Lecture 10
Laplacian of Gaussian (LoG)

ECEN5283 Computer Vision

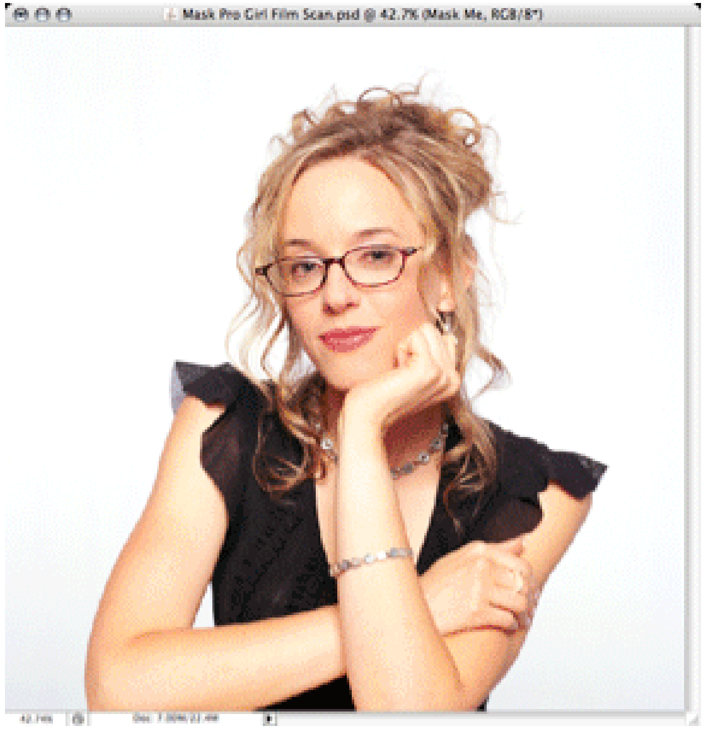
Dr. Guoliang Fan

Oklahoma State University

Goals

To introduce edge detection theory that covers three basic issues.

To introduce the Laplacian-of-Gaussian (LoG) edge detector that uses the second-order spatial differentiation.



Edge Detection Theory

Fundamentals of edge detection

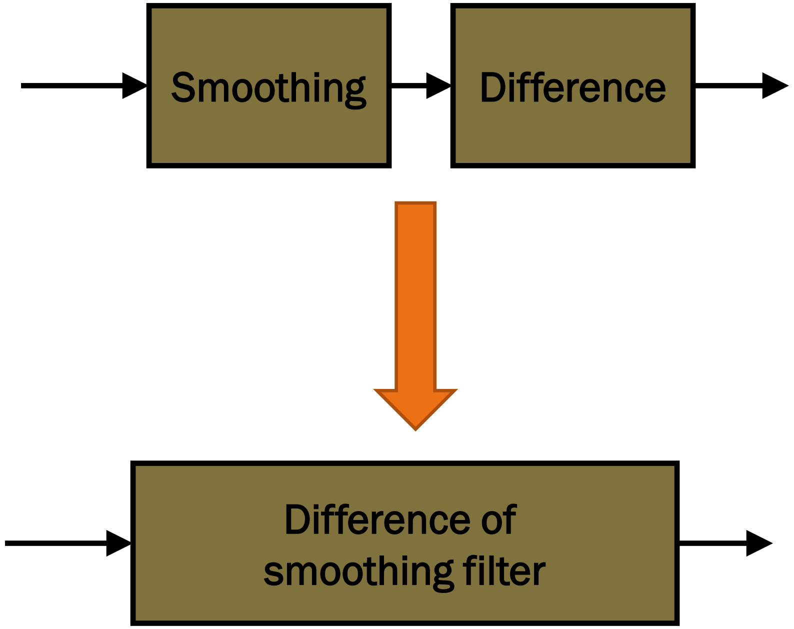
- Edge detection is essentially involving spatial differentiation (difference) that is sensitive to fast intensity changes
- Differentiating a function is the same as emphasizing high-frequency components and deemphasizing low-frequency components.

One key issue of edge detection

- How to reduce the noise effect in spatial differentiation by incorporating a pre-smoothing filter?

Derivates of Smoothing Filters

$$\frac{\partial I}{\partial x} = K_{(\partial/\partial x)} ** I.$$



Important Question:

- *Is differentiation* an LTI/LSI operation?
- If yes, we define a kernel that implements differentiation as a linear filter

We want the derivative of a smoothing function S

$$(K_{(\partial/\partial x)} ** (S ** I)) = (K_{(\partial/\partial x)} ** S) ** I = \left(\frac{\partial S}{\partial x}\right) ** I.$$

This fact appears in its most commonly used form when the smoothing function is a Gaussian

$$\left(\frac{\partial G_\delta ** I}{\partial x}\right) = \left(\frac{\partial G_\delta}{\partial x}\right) ** I.$$ Derivative of Gaussian filters

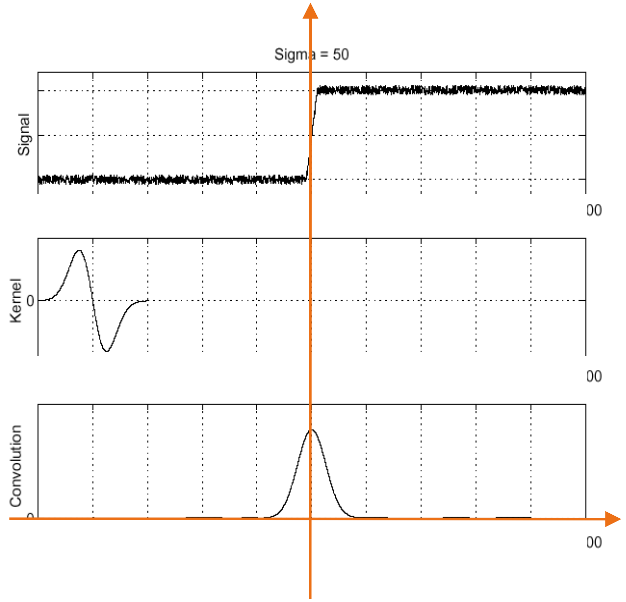
Optimal smooth filter

Canny (1986) established the practice of choosing a derivative estimation filter by optimizing three criteria:

- Signal to noise ratio: the filter should respond more strongly to the edge at $x = 0$ than to noise.
- Edge Localization: the filter response should reach a maximum close to $x = 0$.
- Low false positives: there should be only one maximum of the response in a reasonable neighborhood $x = 0$.

It is a remarkable fact that the optimal smoothing filters tend to look a great deal like Gaussians.

$$edge(x) = AU(x) + B + n(x).$$



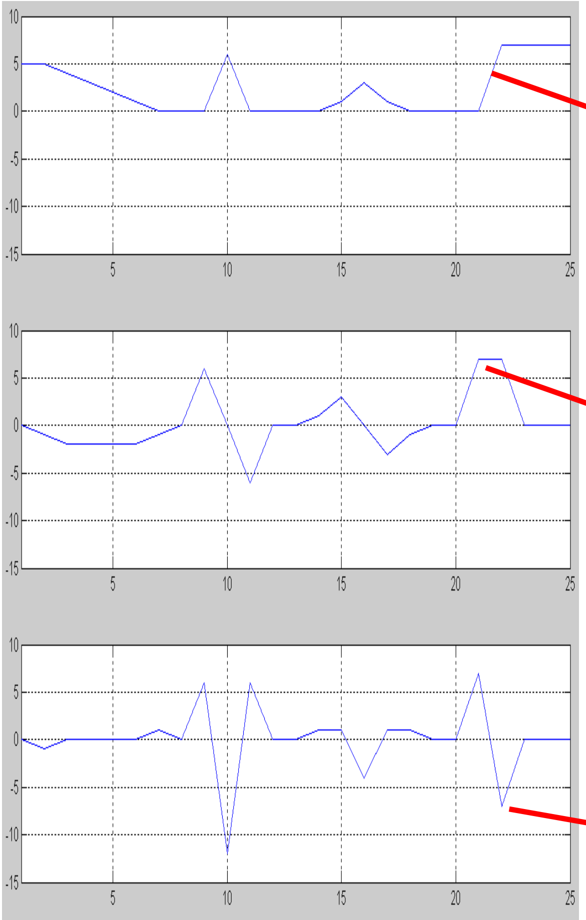
Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Differentiation for Edge Detection

A 1-D signal

First-order derivative

Second-order derivative



Edge point

Local maximum

Zero-crossing

2D Laplacian Function

We develop the 1D first and second order derivatives as follows:

$$\frac{df}{dx} = f(x) - f(x-1)$$

$$\frac{d^2 f}{dx^2} = f(x) - 2f(x-1) + f(x-2)$$

We define a 2-D second order derivative as the Laplacian function which is isotropic

$$\frac{\partial^2 f}{\partial y^2}: \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \frac{\partial^2 f}{\partial x^2}: \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$(\nabla^2 f)(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplacian-of-Gaussian (LoG)

If an image is pre-smoothed by a Gaussian filter, then we have the Laplacian-of-Gaussian (LoG) operation that is defined as

$$(K_{\nabla^2} ** (G_\sigma ** I))$$

$$= (K_{\nabla^2} ** G_\sigma) ** I$$

$$= \left(\frac{\partial^2 G_\sigma}{\partial x^2} + \frac{\partial^2 G_\sigma}{\partial y^2}\right) ** I.$$

$$= (\nabla^2 G_\sigma) ** I.$$

Separable?

where $\nabla^2 G_\sigma(x, y) = \left(\frac{1}{2\pi\sigma^4}\right)\left[\frac{x^2 + y^2}{\sigma^2} - 2\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$

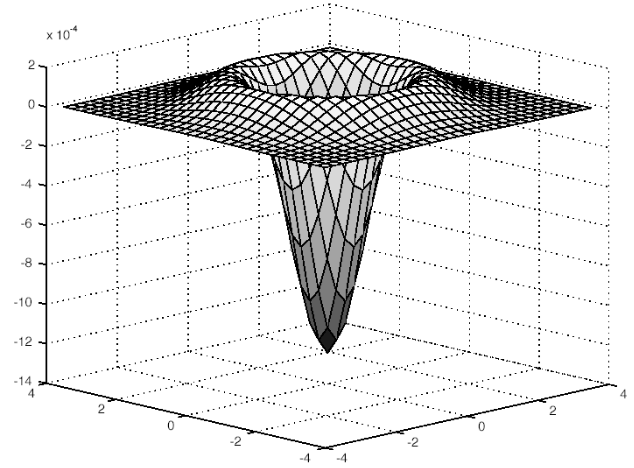


Figure 9.7. The Laplacian of Gaussian filter kernel, shown here for σ one pixel, can be thought of as subtracting the center pixel from a weighted average of the surround (hence the analogy with unsharp masking, described in the text). It is quite common to replace this kernel with the difference of two Gaussians, one with a small value of σ and the other with a large value of σ.

Determining a Discrete LoG Kernel

$$\nabla^2 G_\sigma(x, y) = \left(\frac{1}{2\pi\sigma^4}\right)\left[\frac{x^2 + y^2}{\sigma^2} - 2\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{(continuous LoG)}$$

$$K[i, j] = \left(\frac{1}{2\pi\sigma^4}\right)\left[\frac{(i-k-1)^2 + (j-k-1)^2}{\sigma^2} - 2\right]e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}} \quad \text{(Discrete LoG)}$$

The dimension of the kernel is $(2k + 1) \times (2k + 1)$.

The variance σ^2 will determine the dimension of the kernel.

π is usually ignored in computing the LoG coefficients.

Some rounding may be involved to make coefficient integers to speed up the 2D convolution.

The average of all kernel coefficients must be zero (why?).

A LoG kernel example

$$K[i,j]=\left(\frac{1}{2\sigma^4}\right)\left[\frac{(i-k-1)^2+(j-k-1)^2}{\sigma^2}-2\right]e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}} \quad \text{(Discrete LoG)}$$

$$\delta = 0.5$$

$$\mathbf{K}=\begin{bmatrix} 0.0000 & 0.0065 & 0.0376 & 0.0065 & 0.0000 \\ 0.0065 & 0.8792 & 2.1654 & 0.8792 & 0.0065 \\ 0.0376 & 2.1654 & -16.0 & 2.1654 & 0.0376 \\ 0.0065 & 0.8792 & 2.1654 & 0.8792 & 0.0065 \\ 0.0000 & 0.0065 & 0.0376 & 0.0065 & 0.0000 \end{bmatrix} \Rightarrow \mathbf{K}^*=\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The Steps of LoG Edge Detection

Applying the LoG to the image.

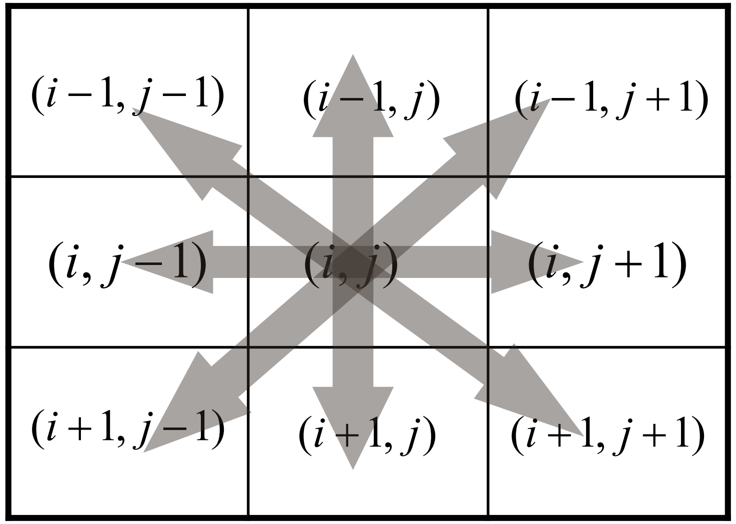
Detection of zero-crossings (ZCs) in the image (how?).

Threshold the ZCs to keep only strong ones with significant difference between the positive and negative values.

Remove isolated edge points caused by noise by counting the number of pixels in each connected component and removing smaller components.

- (To be discussed in more details later.)

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$



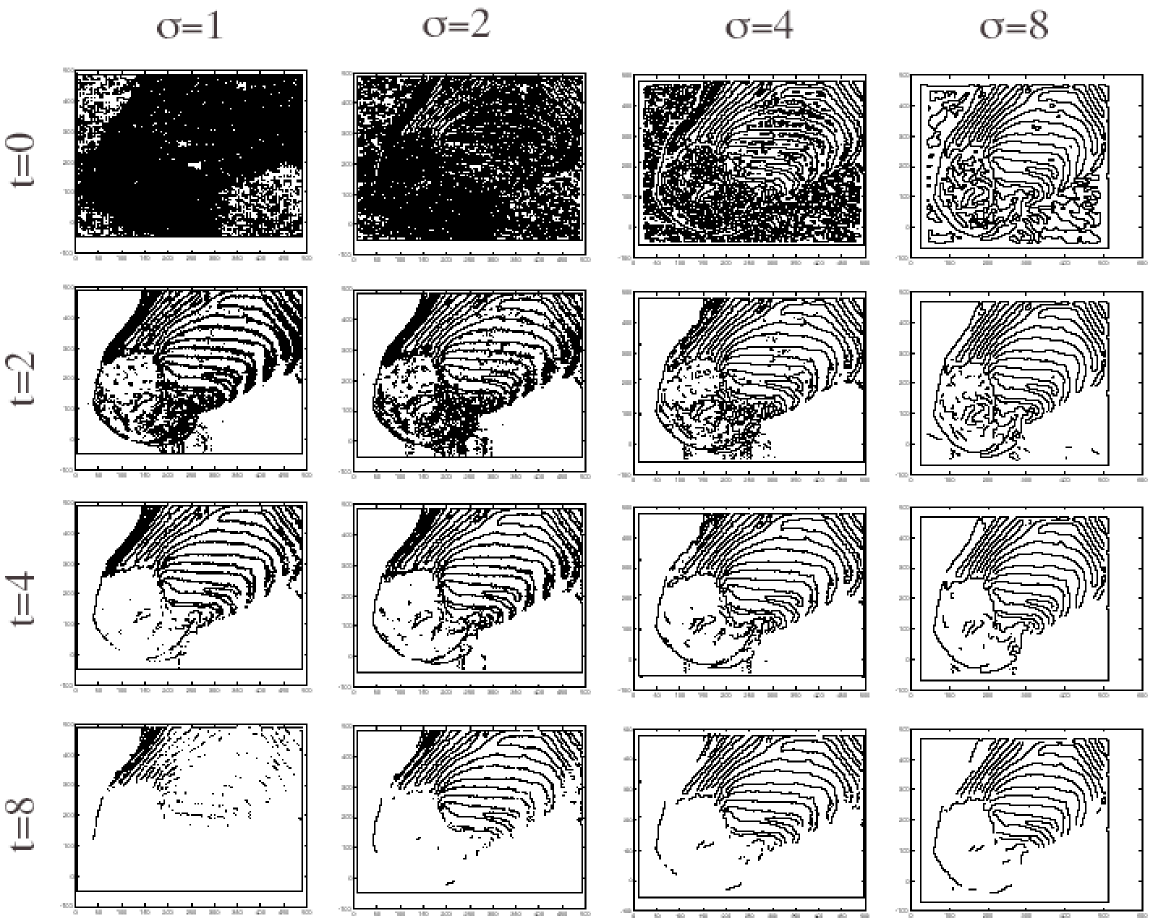
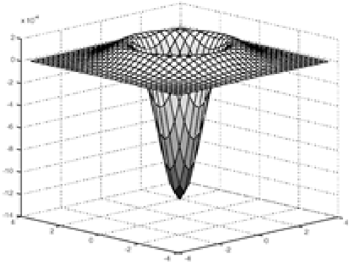


Figure 9.8. Zero crossings of the Laplacian of Gaussian for various scales and at various gradient magnitude thresholds. Each column shows a fixed scale, with t, the threshold on gradient magnitude increasing as one moves down (by a factor of two from image to image). Each row shows a fixed t, with scale increasing from σ one pixel to σ eight pixels, by factors of two. Notice that the fine scale, low threshold edges contain a quantity of detailed information that may or may not be useful (depending on one's interest in the hairs on the zebra's nose). As the scale increases, the detail is suppressed; as the threshold increases, small regions of edge drop out. No scale or threshold gives the outline of the zebra's head; all respond to its stripes, though as the scale increases, the narrow stripes on the top of the muzzle are no longer resolved.

Advantages/Disadvantages of LoG

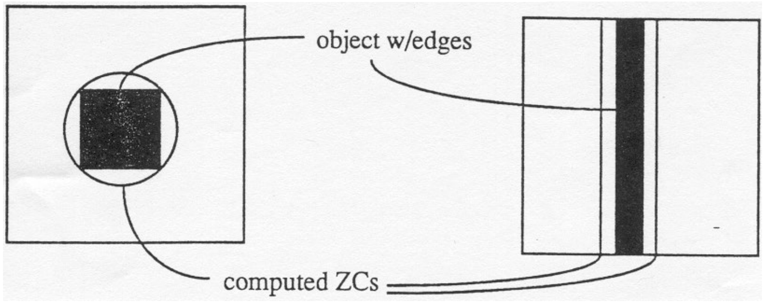
Advantages

- **Location:** ZCs are easier to find compared with peaks. Small variances have high precision while large ones are more robust.
- **Robustness:** the second derivative is much less noise-sensitive when Gaussian smoothing is applied first.

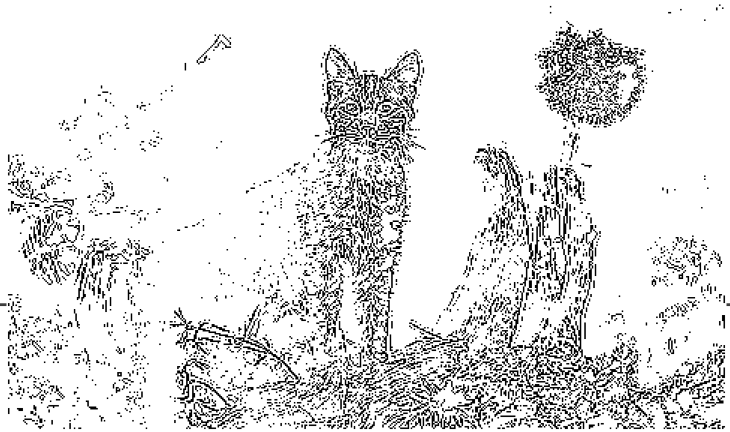
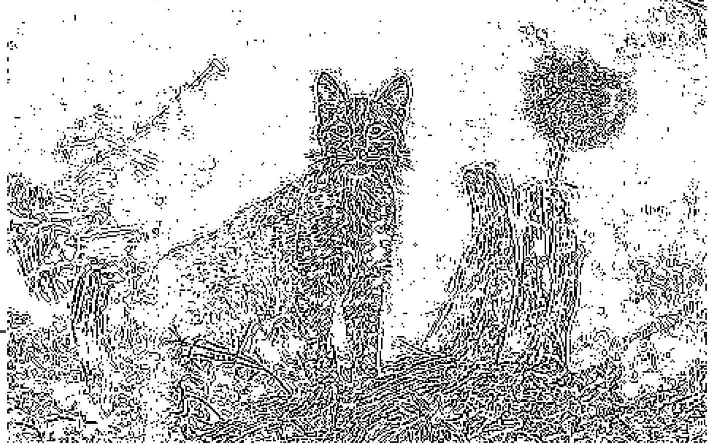


Disadvantages

- The LoG filter is not oriented (isotropic), and its response is composed of an average across an edge and one along the edge.
- ZCs are slightly displaced when the LoG is applied to objects having corners and thin line structures.

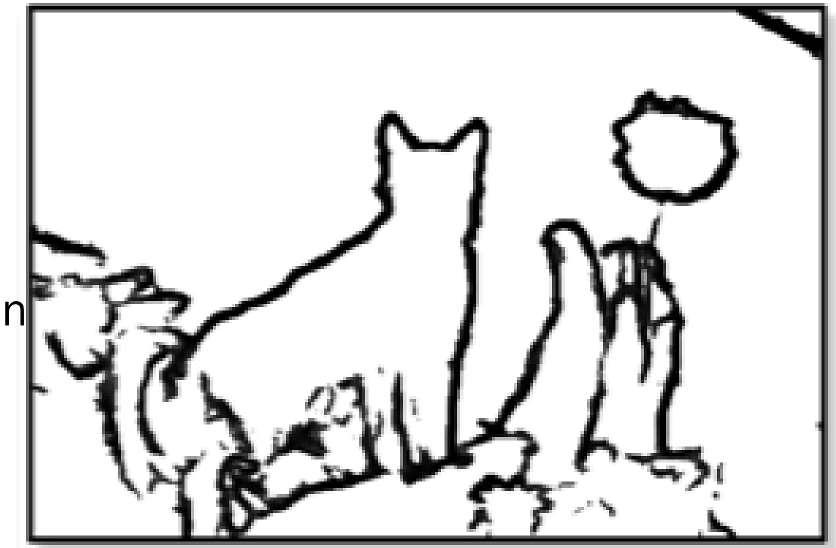


LoG Results

Ground
Truth

Deep Learning
for edge detection



Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 1395-1403).