1 Theory

1. Let the basis $\mathscr{B}_1$ be the standard basis, i.e., $\mathbf{e}^{(1)} = (1\,0)^T$, $\mathbf{e}^{(2)} = (0\,1)^T$, and the basis $\mathscr{B}_2$ be given by the two vectors $\mathbf{v}^{(1)} = (1\,1)^T$, $\mathbf{v}^{(2)} = (-1\,1)^T$. Given $\mathbf{u}_{\mathscr{B}_1} = (1\,1)^T$, find $\mathbf{u}_{\mathscr{B}_2}$.

2 Computing

1. Write a code to generate 1000 random numbers contained on the unit circle. Apply several random matrices to this data and describe your results in the terminology of bases and change of bases. How do your results differ if the multiplying matrix is constrained to be orthogonal? Be sure to present the visual results in your report.

2. Given an algorithm [1] for computing small principal angles between two subspaces given by the real matrices X and Y, where X is in $\mathbb{R}^{n\times p}$ and Y is in $\mathbb{R}^{n\times q}$ (Principal angles are defined to be between 0 and $\pi/2$ and listed in ascending order):

Input: matrices X (n-by-p) and Y (n-by-q).
Output: principal angles θ between subspaces $\mathscr{R}(X) = \mathscr{X}$ and $\mathscr{R}(Y) = \mathscr{Y}$.

(a) Find orthonormal bases Q_x and Q_y for $\mathscr{X}$ and $\mathscr{Y}$ such that

$$Q_x^T Q_x = Q_y^T Q_y = I \quad \text{and} \quad \mathscr{R}(Q_x) = \mathscr{X}, \mathscr{R}(Q_y) = \mathscr{Y}.$$

(b) Compute SVD for cosine: $Q_x^T Q_y = H\Sigma Z^T$, where $\Sigma = diag(\sigma_1, \ldots, \sigma_q)$.

(c) Compute matrix

$$Y = \begin{cases} Q_y - Q_x(Q_x^T Q_y) & \text{if}\, rank(Q_x) \geq rank(Q_y); \\ Q_x - Q_y(Q_y^T Q_x) & \text{otherwise.} \end{cases}$$

(d) SVD for sine: $[H, diag(\mu_1, \ldots, \mu_q), Z] = \text{svd}(Y)$.

(e) Compute the principal angles, for $k = 1, \ldots, q$

$$\theta_k = \begin{cases} \arccos(\sigma_k) & \text{if} \quad \sigma_k^2 < \frac{1}{2}; \\ \arcsin(\mu_k) & \text{if} \quad \mu_k^2 \leq \frac{1}{2}. \end{cases}$$

(a) Implement this algorithm in MATLAB under the function name prinAngles. Your function should have the input and output arguments:

[theta] = prinAngles(X1,X2)

where theta is the principal angles listed from the smallest to the largest, X1 is the first set of images listed by the columns, and X2 is the second set of images listed the columns as well.

(b) To verify that your implementation is correct, download face1.mat and face2.mat from the course website where face1.mat contains 21 distinct images of person 1 in its columns and face2.mat contains 21 distinct images of person 2 in its columns and test your implementation with this data. What are the principal angles? Are they accurate in the context of this data set and why? Note: in case you want to see what the images look like, the images are of resolution 160×138. The following MATLAB commands will display the first image of person 1:

```
>> load face1
>> imagesc(reshape(face1(:,1),160,138)), colormap(gray), axis
        off
```

In the following problems, be sure to include the image results after each transformation along with the specific parameter values used to generate them. Do they seem reasonable? Why or why not?

(3) Write a function in MATLAB using the homogeneous coordinates that scales (enlarge and shrink) a 2D image about a point P = [tx, ty, 1]'. Specifically, the first line of your function will be (other than the comments)

```
function [newImg] = scale(Img, alpha, P)
```

where alpha = [sx, sy]' is the scale parameter that controls how much scaling is applied in x-direction and in y-direction, respectively. Make sure your routine works by applying it to an image of your choice. The MATLAB commands that are useful here:

```
>> meshgrid
>> interp2
>> imread
>> imagesc
>> reshape
```

For example, to find out how to use meshgrid, type

```
>> Help meshgrid
```

in the MATLAB command prompt.

(4) Writ a routine in MATLAB using the homogeneous coordinates that translates a 2D image horizontally and a routine that translates the image vertically. Specifically, one of your functions should have the input and output arguments

[newImg] = translateH(Img, tx)

where tx is the amount of horizontal translation applied (make sure it works for both positive and negative values). Test your routine by applying it to an image of your choice.

(5) Write a routine in MATLAB using the homogeneous coordinates that rotates a 2D image about a point P = [tx, ty, 1]'. Specifically, your function should have the input and output arguments

[newImg] = rotate[Img, theta, P]

where theta is the amount of rotation applied in counterclockwise orientation. Make sure your routine works by applying it to an image of your choice.

References

[1] A. Knyazev and M. Argentati. Principal angles between subspaces in an a-based scalar product: Algorithms and perturbation estimates. *SIAM J. Sci. Comput.*, 23(6), 2002.