Homework 2

CST 205

Background

Analyzing a series of images of the (exact) same location at different times is known as *temporal processing*.[1] One example of a temporal filter is a **median filter**. Given several images, a *temporal* median filter takes a list of pixel channel values at each coordinate and calculates the median. This removes outliers and can act to subtract unwanted elements from an image.In general, to calculate the median of a list of numbers of *odd* length, you do the following:

1. Sort the list
2. Calculate the middle location
3. Return the value at that middle location

If we have a list of length n, where n is odd, the location of the middle value is at $\frac{n+1}{2}$.

For example, if we have the numbers 2, 4, 6, 237, 1, we do the following:

1. Sort the list → 1, 2, 4, 6, 237
2. Calculate the middle location → $\frac{5+1}{2} = 3$
3. Return the value at that middle location → 4

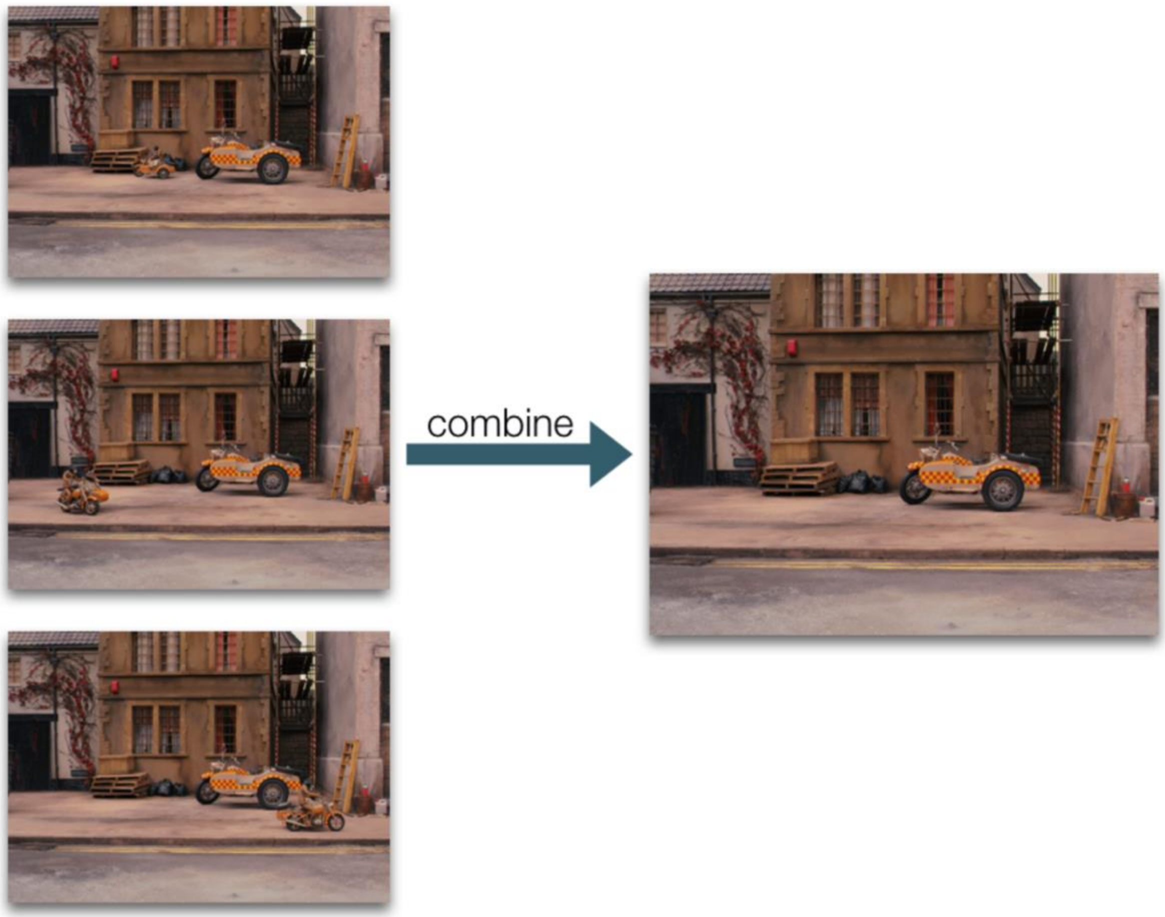
Note: Given that Python lists are indexed at 0, we need to subtract one from the calculated middle location in our program.

Task (20 points)

Imagine you are a motorcycle aficionado and spot a stunning retro sidecar motorcycle on a sidewalk. You are in a hurry, but quickly take multiple pictures of it, only to notice back home that all of your images are ruined — by a fox and opposum on a mini-motorcycle! The *fox/opposum duo* rode through all of the 11 images that you took. These images are provided on iLearn and will also be provided on Slack.

[1] Temporal processing happens over time whereas **spatial processing** analyzes one image and focuses on each pixel's neighbors.

Luckily, you know Python (and Pillow) and can write a median filter program to remove the fox/opposum mini-motorcycle. The basic idea is presented below (though you will use **ALL** of the provided images):



Note: Your program may be tested on different images, but you can assume that your program will be tested on an odd number of images. (**Do not assume that the test images will have the same dimensions.**)

A few more details

- Place the provided images in a sub-directory called `hw2_images`
- Your function to calculate the median should be called `median`

    - Do not use median functionality from packages such as numpy or `statistics`.
- Use glob to load the images. You can follow the example here.
- Your final image should be called `output.png`

Deliverable

Submit all source code files, your final image, and a screenshot of your screen with your final image showing (use Pillow's `show()` method). All source code files should include header comments. **Do not submit the source images.**

All images courtesy of *Fantastic Mr. Fox*, 2009, Twentieth Century Fox