J Braz. Soc. Mech. Sci. Eng. (2017) 39:3103–3120
DOI 10.1007/s40430-017-0807-8

TECHNICAL PAPER

A robust methodology for outdoor optical mark recognition

Douglas Coimbra de Andrade[1] · Luís Gonzaga Trabasso[2] · Carlos César Aparecido Eguti[2] · Ricardo Suterio[2,3]

Received: 30 August 2016 / Accepted: 4 May 2017 / Published online: 17 May 2017

Abstract Outdoor optical mark recognition is an extremely useful tool for recognition of large industrial equipment and application of computer vision-based systems for tracking and positioning. However, current algorithms rely on thresholding and corner detection to identify checkerboard-like patterns, which is not appropriate for non-uniform lighting conditions. This paper presents a robust methodology to identify optical markers in outdoor environments. A GPU-based region filling algorithm automatically detects all contiguous color regions without computing seed points. Post-processing steps extract high-level information from these regions. Analysis of identified contiguous color region allows simultaneous identification of all checkerboard and targets (concentric regions) in the scene. Analysis of variance demonstrates that the proposed methodology is robust to lighting, environment, perspective, and occlusion. Tests indicate that precision and recall for checkerboard and target identification in outdoor conditions are expected to be above 97%. The parallel algorithm implementation using OpenCL yields better results and is two times faster than previous region filling algorithms, taking about 0.6 s to process a full-HD picture using modern hardware

Keywords Computer vision · Optical mark recognition · OpenCL · Parallel processing · Design of experiments · Heterogeneous computing

Technical Editor: Sadek C. Absi Alfaro.

✉ Douglas Coimbra de Andrade
douglasandrade@petrobras.com.br

Luís Gonzaga Trabasso
gonzaga@ita.br

Carlos César Aparecido Eguti
eguti@ita.br

Ricardo Suterio
suterio@lit.inpe.br

[1] Fabrication, Construction and Assembly Technologies, Petrobras, Rio de Janeiro, RJ, Brazil

[2] Department of Mechanical Engineering, Aeronautics Institute of Technology, São José dos Campos, SP, Brazil

[3] Test and Integration Laboratory, National Institute for Space Research, São José dos Campos, SP, Brazil

1 Introduction

Robust object tracking in outdoor environments using computer vision is a tool that can increase productivity and reduce costs in the construction of large equipments, such as pressure vessels, storage tanks, and oil platforms. This task can be performed with markers using cameras and optical mark recognition (OMR). Ideally, a set of cameras strategically positioned could replace the manual labor of tracking parts and alignment checks during the construction process. A challenging task, however, is that the construction environment lacks environmental control that would be available in laboratories and factories, such as controlled lighting and fixed trajectory for parts.

This paper presents a robust methodology for outdoor optical mark recognition, consisting of:

– a novel method of region filling that requires no seed point and finds all contiguous color regions in a scene. This implementation uses OpenCL to take advantage of processing power of current graphics processing units (GPUs) [13] using specific optimization strategies [23, 24];

– identification of characteristics of contiguous regions (area, perimeter, center);
– detection of targets and checkerboards using concentric region analysis and checkerboard regularity.

The main contributions of this work are the development of the parallel filling algorithm in OpenCL, which allows the proposed methodology to have acceptable performance for field use (as discussed in Sect. 4.3), and the checkerboard regularity analysis, which makes it robust to partial occlusion. These features allow the use of commercially available hardware to track and identify objects during field construction and assembly procedures.

The proposed methodology intended application is position feedback for robotic positioning of plates in large construction and assembly environments, as shown in Fig. 1. It could also be used as a reference for programming by demonstration using stereo cameras as an alternative to the Kinect system [15], automatic control of underwater robots [18], and addition of localization capabilities to mobile robots used to inspect power lines [12]. Robust OMR in outdoor environments can be incorporated in end-effector design along with other desirable characteristics [10]. Plates, pipes, and other parts need to be positioned within a tolerance of 3 mm because of welding requirements. Currently, this process is time-consuming, performed manually, and exposes personnel to risks. Optical markers are a cheap alternative to object tracking, since they require no special hardware attached to parts and equipments. In addition, they can easily be printed and replaced. This paper addresses the issue of identifying optical markers under field conditions, in outdoor environments.

This paper is organized as follows: Sect. 2 presents a brief literature review of the current methods used to identify markers under uneven lighting settings and optimization strategies; Sect. 3 describes the implementation of the proposed methodology; Sect. 4 presents an analysis of the collected data; and Sect. 5 presents the work conclusions with a brief overview of future improvements.

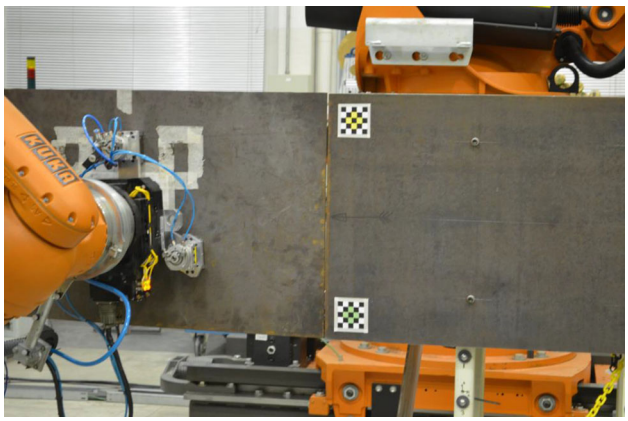


Fig. 1 Optical markers being used for robotic plate positioning system

2 Related work

Detection of patterns in outdoor settings is a challenging task that has deserved research attention in multiple fields, from face recognition to calibration patterns located under irregular lighting conditions [5, 17, 30]. Detection of concentric regions is another robust approach proposed for camera calibration in outdoor environments [16] as are 2D markers [19, 20]. However, little attention has been given to the problem of locating optical markers (not just for camera calibration) in outdoor settings, where lighting may vary not only in intensity but also color and global histogram distribution. Principles of machine vision techniques have been proposed by [21], who claims that accuracy of object detection is enhanced when optical markers are used; however, no implementation is provided.

Edge detection is a preprocessing step used to segregate color regions. Pre-filtering yields better results in edge detection, which may or may not help depending on the quality and noise-to-signal ratio of the camera. Advanced denoising techniques have been presented in the literature (e.g., [8]), [1, 28] and are used to improve edge detection accuracy.

For field applications, performance is an important consideration, because the identification algorithm has to provide information to operators during construction and assembly processes. A possible strategy in embedded systems is the use of FPGAs when custom hardware is made for the application [27]. Another approach used to accelerate computation is to take advantage of high parallelism and low cost of graphic processing units (GPUs) [4, 9]. In this work, GPU implementation was chosen to allow implementation of the robust optical mark recognition at a very low cost, without using custom hardware.

Flood-filling algorithms analyzed require a seed point and propagate regions from there. Many search strategies and methods have been proposed to optimize the process [7]. Flood-filling algorithms without edges are used to detect regions of interest in outdoor images [14]. The method, however, requires blurring images to a level that would not be acceptable for optical marker recognition. A pattern of dots was proposed to increase the robustness of pattern identification [5]. The method entails binarization of the image using Otsu method [25] followed by identification of neighboring dots, as shown in Figs. 2 and 3.

Kang et al. [5] show that Otsu's method, as any binarization, faces problems trying to split regions from a gray-scale image under uneven lighting settings. The ellipses

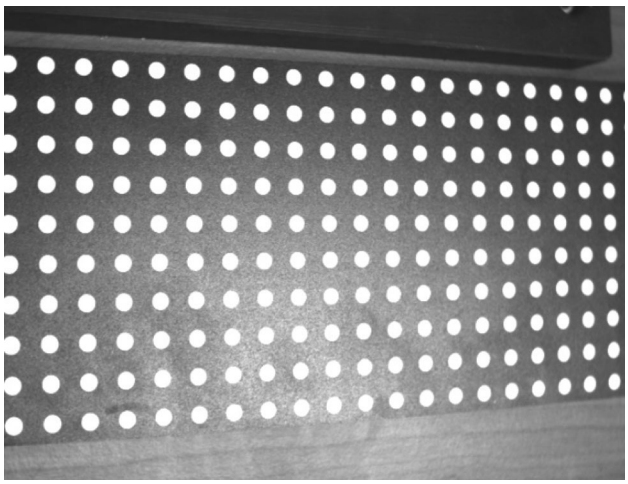


Fig. 2 Robust calibration pattern proposed by Kang et al. [5]

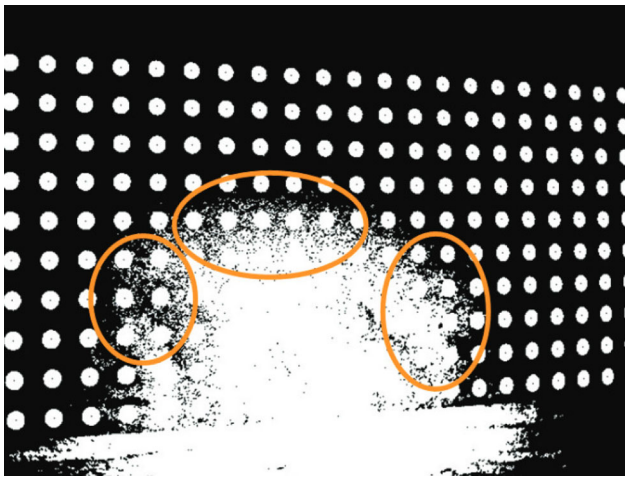


Fig. 3 Otsu binarization of source image, as obtained by Kang et al. [5]

inserted into Fig. 3 mark regions from where noise significantly reduces quality of dot identification. Moreover, color cameras are widely available and there is no reason not to take advantage of the extra information provided by RGB pixel components. For the purpose of comparison, the result of the algorithm developed in this work is shown in Fig. 4, where blue color denotes background color and adjacent regions of contiguous color are clustered together in a group. This result is superior in the sense that no white dots are lost in the binarization processes. Further details of the proposed algorithm are provided in the following sections.

It is also worth noting that Kang et al.'s [5] image sizes were 640×480 and their processing time was 1s using a Pentium IV processor. The exact same image, using modern hardware and GPU computing with a Radeon 7970 GPU, is processed in 0.13 s and better results are obtained.

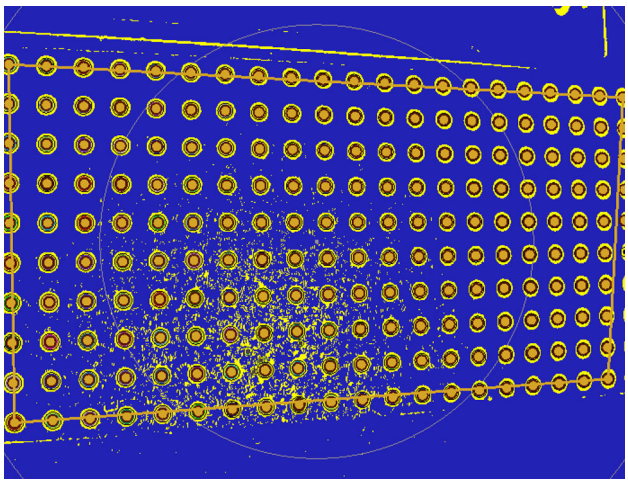


Fig. 4 Otsu binarization of source image, as obtained by [5]

Since no pre-filtering was applied in the original work, this comparison also skipped the pre-filtering step.

OpenCV algorithm for checkerboard detection relies on binarization of input image followed by morphologic procedures (erosion) and quadrant linking heuristics, which have been proposed by Rufli et al. [26]. This procedure has been designed for indoor camera calibration. Thus, for the purposes of outdoor identification of large parts, a more robust alternative is required. For comparison purposes, the typical image (shown in Fig. 5) from [26] has been processed using the proposed methodology, demonstrating that no features are lost: as in the original work, all checkerboard squares are detected. The proposed flood-filling algorithm does not rely on morphologic procedures and allows robust outdoor detection.

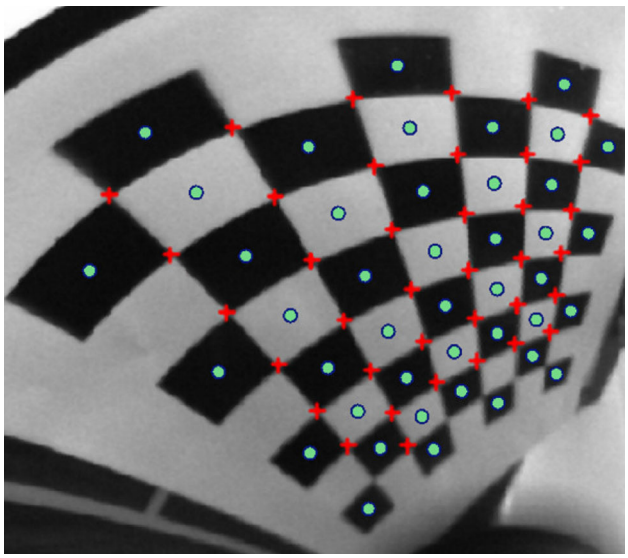


Fig. 5 Comparison of proposed algorithm to OpenCV using the same input image from [26]. *Red crosses* OpenCV output. *Green circles* output of proposed algorithm (color figure online)

3 Methodology for Robust OMR

In this work, a robust methodology has been developed for OMR in outdoor environments. Figure 6 summarizes the methods, tasks, and algorithms of the proposed methodology. The rightmost column provides illustrations of information retrieved during the recognition process.

The main requirement posed is robustness, to find markers in a wide range of lighting conditions. To allow visualization of results, the following graphical elements are created in processed images:

- A yellow circle is used to mark concentric region centers along with an ID (identification) code.
- A circle of the same color is drawn at the center of neighboring contiguous color regions.
- If requested, all pixels belonging to the same color regions are painted using the same randomly generated color. Pixels which belong to edges are marked in yellow.

First color identified in the picture is forced to be blue when identification of contiguous color regions is requested (usually identifies the background). It is worth noting that edge detection and region filling are crucial time-consuming steps for which parallel implementation using OpenCL has been developed.

The methodology goes through the following steps:

- *Identification of contiguous color regions* Image borders are computed using a Sobel filter. Parallel filling is performed (Sect. 3.1) to extract contiguous color regions using pixel-to-region association.
- *Extraction of high-level region information* After extracting pixels of each color region in the previous step, properties such as area and perimeter of each region is estimated (Table 1).
- *Checkerboard identification* Adjacent regions are grouped together and their characteristics are analyzed to predict outer position (using convex hull algorithm) and inner squares position (testing multiple

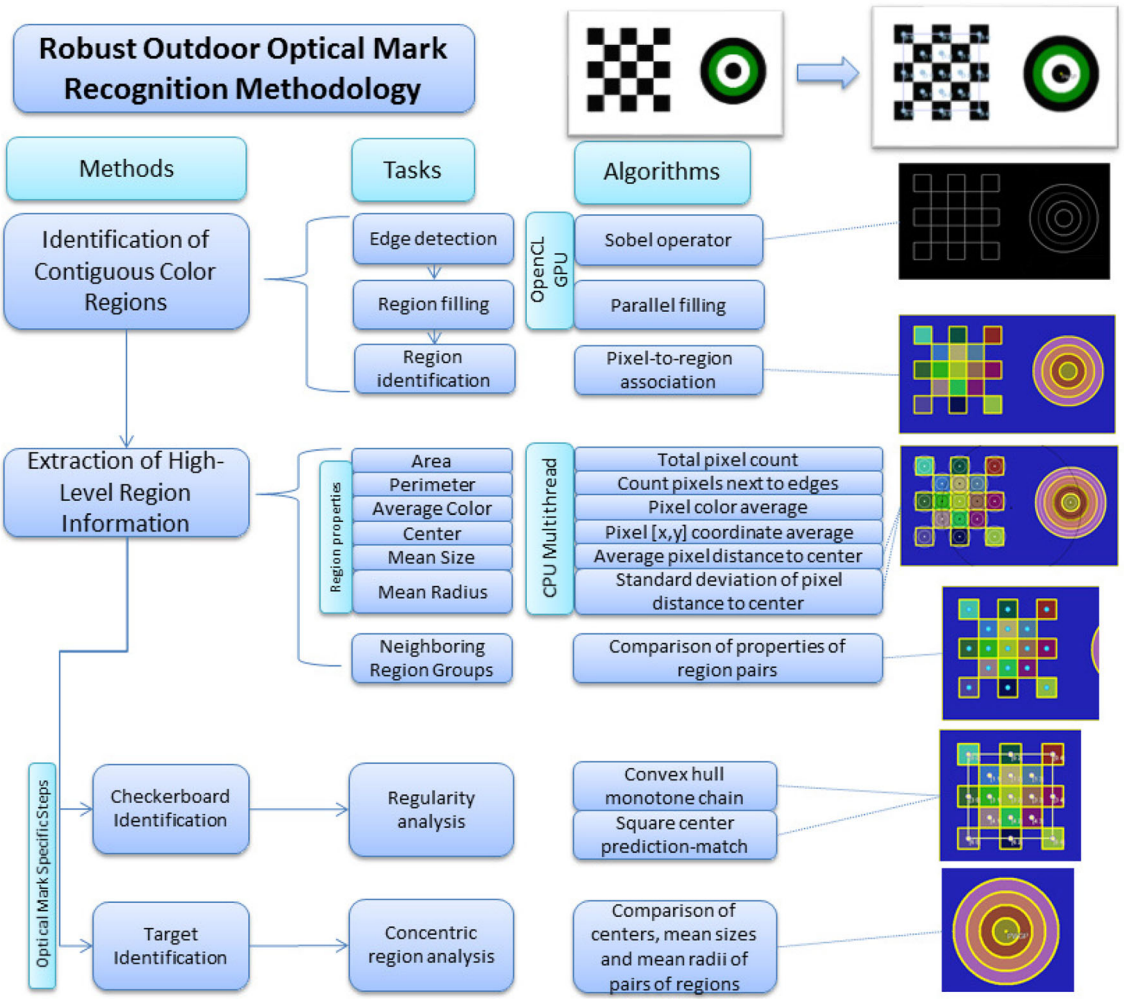


Fig. 6 Robust outdoor OMR structure

Table 1 Region data properties computed in the proposed methodology

Region property	Computation method
Area	Count how many pixels belong to the region
Perimeter	Count how many pixels are next to edges
Average color	Compute color average among pixels which belong to this region
Center	Compute coordinate ([x, y]) average among pixels which belong to this region
Mean size	Compute average pixel distance to region center
Mean radius	Compute standard deviation of pixel distance to center

checkerboard sizes). Details of this procedure are presented in Sect. 3.4.

– *Target identification* Region properties are used to detect which color regions are circular (disk-shaped). Then, circular regions whose center coincides are grouped into a target, as described in Sect. 3.3.

3.1 Identification of contiguous color regions

With invariance to lighting conditions in mind and considering that GPU processing with OpenCL is a very efficient tool to deal with color images, no binarization [5] or seed point usage [16] is necessary. Instead, the proposed methodology groups together all pixels of contiguous color regions compute properties from these pixels and draw higher level conclusions from this information. Identification of regions method is constituted of the following steps:

1. run edge detection algorithm to split regions of different colors;
2. sweep image pixels, clustering together regions which are not separated by edges;
3. retrieve pixels that belong to each region and compute useful region data such as center, area, and perimeter;
4. post process region data to extract higher level information such as location of checkerboards and concentric regions.

Robust edge detection is a crucial preprocessing step for the proposed flood filling algorithm to correctly segregate contiguous color regions. Multiple border detection algorithms have been considered. Although Canny border method [3] yields thinner edges, Sobel method [6] was chosen because of its reasonably good accuracy [2]. In addition, it requires a single pass and its GPU implementation is both robust in the sense that it is possible to consider RGB components and fast, because texture samplers will retrieve RGB components practically as fast as they would retrieve a single value. The edge detection algorithm implemented in this work computes Sobel edge values as the maximum of absolute differences of each RGB component. If the edge detection step fails, the proposed methodology will fail to identify the optical marker; however, in practice, for the experiments conducted, the threshold values set for the Sobel edge detector were sufficient to robustly identify the markers.

Following the edge detection, an adaptation of region flood-filling allows the proposed algorithm to retrieve contiguous color regions.

3.2 Parallel filling algorithm

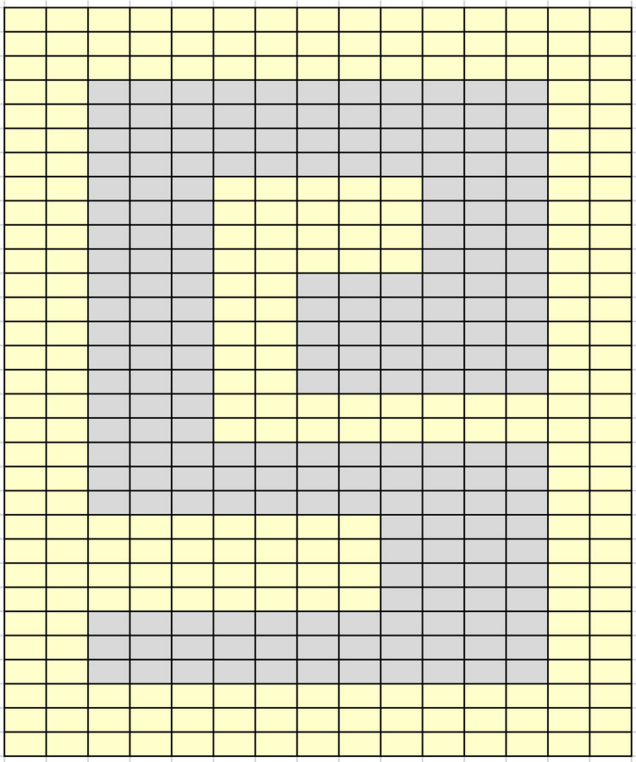
In this work, a new parallel flood-filling algorithm is developed to simultaneously identify all contiguous regions without any seed points. The outline of this algorithm is as follows:

1. Receive edge map and create color region map.
2. Initialize the region map by assigning each pixel a unique color number.
3. Do until region map remains unchanged:
   (a) Sweep region map from left to right, from right to left, from top to bottom, and from bottom to top in parallel.
   (b) If current pixel is not located inside any edge and its index is greater than previous pixel: assign current pixel the same index than previous pixel.

Note that step 3a is pivotal in this implementation, because it allows offloading the algorithm to the GPU. Figures 7, 8 illustrate pixel-to-region association retrieved using the proposed algorithm. The number of parallel workitems is equal to the image height when sweeping columns and to image width when sweeping lines, yielding over 1000 workitems in each case when processing a full-HD (1920×1080 pixels) image, which is appropriate for latency hiding purposes. Sweeping in both directions (left–right, right–left, top–down, and bottom–up) is required, because edge index switch only happens when current pixel color index is greater than previous pixel color index.

3.3 Target identification

After identifying contiguous color regions, it is possible to extract and store region data, such as which pixels belong to each region (list of [x, y] coordinates) and from there



Example image zoom = 75%

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265	266	267	268	269
270	271	272	273	274	275	276	277	278	279	280	281	282	283	284
285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314
315	316	317	318	319	320	321	322	323	324	325	326	327	328	329
330	331	332	333	334	335	336	337	338	339	340	341	342	343	344
345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374
375	376	377	378	379	380	381	382	383	384	385	386	387	388	389
390	391	392	393	394	395	396	397	398	399	400	401	402	403	404
405	406	407	408	409	410	411	412	413	414	415	416	417	418	419
420	421	422	423	424	425	426	427	428	429	430	431	432	433	434
435	436	437	438	439	440	441	442	443	444	445	446	447	448	449
450	451	452	453	454	455	456	457	458	459	460	461	462	463	464

Initialization

Fig. 7 Example image in *gray* and unique color map initialization

compute parameters presented in Table 1. Combinations of these parameters allow identification of targets (concentric regions—rings of the same color, whose center is close, as shown in Fig. 9) and neighboring regions (adjacent color regions without overlap, whose properties are approximately equal). Targets are identified by a string related to their color sequence, according to the following key:

- W: white;
- P: black;
- R: red;
- G: green;
- B: blue;
- ?: unknown color (not identified as W, P, R, G, or B).

For example, a sequence RWP?G means that the color of the inner region is red, followed by white, black, unidentified color, and green.

3.4 Checkboard identification

Checkerboard identification can be accomplished by counting the number of contiguous color regions in the group as well as by checking the regularity of internal places.

A post-processing step applied to the group of neighboring region locates the corners of the checkerboard and fits a rectangle to the point cloud as detailed below:

1. Compute the convex hull of contiguous color region centers grouped to the same region.
2. For each vertex v_n in the convex hull, compute the angle between the lines formed by $v_n \rightarrow v_{n-1}$ and $v_n \rightarrow v_{n-1}$.
3. Sort vertexes by their angle.
4. Keep the first four vertexes and discard regions, whose opposing angles are too different.

The number of squares in checkerboards should not be smaller than 5×7, because the tests showed that cluttered scenes are less likely to generate false positives starting from this checkerboard dimension. Maximum dimension is 21×21 squares, which is appropriate for field construction and assembly applications.

3.4.1 Regularity analysis

Given checkerboard dimensions, it is possible to forecast where the centers of its internal regions should be located. The task of checkerboard regularity analysis is accomplished as follows:

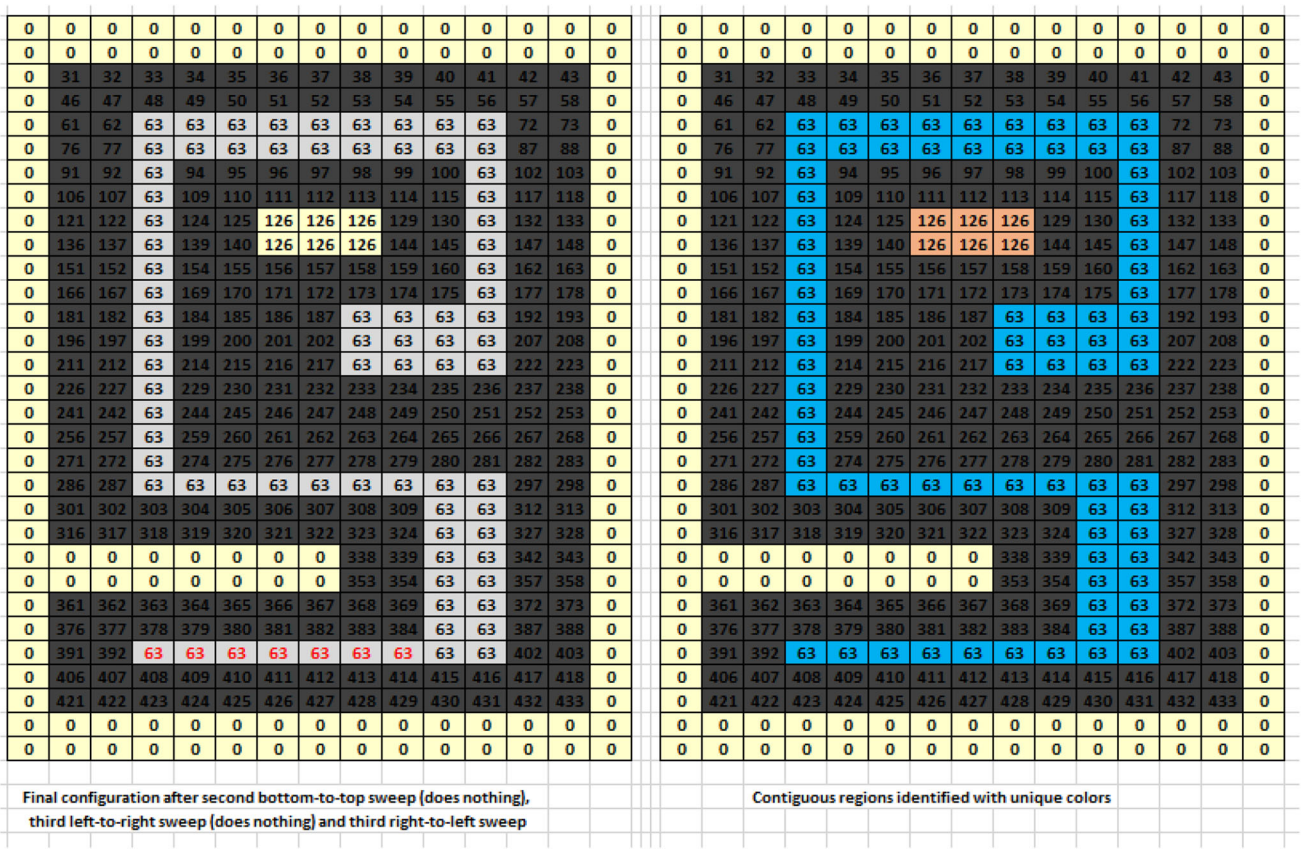


Fig. 8 Final configuration and visual identification of contiguous regions

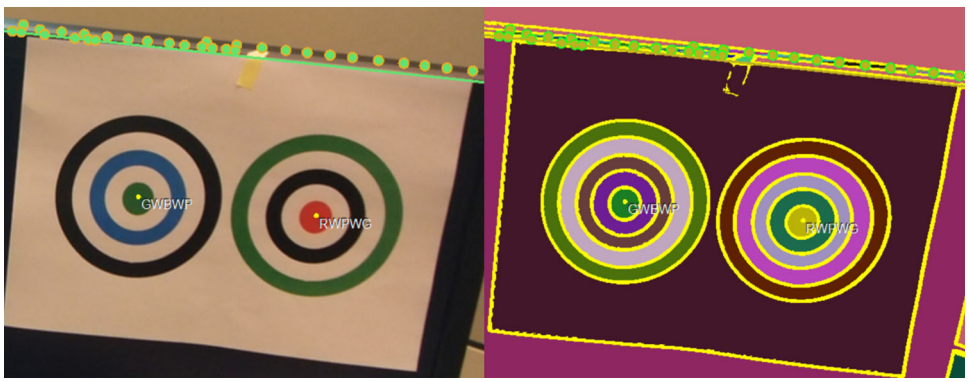


Fig. 9 Identified targets with proper label. *Left* cropped picture. *Right* identified contiguous color regions. GWBWP and RWPWG are target color label strings, denoting the sequence of colors from the inside out

- Use convex hull to find edge squares.
- Compute and store predicted square locations using the four outer black squares as a reference.
- Assign regions to their closest checkerboard position.
- Reject checkerboard if more than one region is assigned to the same position.

Regularity analysis allows automatic identification of checkerboard dimensions by successively trying to fit different sizes. It can also compensate partial occlusion, which is desirable in field applications. Both features are demonstrated in Fig. 10.

4 Results and analyses

4.1 Indoor and outdoor optical mark recognition

An experiment has been designed to check the robustness of the proposed methodology. A total of 270 images containing four checkerboard patterns and two concentric regions containing five concentric regions (see Fig. 11) were photographed using different cameras and settings (see details in Table 2). The markers were printed using regular ink and A4 paper to simulate worst case scenarios.

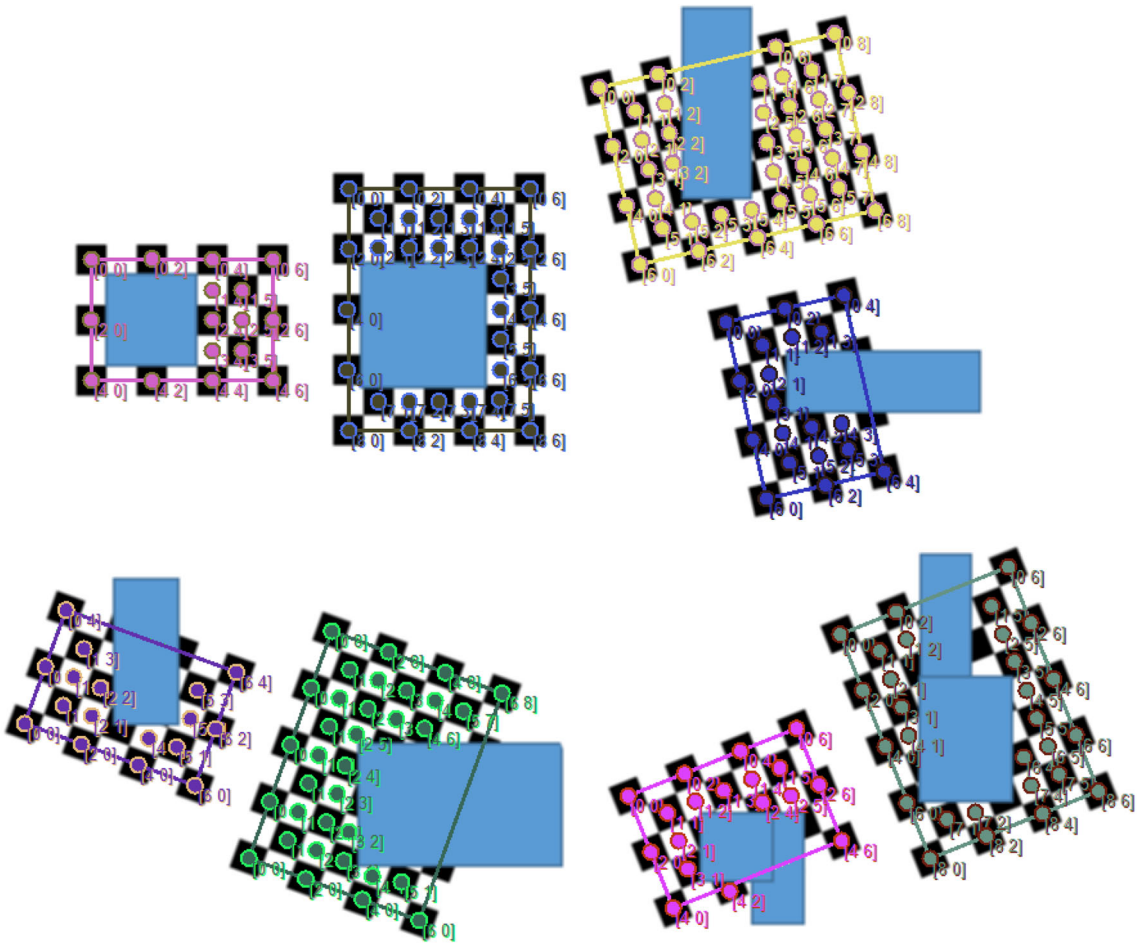


Fig. 10 Robust checkerboard identification under occlusion. Dimensions are identified automatically

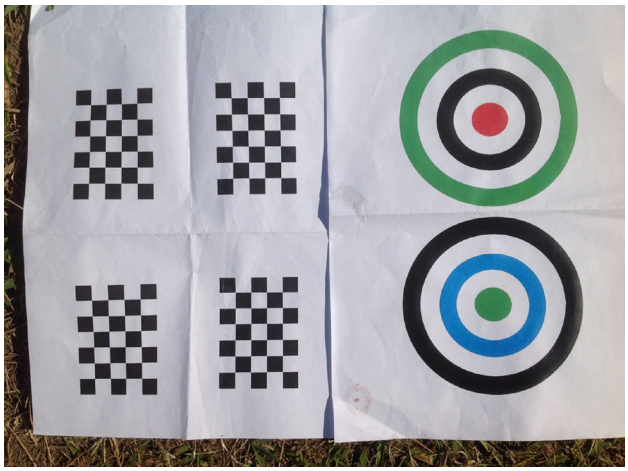


Fig. 11 Checkerboard and target patterns used to test robustness of the proposed methodology

In the tests performed, pre-filtering with a simple homogeneous Gaussian filter was enough to reduce noise while still maintaining edges identifiable at a negligible computational cost due to use of GPU acceleration via OpenCL. Median filtering provides the best noise removal while still preserving borders and allows detection of patterns from greater distances; simple Gaussian filter, however, still provided good results without the large computational burden of median filter. In addition, camera resolution and distance to optical markers can be controlled in an outdoor industrial environment, eliminating the need to identify very small regions.

Four lighting conditions were tested as a means to simulate possible field conditions (Table 3). Worst case scenarios are hard shadows and sunlight reflection on ink (v. failed cases) and these should be addressed using low pixel noise cameras and less reflective ink.

Figures 12, 13, 14 show sample images used to test the methodology and the respective results. When using the system for part identification, one is interested in optical mark recognition as well as its unique identification. As proposed in Sect. 3.3, the combination of targets and checkerboards allows better identification and geometric parameter estimation. Considering these factors, the

Table 2 Relevant parameters for outdoor optical mark recognition

Controlled variable	Possible values
Camera brand	Apple iPad 3 camera / Fuji W1 3D / Sony Cybershot DSC-WX7
Algorithm filter	None/Gaussian/median
Environment	Indoor/outdoor
Perspective	Frontal/sid /skewed
Lighting	Direct light/uniform shadow/soft shadow/hard shadow

Table 3 Field lighting conditions

Condition	Lighting condition on marker	Possible scenarios
Direct light	Direct daylight	Sunlight hitting marker directly
Uniform shadow	Only ambient light	Marker facing direction opposite to sun or cloudy environment
Soft shadow	Smooth transition light / shadow	Shadow from cloud/part partially cast on marker
Hard shadow	Harsh transition light / shadow	Shadow cast from cranes and robots on top of marker

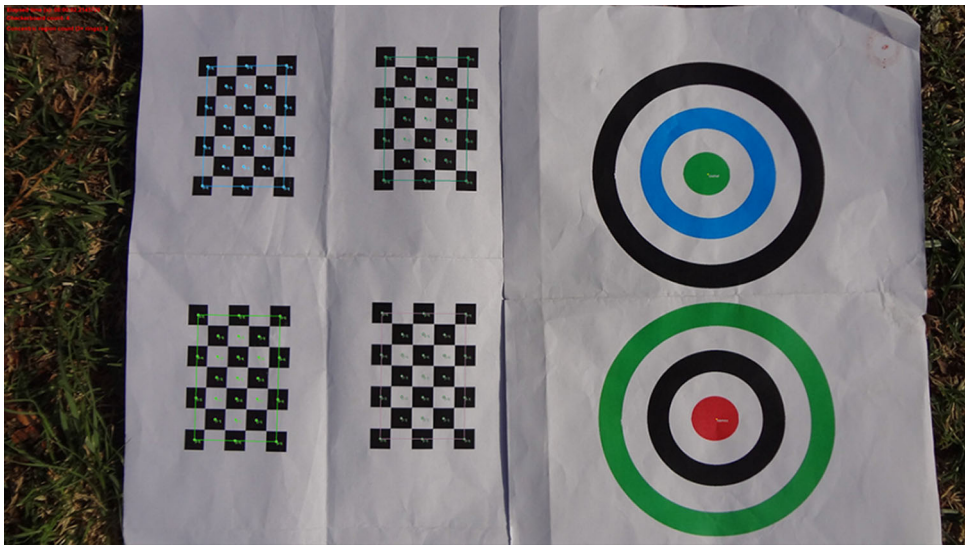


Fig. 12 Sample test image: WX7 camera, outdoor environment, frontal perspective, uniform shadow

performance of the algorithms used to perform OMR has been measured with criteria, as shown in Table 4.

4.2 Analysis of experiment

Table 5 summarizes identification performance of the proposed OMR methodology. Hard shadows are not common field conditions; these could be corrected controlling camera exposure and applying high dynamic range corrections. These settings were included along with using no special ink, paper or camera adjustment to test algorithm robustness under worst case scenarios. For all these reasons, performance figures excluding hard shadows are the ones expected as worst case in the field.

Images obtained were analyzed to verify which of the parameters from Table 2 are important for checkerboard and target identification using performance criteria, as presented in Table 4. One image was obtained for each possible combination of parameters and analysis of variance tests allowed proper assessment of algorithm robustness. Analysis of variance (ANOVA) [22] was then applied to verify the influence of each parameter on algorithm precision, recall, and F-score. Checkerboard F-Score results are as follows:

```
             Df Sum Sq Mean Sq F value Pr(>F)
environment   1 0.6277  0.6277   90.83 <2e-16 ***
lighting      3 0.7366  0.2455   35.52 <2e-16 ***
camBrand      2 0.0105  0.0053    0.76  0.469
filters       2 0.0216  0.0108    1.56  0.212
perspective   2 0.0048  0.0024    0.35  0.705
Residuals   259 1.7900  0.0069
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

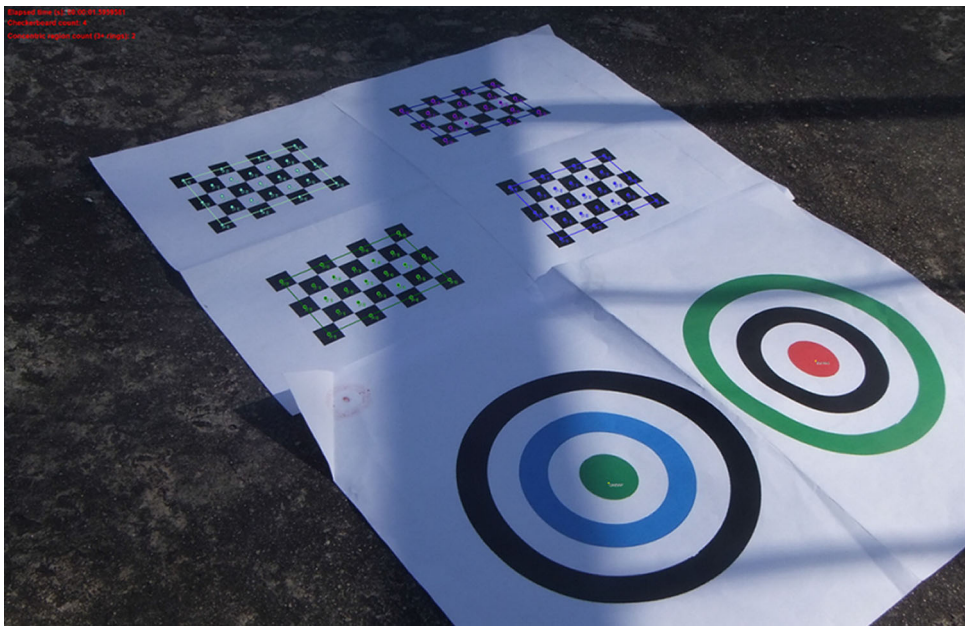


Fig. 13 Sample test image: Fuji W1 camera, outdoor environment, skewed perspective, hard shadow

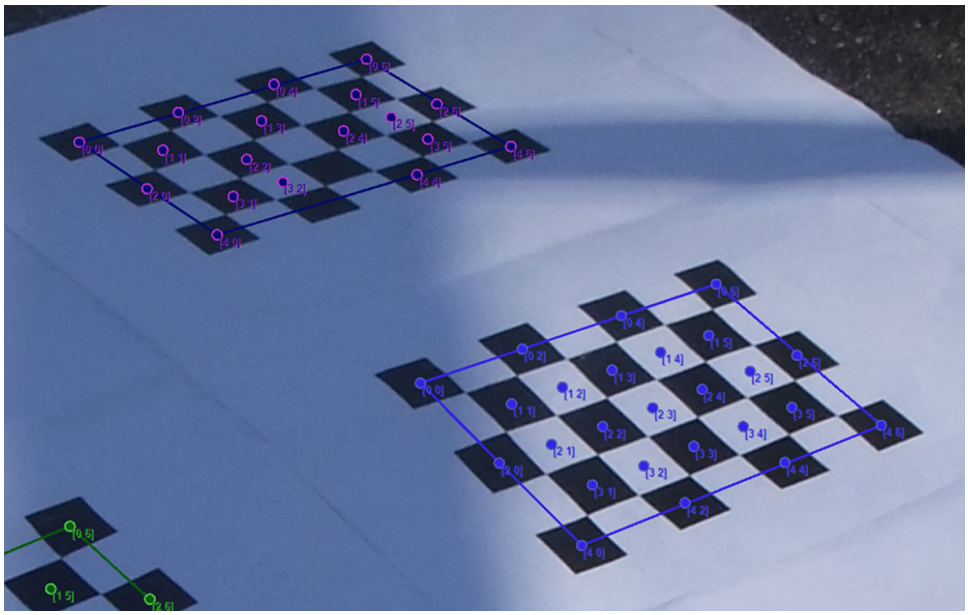


Fig. 14 Identification detail: Fuji W1 camera, outdoor environment, skewed perspective, hard shadow

Table 4 Performance criteria for the proposed identification algorithm

Criterion	Desired value per picture
True 5×7 checkerboards	4
False checkerboards	0
Targets with five concentric regions	2
Correctly identified concentric region in targets (proper color ID)	10
False target with three or more concentric regions	0

Table 5 Summary of identification performance of the proposed algorithm

Parameter	Global results (average) (%)	Results excluding hard shadows (%)
Checkerboard precision	98.2	98.5
Checkerboard recall	94.5	98.8
Checkerboard F-Score	95.5	98.5
Target precision	92.0	98.0
Target recall	89.8	97.2
Target F-Score	90.3	97.2
Target color accuracy	79.5	83.5

If hard shadow conditions are removed, the new analysis yields:

```
             Df Sum Sq  Mean Sq F value Pr(>F)
environment   1 0.0014 0.001414   0.607 0.4369
lighting      2 0.0115 0.005765   2.475 0.0867 .
camBrand      2 0.0070 0.003517   1.510 0.2234
filters       2 0.0114 0.005685   2.440 0.0896 .
perspective   2 0.0033 0.001672   0.718 0.4892
Residuals   206 0.4799 0.002329
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

This analysis shows that it is important to prevent or correct hard shadows should they happen in the field, which is unexpected. The most important influence is filter type which can be chosen to be the one that performs best (Gaussian filter).

Concentric region F-Score:

```
             Df Sum Sq Mean Sq F value   Pr(>F)
environment   1  1.341  1.3411  29.272 1.43e-07 ***
lighting      3  4.321  1.4404  31.440  < 2e-16 ***
camBrand      2  0.542  0.2711   5.918  0.00307 **
filters       2  0.001  0.0003   0.007  0.99320
perspective   2  0.583  0.2914   6.360  0.00201 **
Residuals   259 11.866  0.0458
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Concentric region F-Score below (no hard shadow)

```
             Df Sum Sq Mean Sq F value Pr(>F)
environment   1 0.0015 0.00154   0.116 0.7335
lighting      2 0.0559 0.02794   2.104 0.1245
camBrand      2 0.1133 0.05663   4.265 0.0153 *
filters       2 0.0002 0.00008   0.006 0.9938
perspective   2 0.0929 0.04644   3.498 0.0321 *
```

In the case of targets, camera brand has an important influence. This is due to better optical systems leading to well-defined edges and low noise inside color region. When color identification is important, it is better to use custom-made algorithms for each application and lighting conditions instead of the absolute criteria used in the implementation described herein.

The proposed methodology has limitations when shadows or reflexes cause extremely abrupt variation in the image, as shown in Figs. 15 and 16 . False negatives occurred either because of light reflexes (glare) in the scene or due to the presence of hard shadows. It is important to note that identification errors mostly stemmed from the tests carried out under very severe conditions and that the resulting identification succeeded in the majority of the cases. Light reflex is an important factor when illumination is not uniform and special ink could mitigate these effects, whereas images with hard shadows could be addressed using high dynamic range techniques.

4.3 Considerations about running time

All algorithms have been implemented using Microsoft Visual C# 2010 and OpenCL to offload workload to the GPU. Table 6 shows computing times for a full-HD (1920×1080) picture and describes which hardware is used for each step when extracting region data information that is used for concentric region and neighborhood computation. These figures were obtained using hardware that was available for this research. Note that GPU computation of Sobel borders takes negligible time in comparison with the other steps. In addition, although computation of median filter yields the best results, a simpler Gaussian filter takes negligible time and still produces acceptable results. Note that, while there are very fast optical marker recognition algorithms that run even in smartphones, these are not designed to be robust under different lighting settings or occlusion.

The proposed parallel filling algorithm for contiguous color detection is designed to run in parallel and fits nicely the SIMD (Single Instruction, Multiple Data) structure of GPUs, since all workitems perform approximately the same amount of computation per kernel launch.

Segregation of region pixels and computation of region data properties, on the other hand, are more suited to the MIMD (Multiple Instruction and Multiple Data) structure

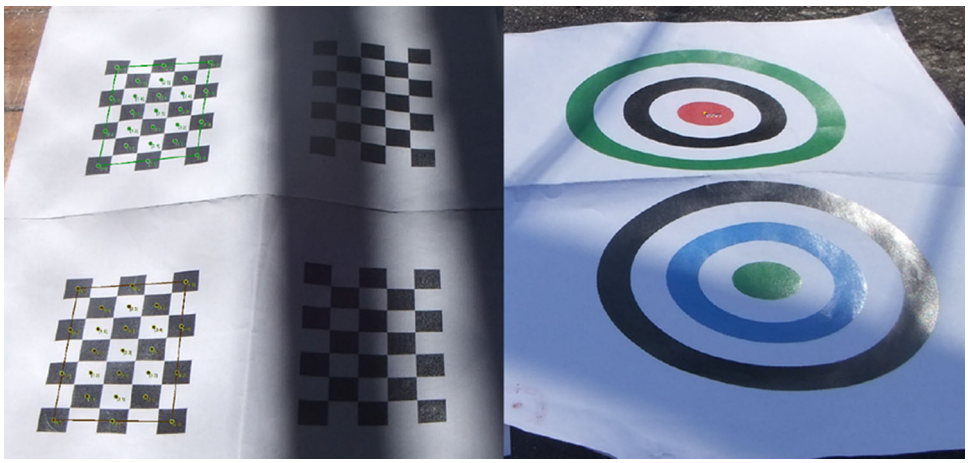


Fig. 15 False checkerboard and target negatives

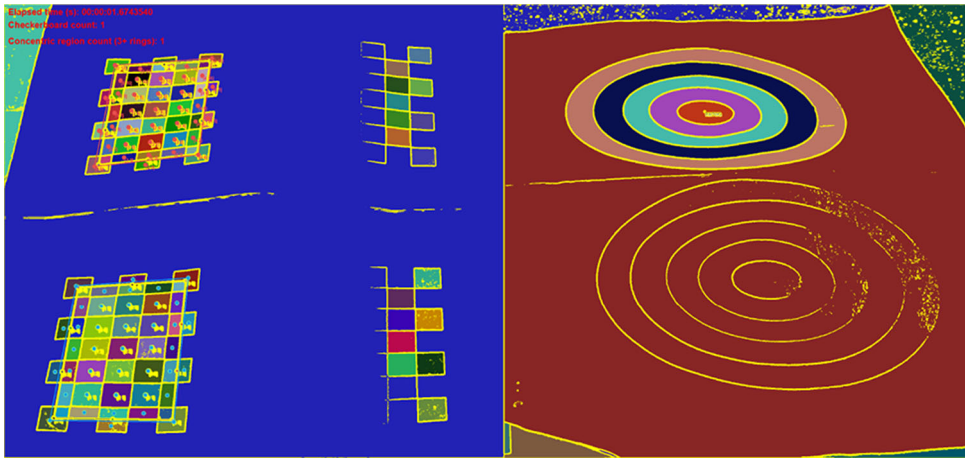


Fig. 16 False negatives color region analysis. *Left* edges were not properly identified. *Right* light reflex impairs edge detection

Table 6 Algorithm runtime when processing full-HD (1920 × 1080) images

Algorithm step and execution hardware	Core i5-3317U 1.7GHz + Intel HD 4000 GPU	Xeon X5650 2.66 GHz + Tesla C1060	Core i7 3820 3.6GHz + Radeon 7970
Median filter (GPU)	0.45	0.18	0.1
Contiguous color detection (GPU)	0.29	0.14	0.1
Pixel list per color region (CPU+GPU)	0.19	0.14	0.15
Region data information (parallel CPU)	0.41	0.26	0.22
Total time	1.34	0.72	0.57

Times in seconds

of the GPU considering that the number of pixels per region may vary considerably.

Runtime of the proposed algorithm was compared to that presented in [7]. Their reported runtime to identify a 100 pixel radius circle, a 100 × 100 square, and a random pattern (their three test cases) that they created was 38017 ms, although that is probably a typo and their real result is 380 ms. To fully exploit GPU parallelism, a joint image with multiple circles, squares, and pattern was created. Its dimensions are 1200 × 1000 pixels, whose region filling is presented in Fig. 17. Note that the proposed methodology requires no seed points and runs in 523 ms using the Intel HD 4000, 482 ms in the Tesla C1060, and 282 ms in the Radeon 7970, which is $380/(282/24) = 32$ times faster

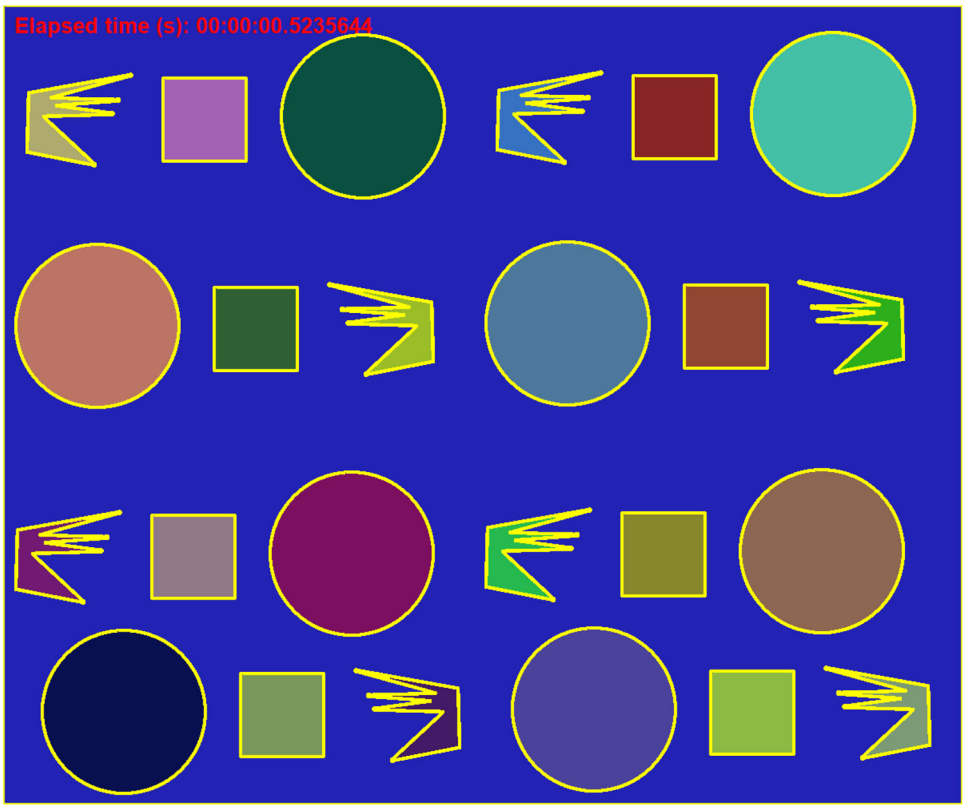


Fig. 17 Region filling image for runtime comparison purposes

Table 7 Performance comparison of the proposed flood-filling algorithm in a 512 × 512 image

RegionCount	Tracing	Classical	Column	OpenCL (proposed)
126.1	144.7	175.9	5681.2	72.1

Times in ms

despite the extra time spent filling background pixels which alone would probably take way longer than filling the patterns due to its having as many pixels as the other regions combined.

To compare results from the proposed algorithm with RegionCount, Tracing method, classical recursive method, and column method presented in [11], results from Table 6 obtained with the Radeon 7970 were downscaled to 512 × 512, which was the image size used in these algorithms. Table 7 shows that, despite not needing any seed points, the proposed algorithm is significantly faster than the other methods.

4.4 Experiments in relevant environment

To demonstrate robustness of the method in a relevant environment, under conditions closer to a real application, experiments were conducted at Petrobras' Research Center (CENPES— Fig. 18) and demonstrated that, even though a simple phone camera[1] was used, it was possible to identify markers from a distance. A custom designed system, with better optics and less distortions than a consumer-level system, is expected to have be able to handle even greater distances using industrial cameras with better optical system and controllable zoom capabilities.

In all tested settings, the obtained results were satisfactory in terms of identifying the presence of the optical markers and their position in the image. Figures 19, 20 show that markers are identified even in the presence of shadows cast by the equipment directly on top of it. Figure 21 demonstrates that the methodology is robust to partial occlusion in a real setting. Figures 22, 23 present correct OMR in curved surfaces under nonuniform lighting. It is worth noting that the placement of the markers for these tests was chosen to test these limit cases. In a real application, the use of industrial cameras and placement and marker placement in uniform regions will increase system accuracy.

4.5 Preliminary optical marker detection

In real applications, machine learning techniques can be used to allow detection of optical markers that appear small in the image and guide the camera automation system to

[1] 13 MP Samsung Galaxy S4 without any special features

Fig. 18 Optical markers placed in simulated industrial environment at Petrobras' Research Center (CENPES). *Left* storage tank. *Center* pressure vessel. *Right* heat exchanger

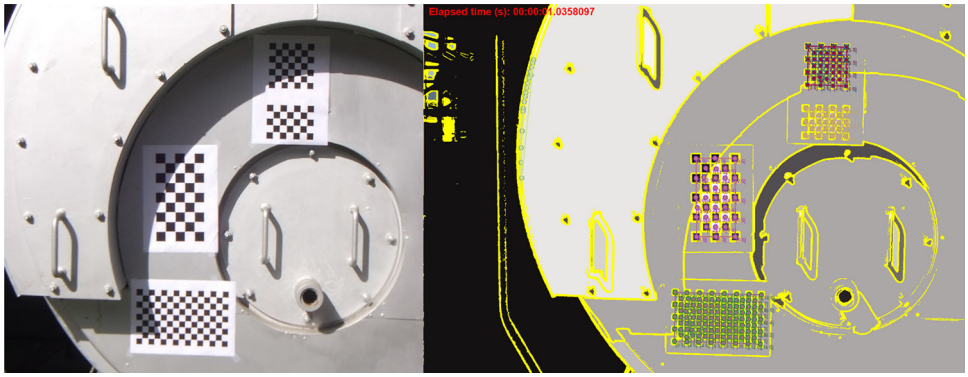


Fig. 19 Correct marker identification in a heat exchanger

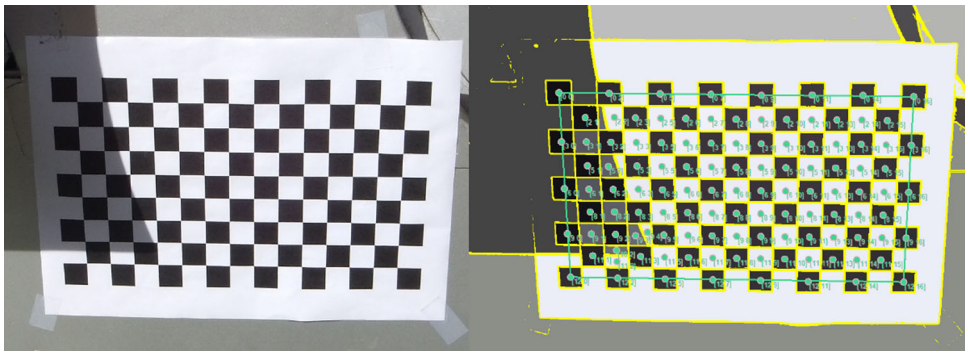


Fig. 20 Detail of identification of optical marker in the presence of shadow cast by the equipment

zoom into image locations, where optical markers are likely to be found, by providing bounding boxes. To obtain a preliminary bounding box of the optical marker, a single hidden layer feed-forward neural network was trained to recognize optical markers. Color Haar features are extracted from from the pictures using an OpenCL implementation of image integrals [29], as shown in Figs. 24 and 25. This method has the advantage of that it does not require edge information, but precise optical marker location still requires the refinements implemented in the proposed methodology to be able perform OMR robustly.

Considering that optical markers are designed to be easily identified, a very high accuracy is expected. In fact, using a neural network with a single hidden layer yields 99.5% accuracy. The classifier's precision is 100% and recall is 90% which indicates that, taking into consideration the higher performance of neural networks in visual tasks, this approach is valid for initial bounding box extraction prior to refinement using the proposed methodology to obtain more precise optical marker position and orientation.

5 Conclusions

This paper has presented a robust methodology for Outdoor Optical Mark Recognition. The OMR methodology applied to planar checkerboard detection allows robust identification under extremely variable lighting conditions and partial occlusion. Experiments demonstrated that, excluding hard shadow scenes, the technique is robust to variations in camera brand, lighting conditions, perspective, and indoor/outdoor environments. Under normal outdoor conditions without hard shadows being projected on top of the optical markers, precision and recall are expected to be above 97%.

Previous methods use binarization and thresholding techniques to identify points of interest in images

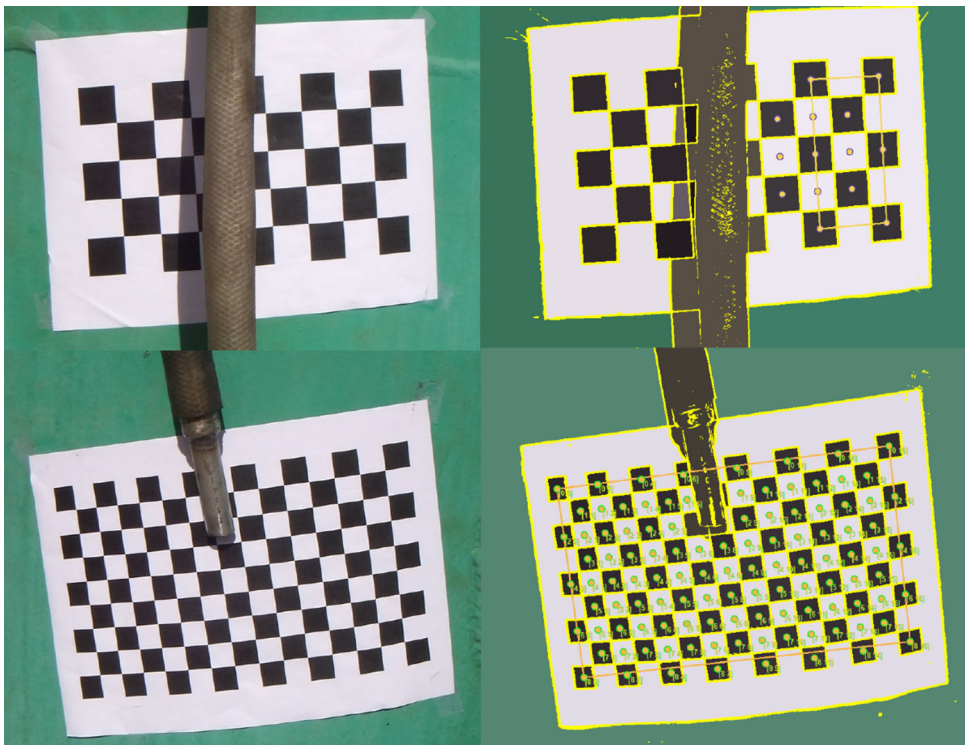


Fig. 21 *Top* marker not identified due to severe occlusion. *Bottom* marker recognized in the presence of partial occlusion

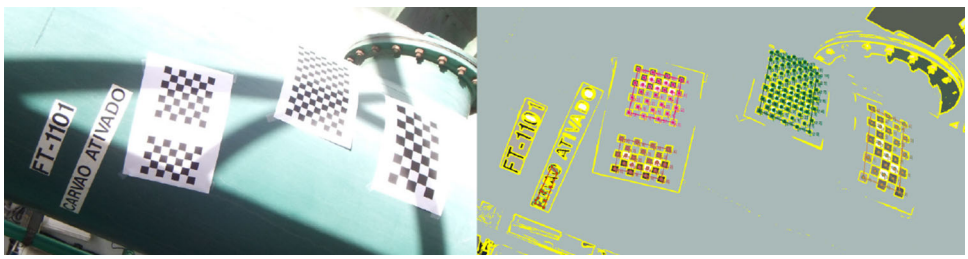


Fig. 22 Correct identification of optical markers in a curved surface, subject to shadow and exposed to light

containing optical markers, notably corners, and white regions [5]. The proposed approach takes advantage of GPU texture processing hardware via heterogeneous computing with OpenCL to implement faster edge detection and region filling algorithms. In particular, the proposed parallel region filling algorithm requires no seed points, identifies all contiguous color regions, and runs at least two times faster than previous methods using color images. Higher level information processing is done using the CPU, because its computing time is small in comparison with edge detection/region filling and its data-dependent structure is more suited to MIMD architectures. This CPU/GPU combination allows robust OMR to be performed under 3 s using current hardware, which is appropriate for field use when positioning large parts.

Experiments conducted at Petrobras' Research Center (CENPES) demonstrated that the methodology is capable of performing OMR in an outdoor industrial environment. Preliminary classification of optical markers using neural networks in Haar image features can provide bounding boxes for marker candidates. This information may be used to guide the camera system to zoom in those areas and for detailed information extraction using the proposed methodology, thus allowing for optical marker identification even when they are small in the original image.

Further investigation is necessary to check whether light reflexes or hard shadows affects the use of the OMR methodology in real applications. However, it is expected that the approach will be even more robust in real settings considering that it was tested in severe conditions, where cameras were used in fully automatic mode and optical markers were printed using regular ink and paper. Future work will also analyze how to utilize region color information for arbitrary image registration.

Although the methodology is capable of detecting small checkerboards and targets in pictures, the influence of their size in identification has not been assessed; because optical markers can be printed in a size comparable to equipment parts and industrial computer vision systems, specification can include high-resolution cameras and high zoom

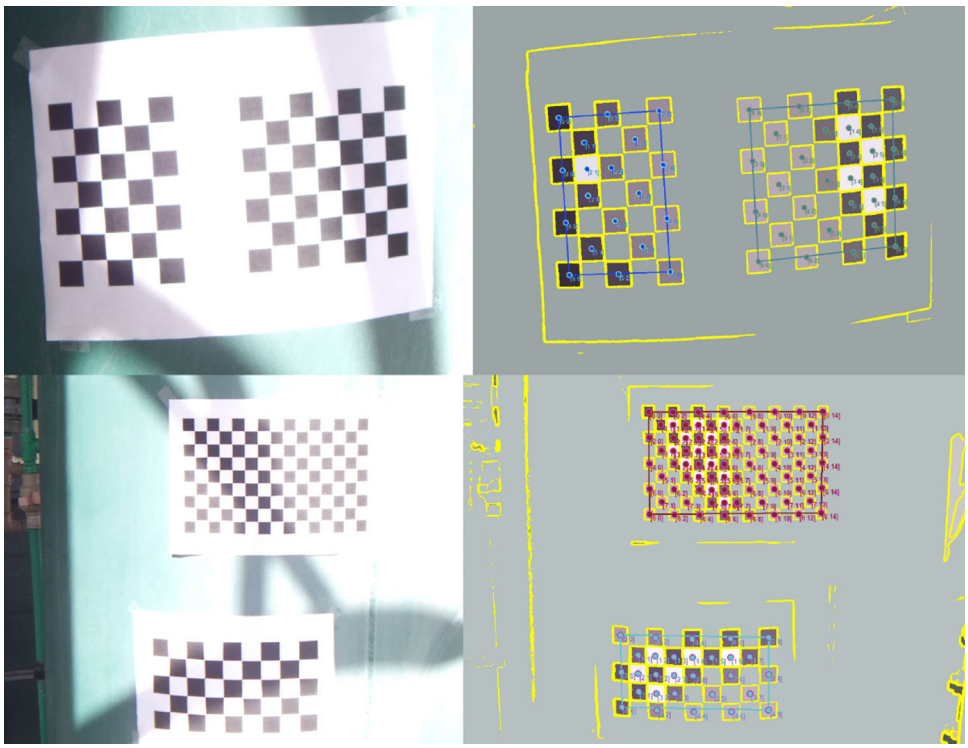


Fig. 23 Details of correct OMR when marker is exposed to light, soft shadow, and object curvature

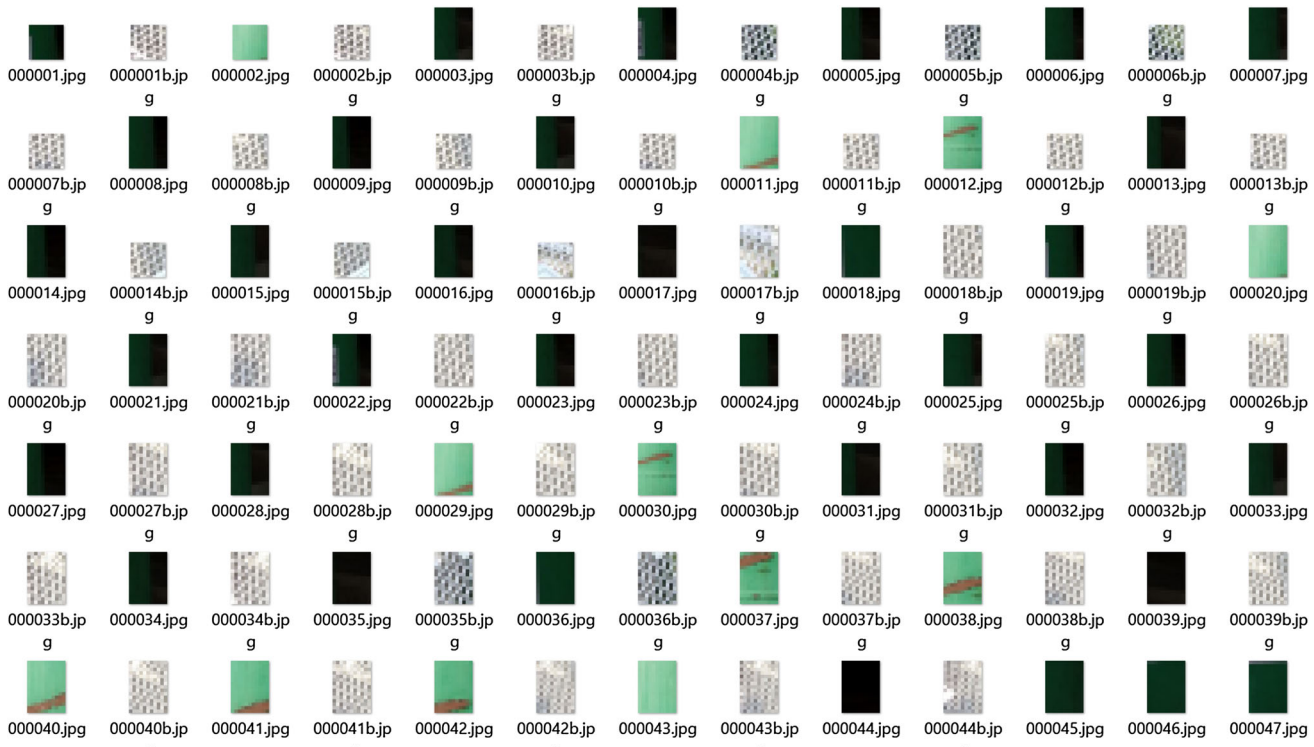


Fig. 24 Extracted images containing samples of the environment

capabilities. Results demonstrate that the OMR methodology performance is not affected by camera brand, preprocessing filters, and perspective (orientation) of the optical markers, which means that there is no special requirement with regard to their positioning considering OMR performance.

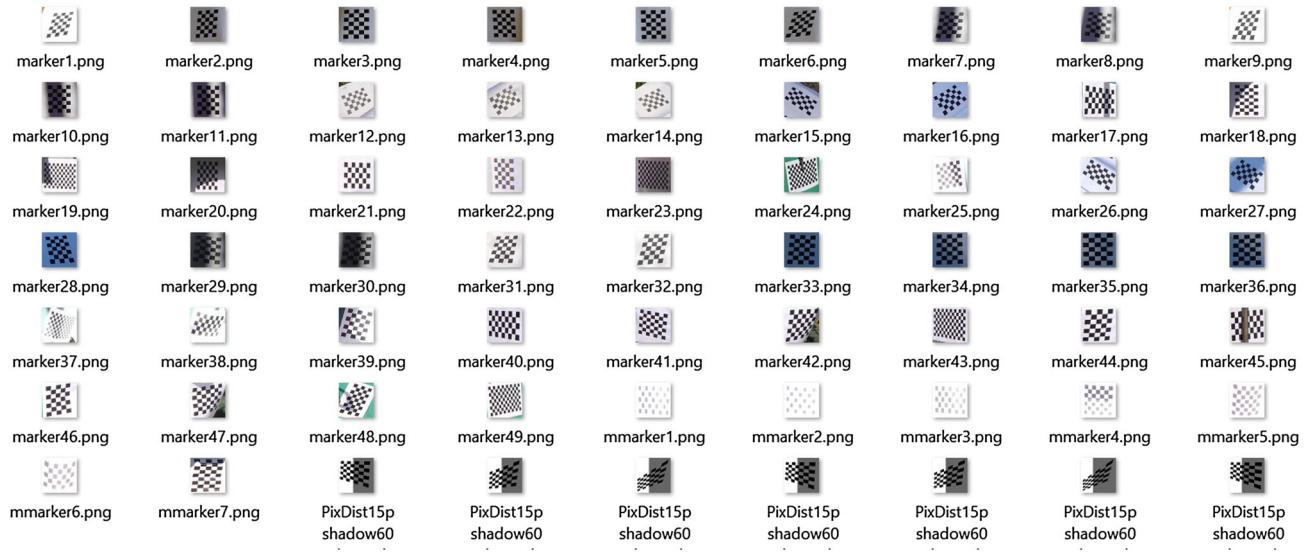


Fig. 25 Extracted images containing samples of optical markers (checkerboards) in the presence of non-uniform lighting and perspective

The proposed methodology can be used with high-resolution high-quality cameras and optical markers to construct a robust, computer vision-based system for outdoor part identification and positioning.

References

1. Aysal TC, Barner KE (2007) Generalized mean-median filtering for robust frequency-selective applications. IEEE Trans Signal Process 55(3):937–948. doi:10.1109/TSP.2006.888882
2. Baglodi V (2009) Edge detection comparison study and discussion of a new methodology. IEEE Southeastcon 2009:446–446. doi:10.1109/SECON.2009.5174124
3. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI 8(6):679–698. doi:10.1109/TPAMI.1986.4767851
4. Carrion R, Mesquita E, Ansoni JL (2015) Dynamic response of a frame-foundation-soil system: a coupled bem-fem procedure and a gpu implementation. J Braz Soc Mech Sci Eng 37(4):1055–1063. doi:10.1007/s40430-014-0230-3
5. Dong-Joong Kang JEH, Jeong MH (2008) Detection of calibration patterns for camera calibration with irregular lighting and complicated backgrounds. Int J Control Autom Syst 6(5):746–754
6. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
7. Duo-le F, Ming Z (2011) A new fast region filling algorithm based on cross searching method. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 380–387. doi:10.1007/978-3-642-22456-0_55
8. Fabijanska A, Sankowski D (2011) Noise adaptive switching median-based filter for impulse noise removal from extremely corrupted images. IET Image Process 5(5):472–480. doi:10.1049/iet-ipr.2009.0178
9. Franco EE, Barrera HM, Laín S (2015) 2d lid-driven cavity flow simulation using gpu-cuda with a high-order finite difference scheme. J Braz Soc Mech Sci Eng 37(4):1329–1338. doi:10.1007/s40430-014-0260-x
10. Furtado LFF, Villani E, Trabasso LG, Silva CEO (2014) Dtw: a design method for designing robot end-effectors. J Braz Soc Mech Sci Eng 36(4):871–885. doi:10.1007/s40430-013-0109-8
11. Geraets W, van Daatselaar A, Verheij J (2004) An efficient filling algorithm for counting regions. Comput Methods Progr Biomed 76(1):1–11. doi:10.1016/j.cmpb.2003.09.004. http://www.sciencedirect.com/science/article/pii/S0169260703001330
12. Gonçalves RS, Carvalho JCM (2015) A mobile robot to be applied in high-voltage power lines. J Braz Soc Mech Sci Eng 37(1):349–359. doi:10.1007/s40430-014-0152-0
13. Group K (2015) The OpenCL Specification Version: 2.0. https://www.khronos.org/registry/cl/specs/opencl-2.0.pdf. Accessed 23 Aug 2016
14. Hsieh SL, Hsiao YJ, Huang YR (2011) Using margin information to detect regions of interest in images. In: Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, pp 3392–3396. doi:10.1109/ICSMC.2011.6084193
15. Jha A, Chiddarwar SS, Alakshendra V, Andulkar MV (2016) Kinematics-based approach for robot programming via human arm motion. J Braz Soc Mech Sci Eng. doi:10.1007/s40430-016-0662-z
16. Jiang G, Quan L (2005) Detection of concentric circles for camera calibration. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol 1, pp 333–340. doi:10.1109/ICCV.2005.73
17. Kang DJ, Lee WH (2010) Automatic circle pattern extraction and camera calibration using fast adaptive binarization and plane homography. Int J Precis Eng Manuf 11(1):13–21. doi:10.1007/s12541-010-0002-7
18. Kuhn VN, Drews PLJ, Gomes SCP, Cunha MAB, Botelho SSdC (2015) Automatic control of a rov for inspection of underwater structures using a low-cost sensing. J Braz Soc Mech Sci Eng 37(1):361–374. doi:10.1007/s40430-014-0153-z
19. Kurka PRG, Delgado JV, Mingoto CR, Rojas OER (2013) Automatic estimation of camera parameters from a solid calibration box. J Braz Soc Mech Sci Eng 35(2):93–101. doi:10.1007/s40430-013-0013-2
20. Liu X, Doermann D, Li H, Lee KC, Ozdemir H, Liu L (2008) A novel 2D marker design and application for object tracking and event detection. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 248–257. doi:10.1007/978-3-540-89639-5_24

21. luzek A (2010) Novel machine vision methods for outdoor and built environments. Autom Constr 19(3):291–301. doi:10.1016/j.autcon.2009.12.002. http://www.sciencedirect.com/science/article/pii/S0926580509001915(**25th International Symposium on Automation and Robotics in Construction**)
22. Montgomery DC (2012) Design and analysis of experiments, 8th edn. Wiley. http://www.wiley.com/WileyCDA/WileyTitle/productCd-EHEP002024.html
23. NVidia (2009) NVIDIA OpenCL Best Practices Guide Version 1.0. http://www.nvidia.com/content/cudazone/CUDABrowser/downloads/papers/NVIDIA_OpenCL_BestPracticesGuide.pdf. Accessed 23 Aug 2016
24. NVidia (2012) NVIDIAs Next Generation CUDA ™ Compute Architecture: Kepler ™ GK110. http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf. Accessed 23 Aug 2016
25. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66. doi:10.1109/TSMC.1979.4310076
26. Rufli M, Scaramuzza D, Siegwart R (2008) Automatic detection of checkerboards on blurred and distorted images. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3121–3126. doi:10.1109/IROS.2008.4650703
27. Sánchez-Ferreira C, Mori JY, Farias MCQ, Llanos CH (2016) A real-time stereo vision system for distance measurement and underwater image restoration. J Braz Soc Mech Sci Eng. doi:10.1007/s40430-016-0596-5
28. Utaminingrum F, Uchimura K, Koutaki G (2013) High density impulse noise removal based on linear mean-median filter. In: Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on, pp 11–17. doi:10.1109/FCV.2013.6485451
29. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol 1, pp I–511–I–518. doi:10.1109/CVPR.2001.990517
30. Wang H, Li SZ, Wang Y (2004) Face recognition under varying lighting conditions using self quotient image. In: Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pp 819–824. doi:10.1109/AFGR.2004.1301635