IMCF Algorithm Derivation

April 4, 2006

The video sequence is represented as a time varying image $I_c(\vec{x}, t)$ where $\vec{x}$ has two spatial dimensions and t represent the time dimension. The sampling process results in signal

$$I_s(\vec{x}, t) = I_c(\vec{x}, t) \cdot \Lambda(\vec{x}, t)$$

where $\Lambda(\vec{x}, t)$ is the standard three dimensionnal lattice sampling kernel (3-D sequence of δ functions).

The reconstruction of the display can be described by convolution of sampled video signal with aperture point-spread function:

$$I_d(\vec{x}, t) = I_s(\vec{x}, t) * A(\vec{x}, t) = (I_c(\vec{x}, t) \cdot \Lambda(\vec{x}, t)) * A(\vec{x}, t)$$

In the frequency domain, we'll have:

$$I_d^f(\vec{f_x}, f_t) = (I_c^f(\vec{f_x}, f_t) * \Lambda(\vec{f_x}, f_t)) \cdot A(\vec{f_x}, f_t)$$

The LCD aperture function can be well modeled as a sample-and-hold display. This results in aparature function

$$A(\vec{x}, t) = \mathit{sinc}(\pi f_t T_h)$$

where T_h is the sample-and-hold hold time.

Note that the sample-and-hold reconstruction function is only a function of time. This makes sense as there should not be any spatial variation in the image reconstruction properties on LCD screen.

A moving image can be represented as

$$I_m(\vec{x}, t) = I_c(\vec{x} + \vec{v}t, t)$$

where $\vec{v}$ is the speed of moving image over screen (in units of pixels/frame, for example). In frequency domain this becomes:

$$I_m^f(\vec{f_x}, f_t) = I_c^f(\vec{f_x}, f_t - \vec{v} \cdot \vec{f_x})$$

The eye tracking property of the HVS results in the user tracking the moving image on screen in order to produce a static image on retina [Wandell]. This result in "inverse" of moving image, and we have

$$I_e(\vec{x},t) = I_d(\vec{x} - \vec{v}t, t)$$

and

$$I_e^f(\vec{f_x}, f_t) = I_d^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x})$$

Putting all pieces together results in:

$$\begin{aligned} & I_e^f(\vec{f_x}, f_t) = \\ & = \left[I_m^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x}) * \Lambda^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x})\right] \\ & \qquad \cdot A^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x}) \\ & = \left[I_c^f(\vec{f_x}, f_t) * \Lambda^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x})\right] \\ & \qquad \cdot A^f(\vec{f_x}, f_t + \vec{v}\cdot\vec{f_x}) \end{aligned}$$

Now, as we have seen in class [Wandell], the HVS system acts a temporal low-pass filter, so only frequencies at $f_t \sim 0$ count. This results in perceived image:

$$\begin{aligned} I_p^f(\vec{f_x}) &= LPF\left[I_e^f(\vec{f_x}, f_t)\right] \\ &= I_c^f((\vec{(}f)_x) \cdot A^f(\vec{f_x}, \vec{v} \cdot \vec{f_x}) \\ &= I_c^f((\vec{(}f)_x) \cdot \mathit{sinc}(\pi \vec{v} \cdot \vec{f_x} T_h) \end{aligned}$$

Therefore, to undo distortion, choose a "compensating filter" - apply a high-pass filter in the **motion direction**:

$$H_{inv}^f(\vec{f_x}) = \frac{1}{\mathit{sinc}(\pi \vec{v} \cdot \vec{f_x} T_h)}$$

MMSE: With noise added to the picture, can use MMSE high-pass filter so as not to suffer from too much noise enhancement.