

Render Algorithm

Function: render_algorithm():

Per ogni pixel dell'immagine di posizione (x,y)

for x in image:

for y in image:

kernel = kernel_building(x,y)

new_pixel = PixelConvolve((x,y), kernel)

new_image[x][y] = new_pixel

return new_image

Function: kernel_building(x, y)

Itero in un kernel di ordine n centrato in (x,y)

for i in range(x - n, x + n):

for j in range(y - n, y + n):

Kernel vuoto

kernel = []

Prendo la PSF con depth di (i,j) centrata in (x,y)

PSFij = upload_PSF(depth[i,j], (x,y))

Prendo il valore del pixel ij nella PSF centrata in (x,y)

valoreij = PSFij[i,j]

Aggiungo il valore al kernel

kernel.append(ijvalore)

Sommo i valori del kernel

kernelSum += ijvalore

Divido il kernel per la somma dei suoi valori (Stitching)

kernel /= kernelSum

return kernel