Final Project

CS3570 introduction to multimedia

Department of Computer Science
National Tsing Hua University

1.Image inpainting

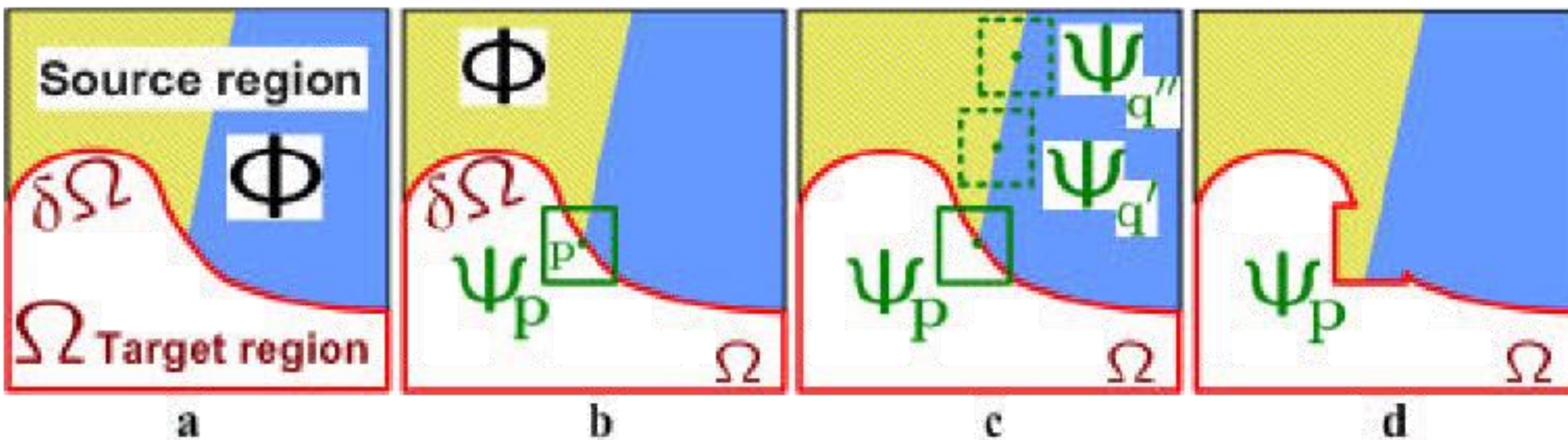
- ***Inpainting*** is the process of reconstructing lost or deteriorated parts of images
- Ref:
  - **Image Inpainting [PDF]**
  - **Object Removal by Exemplar-Based Inpainting [PDF]**

Department of Computer Science
National Tsing Hua University

1.Image inpainting

a) Original image, with the *target region* Ω, its contour *δ*Ω and the *source region* Φ clearly marked.

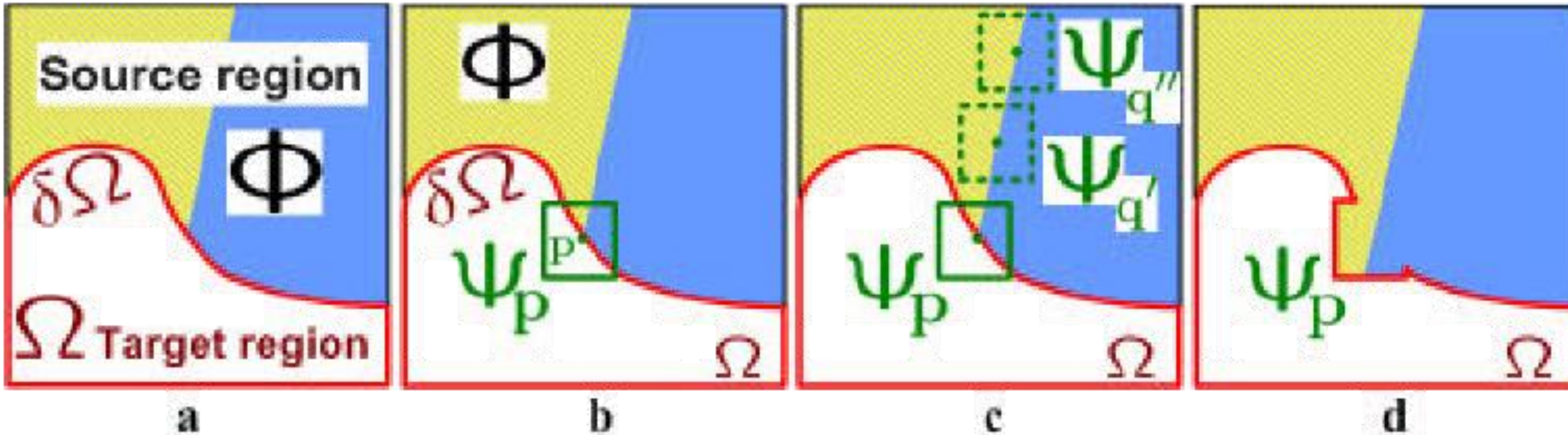
b) We want to synthesize the area delimited by the patch Ψ**p** centred on the point **p** *∈ δ*Ω.



1.Image inpainting

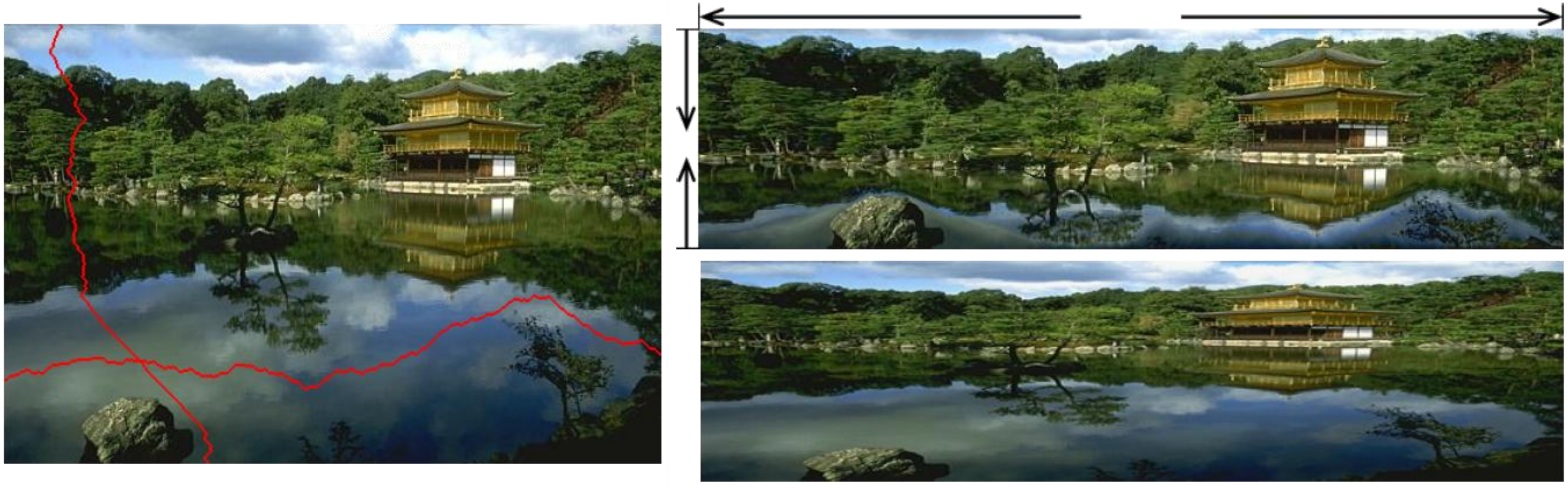
c) The most likely candidate matches for Ψ**p** lie along the boundary between the two textures in the source region, *e.g.*, Ψ**q** and Ψ**q** .

d) The best matching patch in the candidates set has been copied into the position occupied by Ψ**p**, thus achieving partial filling of Ω.



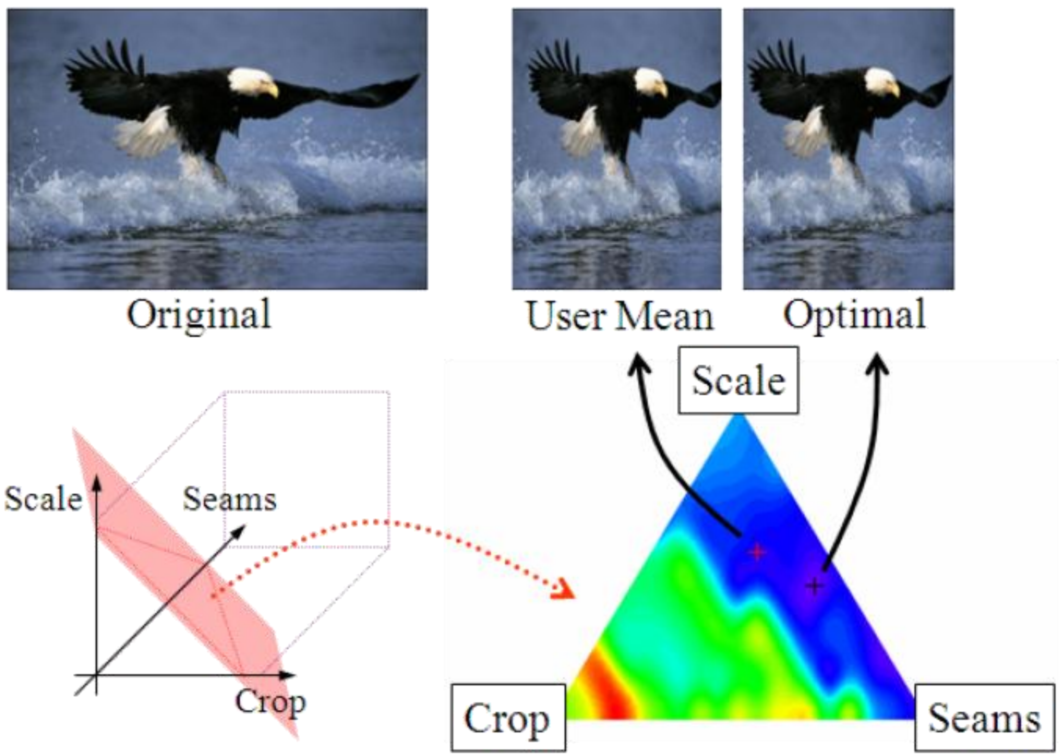
2. Image retargeting

- Resizing images that was aware of the actual photo's contents
- Ref:
  - **Seam Carving for Content-Aware Image Resizing** [PDF]
  - **Multi-operator Media Retargeting** [PDF] [Code]



2. Image retargeting

- Using regular paths we are limited to searching on a plane in resizing space. We find the optimal multi-operator resizing sequence having the minimum BDW distance.



2. Image retargeting

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

3. Image editing

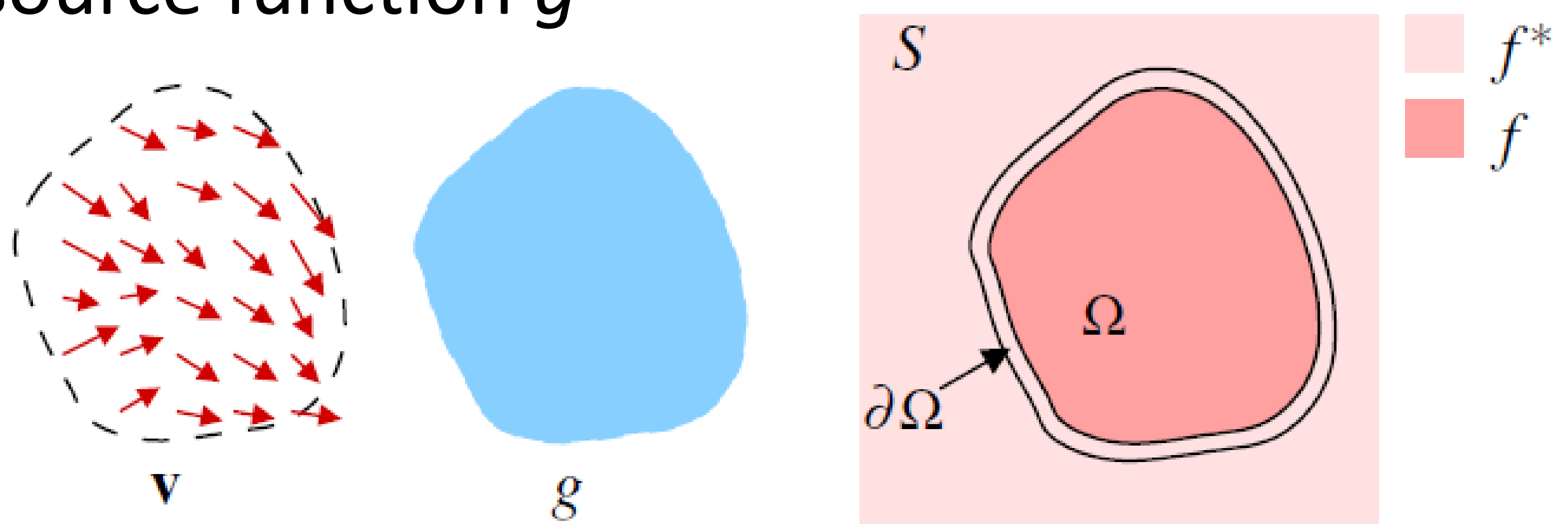
- *Image editing* encompasses the processes of altering images
- Ref:
  - **Poisson Image Editing** **PDF** **Code**
  - **Fast Poisson Blending using Multi-Splines** **PDF**

cloning

seamless cloning

3. Image editing

- **Guided interpolation notations**. Unknown function *f* interpolates in domain W the destination function *f* , under guidance of vector field **v**, which might be or not the gradient field of a source function *g*



4. Non-linear Image Denoising

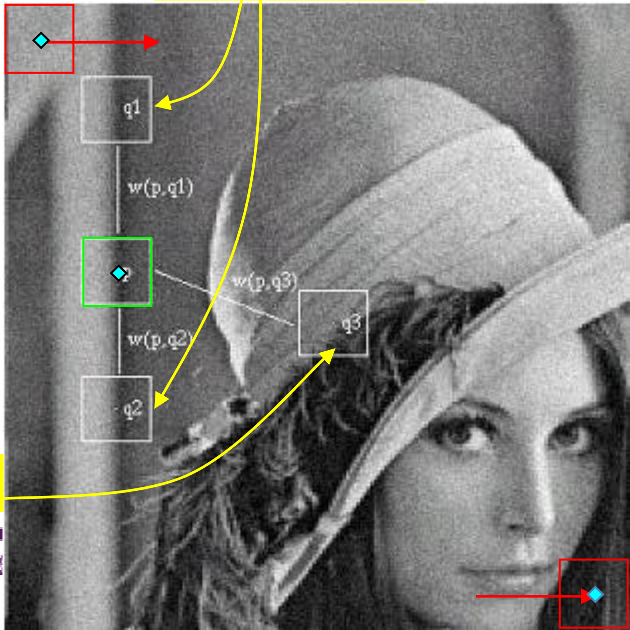
- The goal of image denoising methods is to recover the original image from a noisy image for advanced image analysis
- Ref:
  - **A non-local algorithm for image denoising PDF**
  - **Bilateral filter for gray and color images PDF**

4. Non-linear Image Denoising – non-local mean

- Given a 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

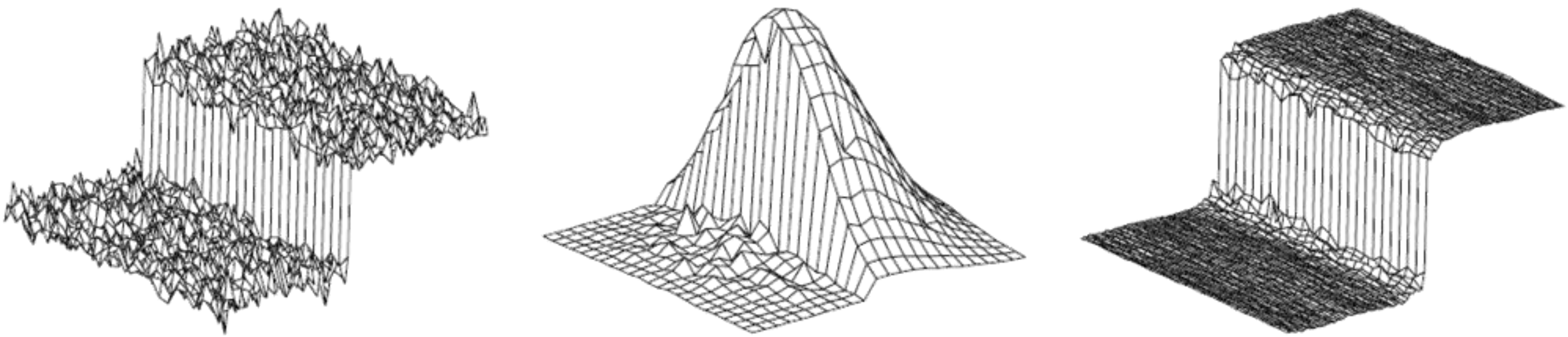
$$NL[v](i) = \sum_{j\in I} \omega(i,j)v(j)$$

- where the family weights $\{w(i,j)\}_j$ depend on the similarity between the pixels i and j, and satisfy the usual conditions $0\le w(i,j)\le 1$ and $\sum_{j\in I} w(i,j)=1$.



4. Non-linear Image Denoising – Bilateral filter

a) A 100-gray-level step perturbed by Gaussian noise with σ= 10 gray levels.
b) Combined similarity weights c(ξ; x)s(f(ξ); f(x)) for a 23x23 neighborhood centered two pixels to the right of the step in (a). The range component effectively suppresses the pixels on the dark side.
c) The step in (a) after bilateral filtering with σ_r = 50gray levels and σ_d = 5pixels.



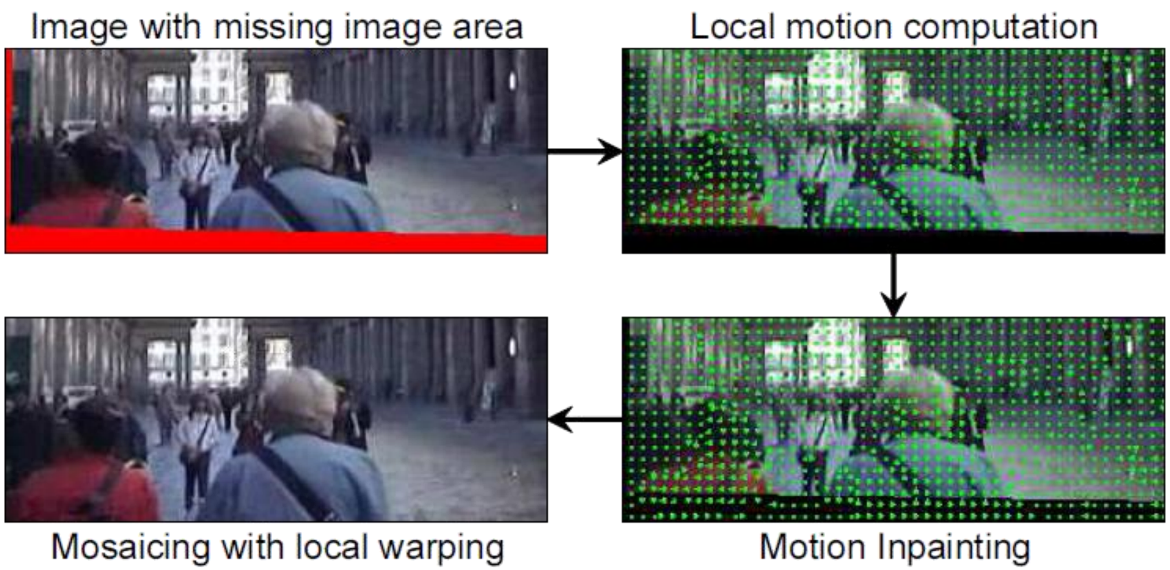
(a) (b) (c)

5. Video stabilization

- Videos retrieved from video devices is affected by unwanted camera shakes and jitters, resulting in video quality loss.
- Video stabilization techniques are important to obtain high quality and stable video footages even in non-optimal conditions
- Ref:
  - Full-frame video stabilization PDF
  - Content-Preserving Warps for 3D Video Stabilization PDF

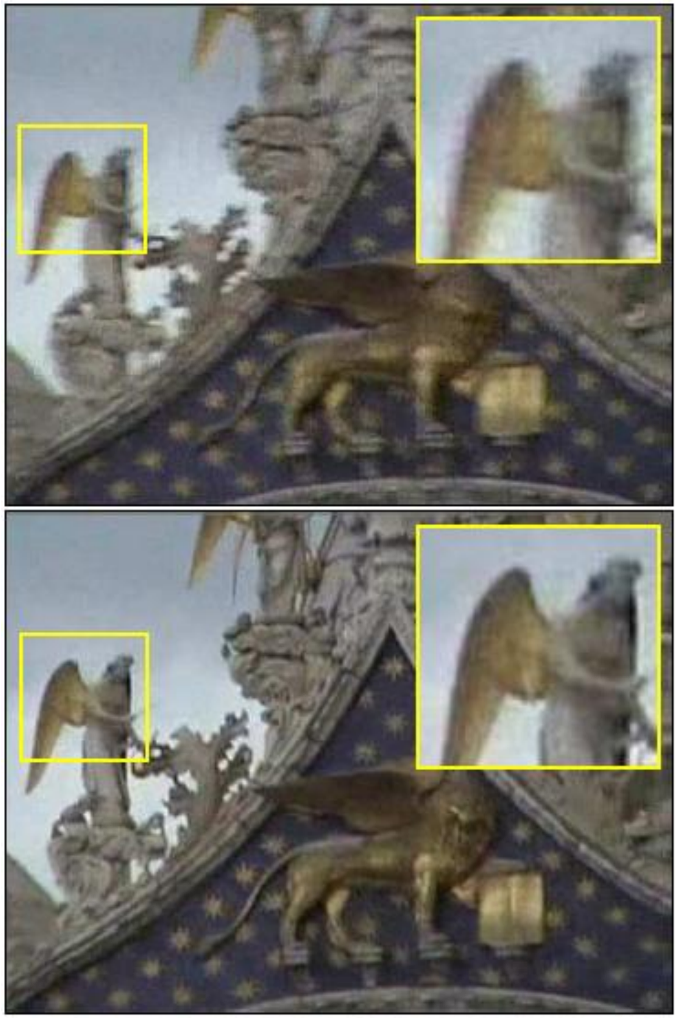
5. Video stabilization

1. **Video completion with motion inpainting**
   - Local motion is first computed between the current frame and a neighboring frame. Computed local motion is then propagated with motion inpainting method. The propagated motion is finally used to locally adjust mosaics



5. Video stabilization

2. **Practical motion deblurring method**
   - While motion blur in original videos looks natural, it becomes an annoying noise in stabilized videos because it does not match the compensated camera motion.



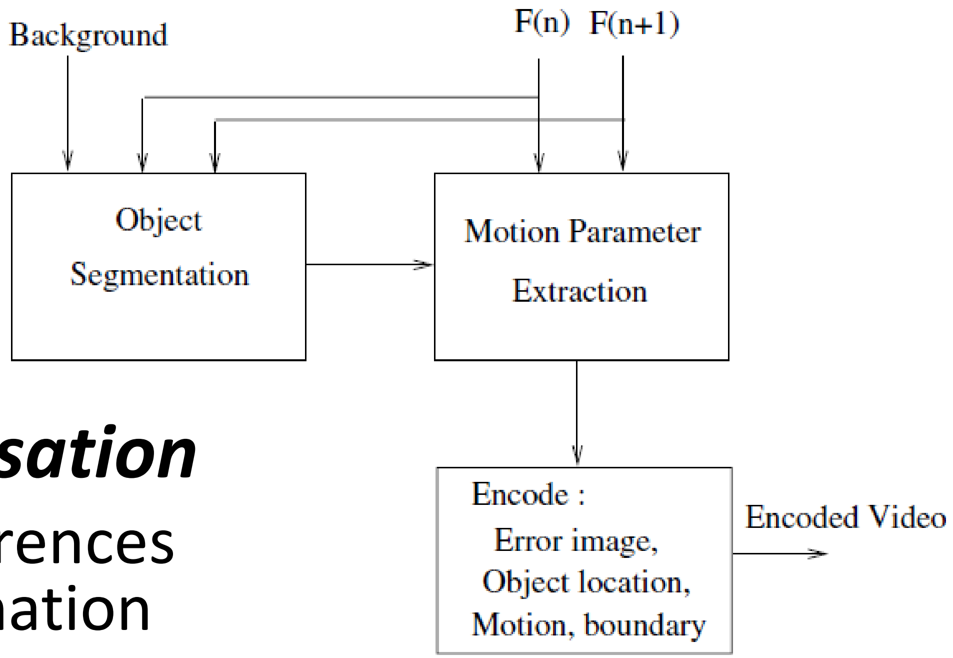
6. Surveillance video compression

- Video surveillance has been widely used in recent years to enhance public safety and privacy protection
- Surveillance video usually has constant background
- State-of-the-art video compression methods such as H.264/AVC often lead to high computational complexity
- The algorithm taught in class can be used to compress surveillance video and compare with other compression algorithms.
- Ref:
  - Object-based Surveillance Video Compression using Foreground Motion Compensation PDF

6. Surveillance video compression

1) *Object segmentation*

- Better segmentation accuracy reduces the amount of bits required to code the video



2) *Foreground Motion Estimation and Compensation*

- Use sum-of squared differences (SSD) based motion estimation for objects

3) *Object Error Coding*

- Error obtained after motion compensation (Δ) is coded using object based SA-DCT procedure

8. Background subtraction / synthesis

- Moving object detection in video sequences is one of the main tasks in many computer vision applications.
- Background subtraction is an common approach for this task. The idea is to compare the current image against background model which learned by GMM
- C. Staufer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking,”*CVPR*, Vol. 2, pp. 246-252, June 1999.

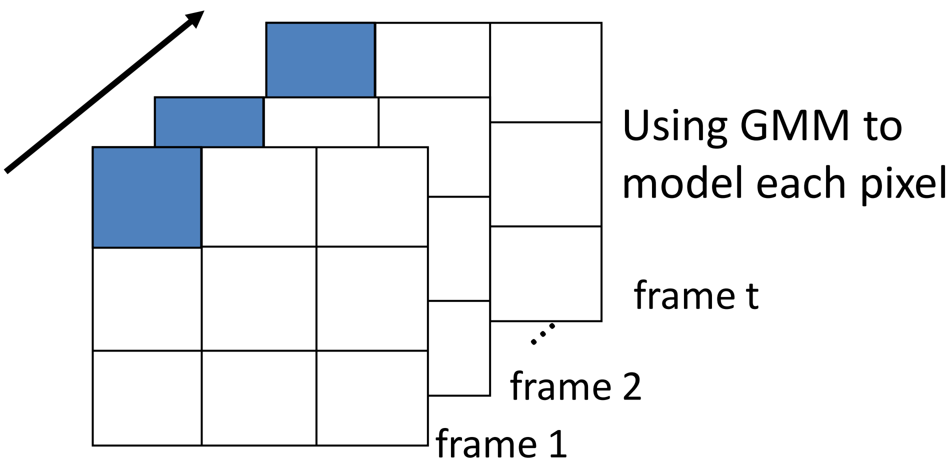
Background subtraction synthesis

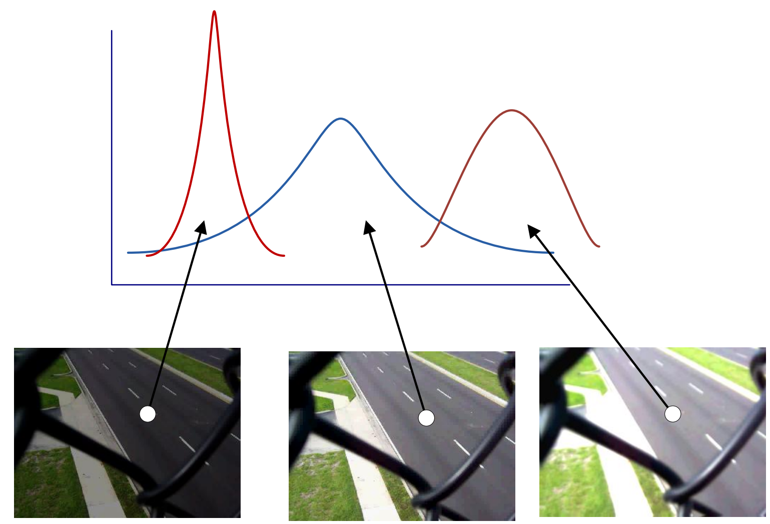
8.Background subtraction / synthesis

- Each pixel modeled with a mixture of Gaussians

$$P(.) = \sum_{i=1}^{K} w_i \eta(., \mu_i, \Sigma_i)$$

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-1/2(X_t-\mu)^T \Sigma^{-1} (X_t-\mu)}$$





8. Background subtraction / synthesis

- Updating the GMM background model

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha M_{k,t}$$
$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t$$
$$\sigma_t^{\,2} = (1-\rho)\sigma^2{}_{t-1} + \rho(X_t - \mu_t)^T (X_t - \mu_t)$$
$$\rho = \alpha\eta(X_t \mid \mu_k, \sigma_k)$$

- First B states are labeled as background states

$$B = \underset{b}{argmin}\left(\sum_{k=1}^{b} \omega_k > T\right)$$

8. Background subtraction / synthesis

- 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 matte α*(x,y)* $I = \alpha F + (1-\alpha)B$
- Automatically replace the background region by another background image in the input image