**Method**

**Symmetric NNMF**:

The symmetric nearest neighbor mean filter is a 2-D filter. In this project, the size of the window is 5\*5. The function of this filter is to select the point which is the most similar to the central pixel in the window from each symmetrically opposite pair. So there are 13 pixels after the selection, and then get the average value of these 13 pixels to represent the value of the central pixel in the new image. Here is an example of symmetric nearest neighbor mean filter.

图片包含 游戏机, 灯光, 钟表, 大

描述已自动生成

Figure 1 An example of symmetric nearest neighbor mean filter.

The formula of the output with respect to Figure 1 is

The symmetric nearest neighbor mean filter can reduce the noise, sharpen edges, and reduce thin lines.

**Anisotropic Diffusion for Image Filtering:**

The implementation of anisotropic diffusion for image filtering is based on the equation

In this equation, N, S, E, W represent north, south, east, west neighbors of pixel (i, j). is a parameter and is set to be 0.25 in our project. refers to difference between pixel (i, j) and one neighbor of it, for example:

are conduction coefficients and updated after every iteration. In this project, we have tried two computation methods of it. How to compute them is shown below:

In this project, we have tried two forms of the function . The first one is the exponential formula:

The second one is the reciprocal function:

The in both functions is a parameter and the value of it is selected by us. We have tried several values of , and the different results are shown in results section.

For cwheelnoise.gif figure, the images after 0, 20, 50, 100 iterations, gray-scale histogram, plot of the line y=128 through the image, and segmented version are also shown. We select pixels which values are between 80 and 110 to be the segmented image which shows the shape of the wheel, and the range of the value is according to the gray-scale histograms, we just pick the second peak in the histogram.

The same process is also conducted on cameraman.tif image.