Video Processing

Final Project Documentation

•••

~ Noam Mansur ~ Yossi Elman ~

Video Stabilization

The process of video stabilization consists of the following flow:

1. Detect the interest points of the first frame using the detectFASTFeatures function.
2. Extract descriptors from each interest point using extractFeatures
3. Initialize the geometric motion model to the Identity Matrix
4. Repeat for each new frame:
    a. Detect the interest points from the new frame
    b. Extract descriptors
    c. Match the descriptors of the new frame to the previous one using matchFeatures
    d. Update the geometric motion model using estimateGeometricTransform
    e. Warp the new frame according the updated model.

Video Stabilization

Parameters and notes:

1. The parameter resizeFactor can be used to reduced the size of the frames and increase performance. The recommended value is 0.5.
2. The parameter ptThresh is used to determine the threshold from which each point is considered a corner. We fixed this value to be 0.1.
3. The model we used for the geometric transformation is 2D affine.

Background Subtraction

The process of background subtraction consists of the following flow:

1. Go over all the frames the separate each one into 3 RGB channels, and store each frame as grayscale.
2. Calculate the background of each frame by calculating a median of window in time, using movmedian.
3. For each frame in grayscale
    a. Apply a gaussian filter on the current frame and current background to smooth the edges (and avoid big distance from each other)
    b. Calculate a difference image from the current frame and the current background
    c. Binarize the difference image using the threshold given
    d. Isolate the figure from the binary image (explained in a different slide)
    e. Use the clean binary image to output re-construct the RGB frame.

Background Subtraction

Figure isolation

We used a low threshold to binarize the difference image (current frame minus the current background), in order to keep the shape of the figure, therefore big amount of noise needed to be cleaned - using the following steps:

1. Apply a first vertical median filter (to keep the overall vertical shape of the figure)
2. Detect the largest connected object in the frame - the person
3. Replace with zeros all the pixels outside of the object
4. Apply a second vertical median filter to detach any connected noise from the figure
5. Remove all connected objects smaller than the size of the object using bwareaopen
6. Apply a final gaussian filter to smooth the edges caused by the median filters.

Background Subtraction

Parameters and notes:

1. We used 2 a different thresholds to binarize the difference image at the first the last frames, since the background on those frames is based on less relevant frames and therefore less accurate.
2. The WindowSize parameter controls the length of the window upon the **median** over time is calculated (a large value will consider more frames of the video to determine the background)

Matting

The process of matting consists of the following flow:

1. Go over the frames in the binary video until you find a figure (This is to protect from the case that the figure only appear after a few frames)
2. Randomly sample coordinates of the background and the figure, these coordinates will be our scribbles
3. Use ksdensity to find the densities map of the foreground and background
4. For each frame in the binary video,
    a. Find the perimeter of the figure using bwperim and widen it using imdilate
    b. Calculate alpha map as seen in class
    c. Use alpha map to construct the new frame
    d. Write the new frame to file

Tracking

The process of tracking was similar to what we've done at exercise 3 and consists of the following flow:

1. Initialize all particles to the first location as given by the user
2. Calculate the signature of the object by creating an histogram, quantifying it and create a 4096-long array of integers as the signature of the object
3. Predict the particles (add noise) and calculate for each one its weight by computing the cosine similarity to the object's signature
4. For each frame:
    a. Sample particles, and calculate weights
    b. Add to the frame a boundary box for the highest weighted particle and the average particle
    c. Write the frame to file

Tracking

Parameters and notes:

1. At the start of the process the user is shown the first frame of the video and is asked to select the object he wants to track by selecting a rectangle. This selection will be used for the calculation of the object's signature.
2. The NumberOfParticles parameter is used to select the number of particles, where the larger the number we expect to see better results but the process will be more time consuming.
3. Through the video we don't expect any change in the object's width and height so these values remain the same for all particles.

GUI - RUNME.m

The values in the text box when starting the GUI are the default values for our input video