Tyler L Wang
Jiacheng Guo

Final Project

EN 256 – Spring 2017
April 26, 2017

Image Combination with Machine Learning

In this section, we introduce our method to linearly combine M lower-resolution inpainted images using a machine learning method.
For each image labeled i, where $i \in [1, M]$, we assign a weight w_i to it. We approximate the ground-truth output image with a linear combination of all input images, that is,

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + ... + w_M x_M$$

where $\hat{y}$ is an approximation of the output image, x_i is the input image, and w_0 is a constant weight. Since each time the set of input images are completely different, it makes sense to first set $w_0 = 0$.
Here we use the least-square regression. The sum-of-squares error function is defined as

$$E = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(w_1 x_1 + w_2 x_2 + ... + w_M x_M - y)^2$$

The optimal solution minimizes the the error.

$$w = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2}(w_1 x_1^{(i)} + w_2 x_2^{(i)} + ... + w_M x_M^{(i)} - y^{(i)})^2$$

Taking the gradients of the error function and setting them to 0, we have the following set of equations

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} (w_1 x_1^{(i)} + w_2 x_2^{(i)} + ... + w_M x_M^{(i)} - y^{(i)}) x_j^{(i)} = 0$$

where $j \in [1, M]$. This is equivalent to

$$\begin{aligned} \sum_{i=1}^{N} y^{(i)} x_j^{(i)} &= \sum_{i=1}^{N} [w_1 \ w_2 \ ... \ w_M] \cdot [x_1^{(i)} \ x_2^{(i)} \ ... \ x_M^{(i)}] x_j^{(i)} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{M} w_k x_k^{(i)} x_j^{(i)} \\ &= \sum_{k=1}^{M} w_k \sum_{i=1}^{N} x_k^{(i)} x_j^{(i)} \end{aligned}$$

Denoting

$$M_{k,j} = \sum_{i=1}^{N} x_k^{(i)} x_j^{(i)}$$

$$z_j = \sum_{i=1}^{N} y^{(i)} x_j^{(i)}$$

our minimization problem is equivalent to solving this linear system:

$$Mw = z \Longleftrightarrow \nabla E(w) = 0$$

For simplicity, the analysis above uses 1 variable to denote each image (x for the input and y for the output). When implementing the algorithm, however, we need to expand x and sum over all pixels. M and z can be rewritten as

$$M_{k,j} = \sum_{i=1}^{N} \sum_{pixel} x_k^{(i)} x_j^{(i)}$$

$$z_j = \sum_{i=1}^{N} \sum_{pixel} y^{(i)} x_j^{(i)}$$

The runtime is $O(N \cdot M^2 \cdot R)$, where R is the resolution of the image.