Image Pyramids

COMPSCI 527 — Computer Vision
Outline

1. Pyramids and Scale
2. (Spatial Frequency) Aliasing
3. Downsampling and Upsampling
4. Bilinear Interpolation
5. Gaussian (and Laplacian) Pyramid
Pyramids and Scale

- Scale:
  - Start with smallest template
  - Look for larger and larger occurrences
- Larger template $\approx$ smaller image!

↑ smallest denticle we look for
Scale Budgets

- $n \times n$ image, $k \times k$ template, scaling $s > 1$
- Processing a large image with progressively larger templates:
  $$n^2(k^2 + k^2s^2 + k^2s^4 + \ldots) = n^2k^2(1 + s^2 + s^4 + \ldots)$$
  - Series diverges
- Processing progressively smaller images with a small template:
  $$k^2(n^2 + n^2/s^2 + n^2/s^4 + \ldots) = k^2n^2(1 + 1/s^2 + 1/s^4 + \ldots)$$
  - Series converges to $k^2n^2s^2(s^2 - 1)$
  - For $s = 2$, the series converges to $k^2n^24/3$
  - About 33% additional cost relative to processing the original image alone
Finer Scales

- Scaling down by $s = 2$ every time may be overly aggressive.
- Let $\phi = 1/s$ be the *downsampling factor*.
- For $0 < \phi < 1$, image shrinks. For $\phi > 1$, the image grows larger.
- How to downsample ($0 < \phi < 1$)?
- Two issues: aliasing and non-integer $s$. 
Aliasing

- Even when $s$ is an integer, pure sampling is a bad idea: \textit{(Spatial frequency) aliasing}
- Colors are sampled at locations on the pixel grid
- Nothing to do with the scene

Original \hspace{1cm} \text{Sampled by } s = 30, \text{ then magnified by 30}
Downsampling = Smoothing + Sampling

- Smooth with a Gaussian blur kernel first, then sample

\[
J(x,c) = \sum_{i,j} G(i,j) I(i \cdot s + x, j \cdot s + c)
\]

\[
s = \frac{1}{4} = 30
\]

\[
J(30a, 30b) = \sum_{i,j} G(i,j) I(30a \cdot i, 30b \cdot j)
\]

- We lose detail (blur), but that's the whole point
- True scale: 
- Every pixel in the low-resolution image is a weighted average of pixel values in the original image
Key Questions

• How much to smooth before resampling?
  • That is, where does $\sigma = 48$ come from for $\phi = 1/30$?
  • Lots of theory for the optimal multiplier
  • Depends on various factors (spectral properties of image and noise)
  • We use what works most of the time, empirically
  • Answer: $\sigma \approx 1.6 s = 1.6/\phi$

• How to “take one out of every $s$ pixels” when $s = 1/\phi$ is not an integer?
Bilinear Interpolation

- What does it mean to “take one out of every $s$ pixels” when $s = 1/\phi$ is not an integer?

$$\xi = \lfloor x \rfloor, \quad \eta = \lfloor y \rfloor$$
$$\Delta x = x - \xi, \quad \Delta y = y - \eta$$

$$\begin{align*}
I(x) &= I(\xi, \eta) (1 - \Delta x) (1 - \Delta y) \\
&\quad + I(\xi + 1, \eta) \Delta x (1 - \Delta y) \\
&\quad + I(\xi, \eta + 1) (1 - \Delta x) \Delta y \\
&\quad + I(\xi + 1, \eta + 1) \Delta x \Delta y
\end{align*}$$
Abstracting Pyramid Operations

\[ J = \text{resize}(I, \phi) : \]

- If \( 0 < \phi < 1 \), image shrinks:
  - Filter with \( \sigma = 1.6 / \phi \),
  - then sample every \( s = 1 / \phi > 1 \) pixels
- If \( \phi \geq 1 \), image grows:
  - No filter. Just sample every \( s = 1 / \phi \leq 1 \) pixels

- Pyramid operators: Pick a \textit{single} value of \( \phi \in (0, 1) \), then define
  \[ \begin{align*}
  \text{down}(X) &= \text{resize}(X, \phi) \\
  \text{up}(X) &= \text{resize}(X, 1/\phi)
  \end{align*} \]

- \textit{up} is \textit{not} the inverse of \textit{down}:
  Cannot restore lost information
A Gaussian Pyramid \( (\phi = 1/2) \)

- A *lowpass* pyramid: Each level contains a subset of the lower spatial frequencies that are in the next-higher resolution level (blurring attenuates high frequencies)
A Laplacian Pyramid ($\phi = 1/2$)

- A \textit{bandpass pyramid}, because each level contains a (more or less) separate band of spatial frequencies
- The Laplacian pyramid is invertible
- \textit{Optional topic, see notes}