# CSCE448 Project 2

Lam Nguyen

February 12, 2023

## 1 Task 1: Exhaustive Method

In the first task, an exhaustive algorithm was implemented to find all possible shifts and compare them to find out which shift was the best using the sum of squared differences (SSD) between the shifted image and the reference image. In addition, in order to avoid errors in calculations due to the borders of the image, a subset of an image was considered when finding SSD. The algorithm goes as follows:

**Pseudocode**:

**Algorithm 1** Exhaustive Approach

```
function find_shift (im1, im2):
    for (tx = -20 to 21) do
        for (ty = -20 to 21) do
            shift = circ_shift(im1, [tx, ty])

            score = np.sum((im2[20:-20, 20:-20] - shift[20:-20, 20:-20])^2)
            if (score < best_score) then
                best_score = score
                best_shift = shift
            end
        end
    end
    return best_shift
```

When running the algorithm above with our 4 jpg images, we get the following:

Figure 1: Monastery Image with Shift of R: [12, 3] G: [5, 2]

Figure 2: Monastery Image with Shift of R: [3, 2] G: [-3, 2]

Figure 3: Nativity Image with Shift of R: [7, 1] G: [3, 1]

Figure 4: Settlers Image with Shift of R: [14, -1] G: [7, 0]

## 2 Task 2: Image Pyramid

Exhaustive method is effective in aligning our images for .jpg files as seen in the figures above, but it is too naive when it comes to handling images of higher resolutions such as ones found in .tif files. An alternative solution involves using image pyramids to resize the original images by a scale and find the best shift at that scale and use it as a starting point for the next scale and repeat until the finest scale is found. The algorithm goes as follows:

**Pseudocode**:

**Algorithm 2** Image Pyramid Approach

```
function find_pyramid (im1, im2):
    pyramid_levels = 4
    offset = 20 // Used to ignore the border

    search_range = 20
    current_shift = [0,0]

    for (level = pyramid_levels to 0) do
        Resize the images by using skimage.transform.resize

        Shift the resized images by the current shift

        Find the best shift using Algorithm 1 given the resized images, a search range, offset, and the
          current shift. The current shift is used to only search around this initial translation vector.

        Set the current shift to the best shift and scale by 2 for next level. Scale the offset by 2 and
          downscale the search range by 2
    end
    return best_shift
```

When running the algorithm above with our 9 tif images, we get the following:

Figure 5: Emir Image with Shift of R: [105, 40] G: [49, 24]

Figure 6: Harvesters Image with Shift of R: [123, 14] G: [60, 16]

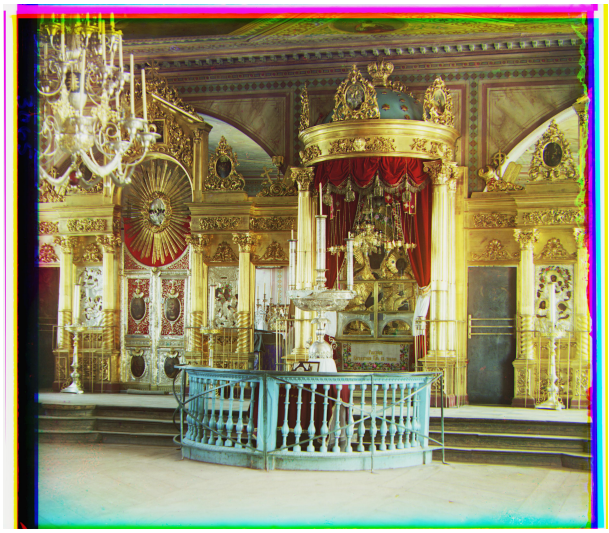

Figure 7: Icon Image with Shift of R: [88, 23] G: [41, 17]

Figure 8: Lady Image with Shift of R: [110, 12] G: [56, 8]

Figure 9: Self Portrait Image with Shift of R: [175, 36] G: [79, 29]

Figure 10: Three Generations Image with Shift of R: [110, 10] G: [53, 13]


Figure 11: Train Image with Shift of R: [86, 30] G: [42, 3]

Figure 12: Turkmen Image with Shift of R: [113, 26] G: [56, 19]

Figure 13: Village Image with Shift of R: [137, 21] G: [65, 12]