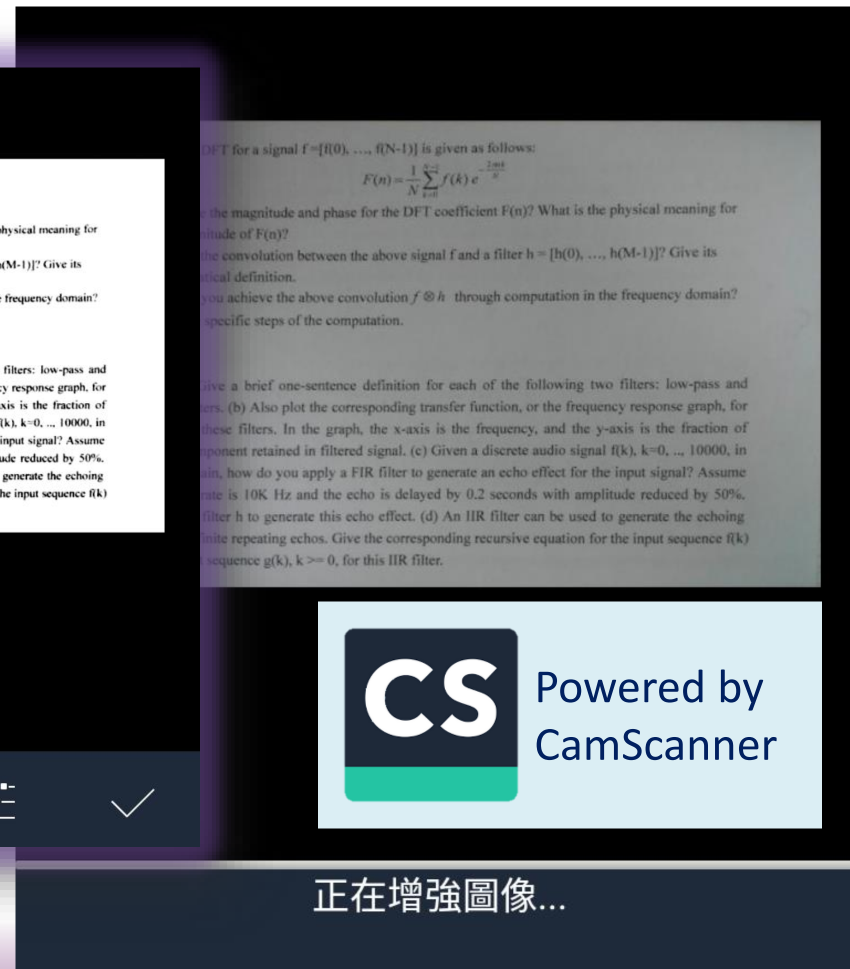
$$F(n) = \frac{1}{N}\sum_{k=0}^{N-1} f(k)\, e^{-\frac{2\pi i n k}{N}}$$

(a) What are the magnitude and phase for the DFT coefficient F(n)? What is the physical meaning for the magnitude of F(n)?

(b) What is the convolution between the above signal f and a filter h = [h(0), …, h(M-1)]? Give its mathematical definition.

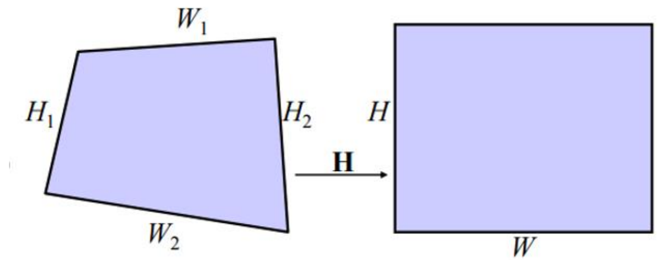
(c) How do you achieve the above convolution $f \otimes h$ through computation in the frequency domain? Give the specific steps of the computation.

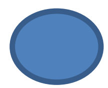
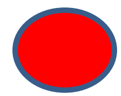
8. (16 pts) (a) Give a brief one-sentence definition for each of the following two filters: low-pass and band-pass, filters. (b) Also plot the corresponding transfer function, or the frequency response graph, for each type of these filters. In the graph, the x-axis is the frequency, and the y-axis is the fraction of frequency component retained in filtered signal. (c) Given a discrete audio signal f(k), k=0, …, 10000, in temporal domain, how do you apply a FIR filter to generate an echo effect for the input signal? Assume the sampling rate is 10K Hz and the echo is delayed by 0.2 seconds with amplitude reduced by 50%. Give the FIR filter h to generate this echo effect. (d) An IIR filter can be used to generate the echoing effect with infinite repeating echos. Give the corresponding recursive equation for the input sequence f(k) and the output sequence g(k), k >= 0, for this IIR filter.



Homography example

input



 = H * 

Left points   Right points

H =

$$\begin{matrix} 0.0084 & 0.0009 & 0.6518 \\ 0.0047 & 0.0052 & -0.7583 \\ 0.0000 & 0.0000 & 0.0042 \end{matrix}$$

Backward warping
(bilinear interpolation)

output

Audio Processing

Music genre classification

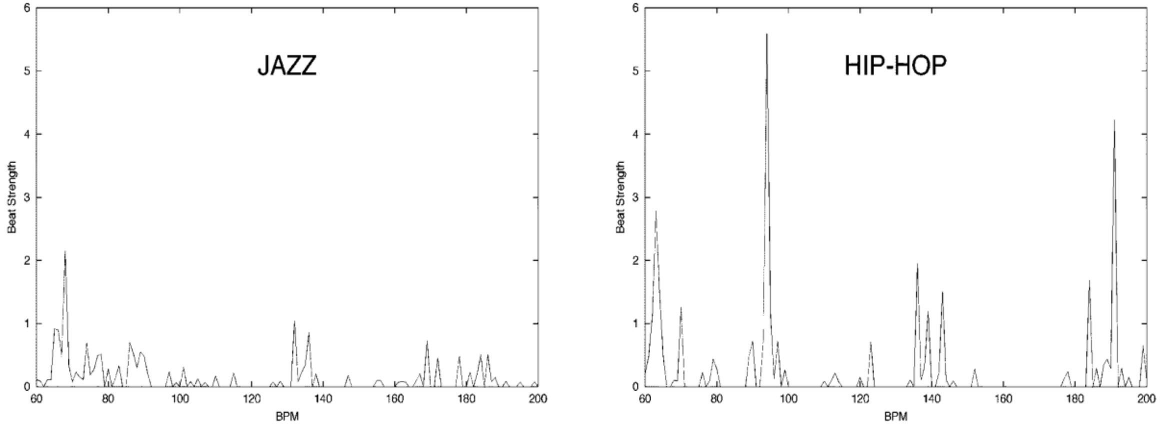
Chord Recognition

Source Separation

Beat tracking

Music 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 on dataset (e.g. **GTZAN**).
- This work involve Pattern Recognition. You can use part of the music 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)

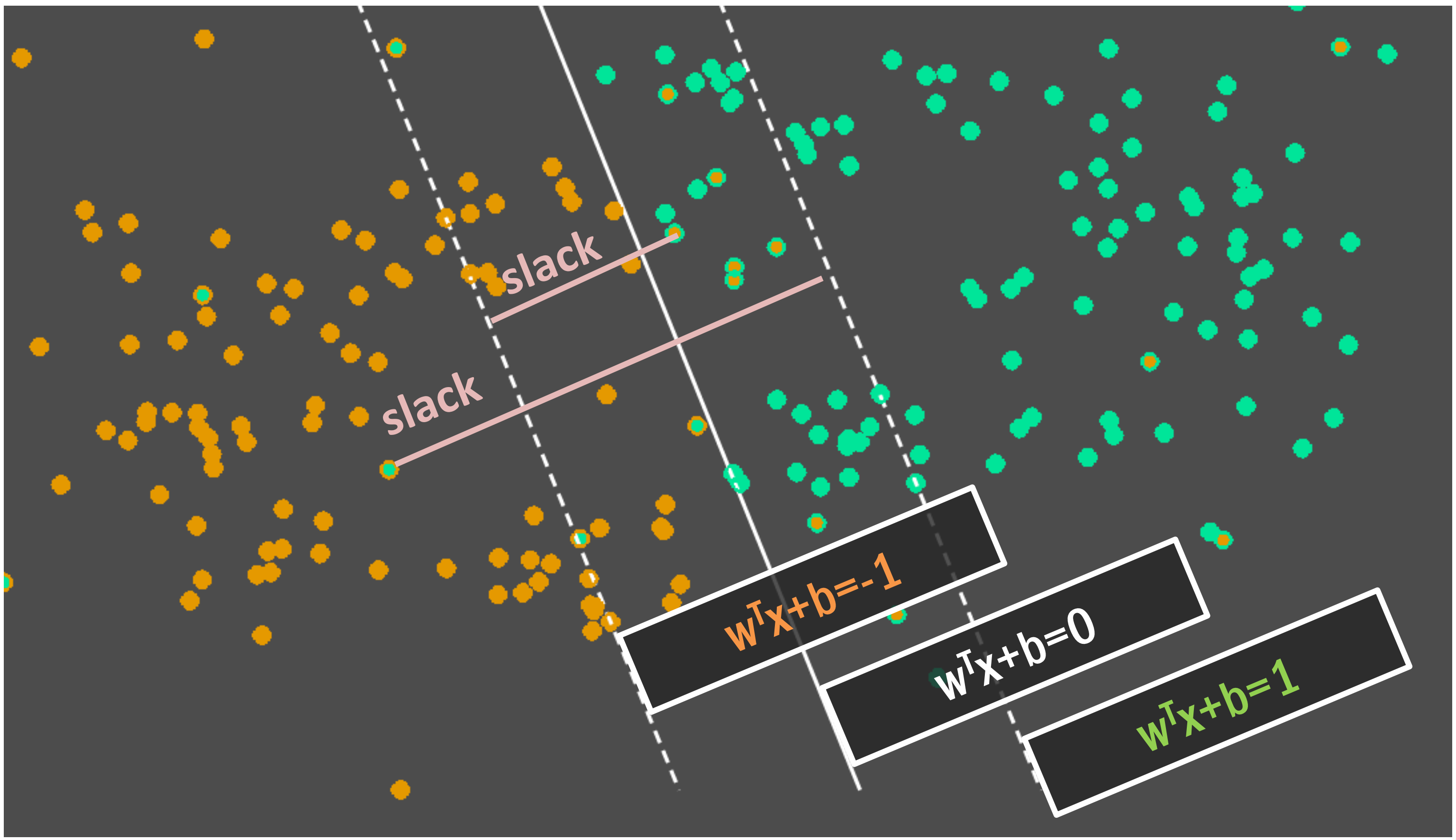
- Ref: George Tzanetakis, Student Member, Perry Cook "Musical Genre Classification of Audio Signals" IEEE 2002

Procedure of Genre classification

1. Pre-processing
   - Filtering
2. Feature extraction
   - ✓ Audio Feature
   - ✓ **Basic feature:** Volume, Pitch, Timbre, Zero-crossing rate…
   - ✓ **Advanced features:** MFCC, Centroid, Rolloff, Flux …
   - ✓ **Others:**
     - Automatic Musical Genre Classification Of Audio Signals
     - Higher-order moments for musical genre classification
   - ✓ **Reference**: MIRtoolbox
3. Classification
   - The classifier divides the feature space into class regions
   - e.g. SVM

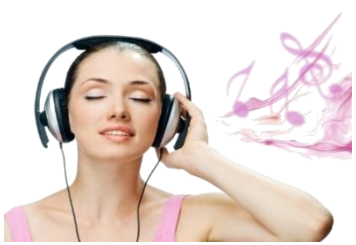
Method of Genre classification

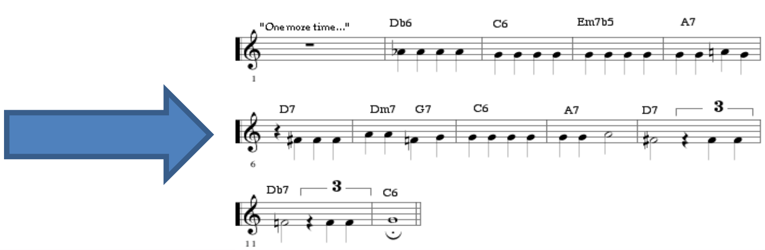
- SVM: Find a line (hyperplane) that can separate the two 2D (High dimensional) data [LIBSVM]



Chord Detection/Recognition

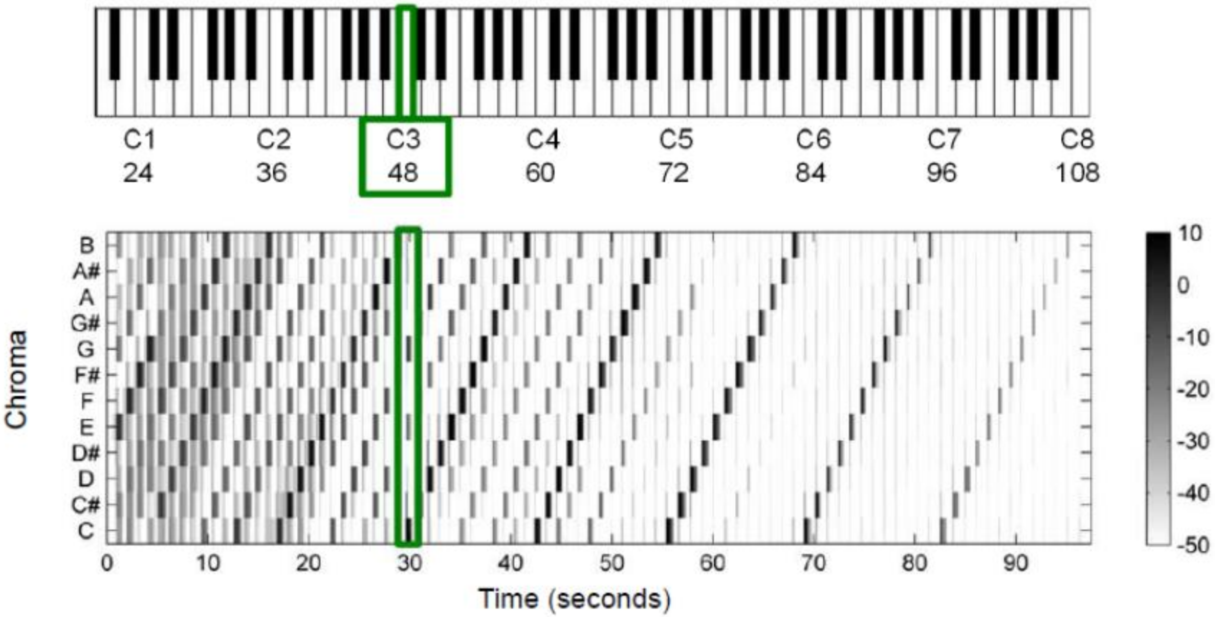
- In music, **Chord Recognition** can mean notating chord to a piece or a sound which was previously unannotated.
- Detect/Recognize what the chord is in a song.
- Musicians who do not have intuitive transcription skills will search for sheet music or a chord chart, so that they may quickly learn how to play a song.
- Related Datasets:
  - Robbie Williams Chord and Key Annotation Dataset as ground truth, there are 5 albums and 65 songs in this dataset.
  - GTZAN dataset and Alexander Lerch's annotation of key on the GTZAN dataset from





Feature in Chord Recognition

- Chromagram
  - Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or Chroma) of the musical octave



Simple Procedure of Chord Recognition

- Extraction feature (Chromagram)
  - Sum the Chroma feature in time domain
- Chord Template 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]
- Calculate correlation
  - x: a clip of a song
  - y: template

$$R(x,y) = \frac{\sum_{k=1}^{12}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{12}(x_k - \bar{x})^2 \sum_{k=1}^{12}(y_k - \bar{y})^2}}$$

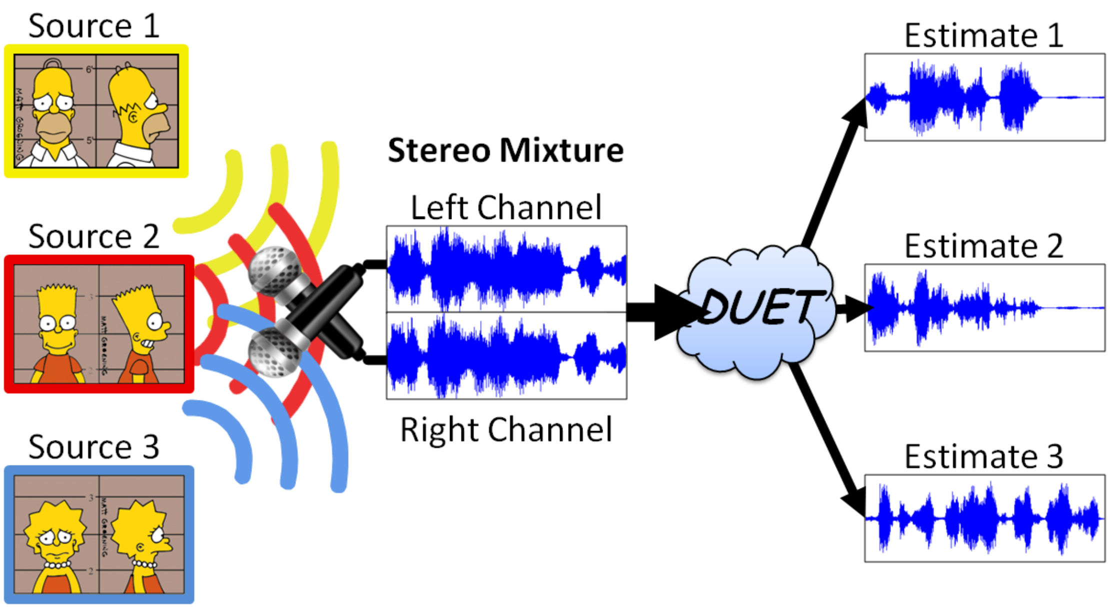
Common Procedure of Chord Recognition

- Extraction feature (Chromagram)
  - Chroma feature in time domain
- Pre-filtering
  - Median filter on the Chroma feature
- Pattern matching (Learning based)
  - Hidden Markov Model / Gaussian Markov Model
  - Predict the chord sequence

Reference:
Chroma Toolbox
Supervised Chord Recognition for Music Audio in Matlab

Source Separation

- **Source separation** problems in digital signal processing are those in which several signals have been mixed together into a combined signal and the objective is to recover the original component signals from the combined signal.



[Dataset related to source separation]

[Spatial sound source separation]

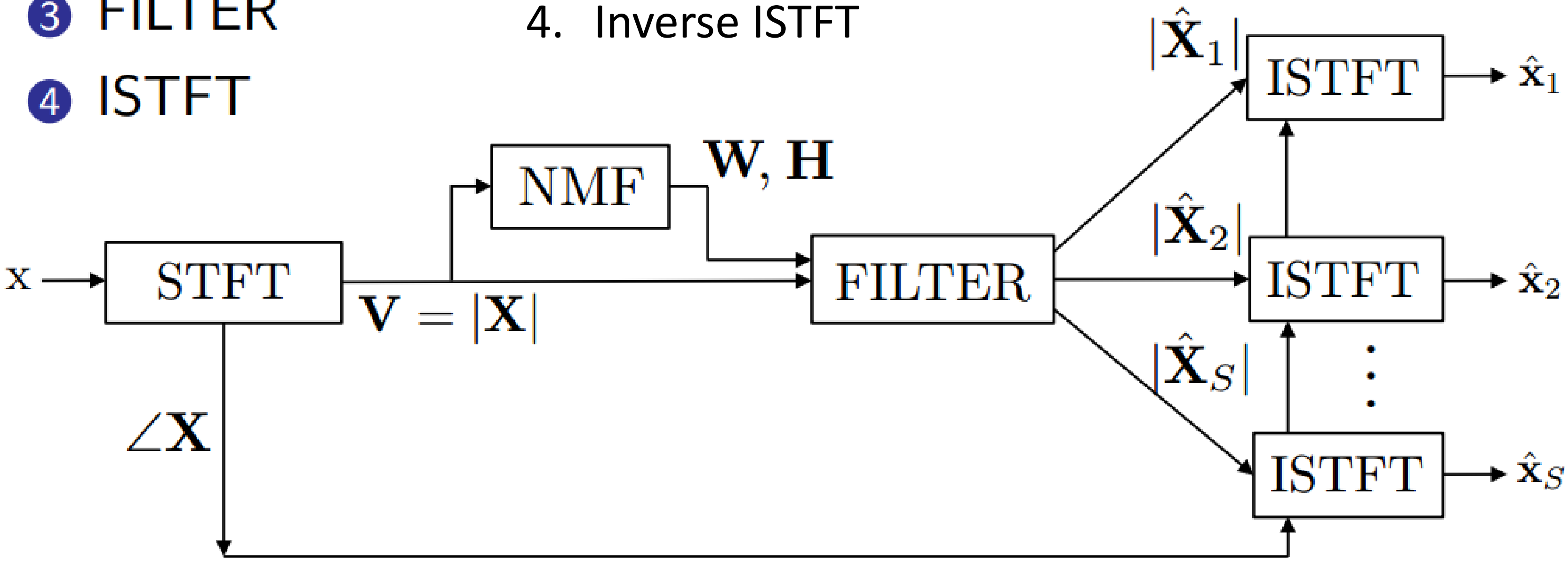
[Flexible Audio Source Separation Toolbox]

Source Separation

- General Source Separation Pipeline

❶ STFT
❷ NMF
❸ FILTER
❹ ISTFT

1. Short Time Fourier Transform
2. Nonnegative matrix factorization
3. Reconstruct after filtering
4. Inverse ISTFT

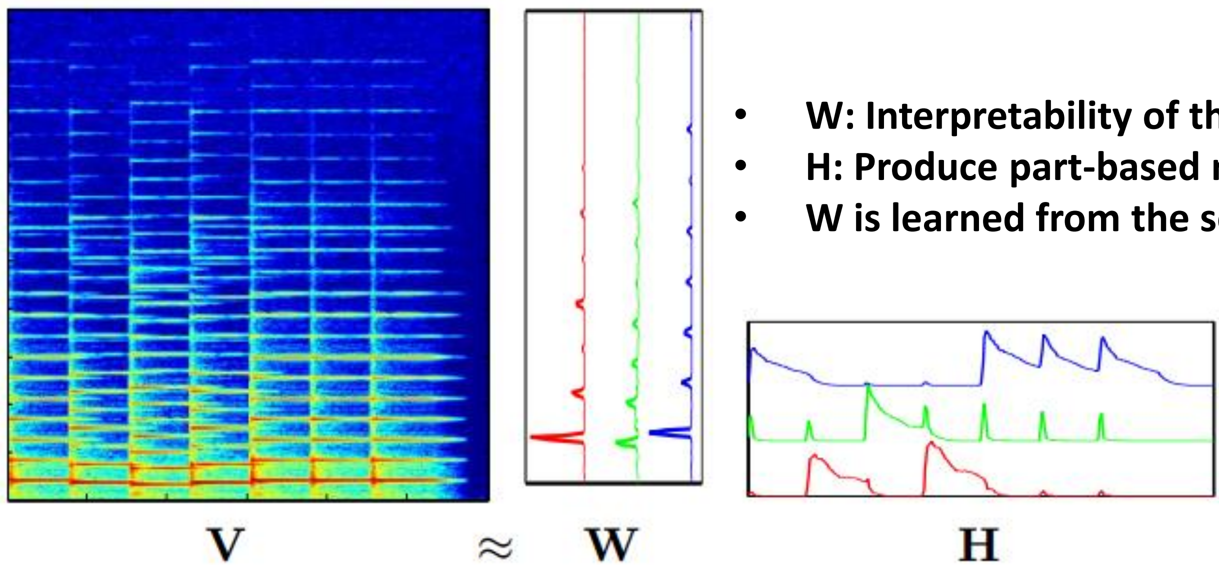


Reference:

[Source Separation Tutorial Mini-Series II: Introduction to Non-Negative Matrix Factorization]

Source Separation

- Nonnegative matrix factorization (NMF)
- We can interpret the column v as a linear combination of different instruments (or voice/noise) and corresponding activation column h



- **W: Interpretability of the dictionary**
- **H: Produce part-based representation**
- **W is learned from the set of data vectors**

Data vector Dictionaries Activation coefficients

Source Separation

- Calculate W and H given mixed audio V
  - Cost function: Euclidean distance

$$\|V - WH\|^2 = \sum_{ij} \left(V_{ij} - WH_{ij}\right)^2$$

  - Optimize H and W : Gradient descent
  - Brief Algorithm
    1. Update H when fixed W
    2. Update W when fixed H
    3. Till the change of cost function since last iteration is small, then it is declared convergence

Algorithm

initialize $\mathbf{W}, \mathbf{H}$

repeat

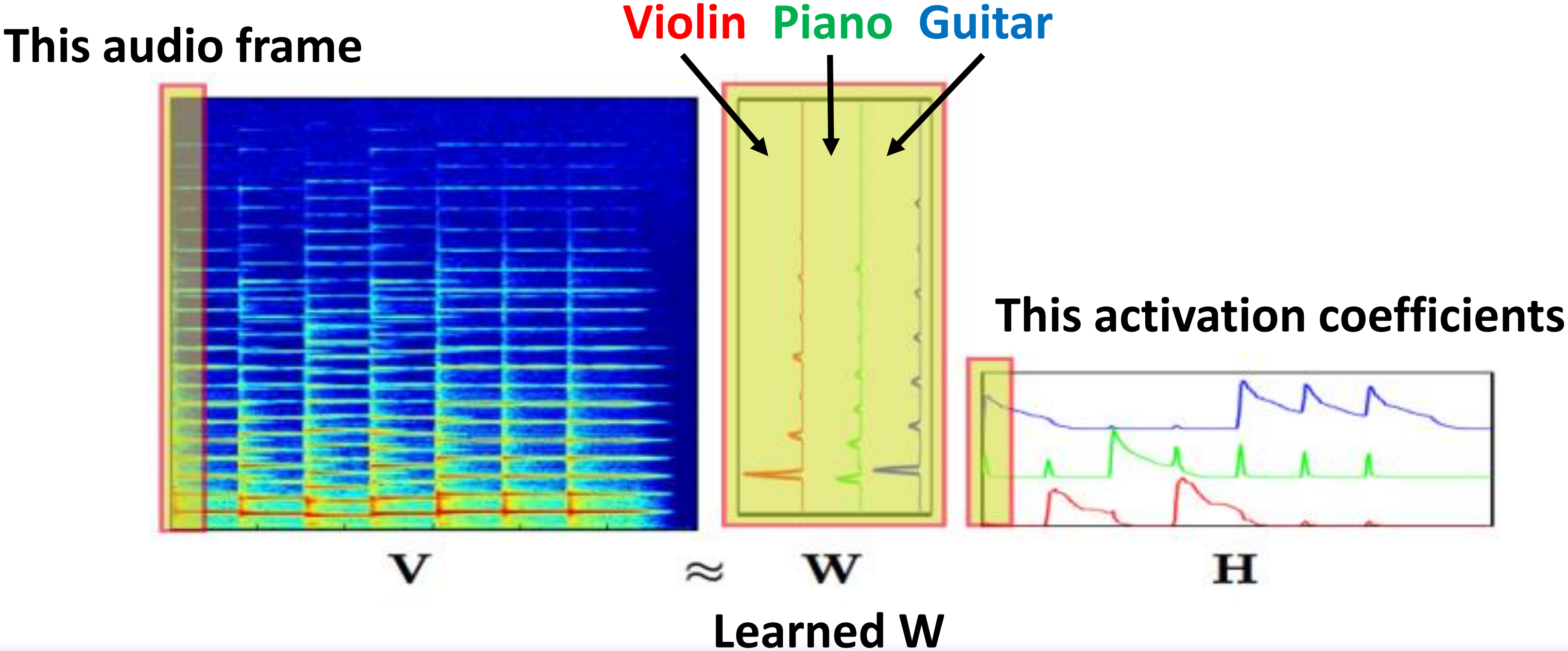
$$\mathbf{H} \leftarrow \mathbf{H} .* \frac{\mathbf{W}^T \frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}}{\mathbf{W}^T \mathbf{1}}$$

$$\mathbf{W} \leftarrow \mathbf{W} .* \frac{\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}} \mathbf{H}^T}{\mathbf{1}\,\mathbf{H}^T}$$

until convergence **return** $\mathbf{W}, \mathbf{H}$

Source separation

- Supervised Nonnegative matrix factorization (NMF)
  - W is learned from clean data
  - We can interpret the column **v** as a linear combination of learned different instruments (or voice/noise) dictionaries and corresponding activation column **h**



Beat tracking

- The aim is to identify each beat location in a sound file.
  - Beat: the unit of time of music
- Available dataset e.g. Ballroom, or you can search more

- Reference
  - Tempo and Beat Tracking (video intro.)
  - beattrack
  - Tempogram toolbox
  - A. Srinivasamurthy, A. Holzapfel, A. Cemgil, and X. Serra. Particle filters for efficient meter tracking with Dynamic Bayesian networks. In ISMIR 2015.
  - S. Durand, J. P Bello, B. David, and G. Richard, “Downbeat tracking with multiple features and deep neural networks,” in ICASSP 2015, pp. 409–413.

Graphics

OpenGL/Direct3D based game / application

Vectorization

Model registration

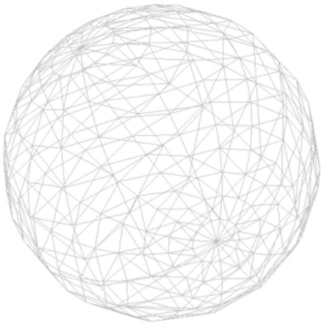
OpenGL/Direct3D based game / application

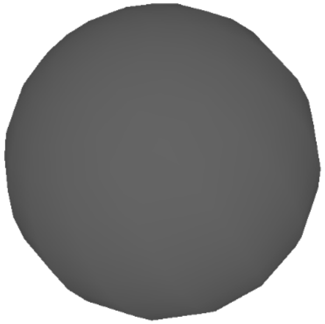
- 3D Graphics Library

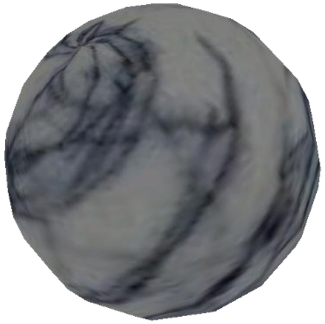
Rendering some 3-D models in the world space

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 ($f \sim 0$)

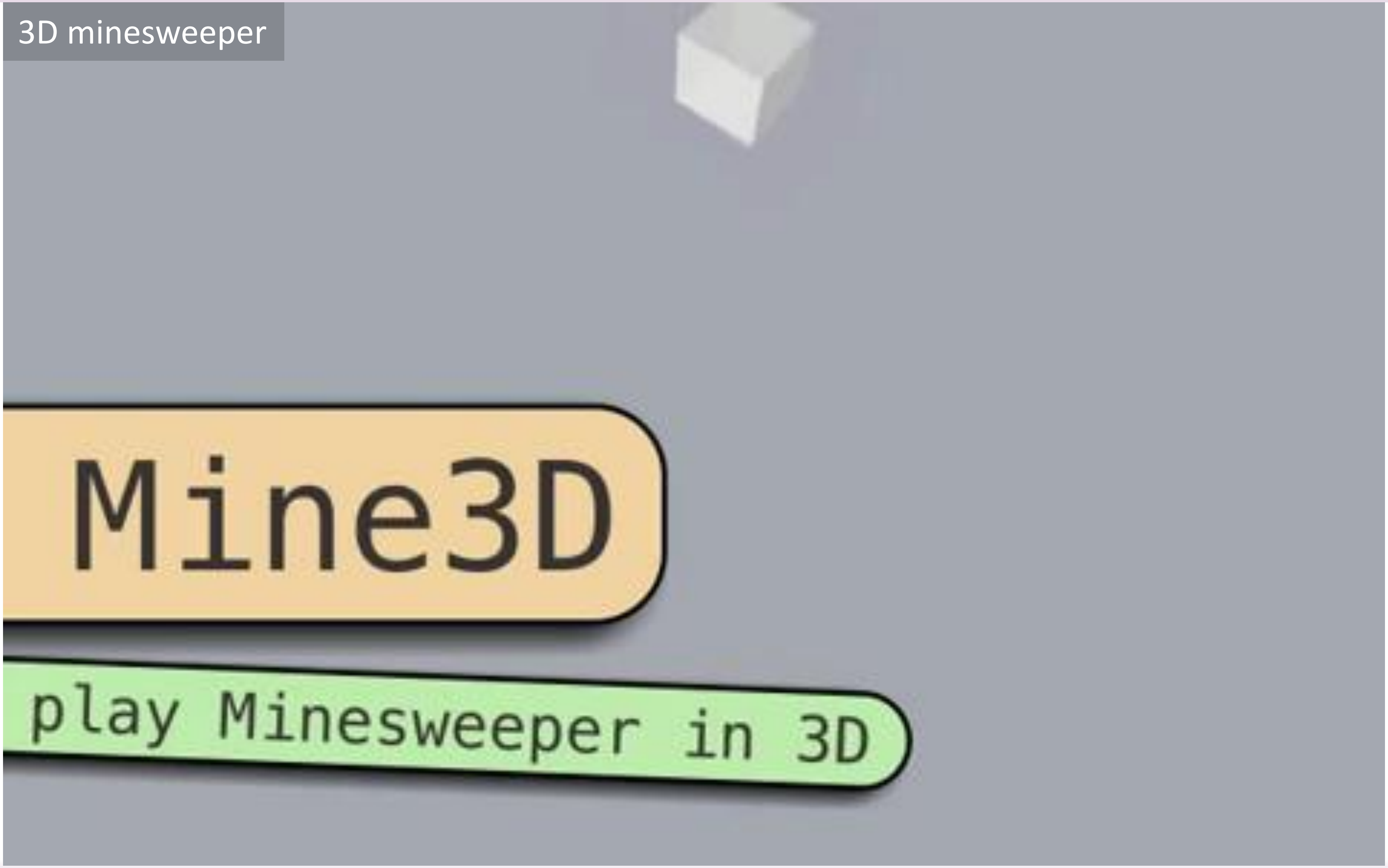
OpenGL/Direct3D based game / application

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

OpenGL/Direct3D based game / application

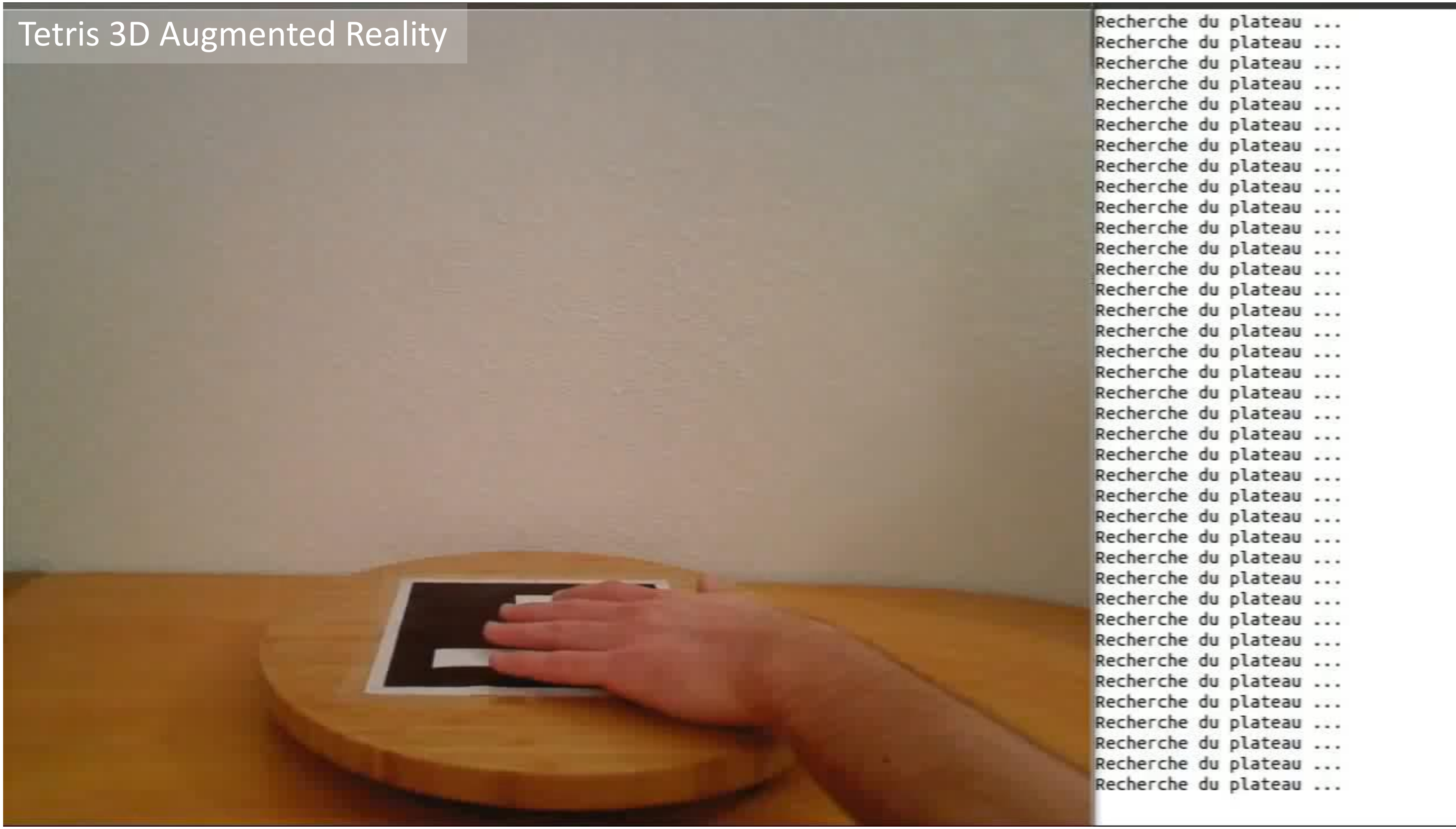
- Lighting
  - Ambient
  - Diffuse
  - Specular
  - Other light effect using shader

OpenGL/Direct3D based game / application



[Youtube]

OpenGL/Direct3D based game / application



[Youtube]

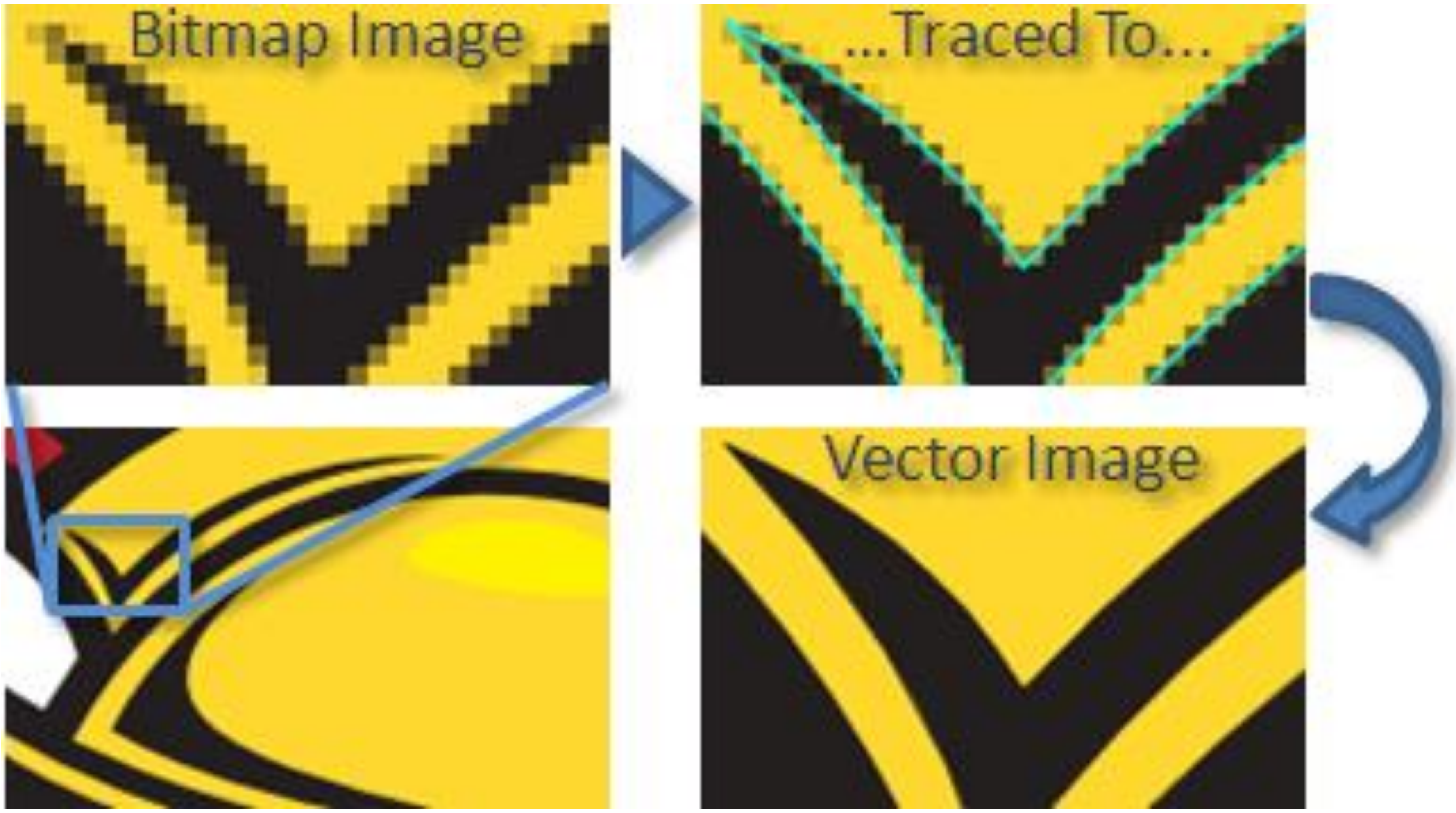
OpenGL/Direct3D based game / application

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

- **Augmented Reality SDK**
- ARToolKit (developed by OpenGL and OpenCV)
  - Support Windows/Mac OS X/Linux/Android/IOS

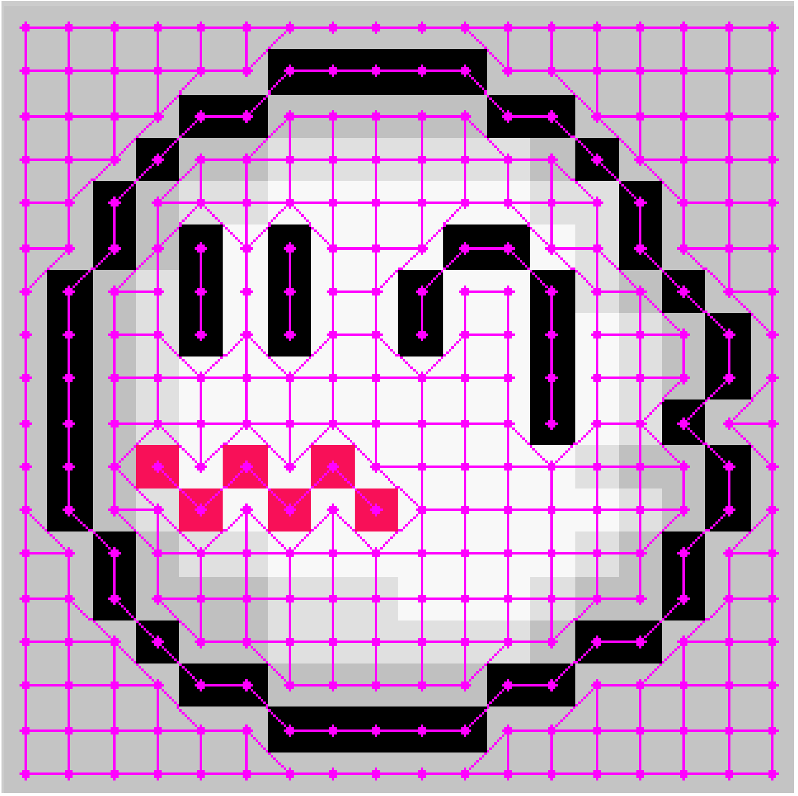
ARTOOLKIT

Vectorization



Example from http://vectormagic.com/

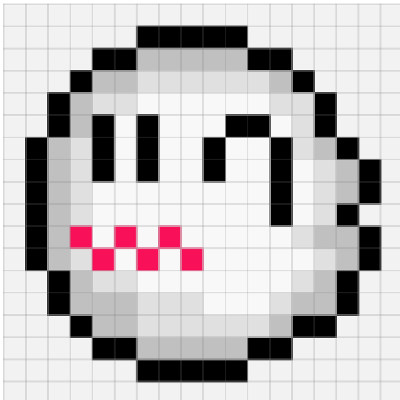
Vectorization



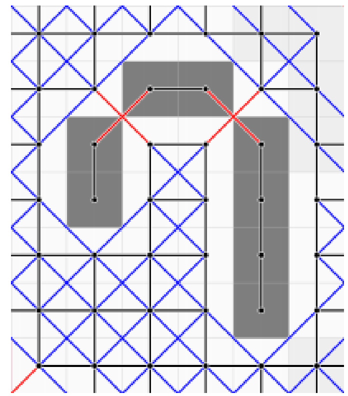
- Input a bitmap
- Extract geometry structure
- Apply curve fitting
- Output a vector image

From 蘇凱煜 and 林雯婷's masterpiece

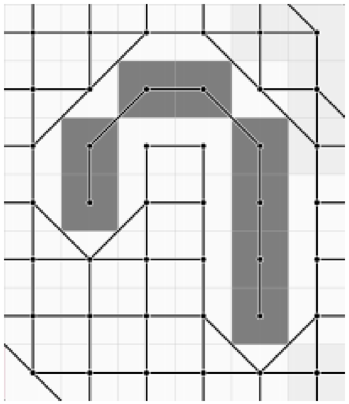
Vectorization



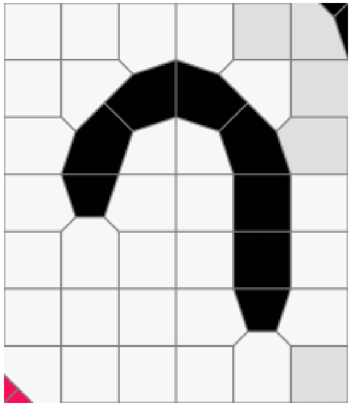
(a) Input image(16*16 pixels)



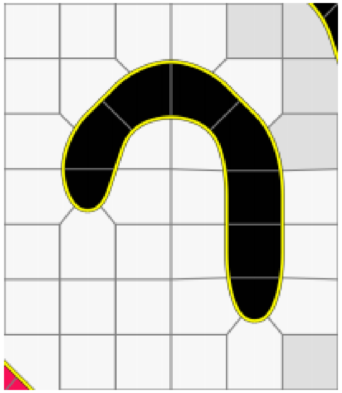
(b) Initial similarity graph with crossing edges.



(c) Crossing edges resolved.



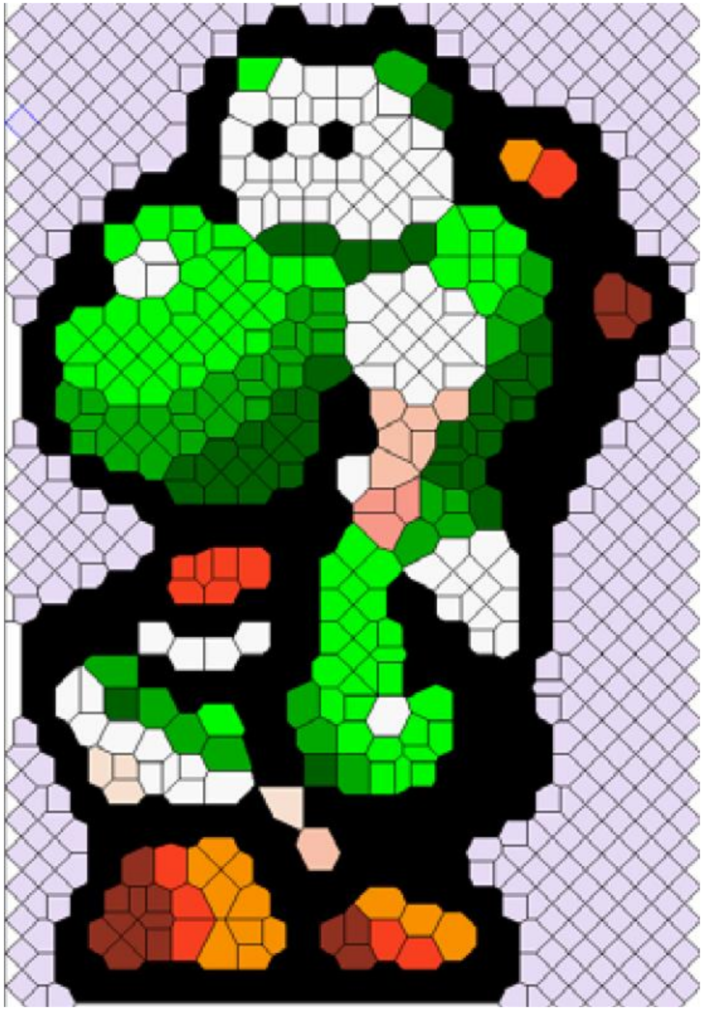
(d) Reshaped pixel cells reflecting the connections in the resolved similarity graph.



(e) Splines fit to visible edges.

(f) Final result

Vectorization

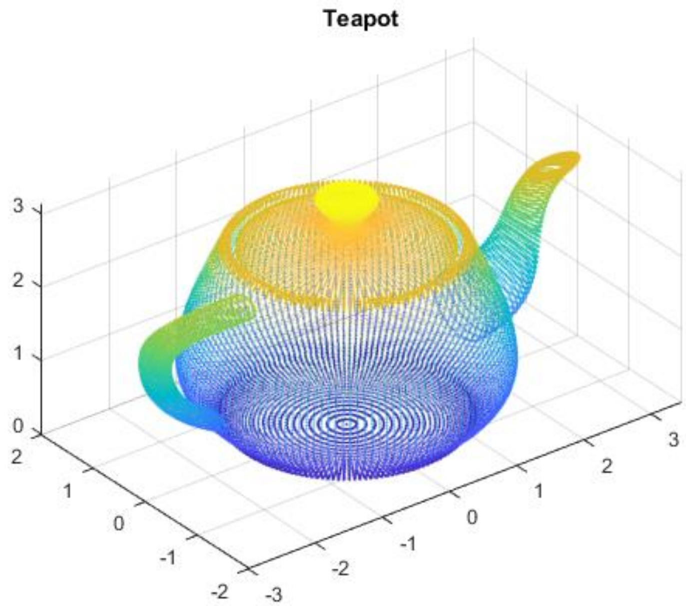


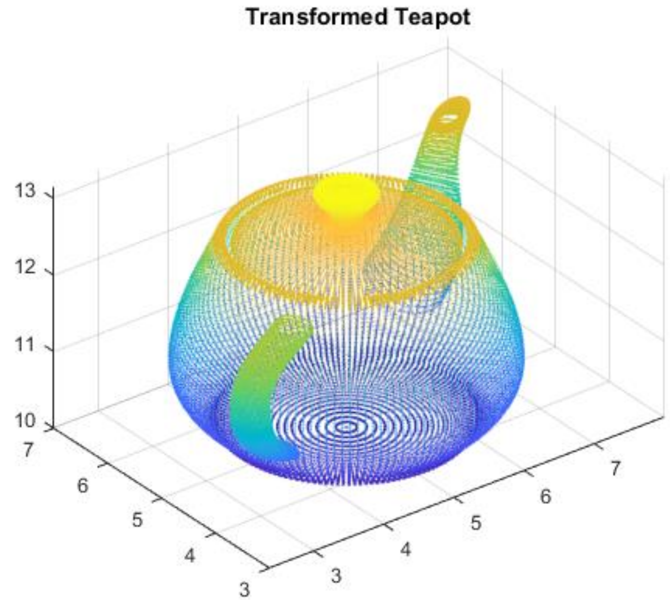
Referenced paper :

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)

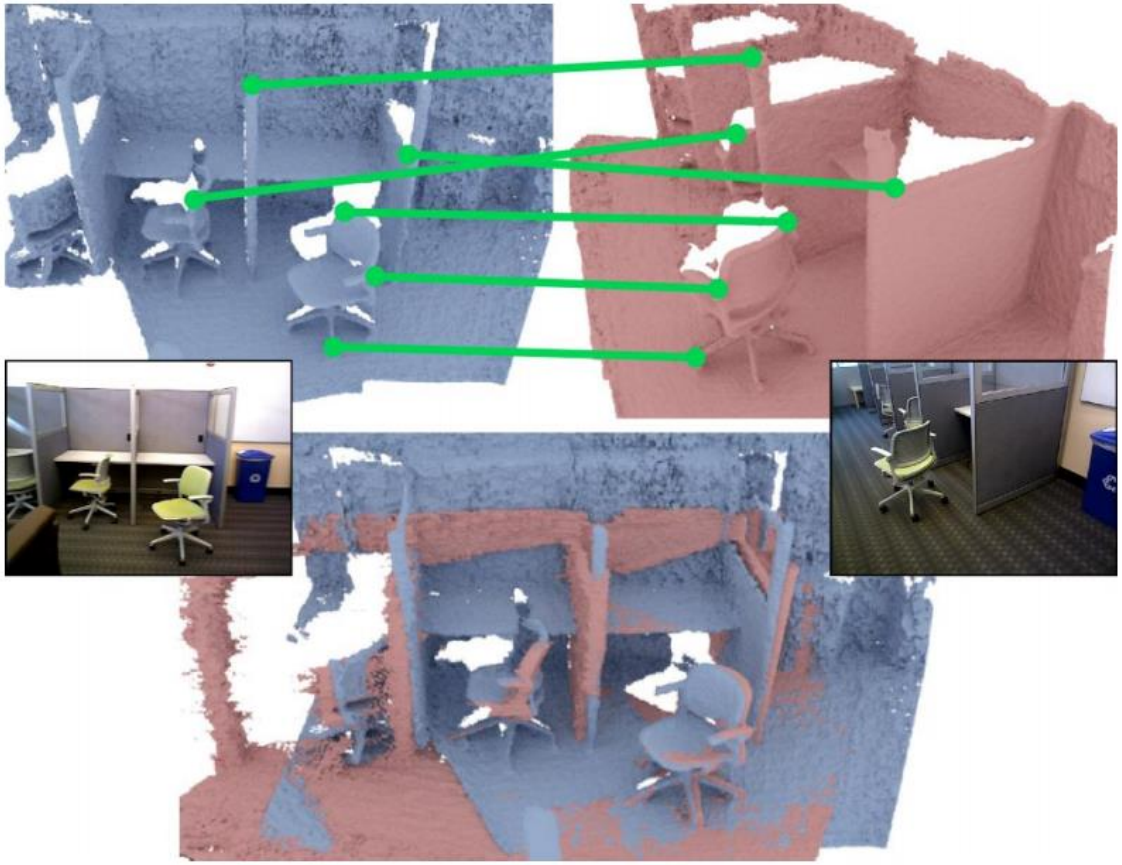
Model registration



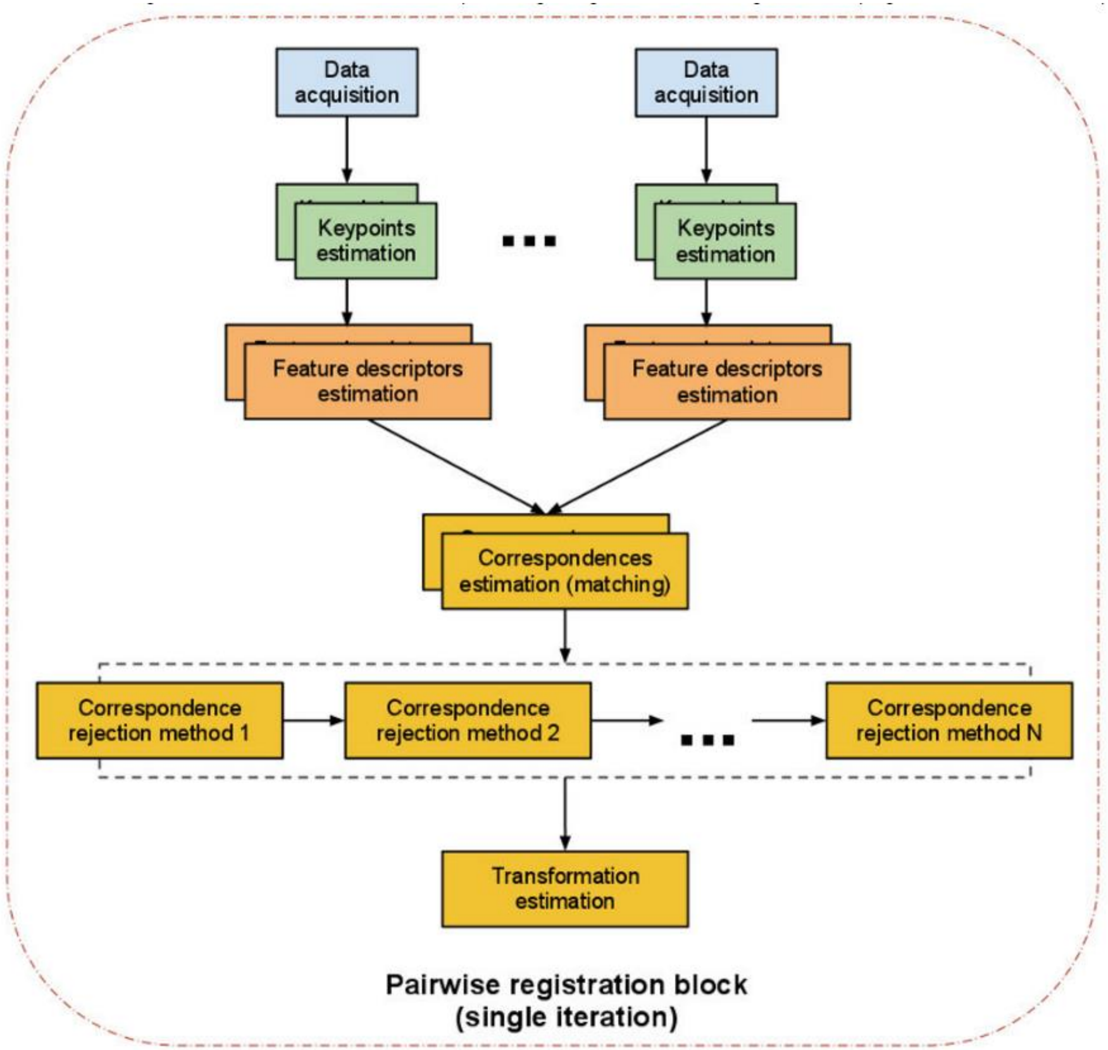


- Model transform from source to target
- Find out the pose estimation

Model registration



- Point cloud registration


Data acquisition
Data acquisition
Keypoints estimation
Keypoints estimation
Feature descriptors estimation
Feature descriptors estimation
Correspondences estimation (matching)
Correspondence rejection method 1
Correspondence rejection method 2
Correspondence rejection method N
Transformation estimation
Pairwise registration block (single iteration)

Model Registration

- Reference Library :
  - PCL regritration API
  - ICP
- Reference Paper :
  - Efficient Variants of ICP Algorithm
  - GOGMA: Globally-Optimal Gaussian Mixture Alignment, CVPR2016

Video Processing

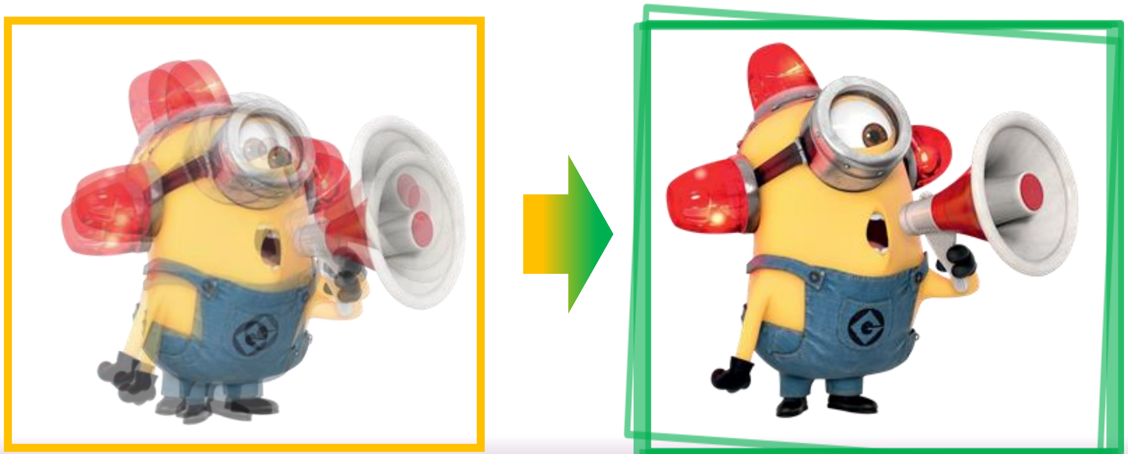
Video stabilization

Background subtraction

Video tracking

Video Stabilization

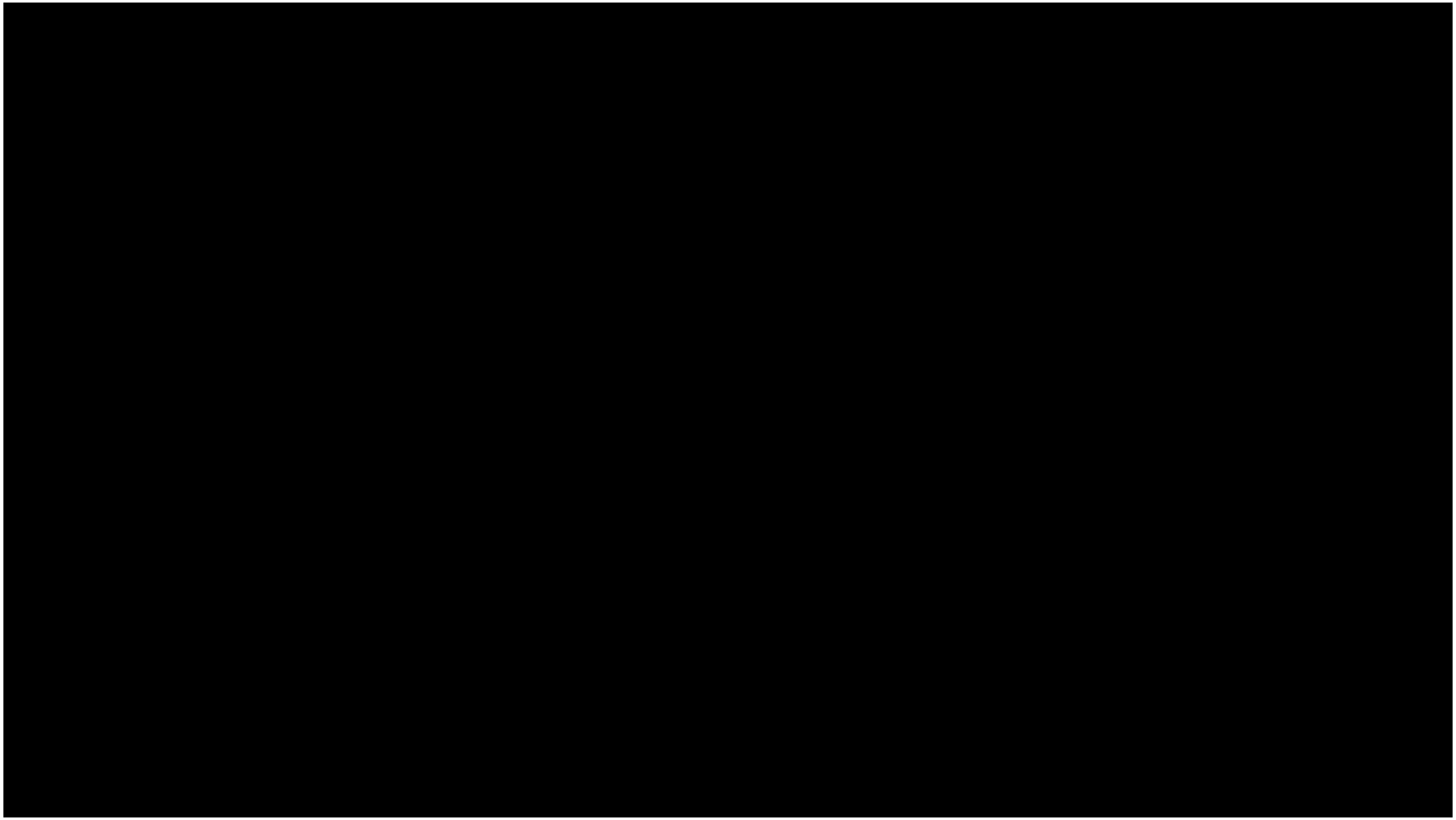
- Most videos which are captured by hand-held cameras are often very shaky and difficult to watch.
- **Video stabilization** techniques have been developed to smooth camera motion in video before viewing.
- This process is aim to estimate and compensate for the image motion occurring due to the ego-motion of the camera.



Video Stabilization

- Techniques are classified as two categories:
  - Feature-based method: extract and match discrete features between two frames and modify the trajectories of these features to fit a global motion model
  - Flow-based method: estimate the optical flow of the image sequence and use it to determine the global model
- The steps of Matlab tutorial:
  - Extract and match local features
  - Eliminate the outliers
  - Compute the Homography
  - Warp frames

Video Stabilization



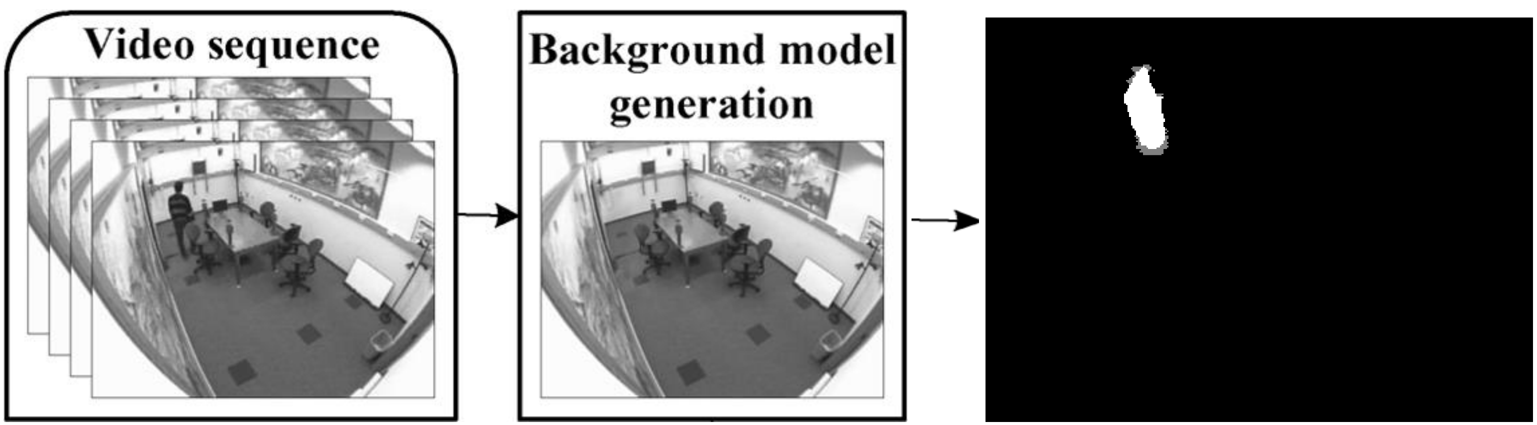
Youtube

Video Stabilization

- Matlab tutorial
  - https://www.mathworks.com/help/vision/examples/video-stabilization-using-point-feature-matching.html
- Related PPT
  - https://www.slideshare.net/soheilash/video-stabilization
- SVW Dataset
  - http://www.cse.msu.edu/~liuxm/sportsVideo/

Background Subtraction

- **Background subtraction** is a technique in the fields of computer vision wherein an image's foreground is extracted for further processing
- BS is a widely used approach for detecting moving objects in videos from static cameras, e.g. surveillance tracking, human poses estimation…

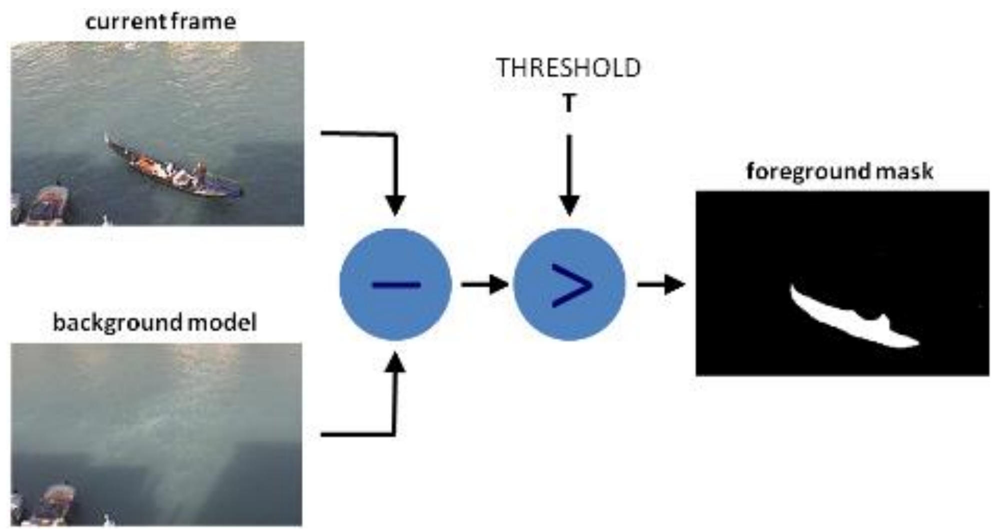


Background Subtraction

- The simplest way is that segment out the objects by using image subtraction technique
- For each pixels in I(t), take the pixel value P[I(t)] and subtract it with the corresponding pixels at the same position on the background image P[B].

$$P[F(t)] = P[I(t)] - P[B]$$

$$|P[F(t)] - P[F(t+1)]| > \text{Threshold}$$



Background subtraction

- Better approach: BackgroundSubtractorMOG

http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html

- Related paper & book

https://sites.google.com/site/backgroundsubtraction/overview

- BS dataset

http://bmc.iut-auvergne.com/?page_id=24

http://wordpress-jodoin.dmi.usherb.ca/#

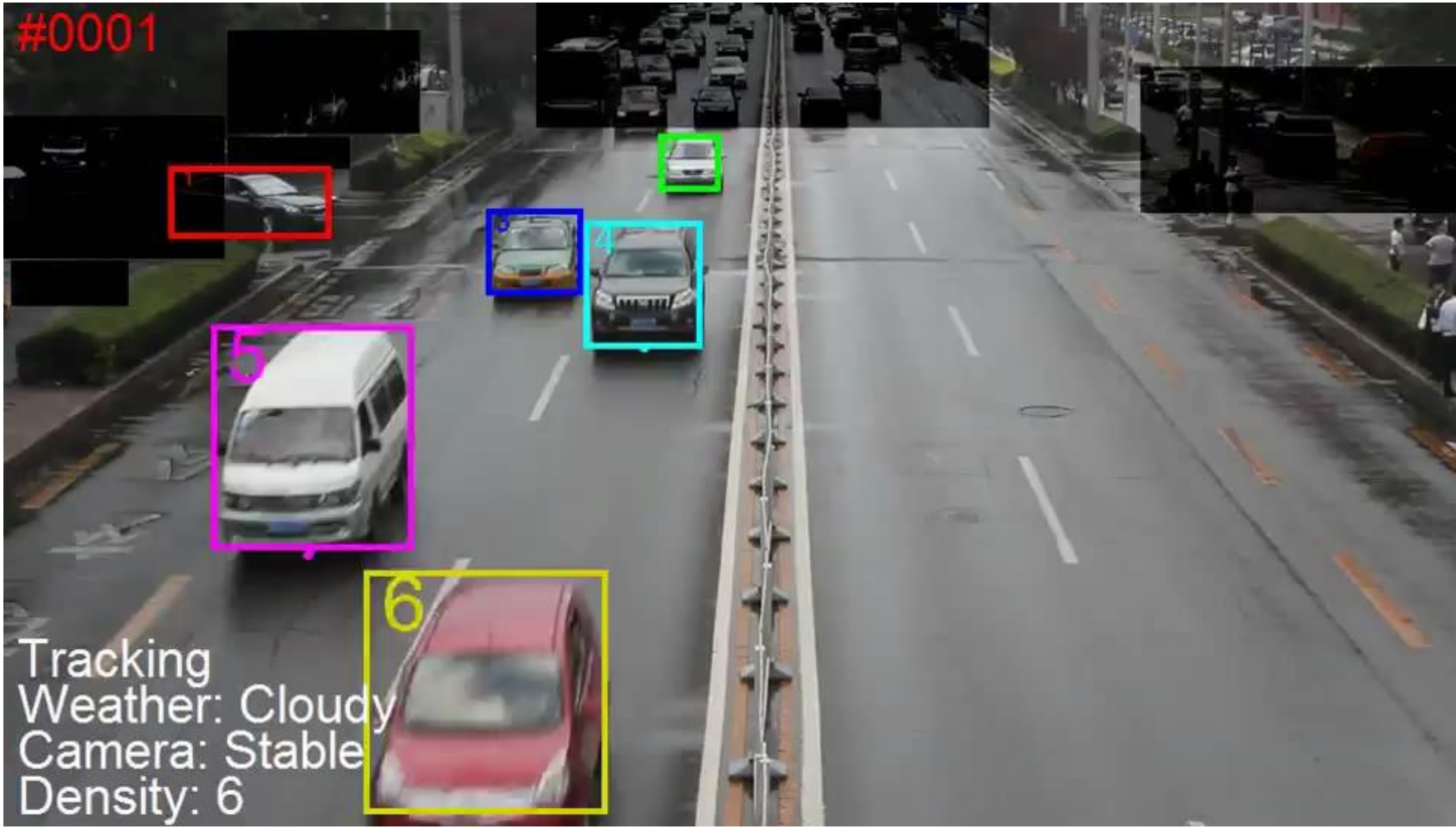
Background subtraction - Dataset

Website

Video Tracking

- Video tracking is the process of locating moving objects, and be used in security, video compression, AR, traffic control, and so on.
- The problem of moving-based object tracking can be divided into two parts:
  - Detect moving objects in each frame.
  - Associate the detections corresponding to the same object over time.

Video Tracking



Website

Video Tracking

- Matlab tutorial
  - https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html
- Dataset
  - The KITTI Vision Benchmark Suite