Deriving Intrinsic Images From Image Sequences

Yair Weiss

ICCV 2001

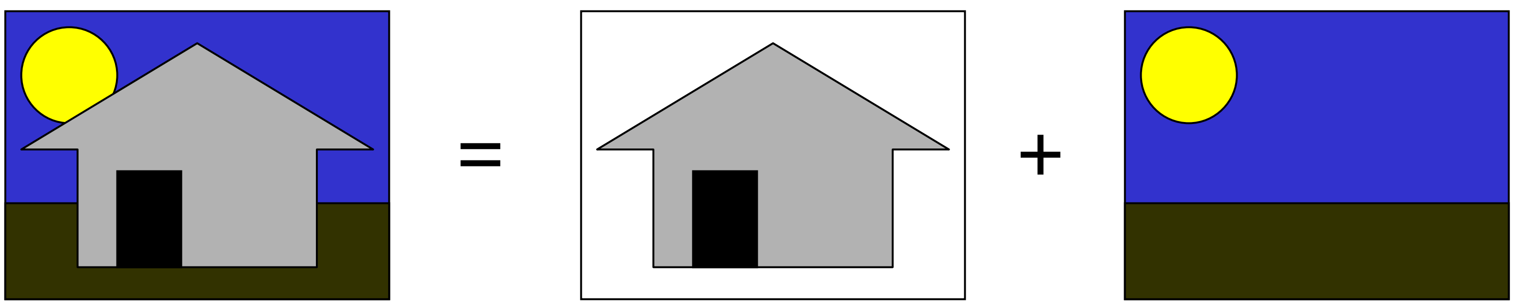
Ben Marlin
Machine Learning Group
Department of Computer Science
University of Toronto

November 24, 2004

Decomposing Images

- Decomposing images into layers, parts, and other types of pieces is often a useful image processing task.

- One of the more common decompositions is layer separation. A simple example is extracting a foreground object F from the background B.



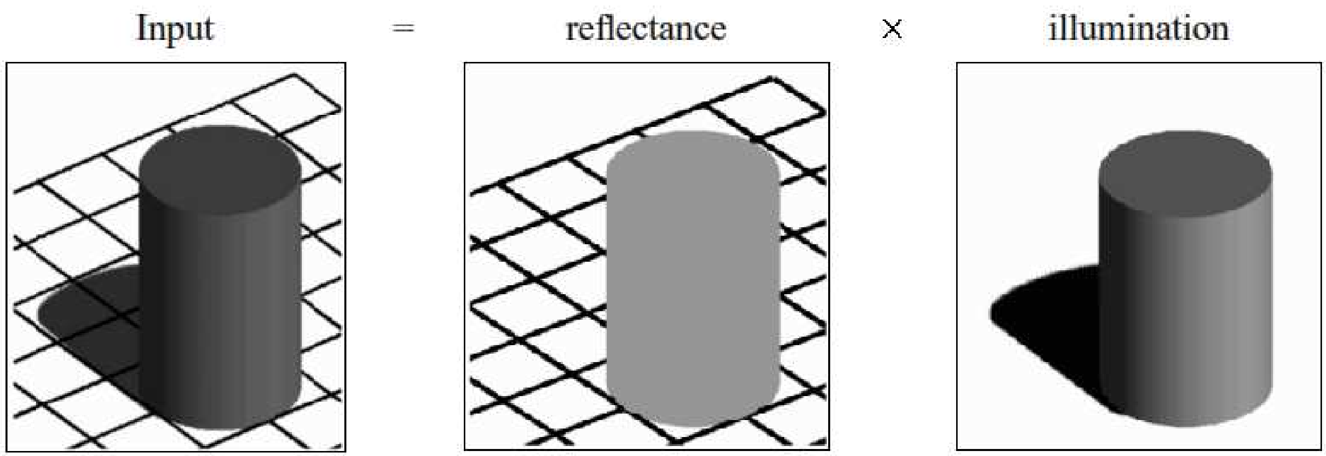
$$I = F + B$$

Intrinsic Image Decomposition

- In an intrinsic image decomposition, the goal is to decompose the input image I into a reflectance image R and an illumination image L such that:

$$I = R \times L$$

$$\log(I) = \log(R) + \log(L)$$

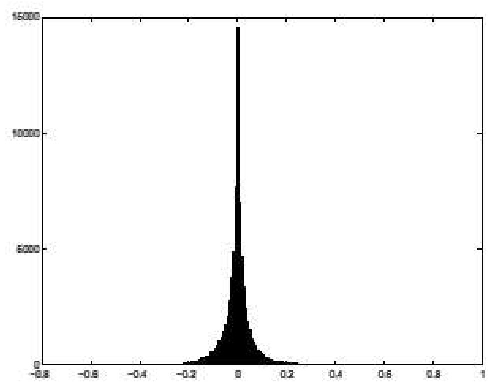


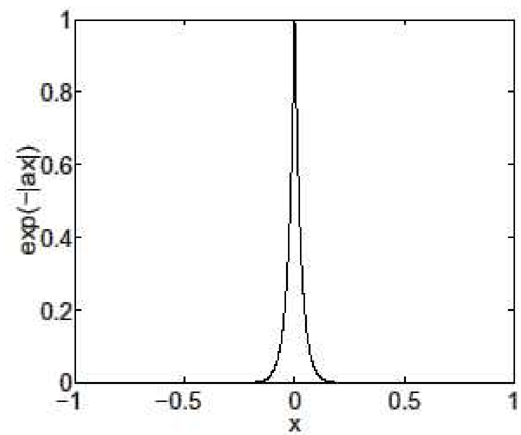
Intrinsic Image Decomposition

- Why is this hard? The problem is completely ill-posed. If we set I=R and L=1, we can always obtain a perfect decomposition.
- The part of the solution proposed in this paper is to use a set of images I_k under different illumination conditions.
- The goal is to recover the single reflectance image R and all the illumination images L_k. This is easier, but its still ill-posed. We need some useful priors.

Natural Image Statistics

- The other idea in this paper is to use a prior over derivative filter outputs. Research into natural image statistics says derivative filter outputs are sparse for natural images.
- The derivative filters used in this paper are simple horizontal and vertical derivatives: f_h=[-1 1], f_v=[-1 1]'.





Model

- Assume we have K images I_k . The proposed model is an IID Laplace distribution on the derivative filter outputs at each pixel (x,y) of each illumination image L_k. We can convert this into a distribution on R and I.

$$L_k^j \;=\; f^j \star L_k = f^j \star (I_k - R) = f^j \star I_k - f^j \star R = I_k^j - R^j$$

$$P(L^j) \;=\; \frac{1}{Z} \prod_{x,y,k} \mathrm{exp}(-\beta |L_k^j(x,y)|)$$

$$P(I^j | R^j) \;=\; \frac{1}{Z} \prod_{x,y,k} \mathrm{exp}(-\beta |I_k^j(x,y) - R^j(x,y)|)$$

Solution

- **NEAT TRICK:** We can now easily get a maximum likelihood estimator for R^j since the median maximizes absolute loss:

$$\hat{R}^j(x,y) = median_k I_k^j(x,y)$$

- To complete the solution, we just need to invert the following over constrained system to obtain the estimated reflectance image:

$$\hat{R}^j(x,y) = f^j \star \hat{R}(x,y)$$

- There is a further neat trick to do this using the Fourrier transform, and pseudo inverses.

Cool, but does it work?