Machine Learning for High-Speed Corner Detection

Edward Rosten and Tom Drummond

Department of Engineering,
Cambridge University, UK
{er258, twd20}@cam.ac.uk

Abstract. Where feature points are used in real-time frame-rate applications, a high-speed feature detector is necessary. Feature detectors such as SIFT (DoG), Harris and SUSAN are good methods which yield high quality features, however they are too computationally intensive for use in real-time applications of any complexity. Here we show that machine learning can be used to derive a feature detector which can fully process live PAL video using less than 7% of the available processing time. By comparison neither the Harris detector (120%) nor the detection stage of SIFT (300%) can operate at full frame rate.

Clearly a high-speed detector is of limited use if the features produced are unsuitable for downstream processing. In particular, the same scene viewed from two different positions should yield features which correspond to the same real-world 3D locations[1]. Hence the second contribution of this paper is a comparison corner detectors based on this criterion applied to 3D scenes. This comparison supports a number of claims made elsewhere concerning existing corner detectors. Further, contrary to our initial expectations, we show that despite being principally constructed for speed, our detector significantly outperforms existing feature detectors according to this criterion.

1 Introduction

Corner detection is used as the first step of many vision tasks such as tracking, SLAM (simultaneous localisation and mapping), localisation, image matching and recognition. Hence, a large number of corner detectors exist in the literature. With so many already available it may appear unnecessary to present yet another detector to the community; however, we have a strong interest in real-time frame rate applications such as SLAM in which computational resources are at a premium. In particular, it is still true that when processing live video streams at full frame rate, existing feature detectors leave little if any time for further processing, even despite the consequences of Moore's Law.

Section 2 of this paper demonstrates how a feature detector described in earlier work can be redesigned employing a machine learning algorithm to yield a large speed increase. In addition, the approach allows the detector to be generalised, producing a suite of high-speed detectors which we currently use for real-time tracking [2] and AR label placement [3].

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part I, LNCS 3951, pp. 430–443, 2006.

To show that speed can been obtained without necessarily sacrificing the quality of the feature detector we compare our detector, to a variety of well-known detectors. In Section 3 this is done using Schmid's criterion [1], that when presented with different views of a 3D scene, a detector should yield (as far as possible) corners that correspond to the same features in the scene. Here we show how this can be applied to 3D scenes for which an approximate surface model is known.

1.1 Previous Work

The majority of feature detection algorithms work by computing a corner response function (C) across the image. Pixels which exceed a threshold cornerness value (and are locally maximal) are then retained.

Moravec [4] computes the sum-of-squared-differences (SSD) between a patch around a candidate corner and patches shifted a small distance in a number of directions. C is then the smallest SSD so obtained, thus ensuring that extracted corners are those locations which change maximally under translations.

Harris[5] builds on this by computing an approximation to the second derivative of the SSD with respect to the shift The approximation is:

$$\mathbf{H} = \begin{bmatrix} \widehat{I_x^2} & \widehat{I_x I_y} \\ \widehat{I_x I_y} & \widehat{I_y^2} \end{bmatrix}, \tag{1}$$

where $\widehat{\ }$ denotes averaging performed over the image patch (a smooth circular window can be used instead of a rectangle to perform the averaging resulting in a less noisy, isotropic response). Harris then defines the corner response to be

$$C = |\mathbf{H}| - k(\operatorname{trace}\mathbf{H})^2. \tag{2}$$

This is large if both eigenvalues of $\mathbf{H}$ are large, and it avoids explicit computation of the eigenvalues. It has been shown[6] that the eigenvalues are an approximate measure of the image curvature.

Based on the assumption of affine image deformation, a mathematical analysis led Shi and Tomasi[7] conclude that it is better to use the smallest eigen value of $\mathbf{H}$ as the corner strength function:

$$C = \min\,(\lambda_1, \lambda_2). \tag{3}$$

A number of suggestion have [5, 7, 8, 9] been made for how to compute the corner strength from $\mathbf{H}$ and these have been all shown [10] to be equivalent to various matrix norms of $\mathbf{H}$.

Zheng et al.[11] perform an analysis of the computation of $\mathbf{H}$, and find some suitable approximations which allow them to obtain a speed increase by computing only two smoothed images, instead of the three previously required.

Lowe [12] obtains scale invariance by convolving the image with a Difference of Gaussians (DoG) kernel at multiple scales, retaining locations which are optima in scale as well as space. DoG is used because it is good approximation for the Laplacian of a Gaussian (LoG) and much faster to compute. An approximation