Sharpening Filters

Unsharp Masking and Highboost Filtering

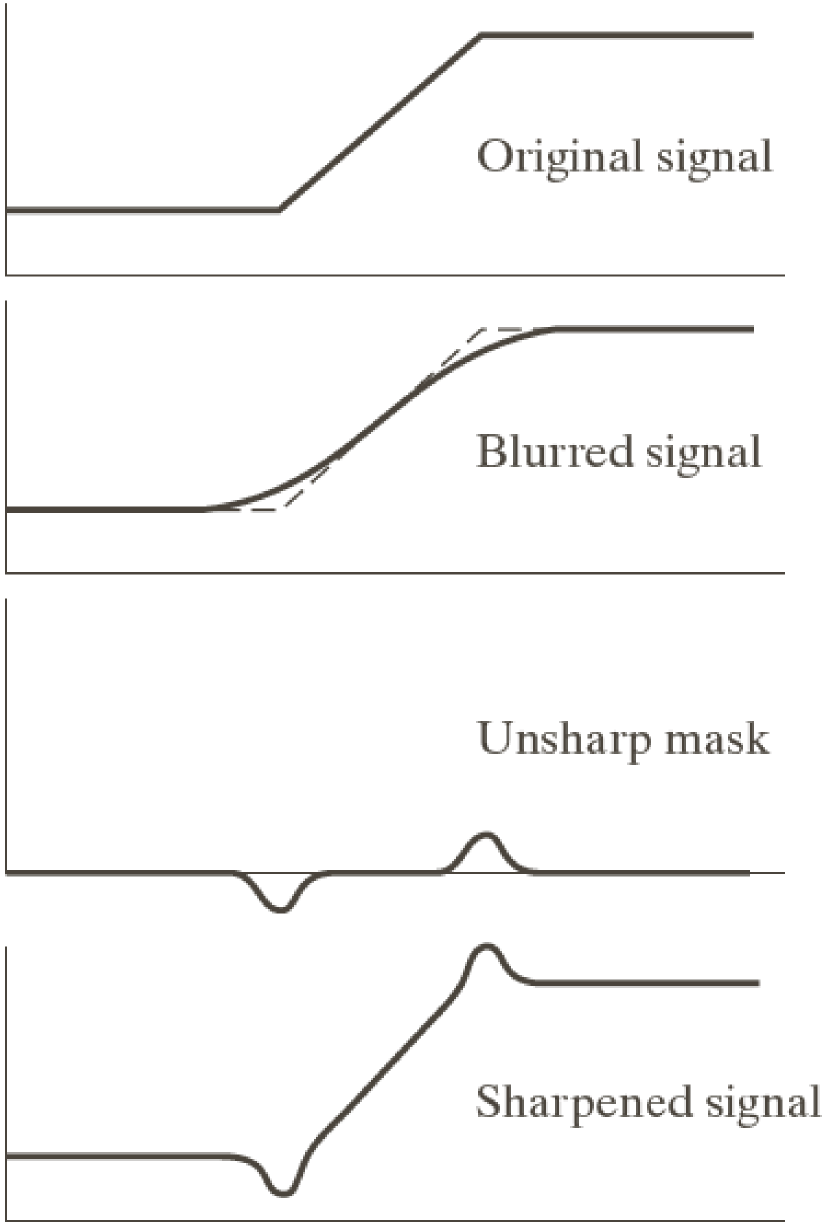
1. Blur the original image
2. Subtract the blurred image from the original (the resulting difference is called the mask)

$$g_{\text{mask}}(x, y) = f(x, y) - \overline{f}(x, y)$$

3. Add the mask to the original

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$

Mechanics of unsharp masking



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Unsharp Masking and Highboost Filtering

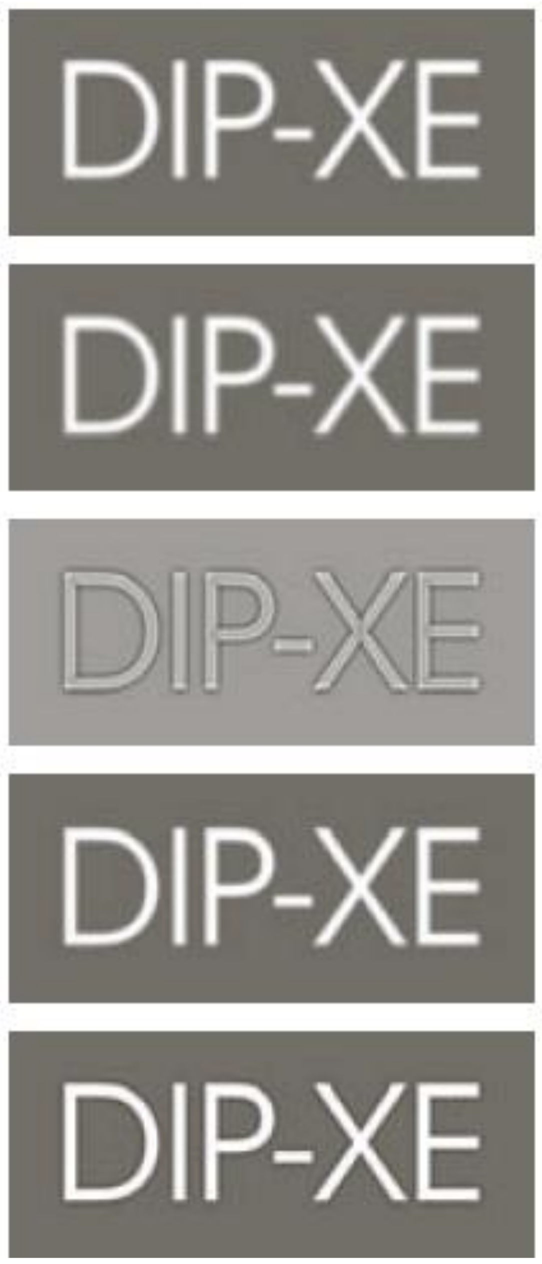


FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

Food for thought!

1. What is the main purpose of unsharp masking?
2. What are the main steps involved in unsharp masking?
3. What is the “mask” in unsharp masking?
4. How does high-boost filtering differ from unsharp masking?
5. Why is blurring performed before subtraction in unsharp masking?

Programming assignment

- Implement unsharp masking and high-boost filtering to enhance edges and fine details in a grayscale image.
- **Concepts Used**
    - Image blurring (low-pass filtering)
    - Unsharp masking
    - High-boost filtering
    - High-frequency components
    - Image sharpening
- **Tasks**
    - Read a grayscale image.
    - Blur the image using an **average or Gaussian filter**.
    - Compute the **mask** by subtracting the blurred image from the original image.
    - Generate the **unsharp masked image** by adding the mask to the original image.
    - Implement **high-boost filtering** using a boost factor $A > 1$.
    - Display the original, blurred, mask, unsharp masked, and high-boost images.
    - Briefly compare the sharpening effect of unsharp masking and high-boost filtering.

AI supported self-learning (Prompts compatible with ChatGPT)

Active Learners (Learning by Doing)

1. Provide a small grayscale matrix and guide me to manually perform unsharp masking: blur the image, subtract to obtain the mask, and add it back to the original.
2. Guide me step by step in writing a Python/OpenCV program to implement unsharp masking and high-boost filtering and compare the outputs.

Reflective Learners (Learning by Thinking)

1. Explain why blurring is performed before subtraction in unsharp masking and summarize the reasoning behind this approach.
2. Conceptually compare unsharp masking and Laplacian sharpening in terms of how edges are enhanced.

Sensing Learners (Concrete & Practical)

1. Use actual pixel values to demonstrate how the mask is generated in unsharp masking.
2. Provide a practical example where high-boost filtering is preferred over simple unsharp masking.

Intuitive Learners (Concepts & Patterns)

1. Explain unsharp masking as a method of isolating high-frequency components and enhancing them.
2. How does increasing the boost factor in high-boost filtering affect edge enhancement mathematically?

Visual Learners (Diagrams & Structure)

1. Show an original image, blurred image, mask image, and sharpened result, and visually explain each stage.
2. Illustrate graphically how subtracting a blurred image enhances high-frequency details.

Verbal Learners (Words & Explanation)

1. Explain unsharp masking using an analogy such as enhancing details by subtracting a smoothed background.
2. Describe the difference between unsharp masking and high-boost filtering in simple language.

Sequential Learners (Step-by-Step Logic)

1. Break down the unsharp masking algorithm step by step, from blurring to mask generation to final enhancement.
2. Explain step by step how high-boost filtering modifies the unsharp masking formula.

Global Learners (Big Picture First)

1. Explain the overall role of unsharp masking in the image enhancement pipeline before discussing mathematical details.
2. Provide a big-picture comparison of all sharpening methods: Gradient, Laplacian, and Unsharp Masking.