Automated Thresholding for Low-Complexity Corner Detection

Nirmala Ramakrishnan, Meiqing Wu, Siew-Kei Lam, Thambipillai Srikanthan
School of Computer Engineering
Nanyang Technological University
Singapore
{rama0056, wume0007}@e.ntu.edu.sg, {assklam, astsrikan}@ntu.edu.sg

Abstract— **Widely-used corner detectors such as Shi-Tomasi and Harris necessitate the selection of a threshold parameter manually in order to identify good quality corners. The recent attempts based on trial-and-error methods for threshold setting are time-consuming, making them unsuitable for low-cost and embedded video processing applications. In this paper we propose a novel automated thresholding technique for Shi-Tomasi and Harris corner detectors based on an iterative pruning strategy. The proposed pruning strategy involves the rapid extraction of potential corner regions and their evaluation for detecting corners. This pruning strategy is applied iteratively until the required number of corners is identified without necessitating the selection of the threshold parameter. As the complex corner measure computations of the Shi-Tomasi and Harris detectors are only applied to very small regions selected by the proposed pruning method, significant savings in computation is also achieved. In addition, the pruning strategy is computationally simpler, making it suitable for deployment in low cost and embedded applications. Our evaluations on the NiOS-II embedded platform show that the proposed automated thresholding technique is able to achieve an average speedup of 67% in Shi-Tomasi and 51% in Harris, with almost no loss in accuracy. The proposed method to identify corners without the manual selection of a threshold parameter makes it ideal for corner detection on a wide range of imagery where the quantity and quality of corners is not known a priori such as in video processing applications.**

Keywords— corner detection, threshold, low-complexity, automated, pruning, Shi-Tomasi, Harris

I. INTRODUCTION

Corner detection is the fundamental step in many computer vision algorithms such as registration and tracking. Increasingly, these algorithms are being applied to low-resource embedded platforms that process video imagery where the quantity and quality of corners is varying across frames and is not known a-priori. Application areas of such platforms include vision-based navigation and landing of unmanned aerial vehicles [1, 2], image guided medical interventions [3] and robot navigation [4]. In all these embedded applications, manual intervention to set the optimal parameters for corner detection is not feasible. In order to increase the robustness of these platforms, there is a need for automated parameter selection for corner detection which can adapt to the image content. In addition, methods employed to automate the parameter selection need to be highly compute-efficient in order to cater to the real-time requirements of these applications.

Shi-Tomasi [5] and Harris [6] are very popular corner detectors and have been used extensively for various real-time video processing applications [1-4, 7, 8]. In practice, the parameter to threshold, which is based on the quality of corners, is set conservatively such that the corner detectors can extract the required number of corners on a wide range of image content. However the conservative parameter is set without taking into account the image content. This could lead to high computation inefficiency, as setting a low threshold can result in unnecessary computations particularly if the images are rich in corners. Fig. 1 shows an example of two images with widely different image content. A high threshold will result in insufficient number of corners detected as in 1(c) whereas a low threshold will incur unnecessary computations for rich images as in 1(f). Automatically setting optimal thresholds based on image content can minimize unnecessary computations, which is critical in real-time systems.

In this paper, we propose an automated thresholding method for low-complexity corner detection with the Shi-Tomasi and Harris algorithms. This method applies the compute-efficient pruning technique in [9] and [10] in an iterative manner such that in each iteration, corners are extracted from a small pool of corner candidates that is selected by the pruning process. The iterations stop when the required number of corners is found. The proposed method eliminates the need to specify the threshold parameter manually and results in adapting the computations for corner detection to the image content. For images with richer image content, the method converges faster using lesser number of iterations to extract the required number of corners, compared to images with fewer corner regions. The proposed method results in corner detection that is both adaptive to the image content and computationally efficient.

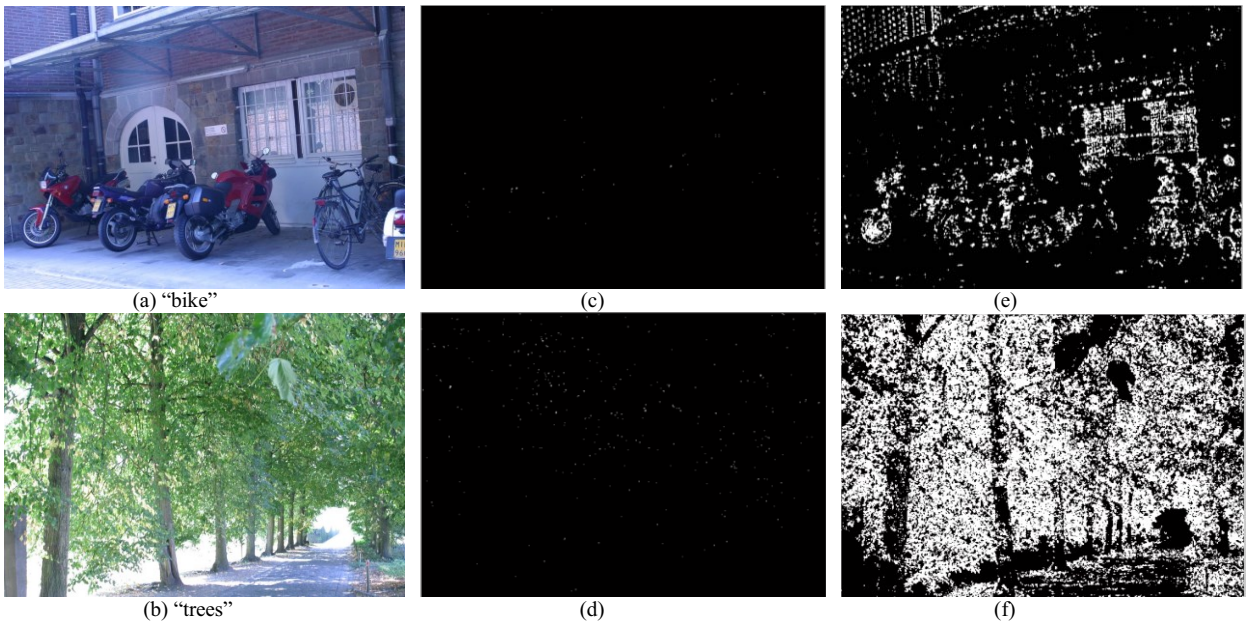


Fig. 1. Example of inappropriate threshold settings (a) "bike" image poor in corners (b) "trees" image richer in corners, Corner regions at "too high" threshold=0.3 for (c) "bikes" resulting in only 92 corners and (d) "trees" resulting in 300 corners, and (e) and (f) Corner regions at "too low" threshold=0.01.

II. Related Work

In this section we review the literature for low-complexity and/or automated corner detection. It is well recognized that corner detection is compute-intensive. There are typically two approaches that have been proposed in the literature for increasing the computation efficiency of corner detectors: the first employs hardware accelerators and the second focuses on algorithmic innovations. Hardware accelerators such as Application Specific Integrated Circuits (ASICs) [11], Field Programmable Gate Array (FPGA) [12] and GPU implementations [13, 14] of Shi-Tomasi and Harris have been proposed. Algorithmic innovations such as [15, 16] that do not depend on the underlying hardware architecture also result in low-complexity corner detectors. In our earlier work [9, 10], we proposed a low-complexity pruning technique for Shi-Tomasi and Harris corner detector.

In all the above techniques, the complexity of corner detection is reduced without being aware of the image content. However in natural images, the number of pixels that are salient and potentially corners is small compared to the entire image. Therefore there is huge opportunity to reduce complexity of corner detectors by adapting to the image content on the fly and working "less" when not needed, such as in smooth regions. In [17], an adaptive sampling technique is proposed for FAST corner detector, that locally adapts the computation steps for corner detection based on the image content and results in lower computations in homogenous regions that are not likely to contain corners. Pruning in [9, 10] also removes the obvious non-corner regions quickly and generates a small corner candidate set that is used for corner measure computations. However, it is dependent on a user defined threshold which is set empirically in Shi-Tomasi/Harris detectors.

The problem of automatic optimal parameter selection has been addressed in the context of edge detection in [18] and later generalized to all types of features in [19]. It involves a statistical analysis of detections by using a range of parameter sets and identifying the parameter set that results in maximum number of useful features and minimum amount of noise. However, it is not practical to use this technique in a real-time resource-constrained system, as the detector is executed as many times as the evaluated parameter sets. The OpenCV implementation that selects the optimal parameters for detecting corners in [20] also requires multiple executions of the detector and hence, it is not a low-complexity solution.

To the best of our knowledge, the work in this paper is the first to focus on a low-complexity automated thresholding technique for Shi-Tomasi and Harris.

III. Proposed Automated Thresholding Technique

In the conventional Shi-Tomasi and Harris corner detectors, the corner measure C is computed on all pixels of the image. A threshold is applied on the corner measure to discard the non-corner regions. The corner candidates are then sorted based on the value of C, and non-maximal suppression is applied to extract the final corners. In [9], we proposed a computationally less complex pruning measure P, to select corner candidates. In [10], we enhanced the pruning process by adding a computationally efficient edge removal step. The complex corner measure was applied to only the small pool of corner candidates selected after the pruning process in [9, 10].

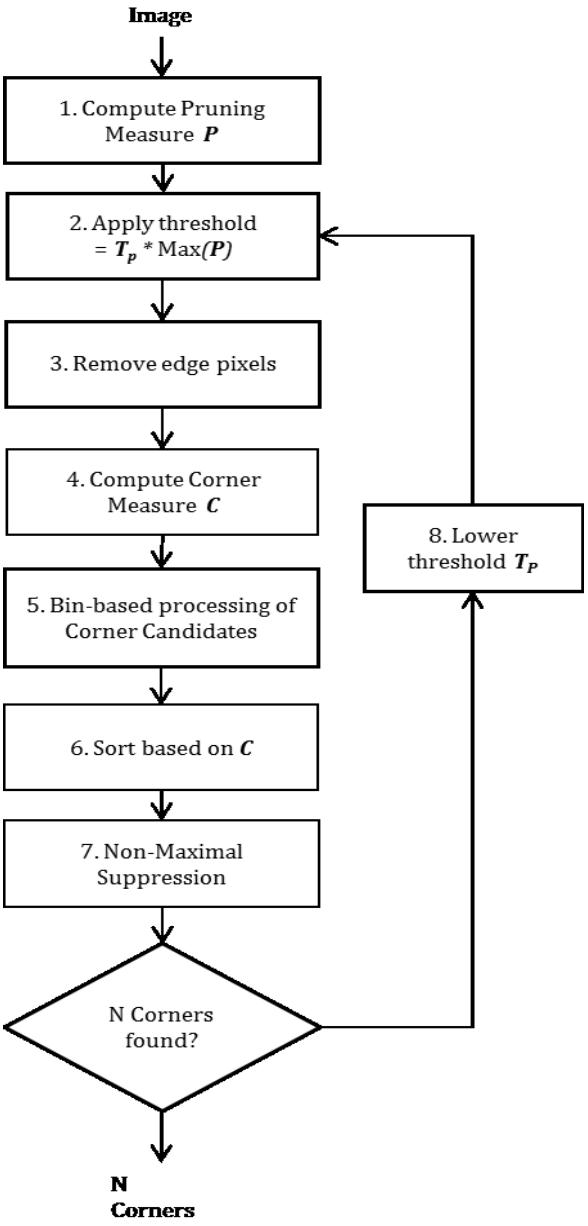


Fig. 2. Proposed automated thresholding technique.

The pruning requires a user defined threshold to be applied to pruning measure *P* to select the corner candidates. In the proposed automated thresholding technique, we apply the pruning methods in [9] and [10] with the Shi-Tomasi and Harris corner detectors, in an iterative manner and eliminate the need to specify the threshold for the pruning and corner measures as user input. Fig. 2 shows the steps in the proposed automated thresholding technique. As in [10], the pruning measure *P* is computed on all pixels in the image. The pruning threshold T_p is lowered in each iteration to release corner candidates in a controlled manner. The corner measure *C* is then computed only on this small pool of corner candidates. A novel bin-based approach is used to collect the corner candidates and select the best corners without the need to threshold based on the corner measure *C*. The iterative

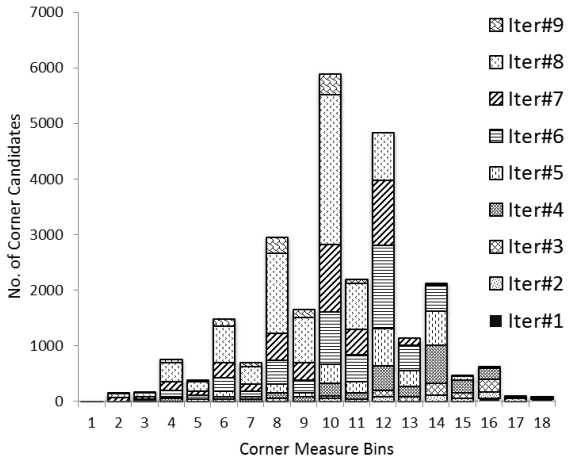


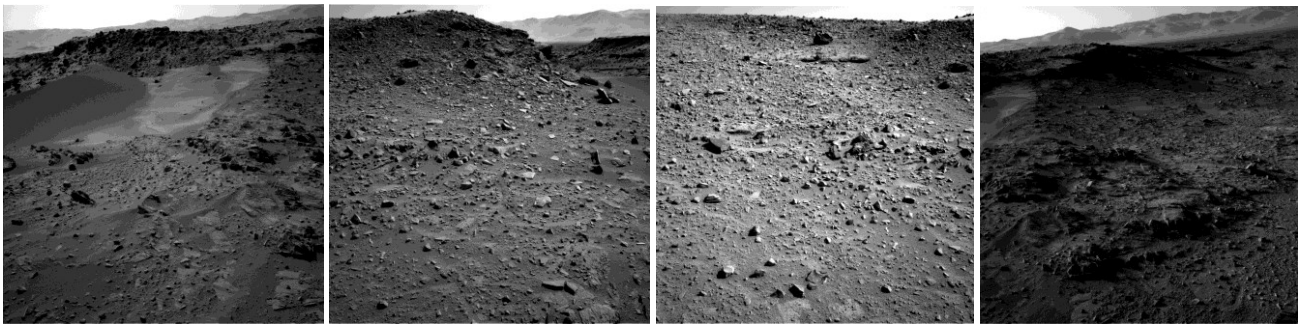
Fig. 3. Corner candidates count in the corner measure bins for successive iterations in the iterative pruning strategy.

processing stops when the required number of corners has been found.

The pruning measure *P* is derived directly from the corner measure *C* of the Shi-Tomasi and Harris detectors and therefore, there is a high degree of correlation between these two measures – i.e., if a pixel has a high value of *P* then it is highly likely that it will have a high value of *C* and if a pixel has a low value of *P* then it is highly likely that it will also have a low value of *C*. Due to this correlation, when T_p is lowered iteratively, initially the corner candidates released have higher values of corner measure *C*, but for subsequent iterations, the released corner candidates have lower values of *P* and they also have lower values of *C*. We exploit this behavior by setting up corner measure bins for the various ranges of values of corner measure *C* to hold the corner candidates temporarily. Fig. 3 shows how the corner candidates released in each iteration of lowering the T_p threshold, fill the corner measure bins. Bin 1 is for the lowest value of *C* and bin 18 is for the highest value of *C*. It is clear that the bins for the higher values of *C* are populated by the initial iterations with higher values of T_p and stop growing for subsequent iterations. When a bin stops growing, it is referred to as a saturated bin. Once the bins saturate, we have effectively extracted the corner candidates in the range of corner measure values represented by those bins without explicitly setting the threshold for the corner measure. Sorting and non-maximal suppression can now be applied to these candidates and final corners extracted. Further iterations of lowering *Tp* and releasing corner candidates are necessary only if the final corners detected in the current iteration are lesser than the required number of corners.

For natural images, the number of corner candidates increases non-linearly when the pruning measure and corner measure decreases as there are very few high quality corner regions compared to smooth regions. This governs the way the corner measure bins are setup and the steps by which the threshold *Tp* is lowered and a similar strategy is employed for both cases. The bin boundaries and the threshold steps are

"bike" "trees" "mars1"



"mars2" "mars3" "mars4" "mars5"

Fig. 5. Image dataset used for evaluation of the proposed automated thresholding technique.

setup as a factor of the maximum value of the corner measure and the pruning measure respectively. For example, the last bin has a range of (1-0.4) and it holds all corner candidates with values ($maxC - 0.4 \times maxC$) where $maxC$ is the maximum value of the corner measure for the given image. Similarly the threshold Tp for the first iteration is $0.4 \times maxP$ where $maxP$ is the maximum value of the pruning measure for the given image. Each lower bin is of a smaller range than its predecessor in order to keep the number of candidates in each bin of roughly an equal number. Similarly, in order to ensure the number of corner candidates released in each iteration does not increase drastically, every subsequent step in the threshold Tp is smaller than the step before. Fig. 4 shows the scheme used for the corner measure bin boundaries and steps for the threshold Tp. First the major bins are setup such that the range of thresholds for every bin is half that of its higher neighbor major bin. Every major bin is then sub-divided into a 2:1 distribution to ensure that there is not a huge surge of pixels for a new major bin. When assigning every corner candidate to a bin, we first find the major bin and then the sub-bin. The threshold Tp is also lowered non-linearly following the steps as setup by the sub-bin boundaries in Fig. 4.

After the corner candidates have been classified into the corner measure bins, at the end of each iteration we monitor the growth of the bins to detect saturating bins. For this, we scan the bins from the highest corner measure, and stop at the first bin that is growing. The corner candidates from the saturating bins are then released and sorting and non-maximal suppression is applied to these candidates to extract the final corners.

The novel bin-based approach to collect the best corners in an image allows the proposed technique to adapt to diverse image content in the following ways. First, there is no user

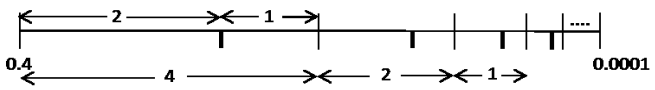


Fig. 4. Threshold setup for pruning threshold in each iteration and bin boundaries for corner measure bins. Major bin boundaries are setup such that they are reduced by half for each bin. Within each major bin, the sub-bin boundary is setup as a 2:1 distribution. The actual threshold steps are given in Table I.

input required for the threshold settings and therefore as long as an image has sufficient number of corner regions the proposed technique will always be able to find the required number of corners. Second, computations incurred for processing the corner candidates are adjusted based on the image content. If an image contains many rich corner regions, then the corner measure bins for the higher ranges of corner measure values saturate with large number of corner candidates and the technique converges faster. For an image that contains fewer high-quality corner regions, the proposed technique takes more iterations to converge. In effect, computations are incurred only when needed by being adaptive to the image content.

IV. Evaluations and Results

We evaluate the proposed automated thresholding technique (referred to as ST+AT for automated thresholding with Shi-Tomasi and H+AT for automated thresholding with Harris) in comparison to the fixed/manual threshold techniques:

1. Conventional Shi-Tomasi and Harris detectors (referred to as ST for Shi-Tomasi and H for Harris), and

2. Pruning based Shi-Tomasi and Harris detectors (referred to as ST+P for Shi-Tomasi and H+P for Harris) – when pruning in [10] is applied to both the detectors.

We use the images – "bike", "trees" and images of Mars from the navigation camera of the Curiosity rover [21] as shown in Fig.5 for our evaluations. Images "bike" and "trees" have an image size of 1000x700 pixels. Images "mars1-5" have an image size of 1024x1024. A maximum of 300 corners is extracted in all cases.

TABLE I. THRESHOLD STEPS FOR THE AUTOMATED THRESHOLDING TECHNIQUE

Iteration No.	*Threshold T_p*	*Step Size*
1	0.4000	-
2	0.2662	0.1338
3	0.1993	0.0669
4	0.1324	0.0669
5	0.0989	0.0335
6	0.0654	0.0335
7	0.0487	0.0167
8	0.0320	0.0167
9	0.0236	0.0084
10	0.0153	0.0084
11	0.0111	0.0042
12	0.0069	0.0042
13	0.0048	0.0021
14	0.0027	0.0021
15	0.0017	0.0010
16	0.0006	0.0010
17	0.0001	0.0005

TABLE II. CORNER MEASURE BIN BOUNDARIES FOR THE AUTOMATED THRESHOLDING TECHNIQUE

Bin No.	*Threshold Factors*		*Bin Width*
	Min	*Max*	
1	0.4000	1.0000	0.6000
2	0.2662	0.4000	0.1338
3	0.1993	0.2662	0.0669
4	0.1324	0.1993	0.0669
5	0.0989	0.1324	0.0335
6	0.0654	0.0989	0.0335
7	0.0487	0.0654	0.0167
8	0.0320	0.0487	0.0167
9	0.0236	0.0320	0.0084
10	0.0153	0.0236	0.0084
11	0.0111	0.0153	0.0042
12	0.0069	0.0111	0.0042
13	0.0048	0.0069	0.0021
14	0.0027	0.0048	0.0021
15	0.0017	0.0027	0.0010
16	0.0006	0.0017	0.0010
17	0.0001	0.0006	0.0005

For the conventional detectors (ST, H) we set a threshold for corner measure of 0.01 for Shi-Tomasi and 0.001 for Harris. For the pruning based detectors (ST+P, H+P) we set the threshold for the pruning measure of 0.01. The threshold steps and corner measure bin boundaries in the proposed automatic thresholding technique are setup to release corner candidates that have a non-linearly increasing distribution as shown in Table I and II respectively. Table I shows that as iterations increase, the step size by which the pruning threshold Tp is lowered reduces and this ensures that corner candidates released in a given iteration are not very large in number compared to the previous iteration. Table II shows the bin boundaries for the corner measure bins used to collect the corner candidates, are also setup such that the width of bins for lower threshold values are much smaller than the higher threshold values.

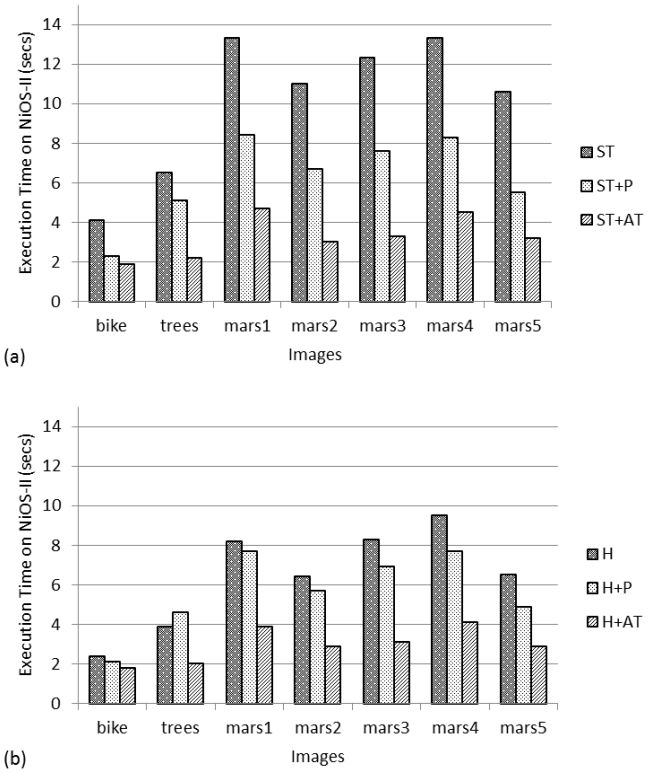


Fig. 6. Efficiency results for the proposed automated thresholding technique for (a) Shi-Tomasi and (b) Harris corner detectors

TABLE III. ACCURACY RESULTS FOR PROPOSED AUTOMATED THRESHOLDING TECHNIQUE

Corner Matches with Conventional Algorithms (%)				
	ST+P	**ST+AT**	**H+P**	**H+AT**
bike	93.1	93.1	90.7	90.7
trees	97.9	97.9	95.1	95.1
mars1	100	100	100	100
mars2	99.7	99.3	99.7	99.3
mars3	100	100	99.3	99.3
mars4	100	100	99.7	99.7
mars5	97	96.7	97	97

The compute-efficiency of the proposed automated thresholding technique is demonstrated by running all the algorithms on the Nios-II embedded platform [22] and measuring the execution time. All the algorithms evaluated are executed as applications on the NiOS-II/f fast core with the on-board cache and floating-point units enabled. Fig. 6 shows the

execution time for Shi-Tomasi and Harris detectors. Although images "bike" and "trees" have the same image size, both the conventional detectors (ST and H) have a longer execution time for "trees" compared to "bike" as it has more textures and hence more corner candidate regions to process before the required number of corners can be extracted. We observe a similar difference in execution times for the Mars images – e.g. "mars1" which has a higher contrast and hence more corner candidates has a higher execution time compared to "mars5". When pruning with the fixed threshold is applied (ST+P and H+P), "bike" shows computation savings because the corner measure is applied to a small pool of corner candidates selected by the pruning process. However, "trees" does not show substantial improvement in the execution time in ST+P and incurs an overhead in H+P. This is because the threshold for the pruning and corner measure is too low for this image and the pool of corner candidates is very large. These results show that if the image content is unknown then setting a fixed threshold in a conservative manner is not computationally efficient as the number of corner regions that are selected by the fixed threshold is dependent on the nature of the image content and therefore, unnecessary computations are incurred for images rich with corner regions.

The automated thresholding technique (ST+AT, H+AT) alleviates this problem and exploits the nature of the image content to achieve high computation savings. For ST+AT, an average speedup in execution time of 67% is achieved compared to the conventional algorithm, ST. HT+AT achieves an average speedup is 51%. Rich images such as "trees" have a low speedup with fixed threshold pruning – 22% with ST+P and -18% with H+P (i.e. 18% overhead). But with automated thresholding, the pruning is adaptive to the image content and the required number of corners is found at a higher threshold resulting in a speedup in execution time of 66% with ST+AT and 49% with H+AT.

The accuracy of the proposed method is measured by counting the number of corner matches between the corner sets from the proposed methods and the conventional Shi-Tomasi and Harris detectors. This metric shows how close the corner sets generated by the proposed methods are with those generated by the conventional detectors. When the number of matched corners is higher, the proposed methods can replace the conventional detectors seamlessly.

We report the corner matches as a percentage – i.e. if 270 corners match out of a total of 300 corners, then we report the match as 90%. Table IV shows that the corner matches for the automated thresholding technique (ST+AT, H+AT) is the same as that of fixed thresholding pruning (ST+P, H+P) and is an average of 98% for Shi-Tomasi and 97.3% for Harris. This shows that the strategy to saturate bins during iterative pruning extracts the highest quality corners resulting in a corner set that is a very close match of the conventional detector.

V. CONCLUSIONS

We have presented an automated thresholding technique in this paper that makes the widely used Shi-Tomasi and Harris corner detectors both adaptive to image content as well as compute-efficient. The proposed technique eliminates the need to explicitly set the threshold for selecting the best corners and employs a novel iterative pruning strategy to automatically arrive at the optimal threshold for corner selection depending on the image content. The technique leads to low-complexity corner detection due to these reasons. First, the pruning process employed generates small pools of corner candidates and the complex corner measure is applied only to this pool of candidates. Second, the iterative pruning strategy enables the proposed technique to adapt the computations to the nature of the image content such that if an image is rich with corners it converges faster to detect the required number of corners. Finally, the pruning steps are themselves computationally efficient compared to the conventional corner measure. The proposed technique shows a speedup of an average of 67% and 51% in execution time compared to the conventional detectors, for Shi-Tomasi and Harris respectively while resulting in corner sets that have an average of 98% match with the conventional detectors. This enables the use of these detectors in modern embedded and low cost video processing applications that need to process imagery that is unknown a priori.

REFERENCES

[1] J. Courbon, Y. Mezouar, N. Guénard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice,* vol. 18, pp. 789-799, 7// 2010.

[2] A. Johnson, J. Montgomery, and L. Matthies, "Vision Guided Landing of an Autonomous Helicopter in Hazardous Terrain," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 3966-3971.

[3] D. J. Mirota, M. Ishii, and G. D. Hager, "Vision-Based Navigation in Image-Guided Interventions," *Annual Review of Biomedical Engineering,* vol. 13, pp. 297-319, 2011.

[4] A. Schmidt, M. Kraft, and A. Kasiński, "An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation," *Computer Vision and Graphics,* vol. 6375, pp. 251-259, 2010/01/01 2010.

[5] S. Jianbo and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994, pp. 593-600.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147-151.

[7] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking," *International Journal of Computer Vision,* vol. 94, pp. 335-360, 2011/09/01 2011.

[8] A. Gil, O. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM," *Machine Vision and Applications,* vol. 21, pp. 905-920, 2010.

[9] M. Wu, N. Ramakrishnan, S.-K. Lam, and T. Srikanthan, "Low-complexity pruning for accelerating corner detection," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, 2012, pp. 1684-1687.

[10] N. Ramakrishnan, M. Wu, S.-K. Lam, and T. Srikanthan, "Enhanced low-complexity pruning for corner detection," *Journal of Real-Time Image Processing,* pp. 1-17, 2014/01/28 2014.

[11] C. Chih-Chi, L. Chia-Hua, L. Chung-Te, and C. Liang-Gee, "iVisual: An Intelligent Visual Sensor SoC With 2790 fps CMOS Image Sensor and 205 GOPS/W Vision Processor," *Solid-State Circuits, IEEE Journal of,* vol. 44, pp. 127-135, 2009.

[12] A. Benedetti and P. Perona, "Real-time 2-D feature detection on a reconfigurable computer," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, 1998, pp. 586-593.

[13] L. Teixeira, W. Celes, and M. Gattass, "Accelerated Corner-Detector Algorithms," *British Machine Vision Conference,* 2008.

[14] S. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "Feature tracking and matching in video using programmable graphics hardware," *Machine Vision and Applications,* vol. 22, pp. 207-217, 2011/01/01 2011.

[15] P. Mainali, Y. Qiong, G. Lafruit, L. Van Gool, and R. Lauwereins, "Robust Low Complexity Corner Detector," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 21, pp. 435-445, 2011.

[16] S. Alkaabi and F. Deravi, "Candidate pruning for fast corner detection," *Electronics Letters,* vol. 40, pp. 18-19, 2004.

[17] M. Ebrahimi and W. W. Mayol-Cuevas, "Adaptive Sampling for Feature Detection, Tracking, and Recognition on Mobile Platforms," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 21, pp. 1467-1475, 2011.

[18] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 25, pp. 1027-1033, 2003.

[19] S. P. DelMarco, V. Tom, and H. F. Webb, "A Theory of Automatic Parameter Selection for Feature Extraction With Application to Feature-Based Multisensor Image Registration," *Image Processing, IEEE Transactions on,* vol. 16, pp. 2733-2742, 2007.

[20] (2013, 9 June). *OpenCV DynamicAdaptedFeatureDetector*. Available: http://opencv.willowgarage.com/documentation/cpp/features2d_common_interfaces_of_feature_detectors.html#dynamicadaptedfeaturedetector

[21] (11/02/2014). *Raw images from Curiosity rover, Mars science laboratory*. Available: http://mars.jpl.nasa.gov/msl/multimedia/raw/

[22] *Nios II Processor: The World's Most Versatile Embedded Processor*. Available: http://www.altera.com/devices/processor/nios2/ni2-index.html