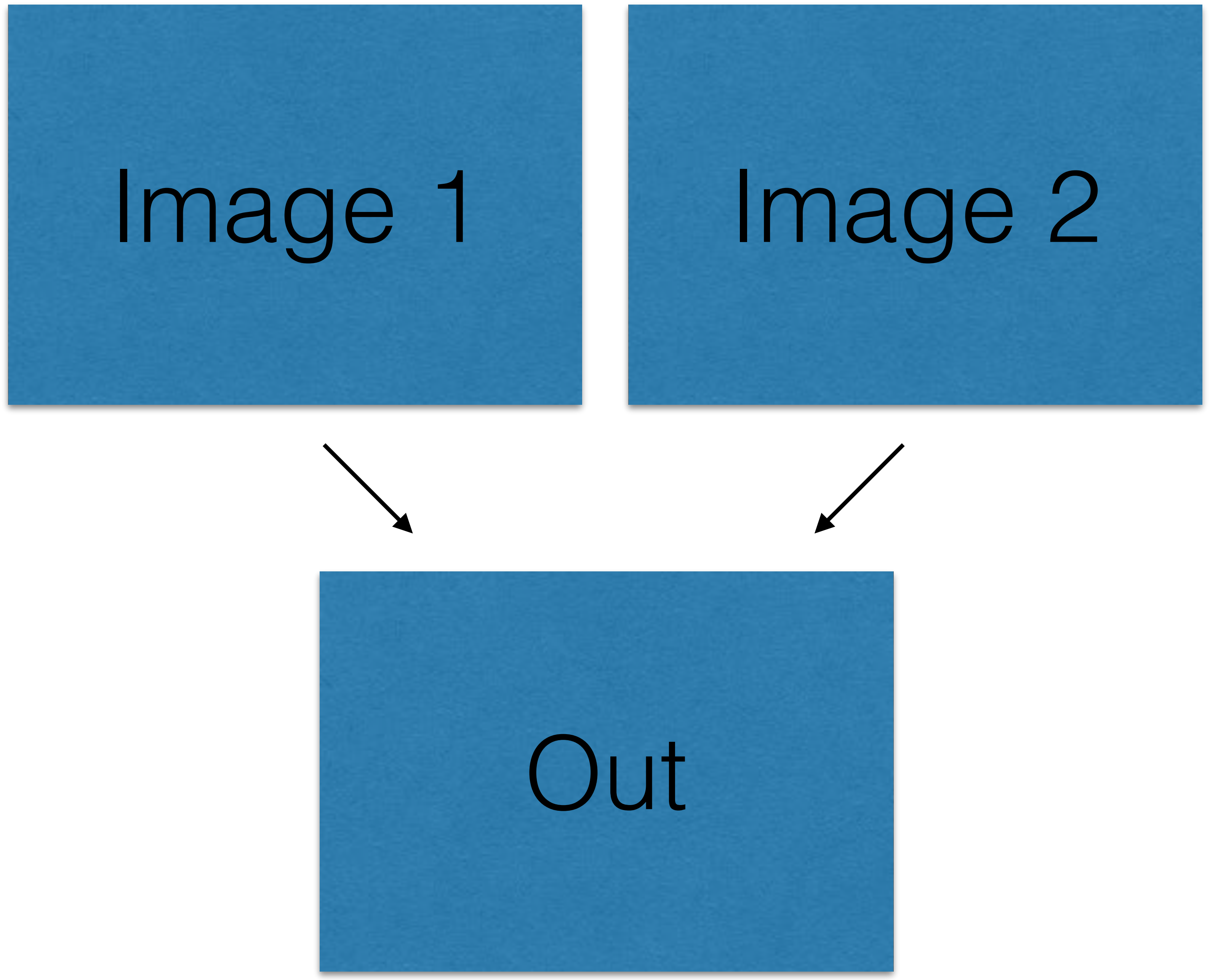
Blending, Differencing, and Masking

CS 355: Introduction to Graphics and Image Processing

Image Arithmetic

- Involve multiple images
- Apply a function pairwise (or more) to the pixels in the input images
- Possibilities:
  add, subtract, and, or, min, max, …

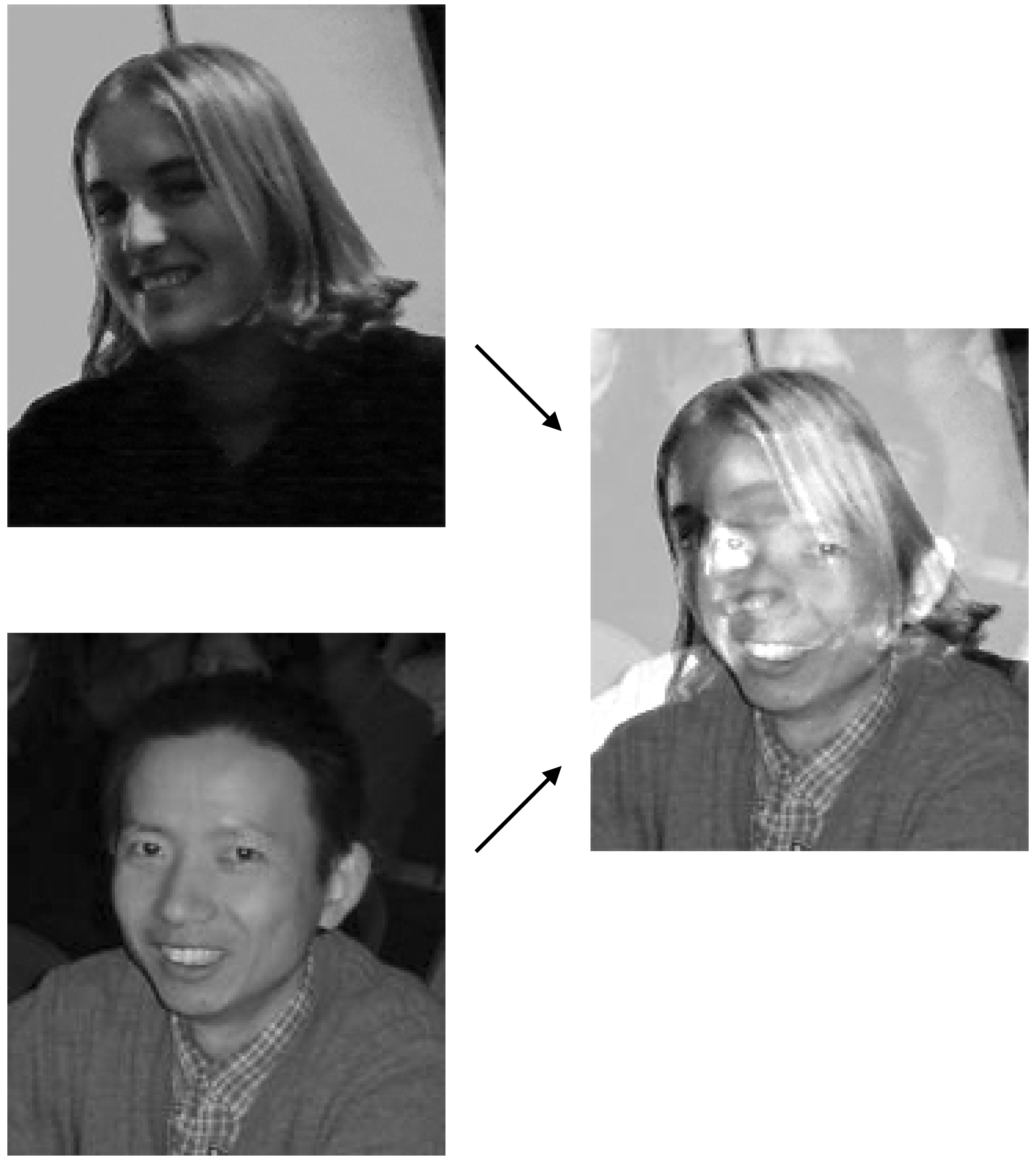


```
for all pixel positions x, y:
out[x,y] = func(in1[x,y],in2[x,y],…)
```

Addition

- Can be used for double exposures or composites

- Often a weighted blend



$$\text{out}(x,y) = \text{in}_1(x,y) + \text{in}_2(x,y)$$

$$\text{out}(x,y) = \alpha_1 \text{in}_1(x,y) + \alpha_2 \text{in}_2(x,y)$$

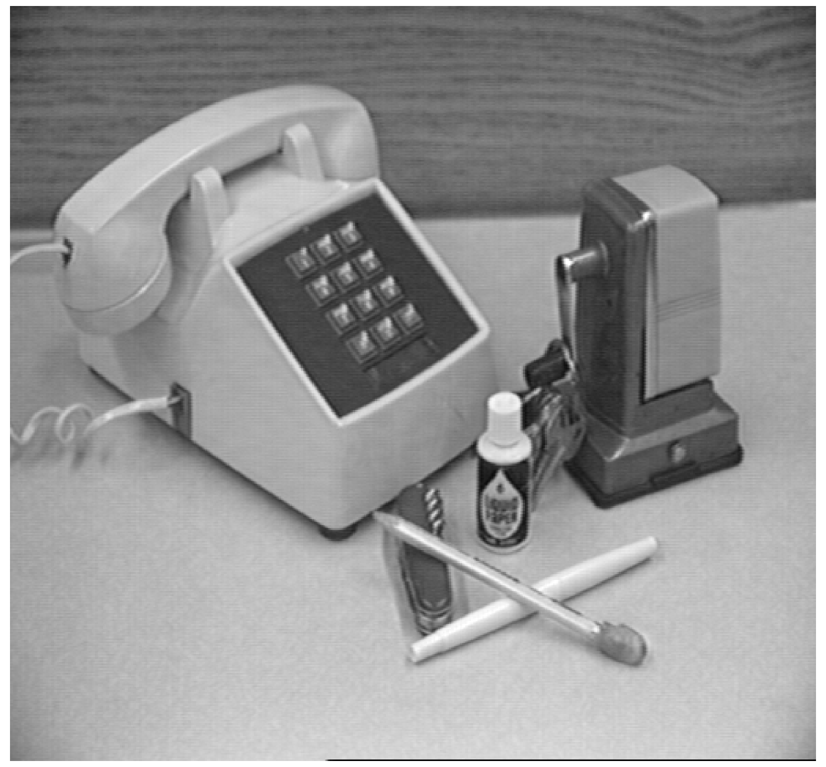
Subtraction

- Useful for finding changes between images

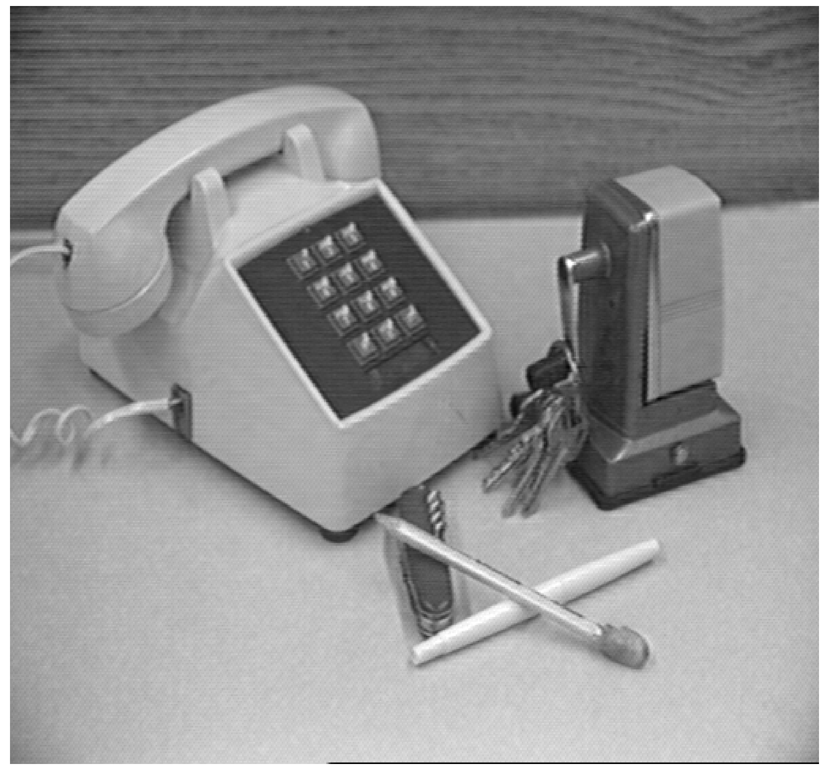
$$\text{out}(x, y) = \text{in}_1(x, y) - \text{in}_2(x, y)$$

- Often more useful to use *absolute difference*

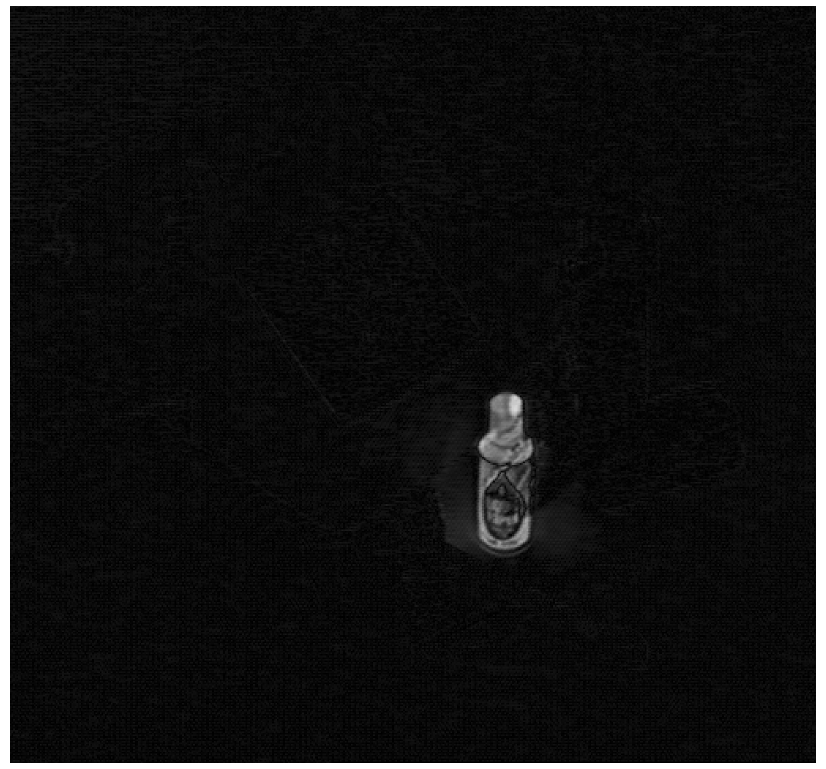
$$\text{out}(x, y) = |\text{in}_1(x, y) - \text{in}_2(x, y)|$$



-

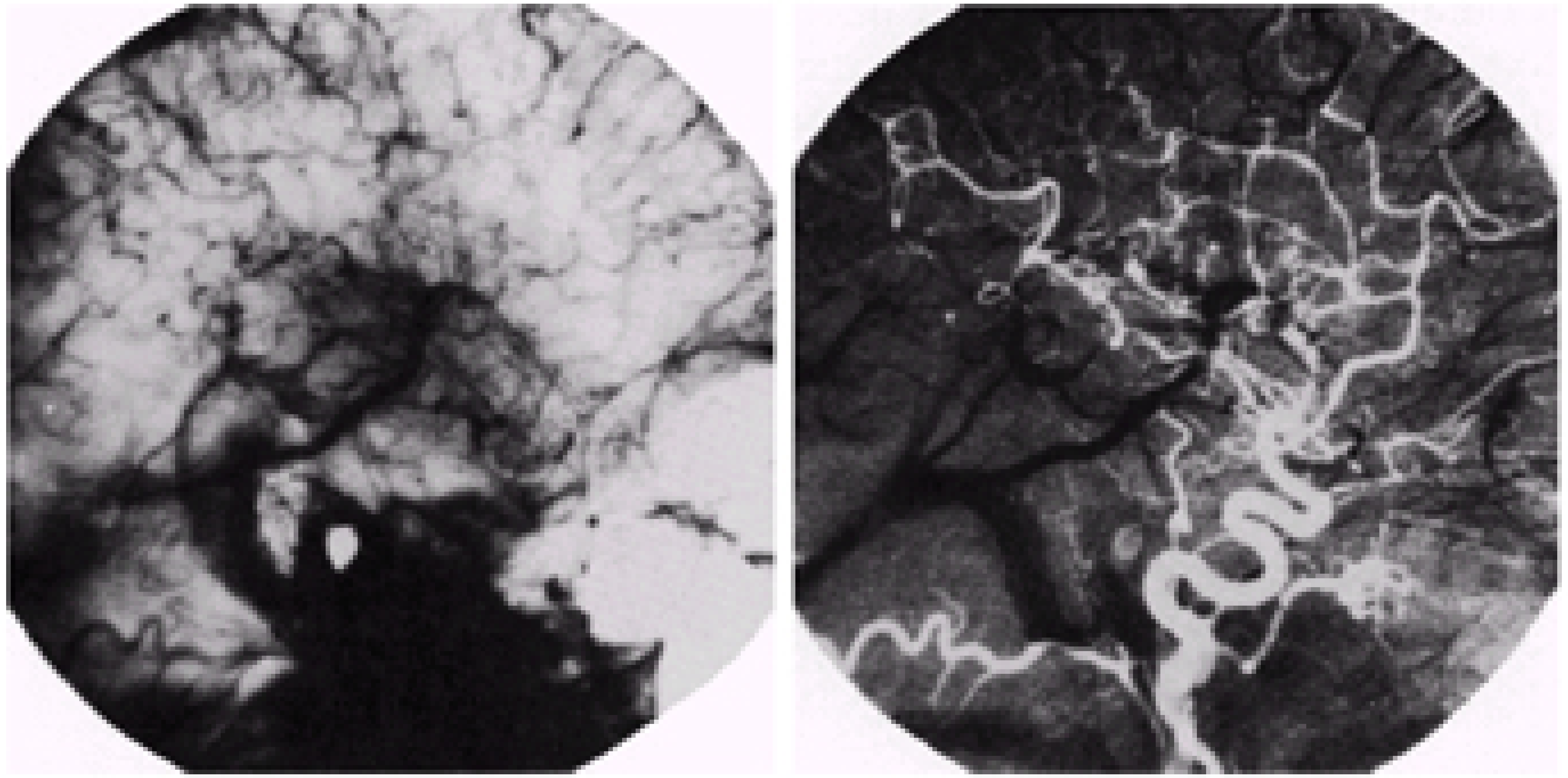


=



Digital Subtraction Angiography

1. Take an x-ray
2. Inject patient with radio-opaque dye (“don’t move!”)
3. Take another x-ray
4. Subtract the two

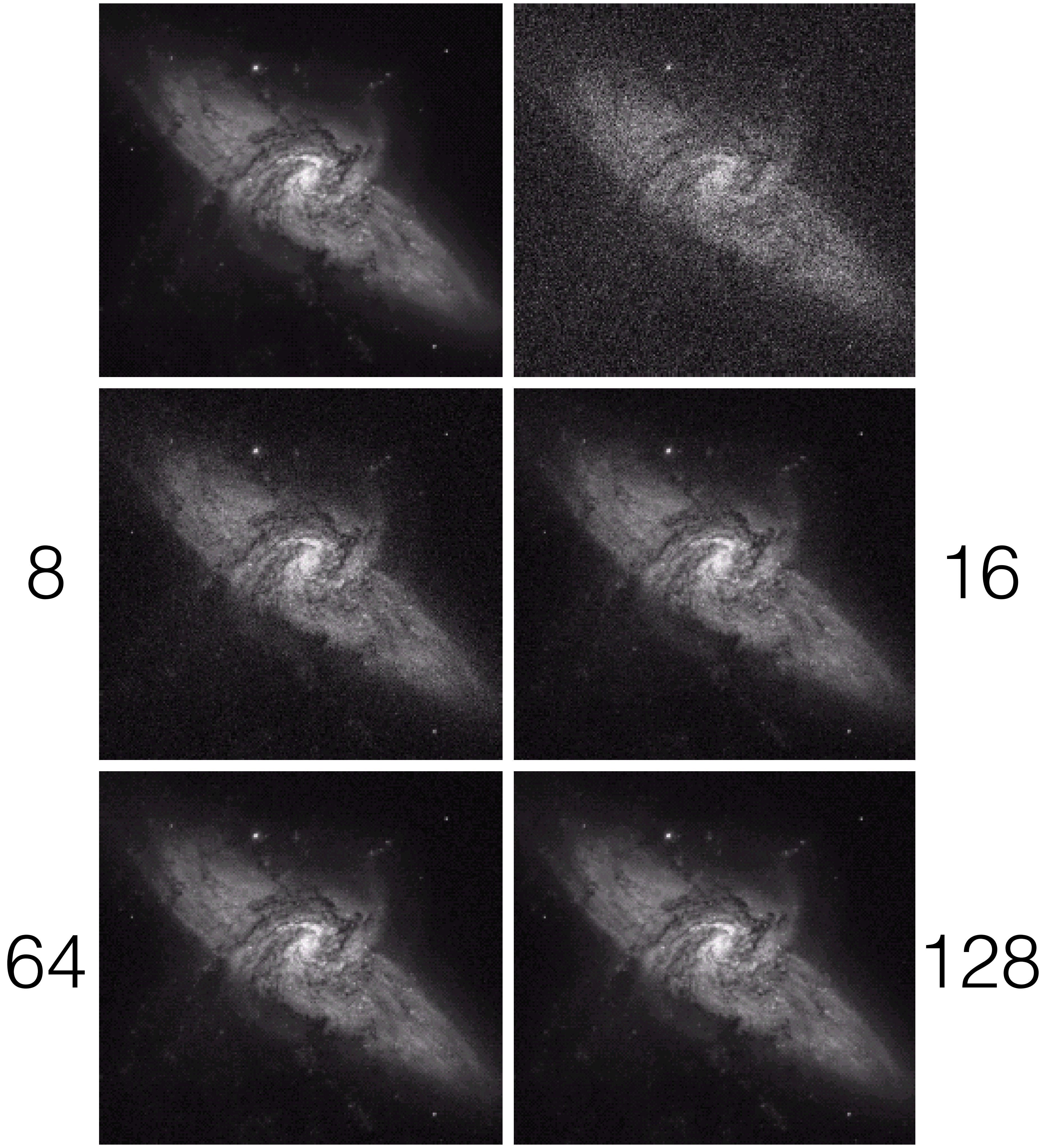


Motion

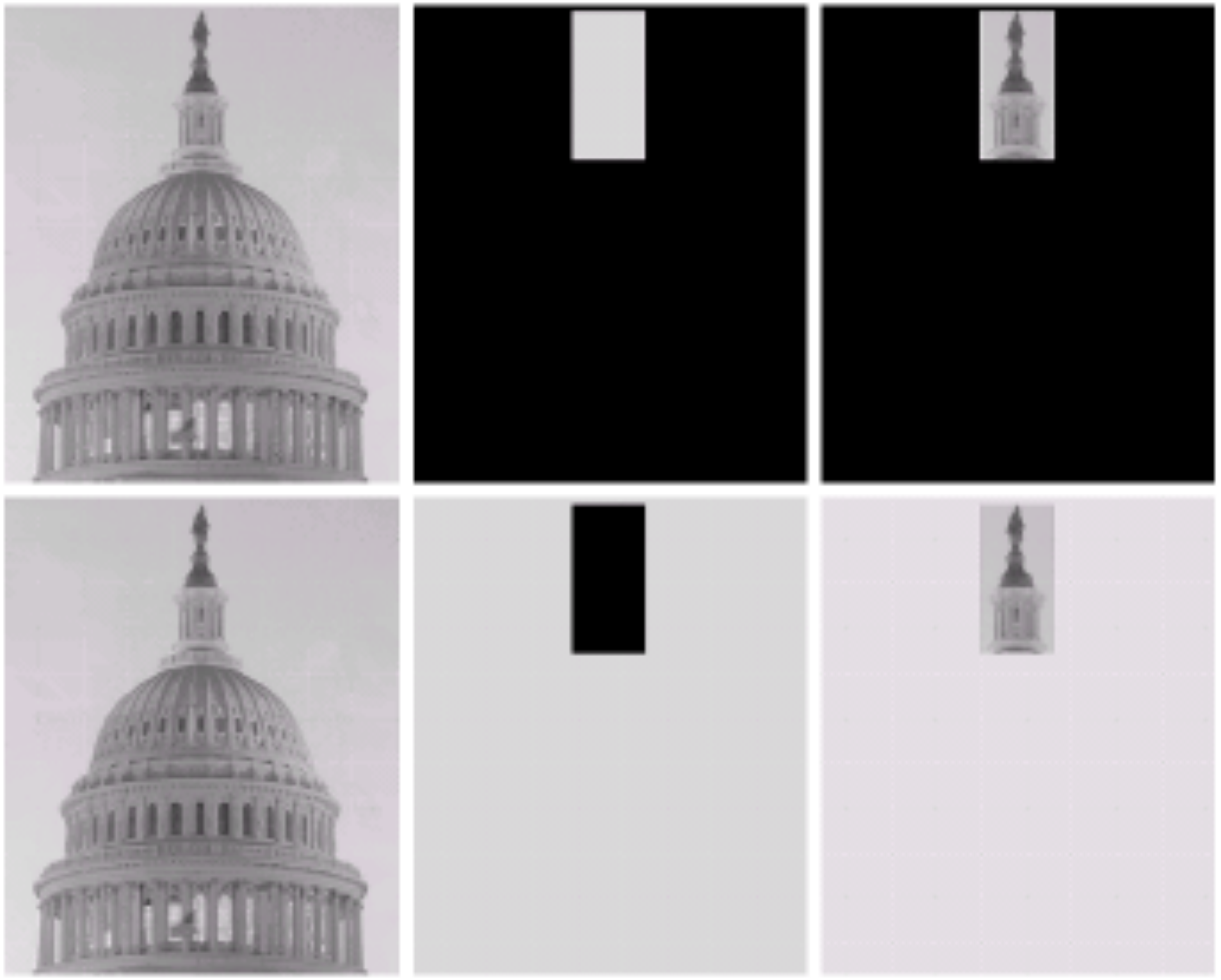
- Use differencing to identify motion in an otherwise unchanging scene (object motion, not camera motion)
  - Basis for motion tracking techniques in computer vision
- Use overall shift (minimum difference) for tracking camera motion
  - Part of a larger process called a “match move” in film making
  - Essential for inserting CGI into a real scene with a moving camera (the virtual camera has to move the same way the physical camera did)
- Useful for video compression
  - Only encode the difference between frames
  - Motion detection/prediction used in video compression (MPEG, etc.)

Image Averaging

- Average multiple pictures of the same static scene to reduce noise
- Similar in principle to acquiring the image for a longer duration



Bitwise AND and OR



Useful for masking

Alpha Blending

- Use *per-pixel weights* to blend two images:

$$\text{out}(x,y) = \alpha_1(x,y)\ \text{in}_1(x,y) + \alpha_2(x,y)\ \text{in}_2(x,y)$$

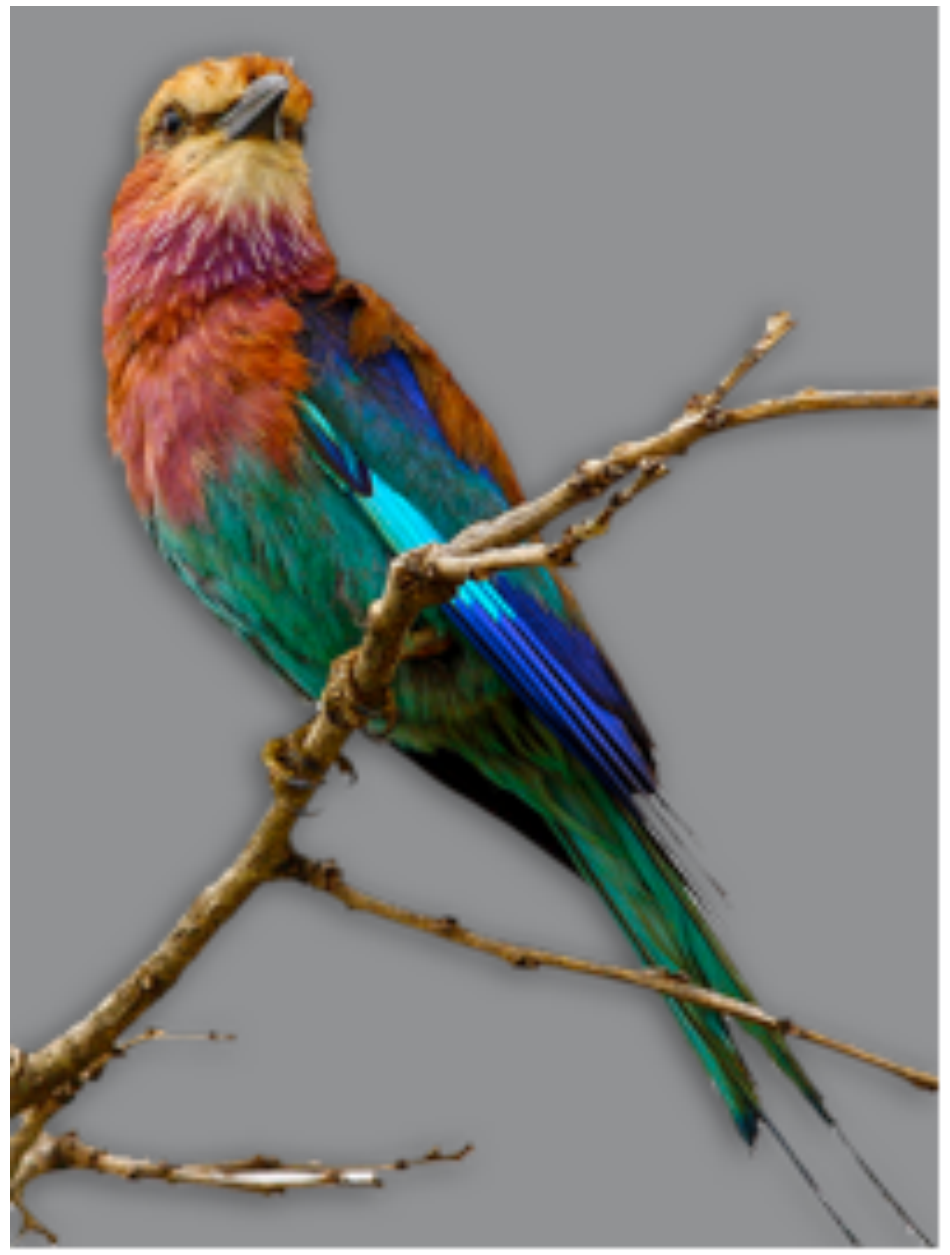
- Or most commonly:

$$\text{out}(x,y) = \alpha(x,y)\ \text{in}_1(x,y) + (1-\alpha(x,y))\ \text{in}_2(x,y)$$

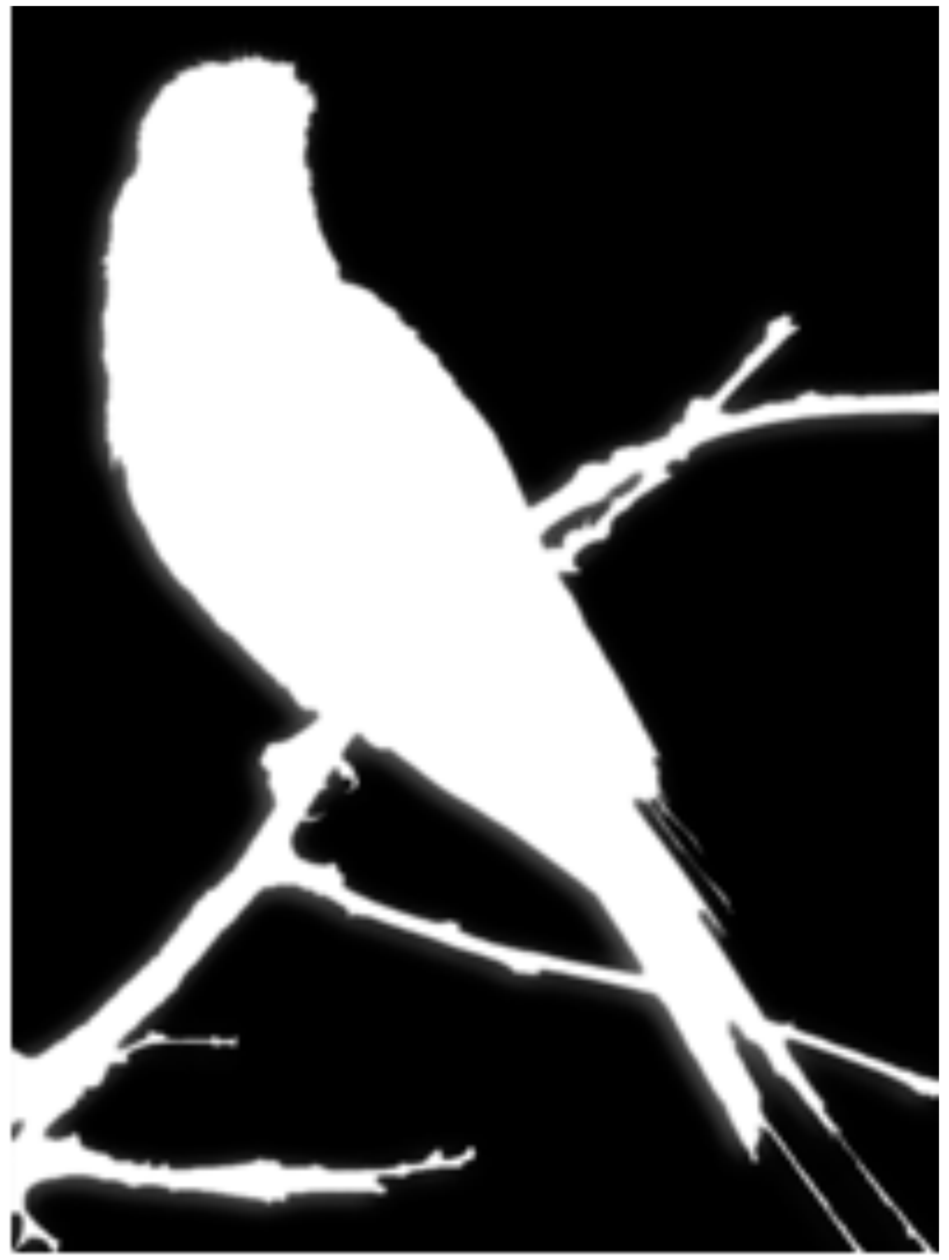
- Useful for transparency, compositing, etc.

Alpha Masks

- Blending often uses an *alpha mask*
- Sometimes also called a *matte*
- Often stored with image as an extra *alpha channel*
- 0 = transparent,
  1 = opaque



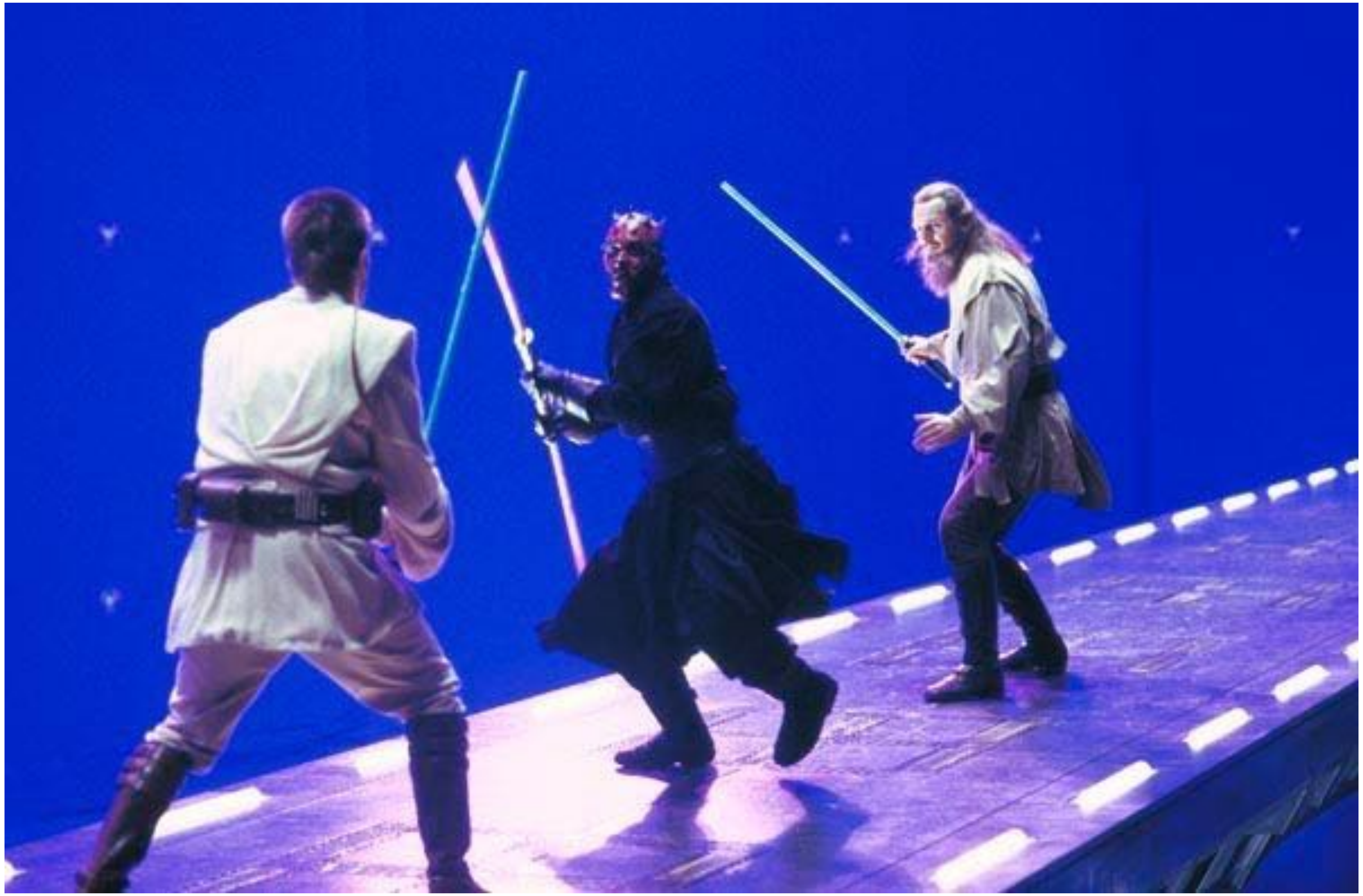
Source Image



Alpha Mask

$$\mathrm{out}(x, y) = \alpha(x, y)\ \mathrm{in}_1(x, y) + (1 - \alpha(x, y))\ \mathrm{in}_2(x, y)$$

Application: Blue Screening



Application: Blue Screening

- Film against blue (or green) background
- Mask out the blue parts
- Use fractional alpha values for partial-pixel effect
- “Decontaminate” the blue (or green) halo
- Store in RGBA format
- Composite onto background using alpha blending

Coming up…

- Neighborhood operations:
    - noise reduction
    - sharpening
    - edge detection