CS268: MACHINE PERCEPTION

Homework 1: Building an Image Mosaic

DUE A WEEK FROM POSTING

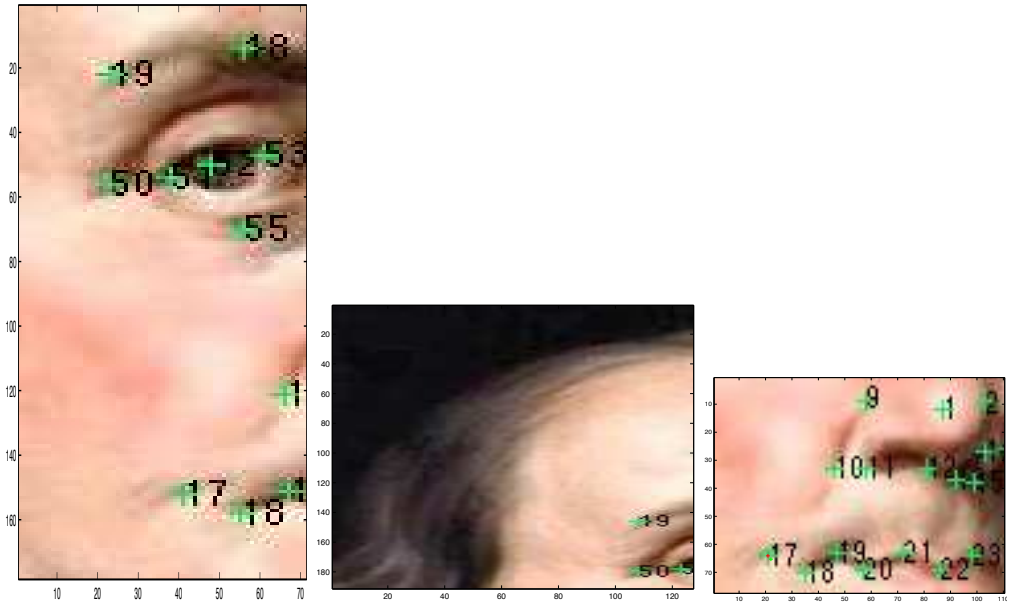


Figure 1: *Point with $j = 17$ is visible on the left and right images, I_1 and I_3, where it has coordinates, respectively, $x_1 = [42,\ 153]$ and $x_3 = [22,\ 65]$. The same goes for point $j = 19$, that is visible in the first and second images, etc. Each image I_j comes with the coordinate vector $x_j \in \mathbb{R}^{2\times N_j}$, where each column is the coordinate of a point in image x_j. Note that the order in this matrix is not necessarily consistent among different images. For instance, point $j = 17$ is visible in both images I_1 and I_3, but its coordinates may be, say, the fifth column of x_1 and the seventeenth column of x_3.*

Constructing a basic image mosaic

In this homework, you will start constructing an image mosaic. You are given a collection of 10 images, $I_1, I_2, \ldots, I_{10}$, and the goal is to compose them into a single image I, much in the same way in which you would "stitch" multiple photographs together.

For simplicity, you are given coordinates of "key points" in different images. You will notice that each image I_j has a certain number of green crosses, with an index written next to each of them. The matrix $x_j \in \mathbb{R}^{2\times N_j}$ contains the two coordinates (row, column) of those points in the image.

You will notice that each point is visible in more than one image, so you will have multiple copies of the coordinates of that point, each written relative to a different reference frame. We will say that the

coordinates in different images of a point with the same index "correspond" to each other. In the final mosaic, corresponding points will coincide.

Your goal is to find the transformations $[G_{ij}]$ that map each point with index "i" in image "j" onto a common canvas, where it will have coordinates $[x_i,\ y_i]$.

Once you have found all the transformations, you will assemble the mosaic image on the canvas and visualize it. You will turn in the Python script that produces the final mosaic image given the original images and the coordinates of corresponding points.

The questions below may help you get the most out of your homework:

- What does it mean for two points in different images to *"correspond"?* Try to distill a definition that is as simple and concise as possible, but unambiguous.
- Based on your definition above, how would you design an algorithm to determine such correspondence automatically?
- Does your mosaic show artifacts? Would you be able to tell from your final product that it has been assembled from different "fragments"? If so, how would you go about eliminating such artifacts?

Background

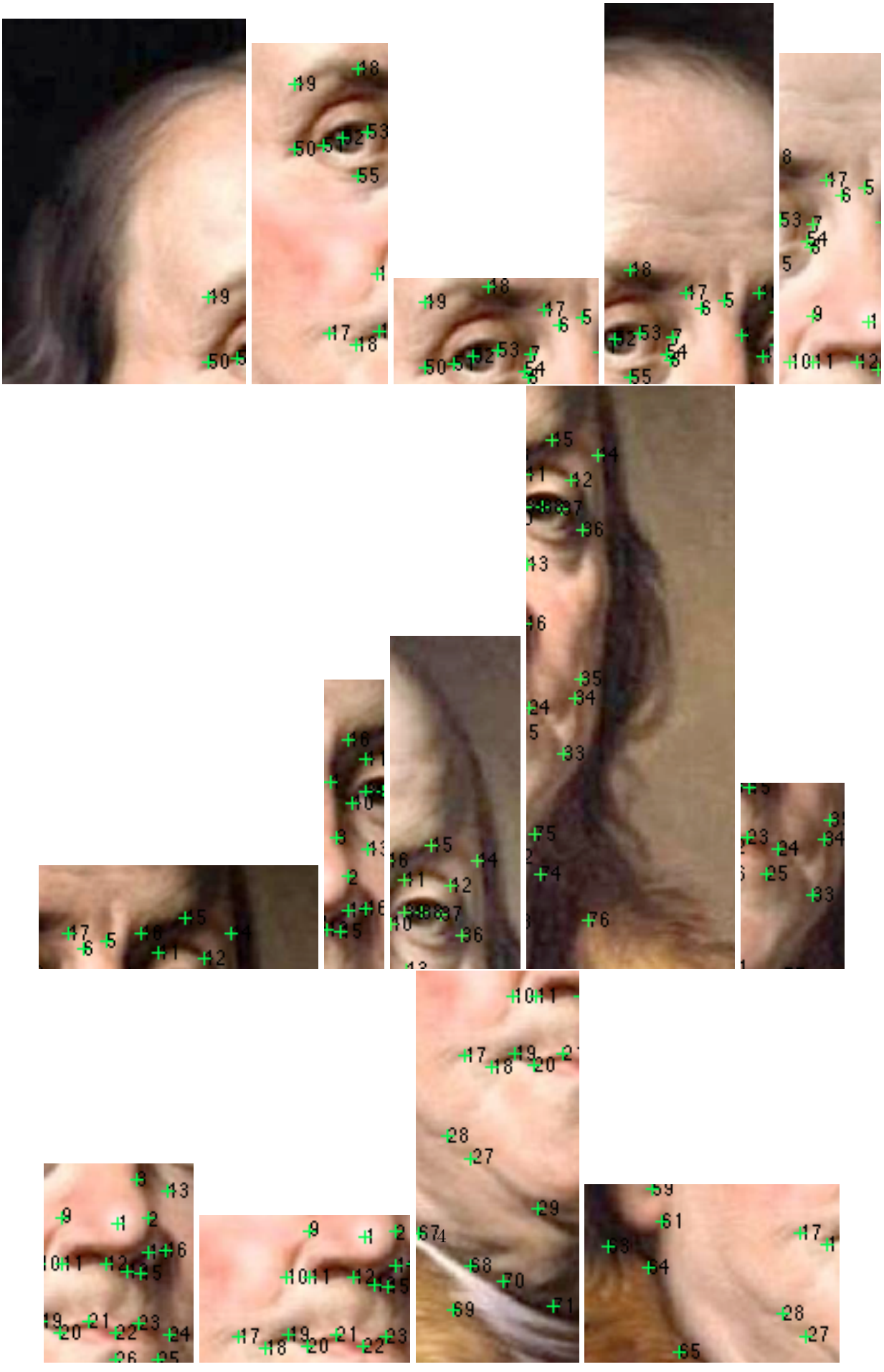
In order to complete this homework, you will need the following skills:

- Solve a linear system of algebraic equations in the least-squares sense.
- Efficiently solve a combinatorial search over index permutations

Below are the numerical values of the coordinates in the 20 images shown in Fig. 2. Note that you should *not* assume that such coordinates are *perfect!*. There may be errors, some small, some perhaps not so small.

```
x1 =
   109   109   124
   147   181   179

x2 =
    24    57    24    39    49    62    57    67    42    56    68
    23    15    57    55    51    48    71   122   153   159   152

x3 =
    18    51    80    88   100    18    33    43    56    73    70    72
    14     6    18    26    22    48    46    42    39    41    50    53

x4 =
     7    15    20    15    34    36    37    44    52    64    73    84    82
   177   141   174   197   185   188   176   154   161   157   175   186   153

x5 =
     2    19    16    18    19    11    19    26    34    46    42    48    53
    89    91    99   103   139   156   163    68    76    72   163   142   167

x6 =
    17    25    37    55    64    79    88   102
    37    45    41    37    47    29    50    37

x7 =
     5    14    23     8    16    23    24    14    14    23    10     3
    55    33    43    84    66    60    90   104   122   122   133   132

x8 =
     9    23    47    33    39    28    17     9
   129   111   119   132   158   147   146   146

x9 =
    39    15    25    31    20    10    30    27    21    34     9     6     3
    38    30    51    77    65    65   155   165   194   282   257   236   170

x10 =
     6    48    45    39    21    15     5
     4    21    31    60    36    49    30

x11 =
    50    66    56    40    11    11    34    46    52    56    68    61    51
    10    16    30    33    30    54    54    58    59    48    91   104    85


    39    39    25    10    34
   104    90    84    90    54

x12 =
    59    88   104   104   101    93    82    59    47    22    36    48    58
```

```
    10    13    10    28    39    38    34    34    34    65    71    64    70

    73    87
    64    70

x13 =
    52    64    78    63    53    41    27    18    30    65    73    47    30
    15    15    45    50    45    52    46    87   100   126   177   164   156

    21    19
   179   151

x14 =
    37    42    14    35   115   129   105   117    52
     4    21    34    45    27    34    69    81    89

x15 =
    95    83    69    60    72    89    72    63    45    22     6
    34    41    35    77    90   153   145   168   128   129    97

x16 =
    23    89   152   182   210   185   170   150   124    39    62    89   106
    21    13    27    31    76    52    43    36    39    52    52    69    77

   132   153   170   210   208   179   118    81
    90    94    77    76   128   121   132    92

x17 =
    14    26    26    17    11    43    61    55    64    69    84   103    86
    18    30    86   108   176    94    56   149   214   108   200   231   158

    91    87    89   107   106   117   123   119
   112    53    44    60    94   138    69    48

x18 =
    12    29    55    76    92   108    93   133   102    72    40   131    50
    29    37    50    54    37    12     3    36    81   102    92    88   157

    69    89   110   143   255   267   251   265
   143   174   144   137   101   121   141   167

x19 =
    21    50    29    62     8    66    75   136   184   170   186   173
    33    40    96    89   126   136   135   160   119    94    73    53

x20 =
    18    39    82   114   145   173   185   152   111    92   131   188   198
    49    50    23    33    12    19    68    75    74    88   105   115   114
```



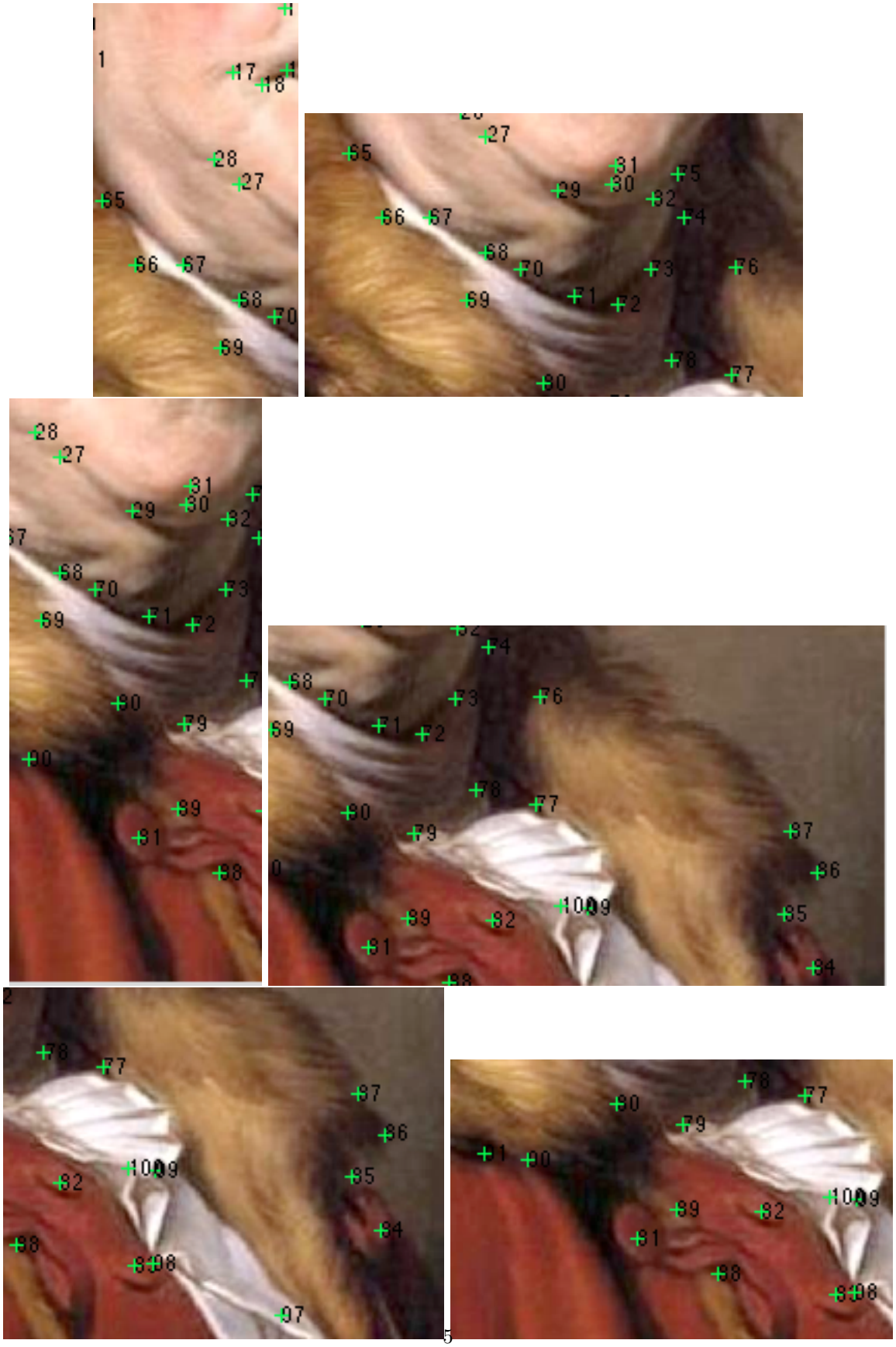


Figure 2: *Images I_1 through I_{20}.*