A 2D gaussian filter is trivially converted into the generating vector by simply setting $y = 0$ and normalizing, and this is done for the detection filters. This is not so easy for the other filter they use:

The spatial filter generation uses:

$$f(x, y) = a_1 e^{-a_2(x^2+y^2)} + b_1 e^{-b_2(x^2+y^2)} \tag{1}$$

As a sum of gaussians, we need to find a way to create two corresponding vectors. $f(x, y)$ is used to create an NxN matrix F that will be the convolution kernel:

$$F_{i,j} = f(i - r, j - r), \tag{2}$$

with $N = 2r + 1$ and i, j spanning $[-r, r]$. Note that F is then normalized into $\tilde{F}$:

$$\tilde{F} = F/S_F \tag{3}$$

$$S_F = \sum_i \sum_j F_{i,j} \tag{4}$$

This notation means every element of the matrix F is divided by S_F.

We construct the correct vectors g_1 and g_2:

$$g_1(x) = \sqrt{a_1} e^{-a_2 x^2} \tag{5}$$

$$g_2(x) = \sqrt{b_1} e^{-b_2 x^2}. \tag{6}$$

They have with the following property:

$$F = g_1 \otimes g_1 + g_2 \otimes g_2 \tag{7}$$

$$F_{i,j} = g_1(j) * g_1(i) + g_2(j) * g_2(i), \tag{8}$$

with the crossed circle denoting the outer product.

Because F is normalized, we need to find the normalization for our vectors too, characterized by:

$$\tilde{F} = \tilde{G} = \frac{g_1 \otimes g_1 + g_2 \otimes g_2}{S_G}. \tag{9}$$

By identification, we trivially have:

$$S_G = S_F \tag{10}$$

We write the normalization:

$$\tilde{g_1} \otimes \tilde{g_1} = \frac{g_1 \otimes g_1}{S(g_1)^2 + S(g_2)^2} \tag{11}$$

$$\texttt{with:} \tilde{g_1} = \alpha_1 g_1 \tag{12}$$

$$\alpha g_1 \otimes \alpha g_1 = \frac{g_1 \otimes g_1}{S(g_1)^2 + S(g_2)^2} \tag{13}$$

$$\alpha^2 g_1 \otimes g_1 = \frac{g_1 \otimes g_1}{S(g_1)^2 + S(g_2)^2} \tag{14}$$

$$\texttt{by identification:} \quad \alpha = \frac{1}{\sqrt{S(g_1)^2 + S(g_2)^2}} \tag{15}$$

$$\tag{16}$$

So, we compute our vectors this way to create a separable convolution of the sum of gaussians.

Here in pseudocode for the convolution:

```
c1=convolve1D(image, alpha*g_1, alpha*g_1)
c2=convolve1D(image, alpha*g_2, alpha*g_2)
result=c1+c2.
```