KECE471 Computer Vision

Pyramidal Image Representation

Chang-Su Kim

Sections 7.7 and 9.2, Computer Vision by Forsyth and Ponce
Note: Most contents were extracted from the lecture notes of Prof. Kyoung Mu Lee.

Image Pyramid: Example

512 256 128 64 32 16 8

A curve corresponds to

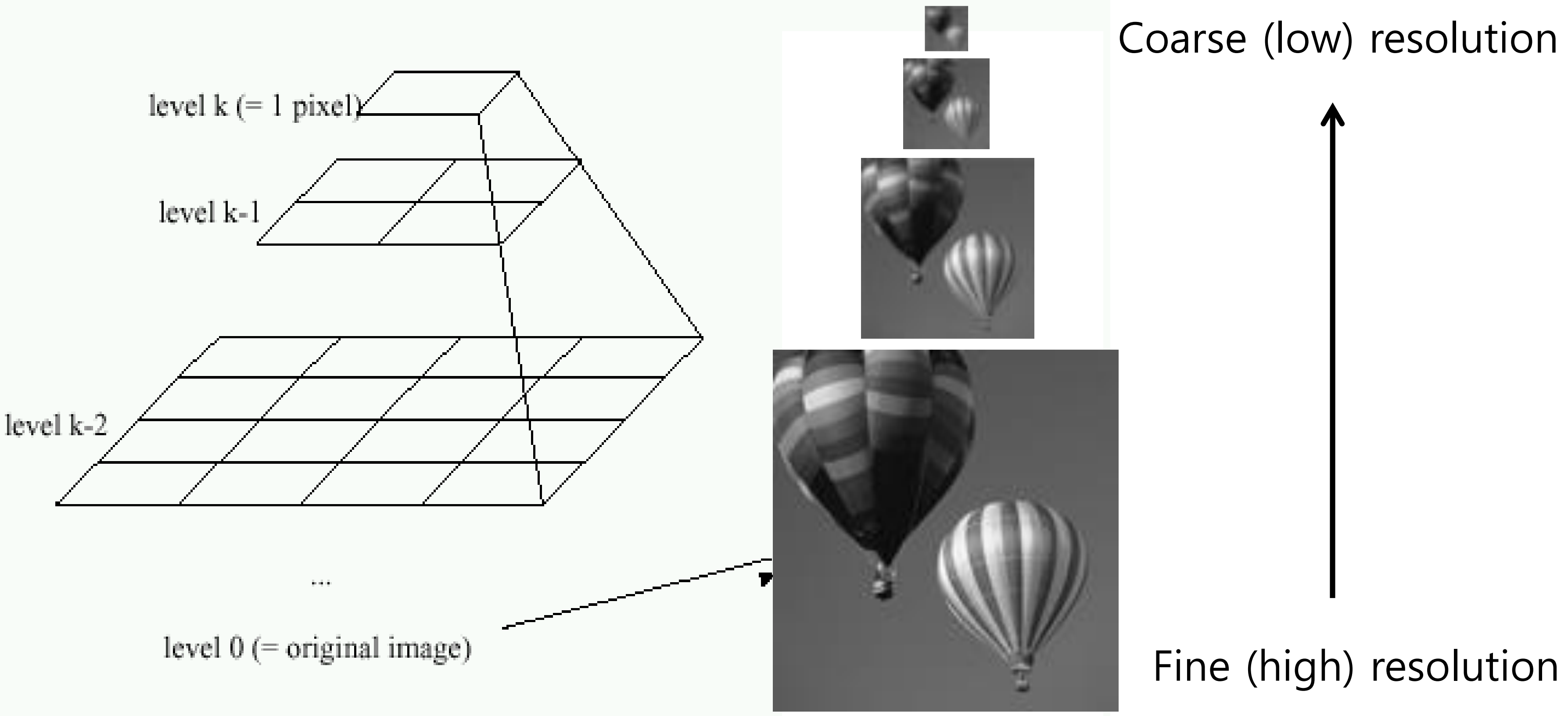
- a hair on the nose in the biggest image
- a stripe in the medium size image
- nose itself in the smallest image


Pyramidal Representation

- Pyramidal representation is a kind of scaled representation
- Both large and small scaled information are interesting
  - Big bars and small bars
  - Stripes and hairs

Image Pyramid (it is not an Egyptian tomb)

Idea: Represent NxN image as a "pyramid" of
1×1, 2×2, 4×4, …, $2^k \times 2^k$ images (assuming $N=2^k$)



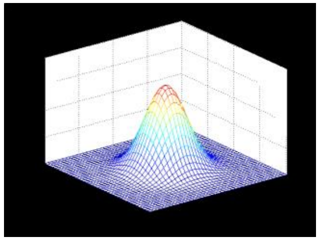
Aliasing

- Lowpass filtering is required before downsampling to avoid aliasing
- Anti-aliasing filtering
  - A Gaussian filter is often used

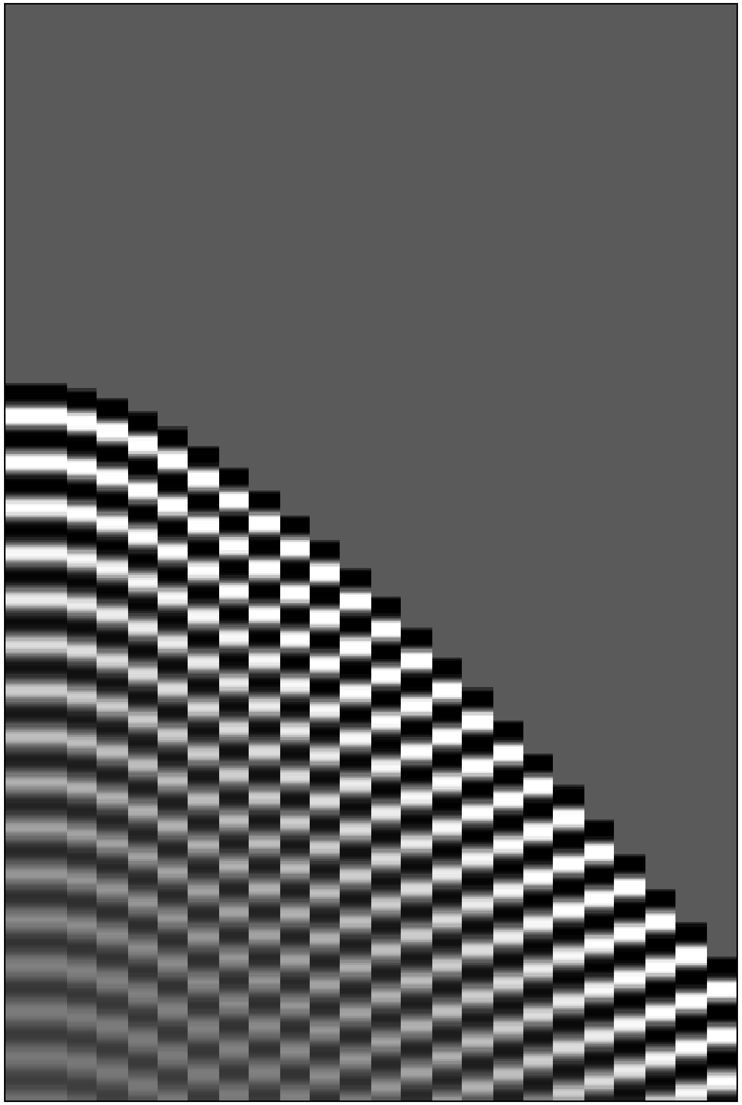
Without anti-aliasing filtering

Aliasing

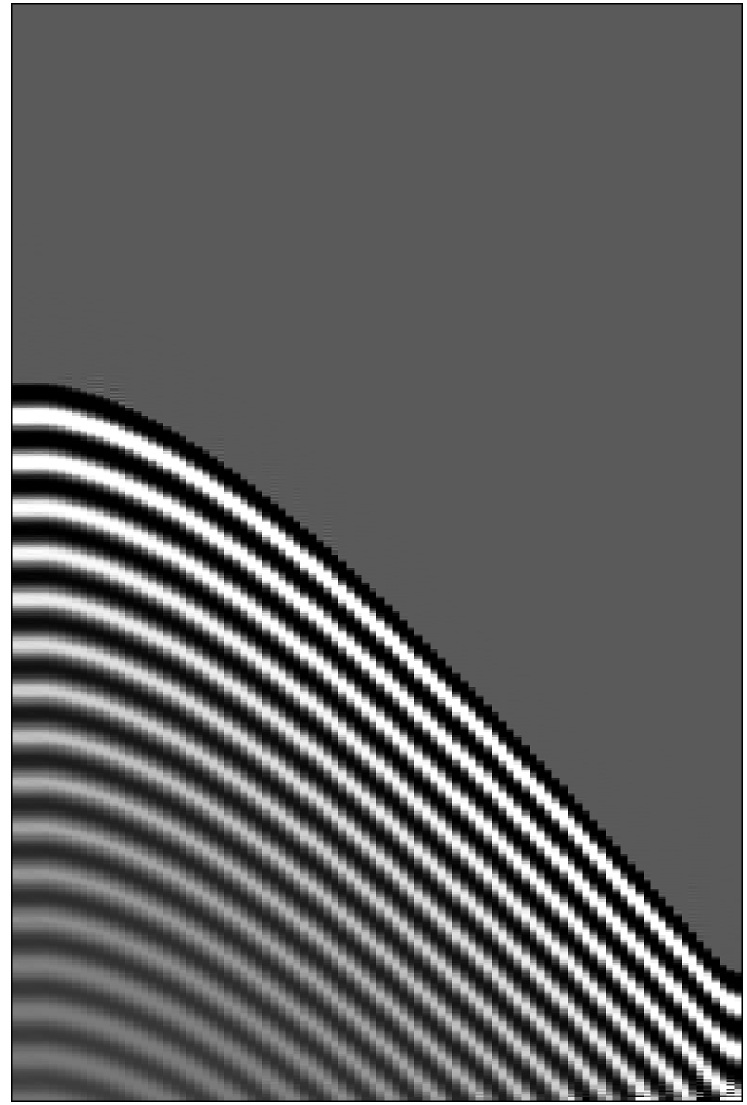
- Lowpass filtering is required before subsampling to avoid aliasing
- Anti-aliasing filtering
  - A Gaussian filter is often used



With anti-aliasing filtering

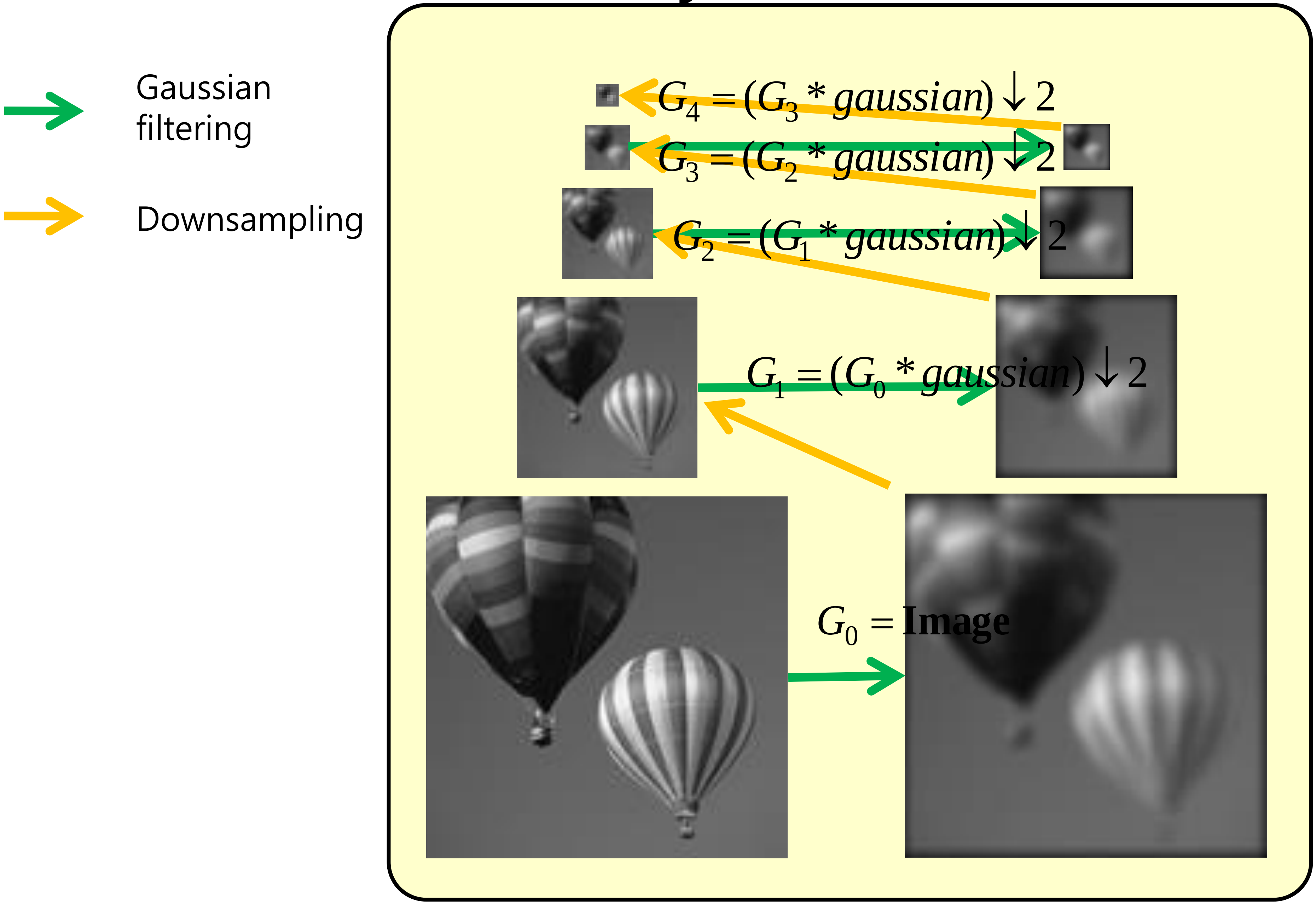


Aliasing

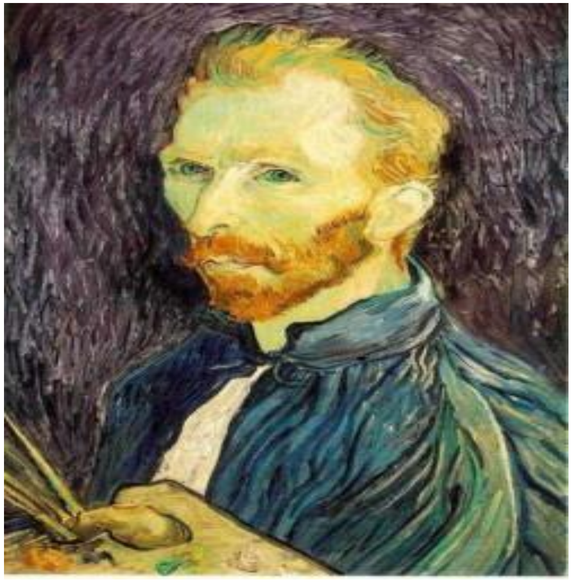


Anti-aliased

Gaussian Pyramid



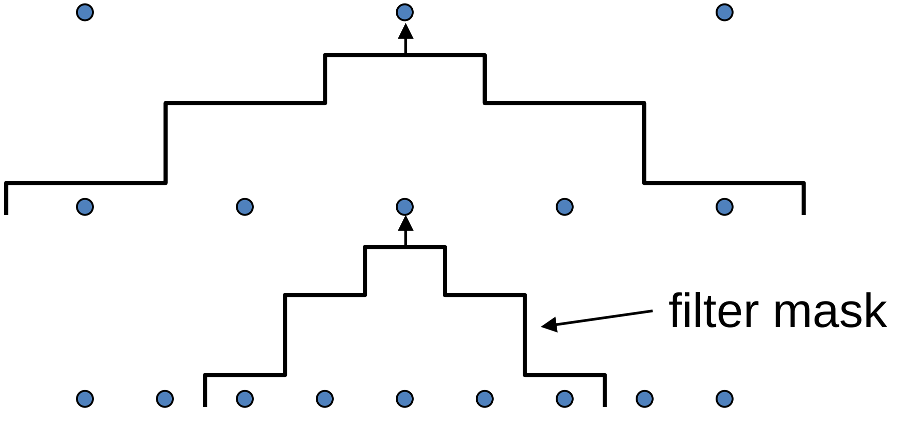
Construction of a Gaussian Pyramid



Gaussian 1/2

G 1/4

G 1/8



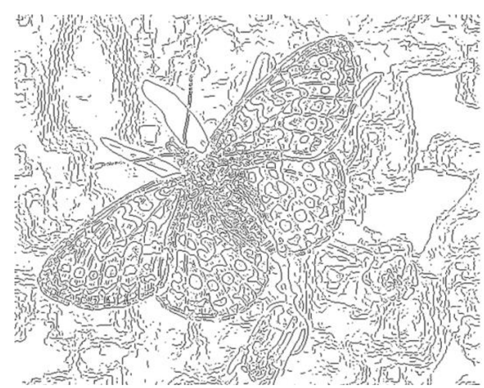
http://web.mit.edu/persci/people/adelson/pub_pdfs/pyramid83.pdf

Applications of Gaussian Pyramids

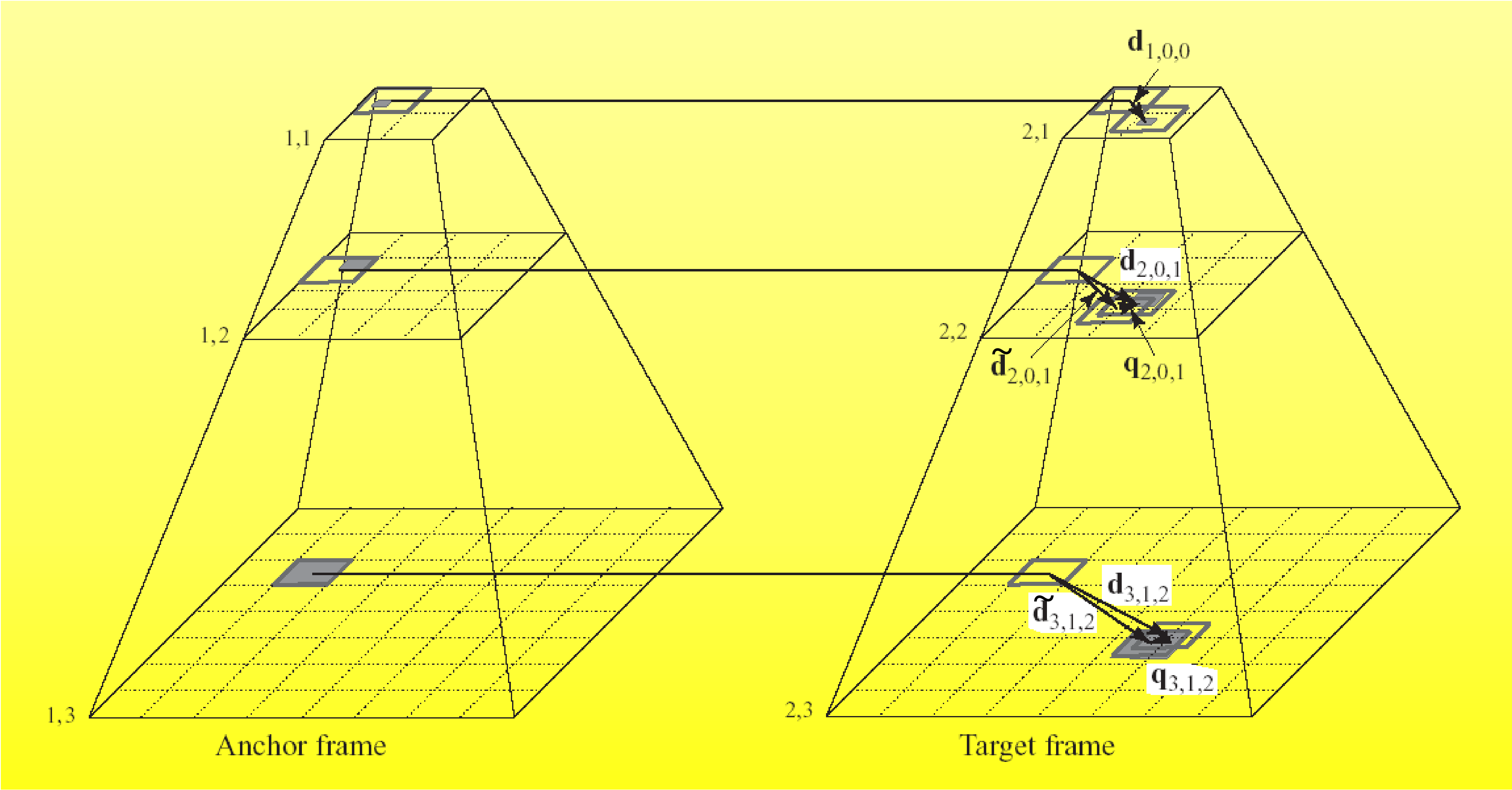
- Search for correspondence
  - look at coarse scales, then refine with finer scales

- Edge tracking
  - a “good” edge at a fine scale has parents at a coarser scale

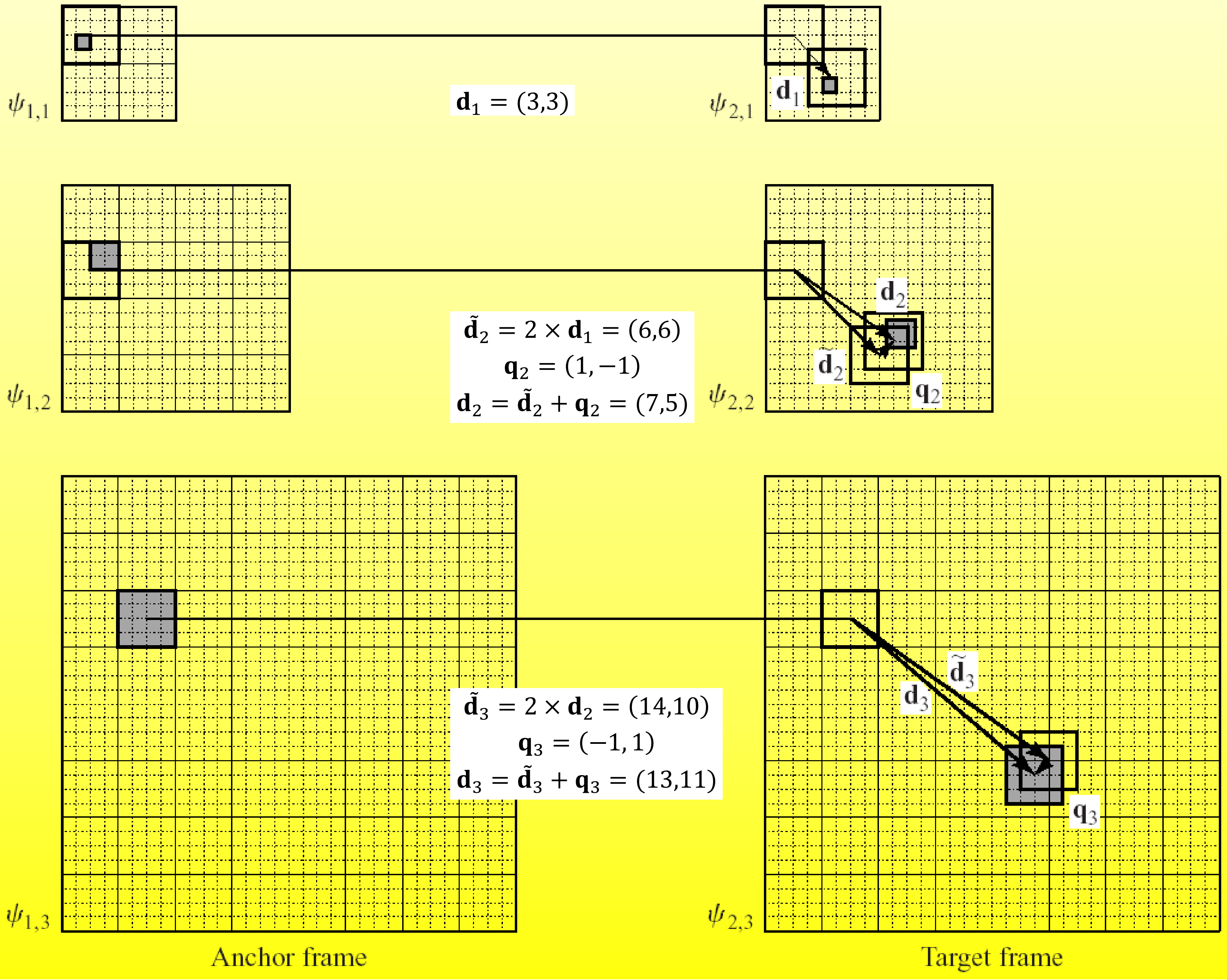
- Template matching
  - e.g. Detecting faces



Hierarchical Block Matching



- Lower resolution motion vector is used to predict higher resolution motion vector (e.g. $d_{2,0,1}$ is used to predict $d_{3,1,2}$)
  - Reduction of computational complexity
  - More reliable motion vector estimation

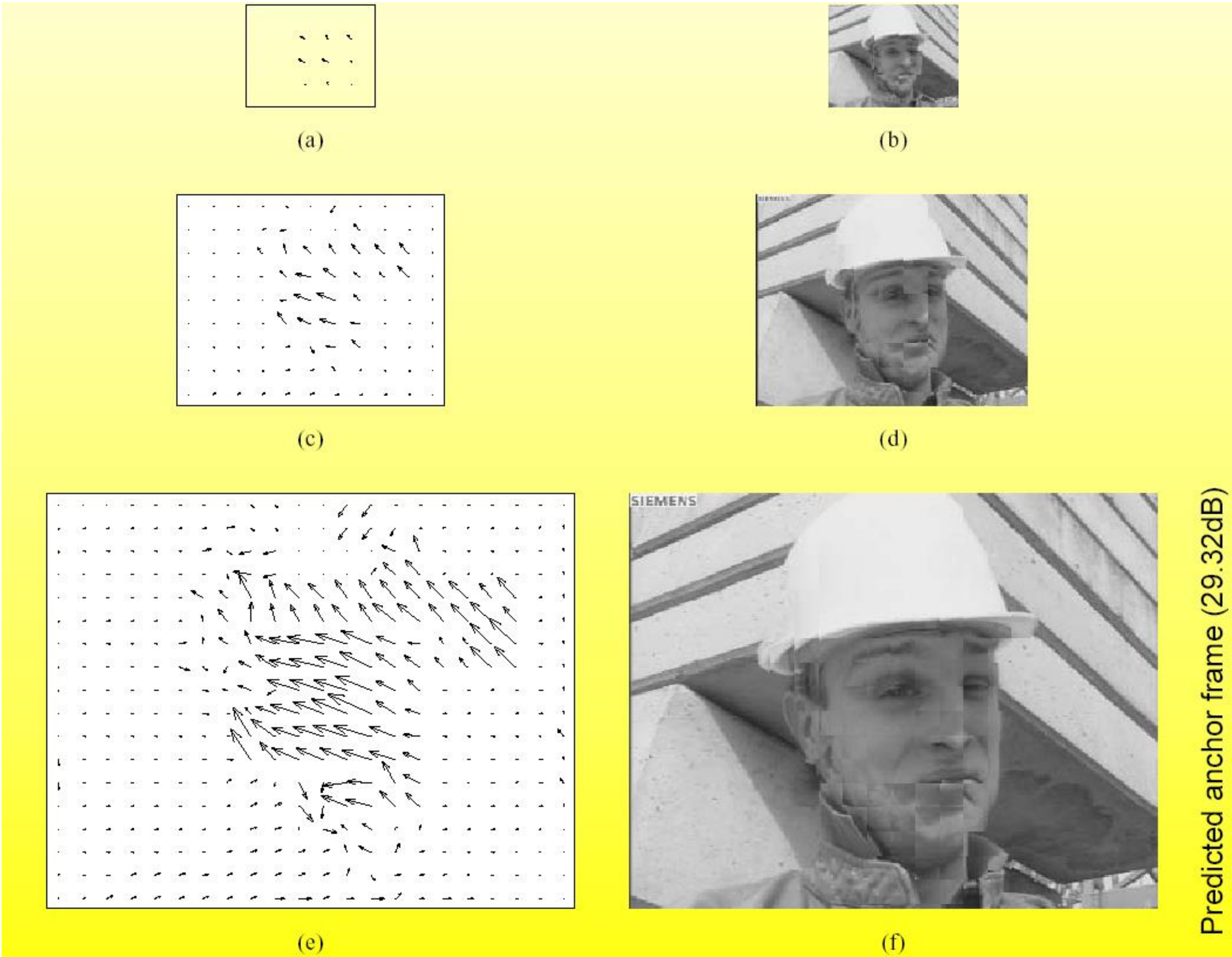


$\psi_{1,1}$
$\mathbf{d}_1 = (3,3)$
$\psi_{2,1}$
$\mathbf{d}_1$
$\psi_{1,2}$
$\tilde{\mathbf{d}}_2 = 2 \times \mathbf{d}_1 = (6,6)$
$\mathbf{q}_2 = (1,-1)$
$\mathbf{d}_2 = \tilde{\mathbf{d}}_2 + \mathbf{q}_2 = (7,5)$
$\psi_{2,2}$
$\mathbf{d}_2$
$\tilde{\mathbf{d}}_2$
$\mathbf{q}_2$
$\psi_{1,3}$
$\tilde{\mathbf{d}}_3 = 2 \times \mathbf{d}_2 = (14,10)$
$\mathbf{q}_3 = (-1,1)$
$\mathbf{d}_3 = \tilde{\mathbf{d}}_3 + \mathbf{q}_3 = (13,11)$
$\psi_{2,3}$
$\tilde{\mathbf{d}}_3$
$\mathbf{d}_3$
$\mathbf{q}_3$
Anchor frame
Target frame

Non-Hierarchical Block Matching Algorithm

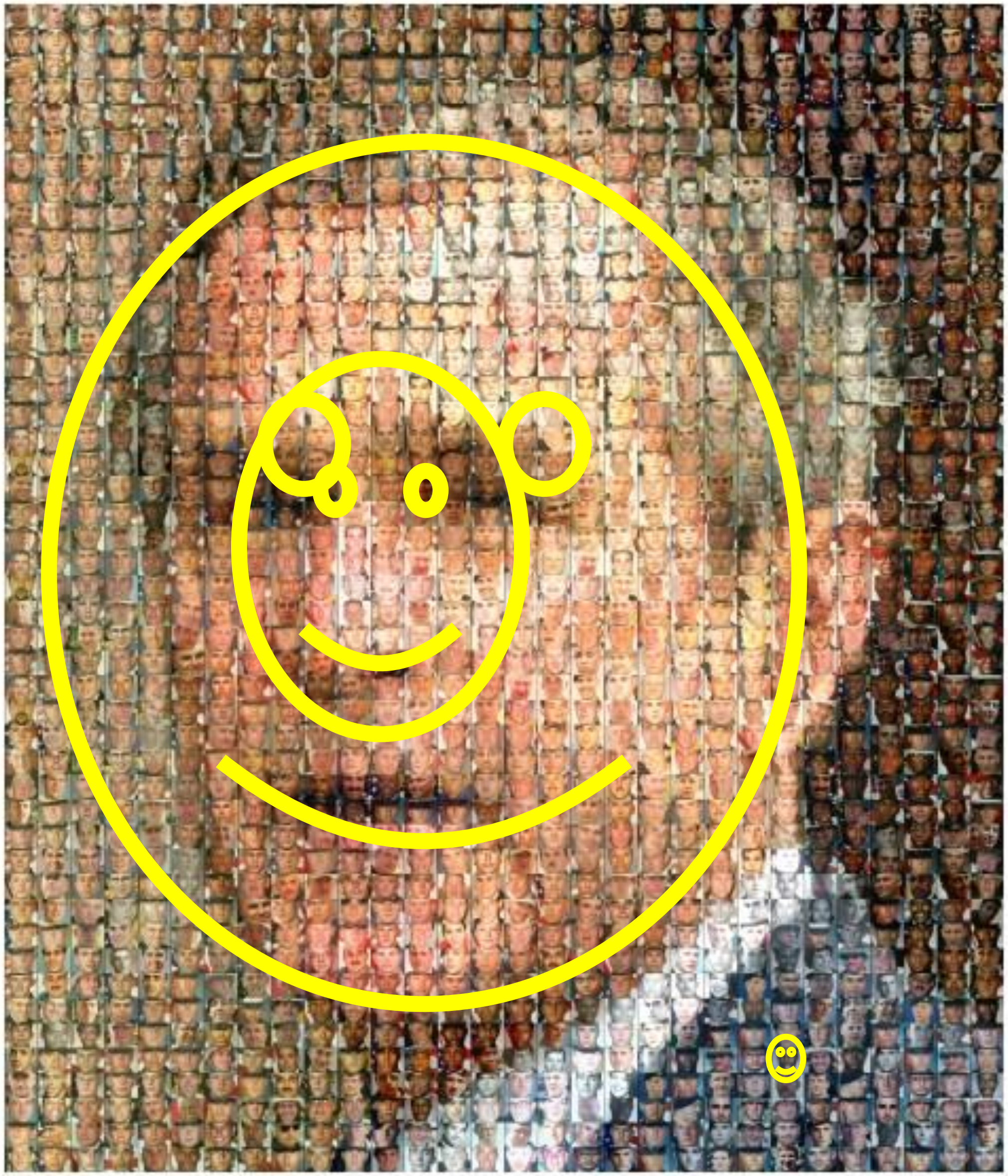
Example: Half-pel EBMA

Hierarchical Block Matching Algorithm



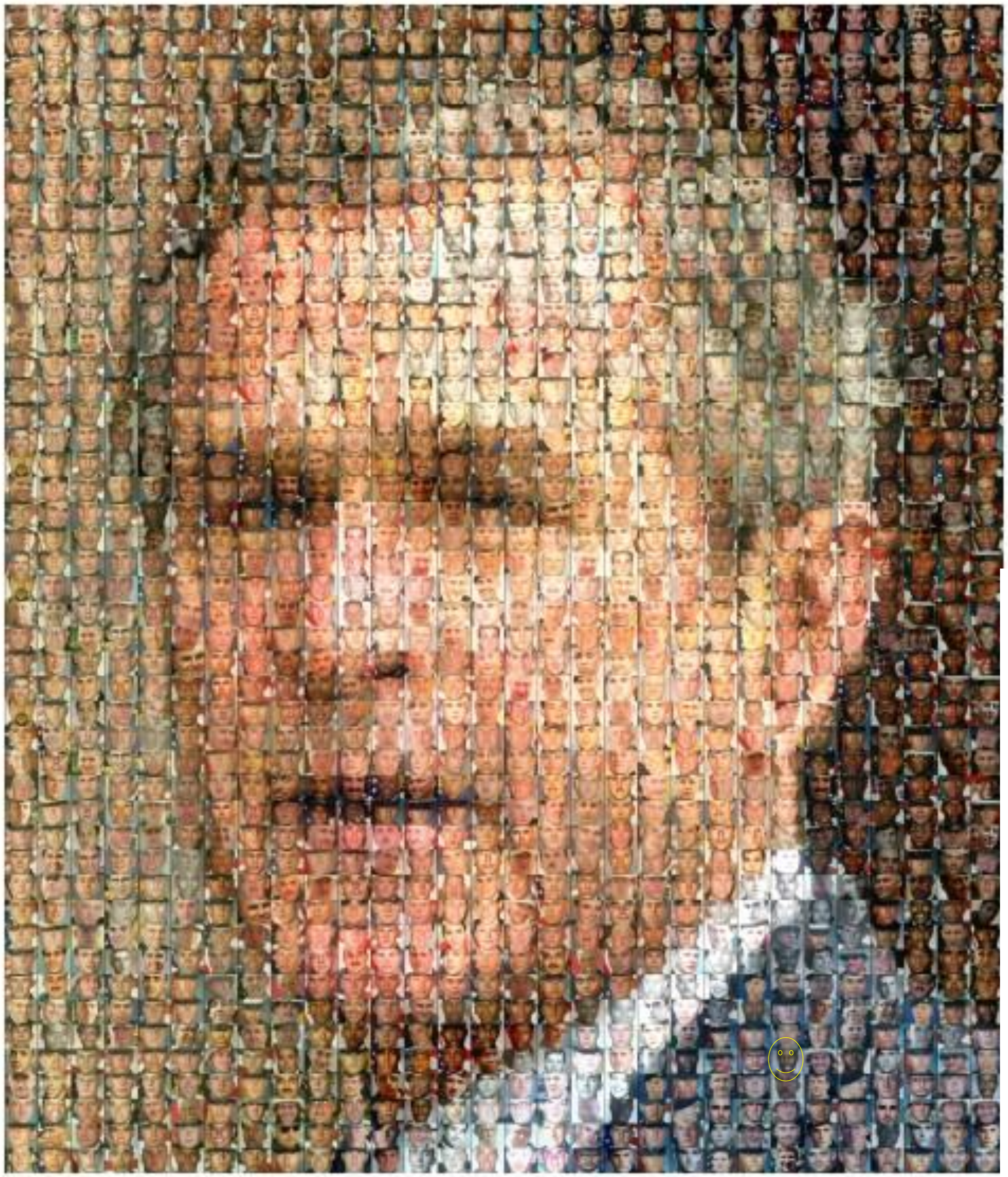
Example: Three-level HBMA

Template Matching

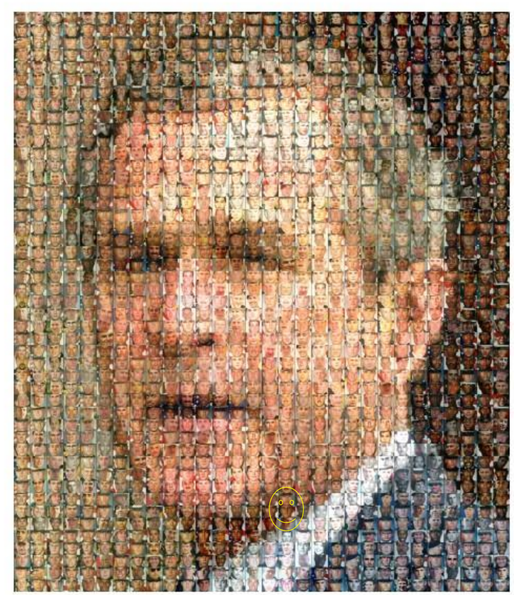


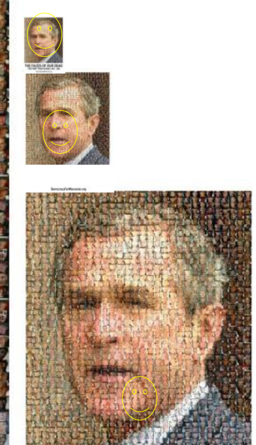
- Strategy 1
  - Use templates of different sizes
  - For large templates, matching is costly

Template Matching



- Strategy 2
  - Apply a fixed-size template to the Gaussian pyramid





Multiscale Saliency Detection Using Random
Walk with Restart

Jun-Seong Kim, Jae-Young Sim, and Chang-Su Kim

To appear in IEEE Trans. Circuits Syst. Video Technol., 2013

Feature Extraction

Random Walk

- Edge weight w_{ij}
  - Feature difference between nodes i and j

- Equilibrium state

$$P\pi = \pi$$

Random Walk

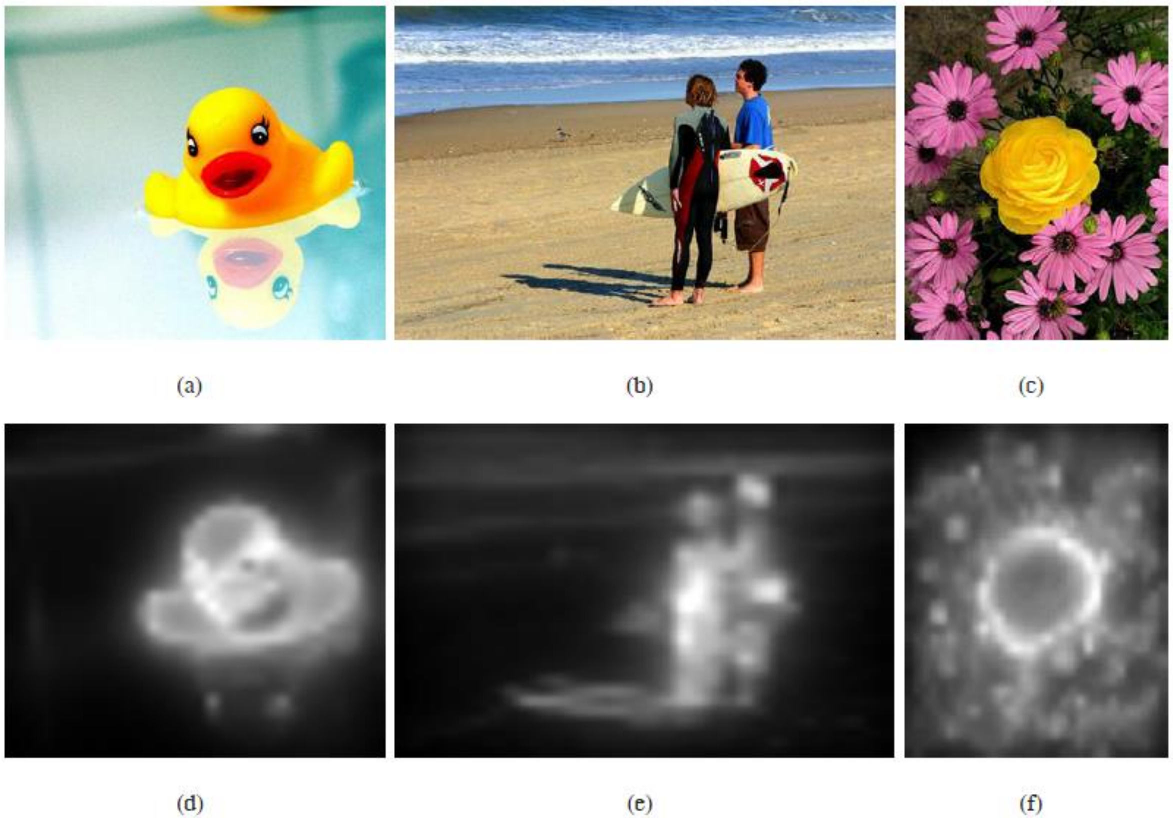
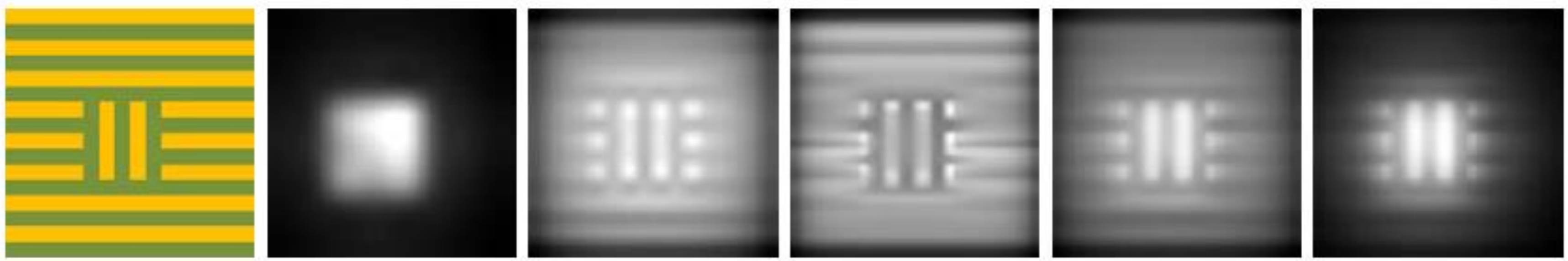


Fig. 3. Results of the single scale saliency detection without the hierarchical refinement: (a)∼(c) input images and (d)∼(f) the corresponding saliency maps.

Scales



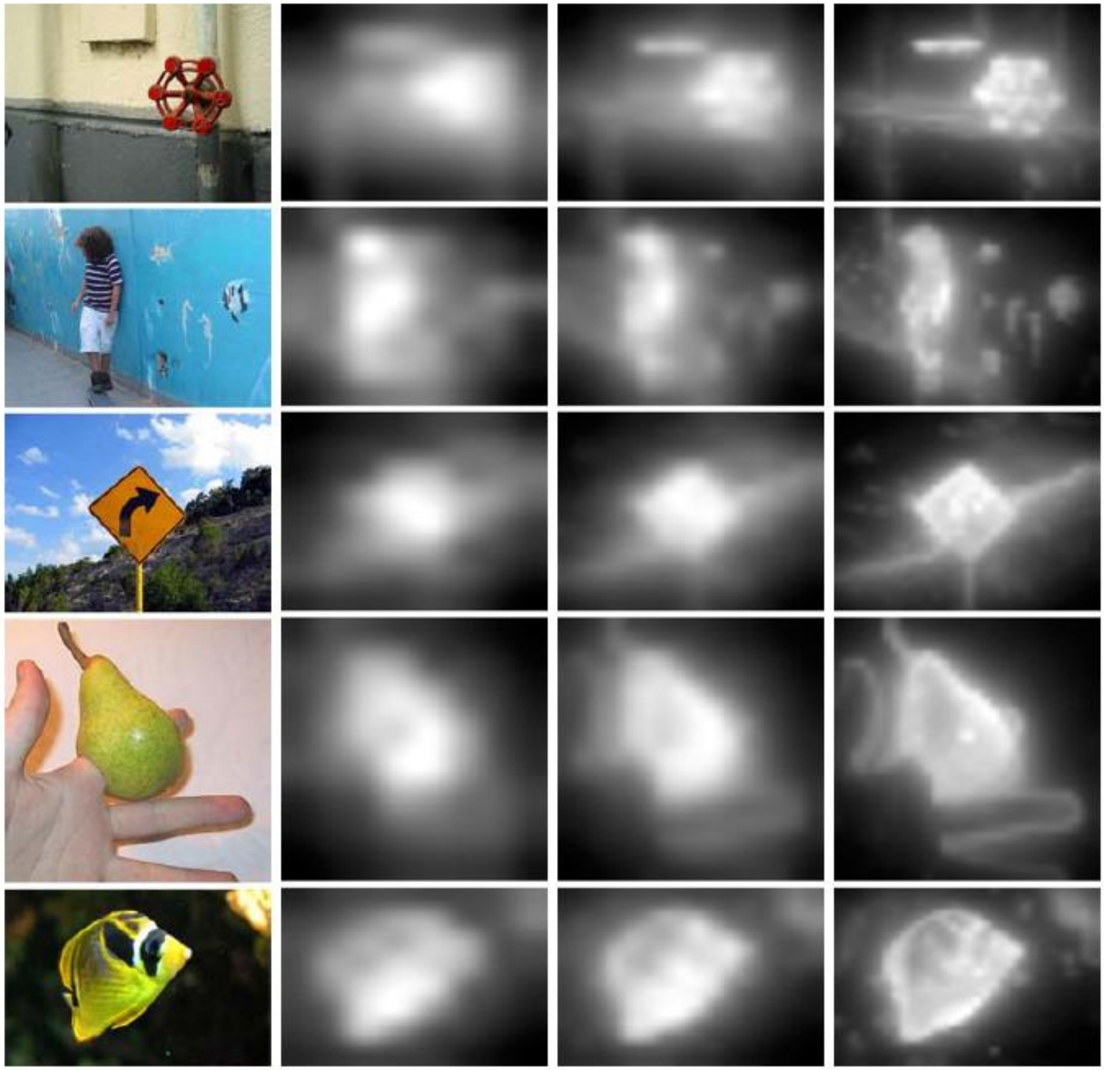
(a) (b) (c) (d) (e) (f)

Fig. 4. Multiscale saliency maps for an input image in (a). The saliency maps at the (b) coarse, (c) medium, and (d) fine scales are combined to yield the weighted average in (e). The weights are 0.1, 0.3, and 0.6 for the coarse, medium, and fine scales, respectively. On the other hand, (f) shows the saliency map obtained by the proposed multiscale saliency detection algorithm.

Hierarchical Saliency Refinement

- Equilibrium state

$$\mathbf{r}_{\text{fine}} = (1-\epsilon)\mathbf{P}\mathbf{r}_{\text{fine}} + \epsilon U(\mathbf{r}_{\text{coarse}})$$



(a) (b) (c) (d)

Fig. 6. Hierarchical saliency refinement. In (a), the first three input images have the size of 400×300, the fourth image 400×400, and the fifth image 400×267. These images yield the saliency maps at (b) the coarse scale (64×64), (c) the medium scale (128×128), and (d) the fine scale (256×256), respectively. Note that the saliency maps are resized to have the same sizes as the input images. These test images are from the MSRA dataset [34].

Laplacian Pyramid

- It removes redundancies in Gaussian Pyramid
- Similar to edge images
- Most pixels are zero
- It can be used in point detection and image compression

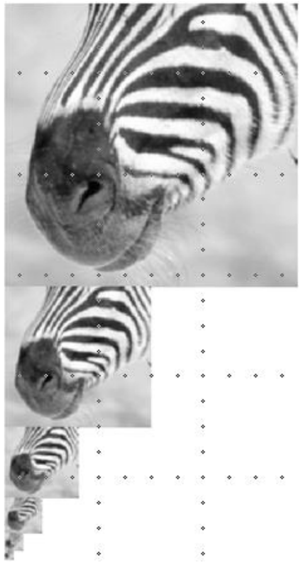
Laplacian Pyramid

- Gaussian Pyramid
  - G_0
  - $G_1 = D(G_0)$
  - $G_2 = D(G_1)$
  - $G_3 = D(G_2)$
- D
  - Gaussian filtering
  - then Downsampling

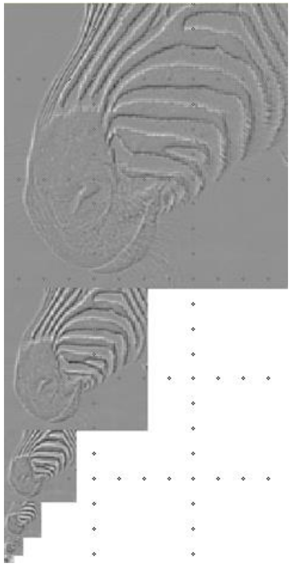
- Laplacian Pyramid
  - $L_0 = G_0 - U(G_1)$
  - $L_1 = G_1 - U(G_2)$
  - $L_2 = G_2 - U(G_3)$
  - $L_3 = G_3$
- U
  - Upsampling

Laplacian Pyramid

- Gaussian Pyramid
  - G_0
  - $G_1=D(G_0)$
  - $G_2=D(G_1)$
  - $G_3=D(G_2)$



- Laplacian Pyramid
  - $L_0=G_0-U(G_1)$
  - $L_1=G_1-U(G_2)$
  - $L_2=G_2-U(G_3)$
  - $L_3=G_3$

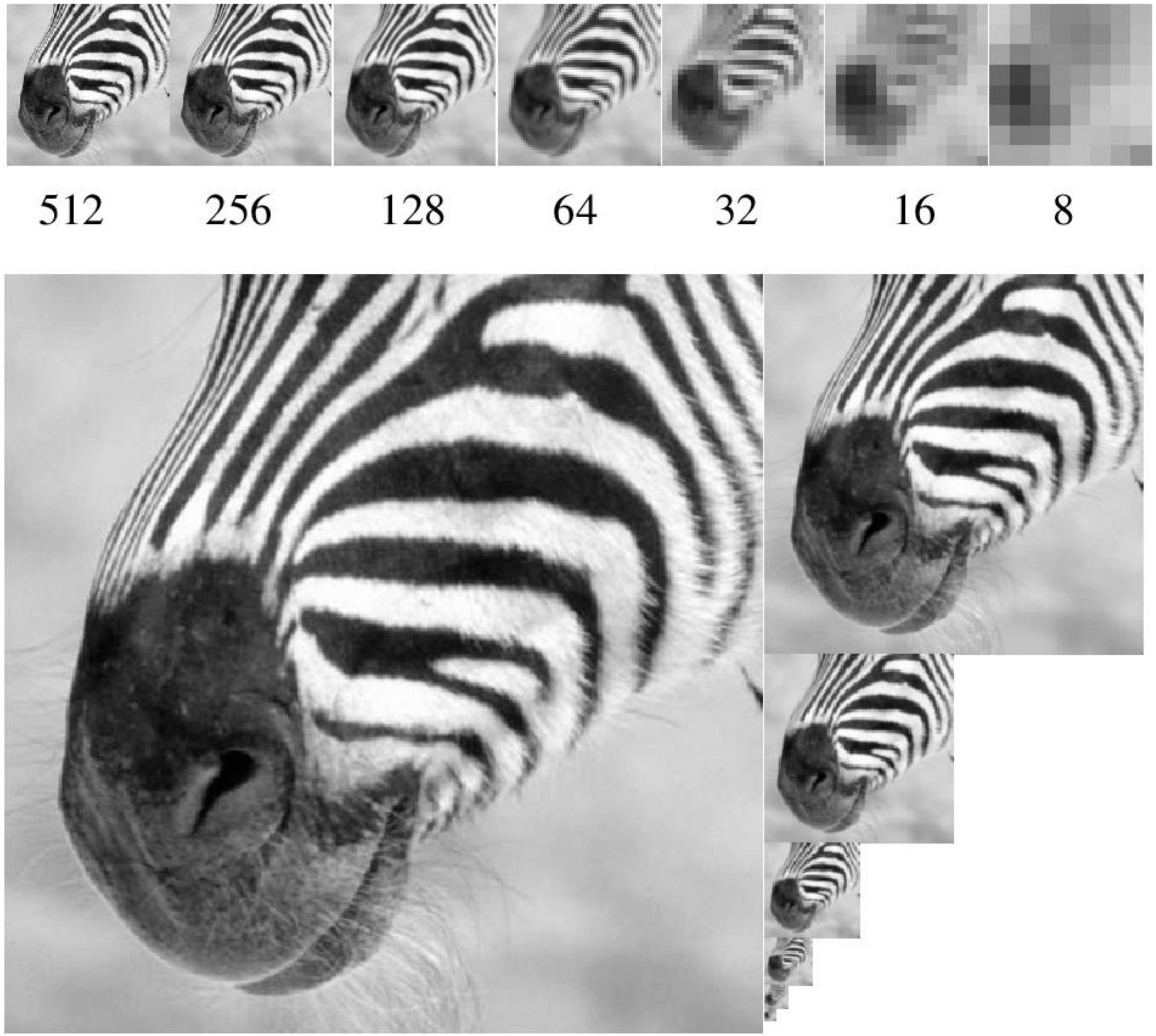


Laplacian Pyramid

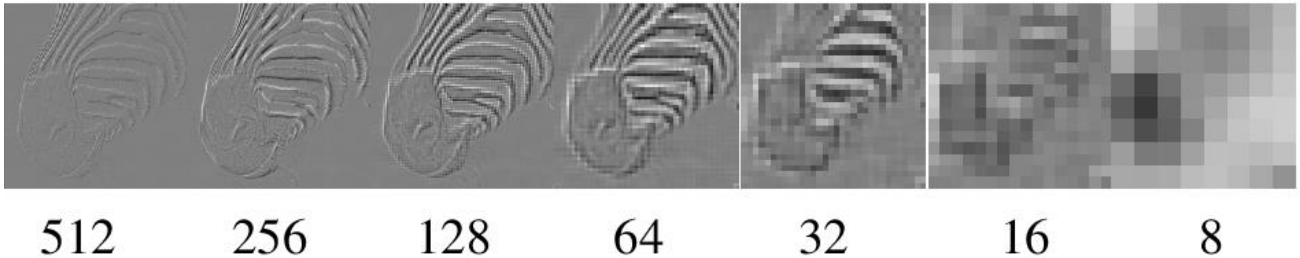
- Analysis
  - $L_0 = G_0 - U(G_1)$
  - $L_1 = G_1 - U(G_2)$
  - $L_2 = G_2 - U(G_3)$
  - $L_3 = G_3$

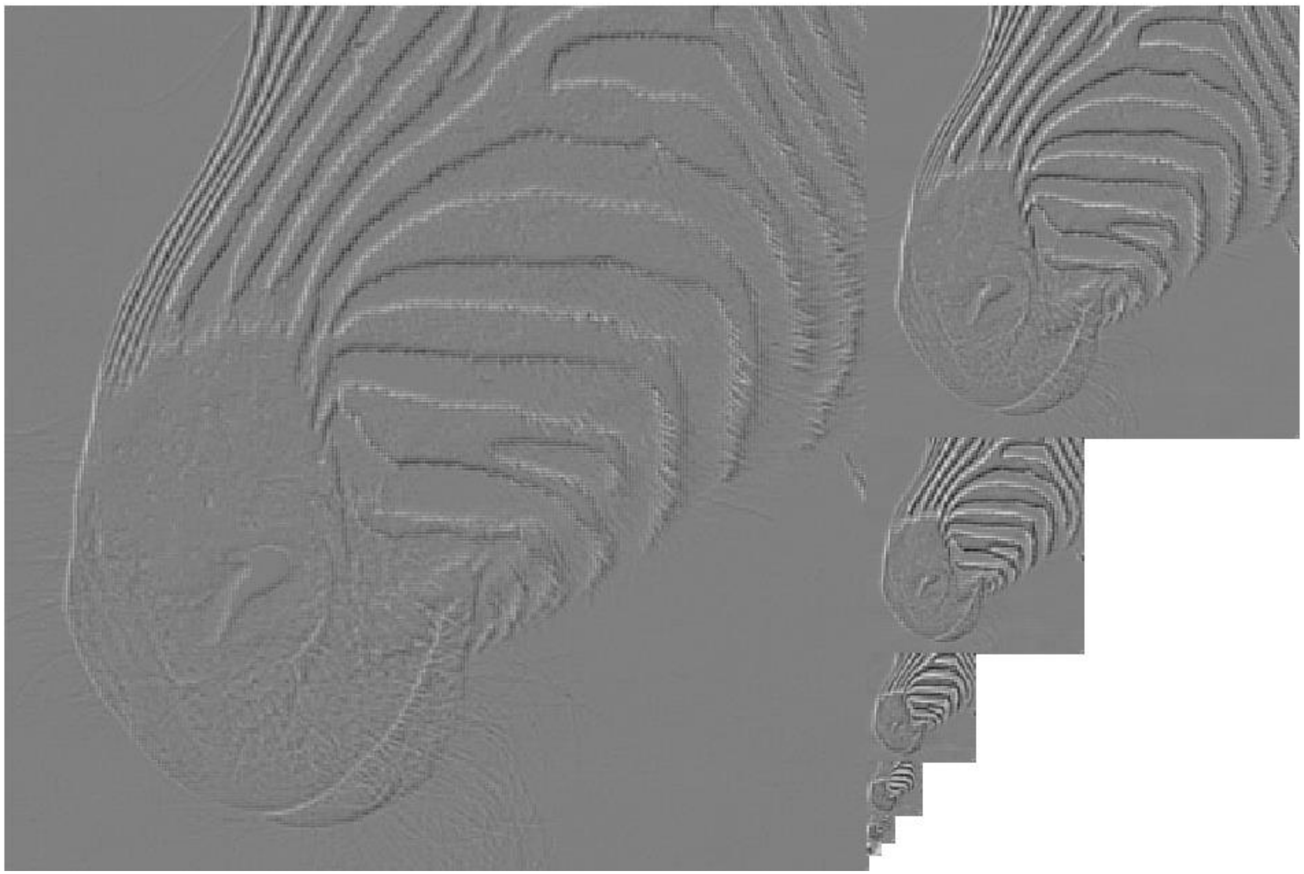
- Synthesis
  - $G_0 = L_0 + U(G_1)$
  - $G_1 = L_1 + U(G_2)$
  - $G_2 = L_2 + U(G_3)$
  - $G_3 = L_3$

Gaussian Pyramid



Laplacian Pyramid





Laplacian Pyramid for Compression