CSL462 - Collage using Hybrid Images (Assignment 1)

Shivam Mittal
www.linkedin.com/in/shivam-mittal

Indian Institute of Technology,
IIT Ropar

The main focus of the project was to create a collage from a given set of input images. The concept of hybrid images was taken because we did not want sudden transition between images in the collage, instead we wanted a smooth transition between images, so the overlapping area should contain information from both the images on either side of the boundary.
Also, hybrid image serve another purpose, the interpretation of the image changes with the viewing distance. We can see in the results that the overlapping area shows this property to some extend as it combines the low-spatial frequency of one picture and high-spatial frequency of the other. A pseudo low pass filter (and correspondingly high pass filter) was created by creating a pseudo center for the 2-D gaussian filter. We will describe the process and ideology simultaneously.
The collage can be built for any number of images, and each time it is run, a different collage might be generated for the same set of images. This is done because one collage might be visually appealing to one person but a different structure might be more appealing to another person. So there is no fixed structure that is visually the best. So we have kept randomization in our collage structure building and hence different collages can be built by running the procedure multiple times.

1 Spatial structure

We have a canvas, and it can be dissected into 2 pieces either horizontally or vertically. Then we would have 2 non-overlapping canvases and we can divide the number of images that we have to place in both them. This divide and conquer strategy can be used until we have a canvas and the number of image to place in it is only 1. (This strategy has been inspired from a research paper PicWall.)
We will generate a random binary tree using this concept and then map our collage layout to this structure. The tree would contain N leaves corresponding to N images. There would be N-1 internal node (Since a node can have only 0 or 2 children), where each internal node would be labelled 'H' or 'V' depending on whether it represents a horizontal dissection or a vertical dissection. At any given point of time, there will be equal probability that the dissection may be horizontal or vertical. We would use a heuristic later to ensure that the resultant collage formed doesn't have all images appended vertically or all images appended horizontally.

We would be maintaining he aspect ratio of all the images. So once a tree is generated, we have aspect ratio information at the leaves (images). We would calculate the aspect ratio of the final tree in a bottom-up fashion. Basically the aspect ratio for a internal node can be calculated by its left child and right child according to given formula:

If the internal node is a vertical cut:

$$ar_{parent} = ar_{left} + ar_{right}$$

If the internal node is a horizontal cut:

$$1/(ar_{parent}) = 1/(ar_{left}) + 1/(ar_{right})$$

Using these formulas, we recursively calculate the aspect ratio for the whole image in a bottom-up format. We have preset the width of collage to be 800px, now the height of the collage would be

$$height = width(800px)/(aspect.ratio.of.the.root.of.tree)$$

. Since, the dissection are selected randomly, the aspect ratio can be extreme, either very large representing all image appended horizontally or very small representing all images appended vertically. To prevent such collages, we run 100 iterations and build different binary tree until the aspect ratio is between 0.8 and 1.8 which gives visually aesthetic result which is seen empirically.

We have calculated the aspect ratio for the image. We will now calculate the positions for each image i.e. where each image will be placed in the new image. We set the position of the root of the tree to be (x,y) = (1,1). We have already calculated the height and width for the root. Now the position, height and width for the other internal node and the leaf nodes is calculated recursively in a top down manner using the formulas given below:
If the parent node is a vertical cut:

$$node_{height} = parent_{height}$$

$$node_{width} = node_{height} * node_{ar}$$

$$node_x = parent_x + parent_{width} - node_{width}$$

$$node_y = parent_y$$

If the parent node is a horizontal cut:

$$node_{width} = parent_{width}$$

$$node_{height} = node_{width}/node_{ar}$$

$$node_x = parent_x$$

$$node_y = parent_y + parent_{height} - node_{height}$$

We will store this position, height and width information in a 2-D matrix names boxes which will be used to build the final image.

2 Collage Wrapper

The main wrapper function (Collage_Wrapper_2015CSB1032) takes as input a directory. It loads all the *.jpg images in the folder in an array images. The main spatial structure placing all the images in the final image is built by generating a binary tree using the procedure described above. Now we have positions of images such that no two images overlap. We scale up each image by a factor of 0.06 so that each image overlaps with multiple images so that we can create a smoothing effect using hybrid images.
We make a 2-D array global_intersection_area of the size of the collage. It is a binary array containing all 0, just ones at places where there is overlap of images. This is done to smooth these area in the end.
We will resize all the individual images to the size as dictated by the binary tree structure. We taper the edges of all individual images using gaussian filters so that the sharp edges are somewhat removed.
We run a loop on each individual image, and keep on placing the image on its respective position in the new image. To do this, we create a hybrid image from the currently formed image and the current image in the iteration. To create the hybrid image, we pass the position information (generated from the binary tree) of the current iteration image , and the intersection_area i.e. the overlap between the currently formed image and the current iteration image. The information is passed in such a format and not the bounding box format given in the question statement because the current iteration image (lets say x) may have overlap with currently formed image in more than one rectangular area and this information is impossible to pass through the bounding box concept, hence we introduce the intersection_area concept. The procedure of creating the hybrid image will be explained in a later section. But now, lets say we have our resultant image. Through this procedure, we will place all our individual images on a initially blank image.
We have maintained the 2-D array global_intersection_area which contains the overlap region of interest. We will dilate this mask global_intersection_area and then apply averaging filter on these region of interest in the resultant image to obtain a further smoothed result. The final image is saved as collage.jpg.

Figure 1: College containing 5 images

Figure 2: College 1 for 11 images

3 Create Hybrid Image

The call to this function can be in 2 formats. One is the format of bounding boxes that was given in the question statement, lets call this case 1. For this format, we have assumed that the height of the two bounding boxes should be same, otherwise there would be an error. To handle general cases, and our use case, as described above, we have kept two more parameters, one is the intersection_area containing the overlap between Image1 (large image) and Image2 (to be placed in Image1) and placement information of Image2 in Image1.
The concept of creating the hybrid image is same for both. We create a gaussian filter as a low pass filter. We set the center of the gaussian distribution as the center of the resultant images in case 1. We set the center as the center of the placement of Image 2 in case 2.
For both the cases, we set

$$2 * sigma^2 = centerI^2 + centerJ^2$$

We'll iterate through the whole new image. For the areas where there is no overlap, the resultant image will contain the pixel information from one of the images. The area where there is an overlap, let's say we are at index (i,j), the pixel intensity there would be given by:

$$resultant(i, j) = weight * Image1(x1, y1) + (1 - weight) * Image2(x2, y2)$$

where the weight is:

$$weight = e^{-(((i-centerI)^2+(j-centerJ)^2)/(centerI^2+centerJ^2))}$$

The effect is that a low-pass filter is created for Image1 and a high-pass filter is created for Image2, and we get a resultant hybrid image.

4 Results

Through this experiment, we can generate a collage for any number of images. Since, we used a randomized algorithm, multiple collages for the same set of images can be generated upon different executions. The individual images are overlapped with each other to create hybrid images. The visual results can be seen, multiple collages for the same set of images are shown. We can't measure our results quantitatively as designing a heuristic for visual appeal won't be justifiable.

Figure 3: College 2 for 11 images

Figure 4: College 3 for 11 images