GITHUB WITH CODE: https://github.com/kerekejo/CVAssignment2

Gaussian filtering is a technique for smoothing images by reducing noise. The process involves taking a kernel and assigns weights to the pixels in said kernel based on the distance from the center pixel.

The code aims to use three different variations of Gaussian filtering using three norms. The l2 norm is considered the “standard Gaussian filter” which minimizes the total squared difference between the filtered and original images. The l1 norm uses absolute values, which produces a different smoothing effect. The final norm, l-infinity, inverts the weights of the kernels, which emphasizes maximum differences in the kernel.

```
def gaussian_kernel(size, sigma=1):
    k = size // 2
    ax = np.arange(-k, k+1)
    xx, yy = np.meshgrid(ax, ax)
    kernel = np.exp(-(xx**2 + yy**2) / (2 * sigma**2))
    return kernel / np.sum(kernel)
```

The above function works on creating the kernel. It uses size as a parameter and sets sigma as 1.

```
def apply_gaussian_filter(image, size, sigma):
    kernel = gaussian_kernel(size, sigma)
    return cv2.filter2D(image, -1, kernel)
filtered_image_l2 = apply_gaussian_filter(image, 3, 1)
```

This function defines the “standard Gaussian filter” which is the l2 norm. It takes the image, the size of the kernel, and a value for sigma. It then applies the standard filter using the cv2 method.

```
def apply_gaussian_l1_filter(image, size, sigma):
    kernel = gaussian_kernel(size, sigma)
    kernel = np.abs(kernel)
    kernel /= np.sum(kernel)
    return cv2.filter2D(image, -1, kernel)
filtered_image_l1 = apply_gaussian_l1_filter(image, 3, 1)
```

This uses the previously described method for using the l1 gaussian filter. It uses numpy methods to get the absolute values, and to then divide by the sum. After this, it filters the changed kernel and displays the image.

```
def apply_gaussian_linf_filter(image, size, sigma):
    kernel = gaussian_kernel(size, sigma)
    kernel = np.max(kernel) - kernel
    kernel /= np.sum(kernel)
    return cv2.filter2D(image, -1, kernel)
filtered_image_linf = apply_gaussian_linf_filter(image, 3, 1)
```

L-infinity Gaussian filtering is found by again adjusting the kernel by using numpy max method and then dividing the new kernel by the sum. It then uses the cv2 filter method to achieve the desired filter.