## Zooming Methods

* Pixel replication or (Nearest neighbor interpolation)
* Zero order hold method
* Zooming K times

1. Pixel replication (Nearest neighbor interpolation)

We produce the zoomed image by **replicating each pixels** of the original image.

**Step 1. Row-wise zooming**

Copy each row’s pixels to an adjacent new cell row-wise

**Step 2. Column-wise zooming**

Copy each column’s pixels to an adjacent new cell column-wise

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | **1** | 2 | **2** |
| **1** | **1** | **2** | **2** |
| 3 | **3** | 4 | **4** |
| **3** | **3** | **4** | **4** |

e.g.

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

**New image size:**

(Original image rows \* zooming factor, Original Image cols \* zooming factor)

**Simple method, but produce blurry zoomed image**

2. Zero order hold method (Zoom twice)

Take “average” of every two adjacent cells and place the result in between.

**Step 1. Row-wise zooming**

e.g. Take first two numbers 1 and 2, where (1 + 2) / 2 = 1.5, we treat it as **1**

|  |  |  |
| --- | --- | --- |
| 1 | **1** | 2 |
| 3 | **3** | 4 |

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

**Step 2. Column-wise zooming**

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| **2** | **2** | **3** |
| 3 | 3 | 4 |

e.g.

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

**New image size:** (2 \* (number of rows) - 1) \* (2 \* (number of columns) - 1)

**Better quality, but can only zoom in a power of 2**

3. Zooming K times

**Step 1. Take ”outputs” (OP)**

Take two adjacent pixels as we did in Zooming Twice. **Subtract the smaller from the greater one to get the OP.**

**Step 2. Process OP with “zooming factor” (K)**

Divide OP with zooming factor K and add the result to the smaller valuepixels[i \* self.K - p - 1, (j - 1) \* self.K]. Then put the result in between those two values

**Step 3. Add OP to the result again until there is (k – 1) values put in between**

Put the first result into the cell next to the smaller number, then put the following result towards greater number.

**Step 4. Repeat the same step for all the rows and columns**

e.g. (Assume K = 3, values to be inserted = k – 1 = 2)

|  |  |  |
| --- | --- | --- |
| 15 | 30 | 15 |
| 30 | 15 | 30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15 | **(30-15) = 15**  **15 + 15/3 = 20** | **20 + 5 = 25** | 30 | **20 + 5**  **= 25** | **(30-15) = 15**  **15 + 15/3 = 20** | 15 |
| 30 | **25** | **20** | 15 | **20** | **25** | 30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15 | 20 | 25 | 30 | 25 | 20 | 15 |
| **20** | **21** | **22** | **25** | **22** | **21** | **20** |
| **25** | **22** | **21** | **20** | **21** | **22** | **25** |
| 30 | 25 | 20 | 15 | 20 | 25 | 30 |

**New image size:** (K \*(number of rows - 1) + 1) X (K \*(number of cols - 1) + 1)

**Able to compute zoom of any factor, less blurry, but it has to be sort so that the computation cost is higher**