Image Pyramids - Overview

Key Concepts to Master

1. What are Image Pyramids?

Definition:

- Multi-scale representation of an image
- Series of images at progressively lower resolutions
- Each level is half the size of previous level

Why use them?

- Detect features at multiple scales
- Coarse-to-fine processing
- Efficient for hierarchical algorithms
- Handle scale invariance

2. Gaussian Pyramid

Construction process:

Level 0 (original):

```
Original image (size: M × N)
```

Level 1:

```
1. Apply Gaussian filter (smooth)
2. Subsample by factor of 2 (size: M/2 × N/2)
```

Level k:

```
Size: M/2^k × N/2^k
```

Algorithm:

```
For each level:
    1. Convolve with Gaussian kernel
    2. Downsample (keep every other pixel)
```

Properties:

- Each level is smoothed version of previous
- Removes high-frequency details progressively
- Used for: multi-scale analysis, image blending

3. Laplacian Pyramid

Construction process:

Requires Gaussian pyramid first!

For each level k:

```
L_k = G_k - expand(G_{k+1})
```

Where:

- G_k = Gaussian pyramid level k
- expand() = upsample and interpolate
- L_k = Laplacian level (difference image)

What it captures:

- Band-pass filtered information
- Details lost between pyramid levels
- Edge and texture information at each scale

Reconstruction:

```
G_k = L_k + expand(G_{k+1})
```

- Can reconstruct original from Laplacian pyramid
- Lossless representation (with proper expand)

4. Comparison: Gaussian vs Laplacian

Property	Gaussian Pyramid	Laplacian Pyramid
Content	Smoothed images	Difference images (edges)
Reconstruction	Cannot reconstruct original	Can reconstruct original
Information	Low-pass filtered	Band-pass filtered
Use case	Feature detection	Image compression, blending

5. Applications

Gaussian Pyramid:

- Multi-scale feature detection (SIFT)
- Fast image search
- Texture analysis

Laplacian Pyramid:

- Image compression
- Image blending (seam carving)
- Edge detection at multiple scales

Visual Example

Gaussian Pyramid (3 levels)

```
Level 0: [Original 512×512]
           ↓ (smooth + downsample)
Level 1: [256×256]
```

```
           ↓ (smooth + downsample)
Level 2: [128×128]
```

Laplacian Pyramid

```
L0 = G0 - expand(G1)    [512×512 detail]
L1 = G1 - expand(G2)    [256×256 detail]
L2 = G2                  [128×128 residual]
```

MATLAB Quick Reference

```
% Create Gaussian pyramid
impyramid(img, 'reduce');  % One level down
impyramid(img, 'expand');  % One level up (upsample)

% Manual Gaussian pyramid
function pyr = gaussian_pyramid(img, levels)
    pyr{1} = img;
    for i = 2:levels
        g = fspecial('gaussian', [5 5], 1);
        smoothed = imfilter(pyr{i-1}, g);
        pyr{i} = smoothed(1:2:end, 1:2:end);  % Downsample
    end
end

% Laplacian pyramid
function lpyr = laplacian_pyramid(gpyr)
    for i = 1:length(gpyr)-1
        expanded = imresize(gpyr{i+1}, size(gpyr{i}));
        lpyr{i} = gpyr{i} - expanded;
    end
    lpyr{end} = gpyr{end};  % Residual
end

% Reconstruct from Laplacian
function img = reconstruct(lpyr)
    img = lpyr{end};
    for i = length(lpyr)-1:-1:1
        img = imresize(img, size(lpyr{i}));
        img = img + lpyr{i};
    end
end
```

Study Checklist

- [ ] Understand purpose of image pyramids
- [ ] Know Gaussian pyramid construction
- [ ] Know Laplacian pyramid construction

- [ ] Can explain difference between them
- [ ] Understand expand/downsample operations
- [ ] Know reconstruction process
- [ ] Understand multi-scale representation concept

Common Exam Questions

Q1: How is Gaussian pyramid constructed?

- Smooth with Gaussian, then downsample by 2

Q2: How is Laplacian pyramid different?

- Stores differences (details) between levels
- Can reconstruct original image

Q3: What is the size at level k?

- Original: M × N
- Level k: M/2^k × N/2^k

Q4: Why use image pyramids?

- Multi-scale feature detection
- Handle different object sizes
- Efficient coarse-to-fine processing

Q5: Can you reconstruct from Gaussian pyramid?

- No (information lost in smoothing)
- Yes from Laplacian pyramid

Answers to Common Exam Questions

Q1: How is a Gaussian pyramid constructed? Start with the original image (level 0). For each subsequent level: (1) convolve with a Gaussian filter to smooth, (2) downsample by a factor of 2 (keep every other pixel in each dimension). Level k has size `M/2^k x N/2^k`. Each level is a blurrier, smaller version of the previous.

Q2: How is the Laplacian pyramid different? The Laplacian pyramid stores the *difference* (detail) between consecutive Gaussian levels: `L_k = G_k - expand(G_{k+1})`. It captures band-pass information (edges and textures at each scale). Unlike the Gaussian pyramid, the Laplacian pyramid can perfectly reconstruct the original image.

Q3: What is the size at level k? For an original image of size M x N, level k has size `M/2^k x N/2^k`. Example: 512x512 original -> level 1: 256x256 -> level 2: 128x128 -> level 3: 64x64.

Q4: Why use image pyramids?

- Multi-scale feature detection (objects/features appear at different sizes)
- Coarse-to-fine processing (fast approximate search at low res, refine at high res)
- Handle scale invariance in recognition tasks
- Efficient hierarchical algorithms (e.g., optical flow, template matching)

Q5: Can you reconstruct from a Gaussian pyramid? No. Gaussian smoothing discards high-frequency information permanently. You cannot recover the original from a Gaussian pyramid. **Yes from a Laplacian pyramid:** `G_k = L_k + expand(G_{k+1})`, working from the coarsest level back up to full resolution.

Related Topics

- Lecture 7: Image Pyramids
- SIFT uses scale-space (similar concept)
- Multi-scale edge detection