

Subpixel Corner Detection for Tracking Applications using CMOS Camera Technology

Christoph Stock, Ulrich Mühlmann, Manmohan Krishna Chandraker, Axel Pinz
Institute of Electrical Measurement and Measurement Signal Processing
Graz Univ. of Technology, Austria
email: {stock, muehlmann, manu, pinz}@emt.tugraz.at

Abstract

A multistage approach to gray-level corner detection is proposed in this paper, which is based on fast corner extraction using the Plessey corner detector combined with CMOS image acquisition technologies and localization refinement using a spatial subpixel analysis approach. The proposed corner detector detects corners as the intersection points of the involved edges, only by using a small neighborhood of the estimated corner position. With this approach it is also possible to compute the corner orientation and the dihedral angle of the corner. In comparison to the standard Plessey detector, which can show localization errors of several pixels, experimental results show an average error of only 0.36 pixels for our algorithm.

1. Introduction

This contribution is motivated by experimental results we obtained when trying to change our existing real-time tracking system to track passive landmarks (i.e. corners) instead of active retro-reflective targets (i.e. blobs, as described in [1]). Corner-based tracking leads to performance problems due to:

1. A lack of accuracy in corner position. Common corner detectors are imprecise in the localization of the exact corner position, especially for corners with a small dihedral angle. A localization error of a few pixels in the image can lead to a significant position error (especially in depth) in the calculated 3D pose, depending on the distance between the camera and the object.
2. Lower tracking rates. In an image of a “natural”, cluttered scene, many corners are present. More features have to be tracked, and complexity increase.

These problems are most severe in inside-out tracking applications, i.e. camera pose estimation from stationary natural landmarks. However, because corners are local features, they can be computed very fast, by using only a small subset of the whole image. Furthermore, CMOS cameras support random pixel access, which allows direct access to the requested sub-image.

We thus propose a multistage approach to gray-level corner detection, which is based on fast corner extraction using the Plessey corner detector [3] combined with CMOS image acquisition technologies and localization refinement using a spatial subpixel analysis approach. The proposed corner detector can detect corners as the intersection points of the involved edges, only by using a small neighborhood of the estimated corner position. With this approach it is also possible to compute the corner orientation and the dihedral angle of the corner.

A wide variety of model-based corner detectors which claim subpixel accuracy exist in literature. Most of them, like [6] or [11] directly parameterize the corner position, by modeling the center of the corner and different angles. Our approach first parameterizes the involved edges and then calculates the corner position as the intersection point of the edges detected at subpixel accuracy. This yields better localization results for corners with small dihedral angles (as also described in [5]), because the support of the edges is more reliable than the instable point-like corner feature support.

2. CMOS Camera Technology

True random access of pixels requires full asynchronous reading of the sensor. In this case fixed integration time concepts cannot be maintained. The pixels should therefore be not of the classic charge integrating type. Instead of CCD sensors an active pixel structure type is used that yields a continuous signal which is proportional to the instantaneous light intensity. Conversion of a photo current to an output voltage can be achieved by using a MOSFET which operates in weak inversion and the photo current-to-voltage conversion is logarithmic. The resulting active pixel sensor (APS) architecture with logarithmic response, shown in figure 1.a, is easy to implement in a CMOS process [8].

The overall architecture of a CMOS sensor array (see Figure 1.b) consists of a 2-D arrangement of active pixels, each provided with its own selection transistor. A pixel is selected by addressing the corresponding row and column transistors, providing a conducting line from the pixel to the output

analog-to-digital converter (ADC). The digital output of the ADCs is selected for read out by the column-select logic.

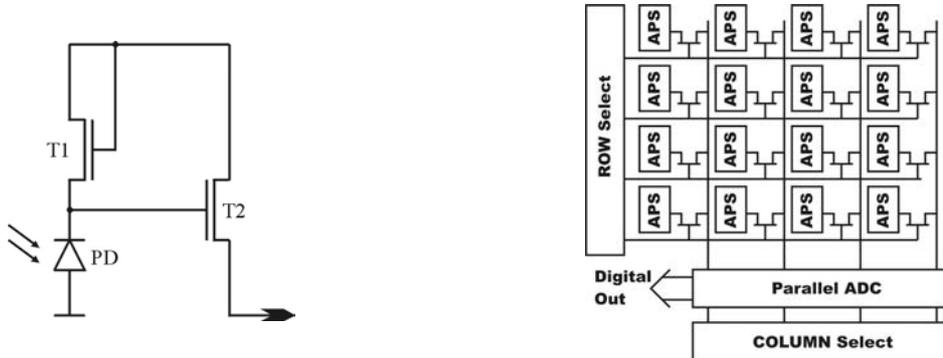


Figure 1: a) Schematic diagram of an APS with logarithmic response, consisting of just one photodiode and two MOSFETs. b) Block diagram of an 4x4 APS array with row and column select logic, and output analog-to-digital converter (ADCs).

Active pixel structures allow random access (e.g. [10]), which provides high-speed imaging of a new dimension in combination with fast self-setting bandwidth control. This means the region of interest (ROI) of an image can be individually set by the user. Reducing the window size speeds up the image acquisition. Hence, frame rates of several 100 Hz for the tracking of several windows can be achieved [2]. The possibility to examine parts of an image much faster than at video frame rates of 25 Hz or 60 Hz is the real advantage of a random access camera.

3. Spatial Subpixel Analysis of Corners

By using spatial subpixel analysis of the involved edges (similar to the approach proposed in [9]), high localization accuracy can be achieved. The idea is that mixed pixels receive their color information from simple real world situations, which can be described by a simple model, e.g. homogenous regions separated by a straight boundary. The aim of spatial subpixel analysis is to estimate the model parameters by analyzing the gray-levels (with respect to a CMOS camera with logarithmic characteristic) of the involved pixels within a local neighborhood. Our approach extends the algorithm of Steinwendner and Schneider [9] from edges to corners. In consideration of the fact that corners are intersections of two or more edges which are borders between different areas, one can use this approach to improve the localization accuracy. Several steps are necessary to estimate the correct corner position:

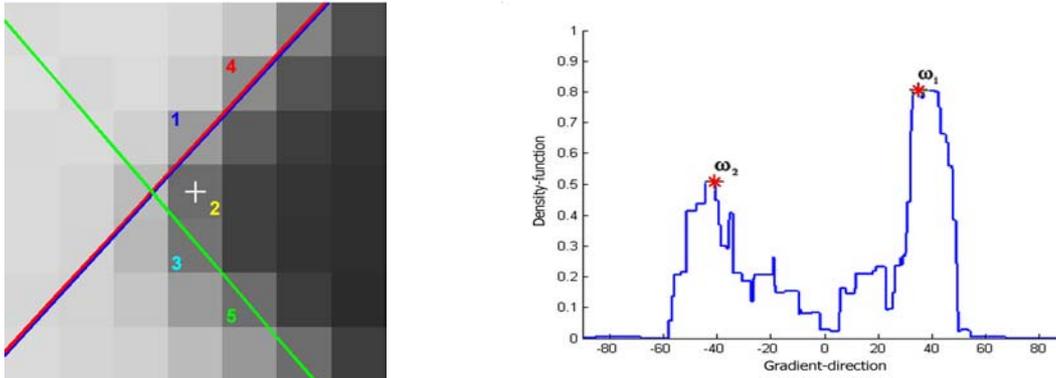
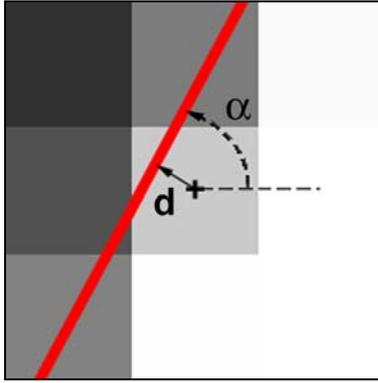


Figure 2: a) 7×7 neighborhood with the estimated corner (Plessey Detector) in the center of the image; The used boundary pixels are labeled from 1 to 5; The weighted average of the computed intersection points leads to the improved corner position. b) Associated density function with labeled dominant gradient directions ω_1 and ω_2 .

- a. *Fast corner detection:* The well-known Plessey detector [3] computes a first estimate of the corner position very fast. However, the positional accuracy is insufficient for a 3D reconstruction, therefore additional steps are necessary. The cross in Figure 2.a depicts the corner position computed by the Plessey detector. This step is only necessary for the start frame and after a reinitialisation of the tracking sequence, otherwise the prediction module of the tracker delivers the approximated corner position. Some sub-results during the corner computation using the Plessey detector can be used for further steps, e.g. gradient-directions to estimate the dominant gradient-directions of the sub-image.
- b. *Calculation of the dominant gradient-directions:* By counting the gradient-direction of all pixels in a 7×7 neighborhood of the estimated corner position and using an interpolation function [12], a weighted density function of the gradient-directions can be obtained. The gradient-directions are weighted with the gradient-magnitude to suppress gradient-directions from nearly homogeneous regions. Local maxima of the density function (see Figure 2.b) indicate the orientation (gradient direction) of the involved edges. This estimated gradient direction yields good initial points for the spatial subpixel analysis.
- c. *Extract edge pixels in a 7×7 neighborhood of the estimated corner:* Each pixel with a sufficiently high gradient-magnitude will be chosen for further calculation. These edge pixels are boundary pixels of the adjacent regions. The used boundary pixels in Figure 2.a are labeled from 1 to 5. Choosing the threshold such that each gradient-direction is represented by two edge pixels can dramatically increase the processing speed of the whole detector.
- d. *Compute line parameters using spatial subpixel analysis:* The pixel value of each extracted edge pixel is a mixture of the pure gray-values of the adjacent regions. The orientation α and the distance d of the edge to the pixel center is computed by using the parameter model from [9] on a 3×3 neighborhood.



$$\sum_{i=1}^9 \left(p_i^{original} - p(d, \alpha)_i^{estimated} \right)^2 \xrightarrow{d, \alpha} \min \quad (1)$$

Figure 3: 3x3 neighborhood with labeled edge model; d denotes the normal distance between the center of the 3x3 neighborhood and α denotes the angle between the edge and the horizontal line of the image.

In Equation (1) $p_i^{original}$ denotes the original pixel-value at position i within the 3x3 neighborhood and $p(d, \alpha)_i^{estimated}$ denotes the pixel-value at position i originating from the estimated edge, which can be described by distance d and angle α .

To decrease the computational cost of estimating the line parameters, the previously calculated dominant gradient directions can be used as initial values for the optimization problem.

- e. *Removing wrong edge candidates:* The previously described parameter model is inappropriate at pixels nearby the exact corner position, because more than one edge can be found within this neighborhood. The algorithm tries to minimize the error function (see Equation 1), which yields wrong parameters for the edge model. These edge candidates can be removed by comparing the angle α with the dominant gradient-direction. Each edge which is not aligned with one dominant gradient-direction (calculated in a previous step) will be removed. (see Figure 2.a: The edges according to pixel 2 and 3 are removed).
- f. *Calculation of the intersection point:* Each point of intersection of the remaining edges is a possible candidate for the exact corner position. Experimental results show that these points vary only little (see Figure 2.a). However, to further increase accuracy, a consensus (e.g. weighted average) can be calculated.

With this procedure it is possible to detect the corner position at subpixel accuracy. Furthermore, it is possible to calculate the dihedral angle and the orientation of the detected corner, only by analyzing the dominant gradient-directions.

4. Evaluation Results

In order to determine the claimed capability for tracking applications, this section analyzes the localization accuracy and the real-time behavior of the subpixel corner detector. Several methods are known (as described in [7]) to evaluate a corner detector. In this work we analyze the localization performance through groundtruth-verification.

Subsections 4.1 and 4.2 show results for the localization accuracy (for synthetic images as well as for real images) and finally subsection 4.3 deals with the runtime-behavior of the algorithm.

4.1 Localization Accuracy on Synthetic Images

Based on a well known groundtruth, which is generated by a synthetic rendering algorithm, the localization error is computed. The rendering algorithm creates images at subpixel coordinate locations. The behavior of the corner detector is analyzed for different dihedral angles of the corner.

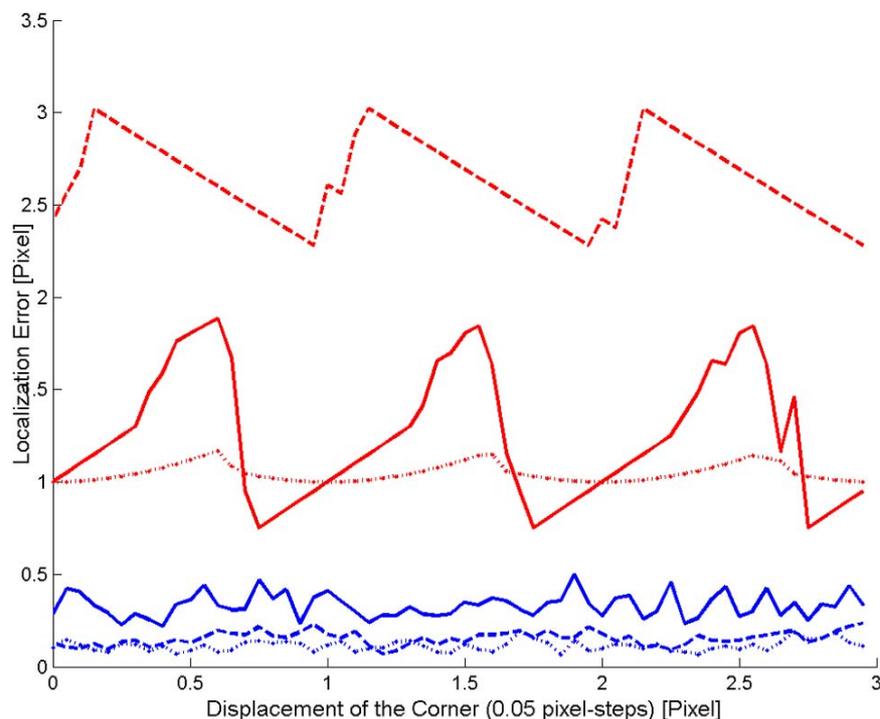


Figure 4: Localization error for a synthetic rendered corner position which is shifted over 3 pixels (step-size 0.05 pixel), with different dihedral angles of the corner; The 3 lines at the top of the graph are derived from the Plessey operator, the 3 lines below are originated by the new subpixel corner detector; The dotted lines describe the error for 90° corners, the solid ones for 60° corners, and the dashed lines describe the localization error for corners with a dihedral angle of 30°.

Figure 4 describes the localization performance of the subpixel corner detector compared with the used Plessey detector. A synthetic corner with a known position is shifted along the x-axis of the image, with a step-size of 0.05 pixel. The distance between the true corner position and the detected corner position is measured for each image. To each image Gaussian noise ($\mu=0$, $\sigma^2=0.005$) is added and the distances are averaged over 10 images. One can see that the new subpixel corner detector yields up to a factor of 10 better results, especially for corners with a small dihedral angle. Due to the fact that the image is shifted over 3 pixel, the graph for the Plessey detector has a very similar pattern for each pixel, which only varies by noise.

4.2 Localization Accuracy on Real Images

By using an xy-table (as shown in Figure 5.a) with a minimal step-size of 7.5 μm and simple targets (with different orientations and dihedral angles, see Figure 5.b), a precise groundtruth can be generated. The localization accuracy can be determined by comparing the detected corner locations with the groundtruth. 70 images shifted by 75 μm along the x-axis of the xy-table were captured with the CMOS camera.

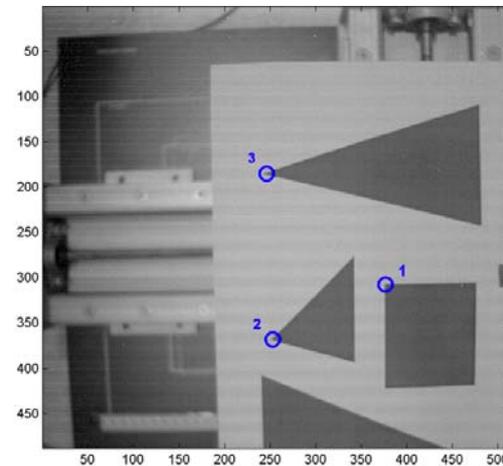


Figure 5: a) Experimental set-up consisting of an xy-table and a stationary CMOS camera, to evaluate the localization accuracy on real images. b) Image of the shifted target, captured by the CMOS camera.

Figure 6: Experimental set-up to determine the runtime performance of the subpixel corner detector

Figure 7 illustrates the localization performance by comparing the detected positions with the real positions. One can see that the results of the subpixel corner detector (solid line) are much better aligned with the real position (dash-dotted line) than the positions of the Plessey detector (dashed line).

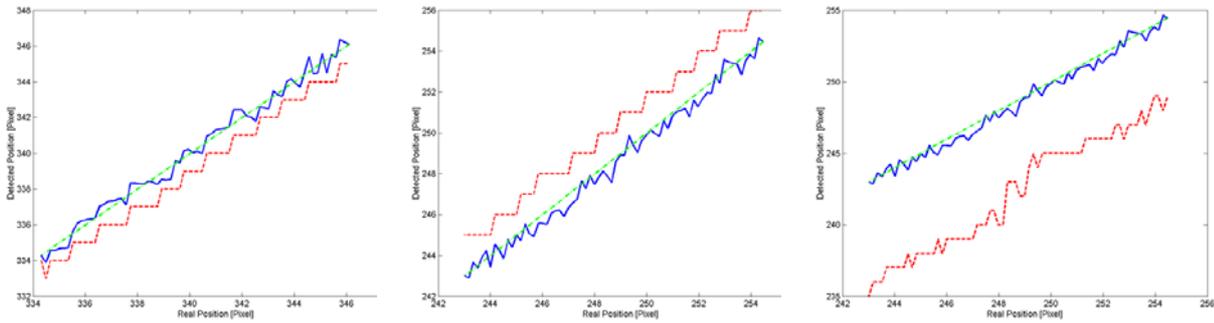


Figure 7: Localization error for different dihedral angles a) 90°, b) 60°, c) 30°; The steps of the Plessey function depend on the image quantization; The dash-dotted line shows the exact displacement (gradient=1); The solid line shows the results from the subpixel corner detector.

The average localization error of the subpixel corner detector is 0.3572 pixel with a variance of 0.0572 pixel. This averaged localization error is independent of the dihedral angle of the analyzed corner. In contrast to this results, the Plessey detector yields significantly higher localization errors (0.6714 - 6.4209 pixels and σ^2 varies between 0.0980 and 0.9427), which depends on the dihedral angle of the corner.

4.3 Runtime Analysis

A first experimental set-up for tracking applications has been developed (see Figure 6). Special targets with unique cross-ratios and distinct corners are used for subsequent pose estimation. These features simplify the correspondence search between the detected corners and the known 3D positions of the targets in world coordinates. A CMOS camera, with a spatial resolution of 512x512 pixel, mounted on a pan-tilt-head captures up to 10 corners and uses at least 4 corners to compute the pose.

With this experimental set-up a processing speed for image acquisition and subpixel localization of approximately 240 corners/sec can be achieved. The experiment runs on a Linux-based AMD 1700XP processor. This satisfies the claimed real-time performance of the system.

Another experiment combines CMOS sensor-readout, subpixel corner detection and subsequent pose estimation. Depending on the number of corners used for the pose estimation tracking rates of 20 – 40 Hz can be achieved. During this experiment no attention was turned on the exact pose verification, due to the fact that no reliable groundtruth was available.

5. Conclusion and Outlook

We have presented a new model-based corner detection scheme combining the advantages of CMOS camera technology and spatial subpixel analysis which achieves high localization accuracy and is able to increase the reliability of tracking applications. By using the random access of modern CMOS arrays real-time behavior can be achieved.

Future work focuses on adapting the spatial subpixel algorithm to the logarithmic response of the CMOS sensor. By doing so, the linear approximation of the modeled gray-values will be removed by a logarithmic one. Some modifications can be done on the optimization problem, to minimize the number of steps needed to reach the demanded edge-fitting error. This will lead to a further increase in processing speed of the algorithm.

Acknowledgement:

This research was supported by the following projects.: VAMPIRE – Visual Active Memory Processes and Interactive REtrieval (EU-IST Programme – IST-2001-34401), FWF Smart Tracking – Autonomous Exploration of Unknown Environment by Fusion of Stereo Vision and Inertial Sensors (P 15748).

References

- [1] Brandner M., Ganster H., Pinz A., Ribo M.; 3D Optical Tracking of Retroreflective Targets for AR Applications; 25th Workshop of the Austrian Association for Pattern Recognition (ÖAGM/AAPR); pp 95-102, 2001.
- [2] Ganster H., Brandner M., Pinz A., Ribo M.; Optical tracking with CMOS cameras; 25th Workshop of the Austrian Association for Pattern Recognition (ÖAGM/AAPR); pp 103 – 109, 2001.
- [3] Harris C. and Stephens M.; A combined corner and edge detector, Proc. of 4th Alvey Vision Conference., pp 189-192, 1988.
- [4] Jähne B., Haußecker H., Geißler P.; Handbook of Computer Vision and Applications; Vol. 1; Academic Press, 1999; ISBN 0-12-379771-3
- [5] Rohr K.; Localization Properties of Direct Corner Detectors; Journal of Mathematical Imaging and Vision, 4; pp 139-150; 1994
- [6] Rohr K.; Recognizing Corners by Fitting Parametric Models; Int. Journal of Computer Vision, 9:3; pp 213-230; 1992
- [7] Schmid C., Mohr R., Baukhage Ch.; Evaluation of Interest Point Detectors; Int. Journal on Computer Vision; 37(2); pp 151-172; 2000
- [8] Seitz P.; Solid-State Image Sensing, in [4]; pp 165-222
- [9] Steinwendner J. and Schneider W.; Algorithmic improvements in spatial subpixel analysis of remote sensing images; Proc. of 22nd ÖAGM Workshop; pp 205-213, 1998.
- [10] Ooi R., Hamamoto T., Naemura T., Aizawa K.; Pixel independent random access image sensor for real time image-based rendering system; Proc. ICIP, Vol. 2, pp 193 –196, 2001.
- [11] Parida L., Geiger D., Hummel R.; Junctions: Detection, Classification and Reconstruction; IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20; pp 687-698;1998
- [12] Yin and Balchen; Corner characterization by statistical analysis of gradient directions; Proc. ICIP; Vol. 2; pp 760-763, 1997.