5.1 Gaussian Conditioning

The template update criterion that is being used is, as mentioned, a very simple one. Perhaps some optimization of this routine could improve the Template Tracking mechanism. Nonetheless, another approach has been attempted. The idea is to condition TMR, i.e. to weight it by some other matrix. This other matrix would supply additional information to the tracking mechanism, other than the absolute measure of similarity inherent to template matching.

Relevant information is related to the positional continuity of objects in time, i.e. objects do not warp from one place to another without following some connecting trajectory between these points. Of course that frame by frame analysis, as is the case, is always a discrete process and warping might occur due to low sampling frequency. However, assuming that the frame rate is high enough, it can be fairly trusted that an object early positioned at some coordinates $\vec{r}^{N-1}$ at instant $N-1$, will have, in the following iteration, a higher probability of its position $\vec{r}^{N}$ being in the neighborhood of $\vec{r}^{N-1}$.

$$P(\vec{r}^{N} \mid \vec{r}^{N-1}) = f\left(\frac{1}{\left\|\vec{r}^{N} - \vec{r}^{N-1}\right\|}\right) \tag{29}$$

In order to embed this "common sense" into our system, it was decided to use a 2D Gaussian probability distribution centered at $\vec{r}^{N-1} = \left[x^{N-1}, y^{N-1}\right]^{T}$, embedded into a matrix with the same size as TMR that will be referred to as Gaussian matrix (GM). GM is calculated as follows:

$$G(a,b) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{a^2+b^2}{2\sigma^2}\right)} \tag{30}$$

Where $G(a,b)$ is the function's value considering $a = x^{N} + x^{N-1}$, $b = y^{N} + y^{N-1}$ while σ is the standard deviation expressing how "wide" the Gaussian is defined. This enables centering the matrix at the previous template position, as in Figure 16.

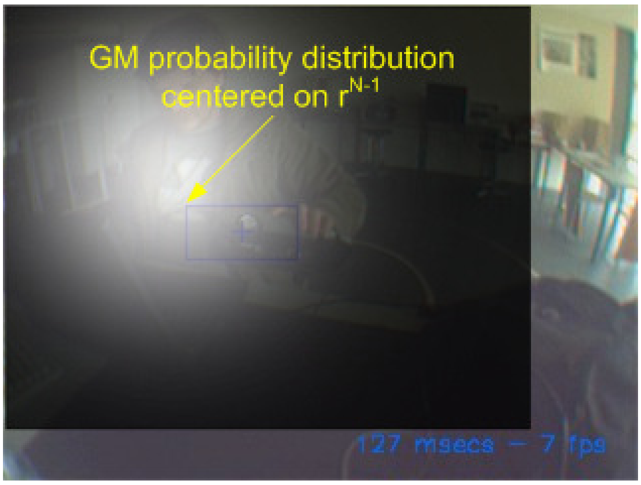

Figure 16. GM overlapped onto its corresponding frame.

After being properly normalized, GM is used to condition TMR by performing a simple pixel by pixel multiplication, thus decreasing the match score for pixels that where distant from the previous $\vec{r}^{N-1}$.

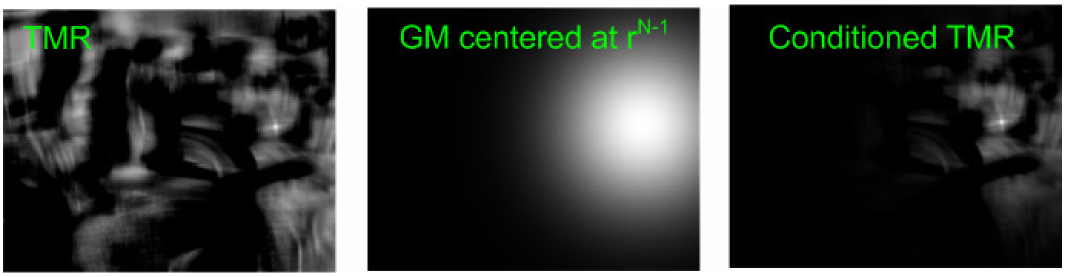

Figure 17. TMR (left), GM centered at $\vec{r}^{N-1}$ (middle) and conditioned GM (right).

This technique was able to considerably improve tracking performance. The phenomena of warping disappeared completely. The tracking works even when objects do not have a high contrast with the background.

5.2 Fast Gaussian Computation

The Gaussian function calculation is not fast to compute, especially considering that any tracking mechanism should be fast when compared to an object recognition technique. Calculating a new GM in all iterations (for every new $\vec{r}^{N}$) would considerably decrease the frame rate. Hence, an alternative method had to be developed. As a part of the program initialization processes, an extended Gaussian matrix (EGM) is calculated. To calculate EMG's minimum size, note that, by absurd, the smallest template one can use is of size 1×1. Therefore, recovering chapter 4.1 template matching results matrix size equation (18), the biggest size that the template match results matrix can have is $W \times H$, i.e., the image's size. The EGM is also a Gaussian 2D function represented in an image of size $2W \times 2H$, with the function centered in the center of the image. For every given T_p and TMR size, a specific sub-window of the EGM can be used without having to reposition the probability distribution thus recalculating the probability value of every pixel. The EGM sub-window's upper left corner's coordinates are given by:

$$EGM_{upperleft} = \begin{bmatrix} EGM_{height} - \dfrac{TMR_{height}}{2} - T_{px} \\ EGM_{width} - \dfrac{TMR_{width}}{2} - T_{py} \end{bmatrix} \tag{31}$$

Its size is equal to the size of TMR. Using this technique, the Gaussian matrix's calculation is limited to the definition of a particular region of interest of the EGM, therefore saving precious computation time (Figure 18).

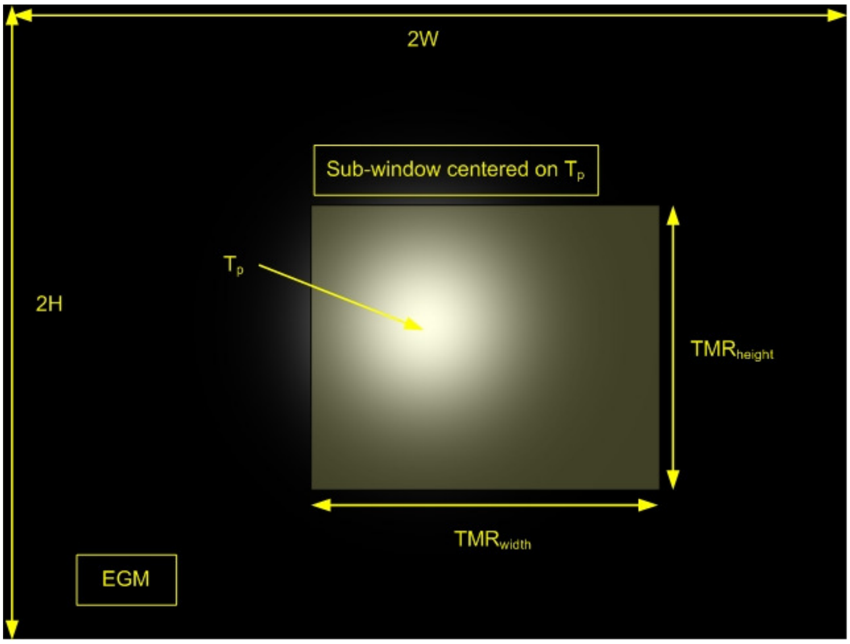

Figure 18. Fast Gaussian computation.