

Article

Braille Block Detection via Multi-Objective Optimization from an Egocentric Viewpoint

Tsubasa Takano ¹, Takumi Nakane ¹, Takuya Akashi ² and Chao Zhang ^{1,*}¹ Department of Engineering, University of Fukui, Fukui 910-8507, Japan;

t-takano@monju.fuis.u-fukui.ac.jp (T.T.); t-nakane@monju.fuis.u-fukui.ac.jp (T.N.)

² Faculty of Engineering, Iwate University, Iwate 020-8550, Japan; akashi@iwate-u.ac.jp

* Correspondence: zhang@u-fukui.ac.jp

Abstract: In this paper, we propose a method to detect Braille blocks from an egocentric viewpoint, which is a key part of many walking support devices for visually impaired people. Our main contribution is to cast this task as a multi-objective optimization problem and exploits both the geometric and the appearance features for detection. Specifically, two objective functions were designed under an evolutionary optimization framework with a line pair modeled as an individual (i.e., solution). Both of the objectives follow the basic characteristics of the Braille blocks, which aim to clarify the boundaries and estimate the likelihood of the Braille block surface. Our proposed method was assessed by an originally collected and annotated dataset under real scenarios. Both quantitative and qualitative experimental results show that the proposed method can detect Braille blocks under various environments. We also provide a comprehensive comparison of the detection performance with respect to different multi-objective optimization algorithms.

Keywords: Braille block detection; egocentric vision; multi-objective optimization



Citation: Takano, T.; Nakane, T.; Akashi, T.; Zhang, C. Braille Block Detection via Multi-Objective Optimization from an Egocentric Viewpoint. *Sensors* **2021**, *21*, 2775. <https://doi.org/10.3390/s21082775>

Academic Editor: Daeun Kim

Received: 3 March 2021

Accepted: 13 April 2021

Published: 14 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last decade, wearable devices have become widespread in a wide range of applications from healthcare to monitoring systems due to the development of miniaturization and computational power. Recent interests in navigation aid for blind people have spurred research aimed at the detection of obstacles and detecting the distance to nearby objects [1]. On the other hand, besides assistive techniques such as white canes and guide dogs, tactile paving (also known as Braille blocks or tenji blocks) is ubiquitous in Japan, which is a system of textured ground surfaces to assist pedestrians who are visually impaired (e.g., Figure 1). As one of the most important usages, the surface of Braille blocks is designed to be uneven such that people can be guided along the route by maintaining contact with a long white cane. However, cane travel can be cumbersome and not as fluid because of its weight and the physical effort required to swing. To eliminate the inconvenience brought by the Braille blocks, one possible solution is to develop a head-mounted device embedded with a sensory substitution system for cane-free walking support. As the first step, the device is required to automatically locate the region of Braille block in the image taken by the first-person camera, which is also the main purpose and motivation of this paper.

In real-world problems, there may exist multiple objectives to be optimized simultaneously in order to solve the task. Multi-objective optimization (MO) is a technique to solve such tasks with the results represented by the Pareto optimal solution, which is a set of non-dominated solutions. The Pareto optimal solution allows for compromises between different evaluation criteria, without favoring one over the other, and thus gives a reasonable solution considering the trade-off. In this paper, the MO technique can be applied to the problem of Braille block detection by assessing multiple types of features of Braille blocks in the form of calculating multiple objective functions. As the basic strategy, we consider that the task of Braille block detection can be effectively solved under an

optimization framework due to the simple geometric and appearance features. Specifically, in this paper, the popular multi-objective genetic optimization algorithm, non-dominated sorting genetic algorithm-II (NSGA-II) [2], was used to optimize multiple validity measures simultaneously. The main contributions of this paper are threefold.

- A Braille block detection framework with the egocentric images as input is proposed.
- We formulate the block detection as a multi-objective optimization problem by considering both the geometric and the appearance features.
- A Braille block detection dataset is originally built with annotations.

The paper is organized as follows. In the next section, we introduce related work. Section 3, we present the proposed framework using MO. Section 4, we describe the qualitative and quantitative experimental results of Braille block detection in egocentric images. The conclusion is presented at the end of this paper.



Figure 1. Examples of the Braille blocks from an egocentric viewpoint.

2. Related Work

In the field of egocentric vision, object detection and recognition [3–8] is a popular problem. To the best of our knowledge, Braille block detection in the form of egocentric vision has been sparsely treated so far. Yoshida et al. [9] propose a strategy to recognize Braille blocks using a sensor to detect bumps on road surfaces in autonomous mobile robot navigation. This method requires a particular sensor that cannot be used for the detection of Braille blocks from images. Okamoto et al. [10] used a convolutional neural network that learned from more than 10,000 images of training data to detect Braille blocks in images. This method requires a large amount of computational and labor costs in training, collecting data and tuning parameters despite the fact that the pattern of Braille blocks is fairly simple. Therefore, instead of collecting large amounts of data to improve accuracy, we propose the extraction of the geometric feature (linearity) and the appearance feature (yellow color) of the Braille blocks. To measure the validity of each feature, two objectives were designed and the optimal solution was achieved under the MO framework.

On the other hand, geometric feature extraction (shape recognition) research using evolutionary algorithms (EA) has been studied for a long time. Ever since Roth et al. showed that geometric primitive extraction can be treated as an optimization problem and genetic algorithm (GA) can be applied to it [11,12], various methods using EA have

been proposed. Generally, in these methods, a candidate shape for a solution (i.e., an individual) is represented as a combination of multiple points, and an objective function is designed to verify whether the solution candidates actually exist on the feature space or not. Chai et al. [13] proposed an optimization method called evolutionary tabu search (ETS), which is a combination of GA and tabu search (TS) algorithm, for geometric primitive extraction. The experimental results show the superiority of ETS in detecting ellipses from images and comparing it against optimization algorithms such as GA, simulated annealing and TS. Yao et al. [14,15] proposed Multi-Population GA, which optimizes a large number of subpopulations by evolving them in parallel, instead of evolving a single population as in the conventional GA, and showed its superiority compared to randomized Hough transform and shared GA in ellipse detection. Ayala et al. [16] proposed circle detection using GA. This method encodes an individual as a circle passing through three points and evaluates whether the circle actually exists in the edge image with an objective function. Their objective function evaluates the completeness of the candidate circle by assessing the percentage of pixels existing in the edge feature space. Değirmenci [17] showed that the parallelization capability of GPU can be used to extract geometric primitives using GA, resulting in a speedup compared to CPU. Raja and Ganesan [18] proposed a fast circle detection based on GA that reduces the search space by avoiding infeasible individual trials. Also, there are several works for line detection. Lutton and Martinez [19] proposed to use GA for geometric primitive (segment, rectangle, circle and eclipse) extraction from image. Their method uses a distance transformed image to compute the objective function. Using distance transformed images, the landscape of the objective function can be smoothed and the similarity between an individual and the original image can be measured. Mirme-hdi et al. [20] presented line segment extraction method using GA. The algorithm computes a quality scale from the statistics of gray-level values in the boxes on either side of the line segment. Kahlouche et al. [21] presented a method of geometric primitive extraction using objective function that is the sum of the average intensity of the distance transform image and the number of edge pixels on the trace of the primitive. GA is the most commonly used algorithm for geometric primitive extraction [12–21]. Besides, techniques that combine the advantages of particle swarm optimization (PSO), GA, chaotic dynamics [22], bacterial foraging optimization [23] and artificial bee colony optimization [24] are also effective alternatives. Also, other meta-heuristic search algorithms are adopted for shape search such as differential evolution (DE) [25,26], adaptive population with reduced evaluations [27] and harmony search [28].

Real-world objects can also be detected by detecting geometric shapes. Many studies have been conducted using geometric feature extraction with EA for real-world problems [29–32]. Soetedjo et al. [30] proposed to detect circular traffic signs from images using GA based eclipse detection. Cuevas et al. [31] proposed to detect white blood cells from medical images using elliptic detection with DE algorithm. Alwan et al. [32] adopted GA-based primitive extraction in vectorizing paper drawings. To solve multi-objective optimization problems (MOP) with two or three objectives, many multi-objective evolutionary algorithms (MOEA) have been proposed, such as strength Pareto evolutionary Algorithm2 (SPEA2) [33], NSGA-II [2], indicator-based evolutionary algorithm (IBEA) [34], generalized differential evolution 3 (GDE3) [35], multi-objective evolutionary algorithm with decomposition (MOEA/D) [36], non-dominated sorting genetic algorithm-III (NSGA-III) [37], improved decomposition-based evolutionary algorithm (DBEA) [38], etc. MO algorithms are generally designed to solve problems that require optimizing multiple objectives, and have been applied in the field of computer vision [39]. For example, Bandyopadhyay et al. [40] proposed land cover classification in remote sensing images with NSGA-II. This approach solves the problem by simultaneously optimizing a number of fuzzy cluster viability indexes. Mukhopadhyay et al. [41] proposed a multi-objective genetic fuzzy clustering scheme utilizing the search capability of NSGA-II, and applied it to the segmentation of MRI brain images. Nakib et al. [42] proposed image thresholding method based on NSGA-II. This method argues that optimizing multiple segmentation cri-

teria simultaneously improves the quality of the segmentation. Shanmugavadivu et al. [43] proposed multi-objective histogram equalization using PSO to achieve two major objectives of brightness preservation and contrast enhancement of images simultaneously. In addition, image segmentation using NSGA-II [44], MOEA/D [45], and watermarking algorithms using multi-objective ant colony algorithms [46] have been proposed. Among them, NSGA-II is a commonly used MO algorithm for real-world problems when the number of objectives is small.

3. Detection of Braille Block

3.1. Problem Setting and Overview

The proposed method uses both appearance and shape features to extract the Braille block region from images taken by an egocentric camera mounted on a walking person. In our problem setting, we especially aim at extracting yellow Braille blocks with linear boundaries for preventing the blind people from straying from the route as shown in Figure 1. By assuming that the walking person is initially on the Braille block, we can observe from the images that the Braille blocks extending from the bottom to the top in a perspective view. Also, as the Braille blocks from the user's egocentric viewpoint appear as regions bordered by two boundaries, the detection problem can then be treated as a task to locate yellow regions with a line pair as boundary lines. Detecting a line (segment) from an image can be understood as extracting a geometric primitive. The overview of our proposed MO based Braille block detection is shown in Figure 2. Each solution (i.e., individual) encodes parameters to define a pair of boundary lines. MO algorithm plays a role in finding solutions on a Pareto front to provide quality candidates considering both the geometric and color characteristics. After removing inferior individuals, the final solution is determined by averaging the survived individuals.

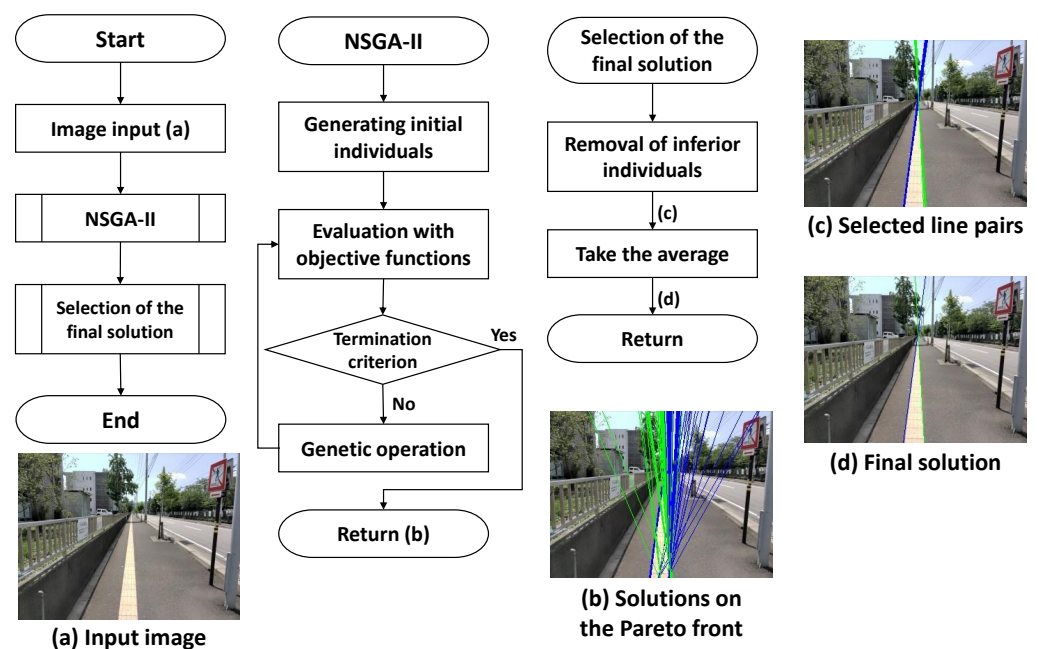


Figure 2. Overview of our proposed method.

3.2. Individual Representation and Population Initialization

We represent each individual with a pair of boundary lines, which implicitly defines the region of Braille blocks from an egocentric viewpoint. Each individual LP is real-coded, which means real variables are directly dealt with. In the perspective view, since the two boundary lines can be extended infinitely, and each extended line will have intersection points with the upper and lower boundaries of the image, respectively, only four x -axis

coordinates $LP = \{x^{lt}, x^{lb}, x^{rt}, x^{rb}\}$ are needed in total to define a line pair as shown in Figure 3. x^{lt} and x^{lb} represent the left-top and the left-bottom points of the left boundary line, and similarly, x^{rt} and x^{rb} represent right-top and the right-bottom points of the left boundary line. Each candidate solution LP in the initial population is randomly generated by sampling x -axis coordinates from the upper and the lower boundaries of the image. To accelerate the convergence and remove unpromising solutions in advance, individuals are initialized with limitations. That is, for each valid solution, the interval between the two lines at the image bottom is limited within $[20, 50]$ pixels, and the slope of the left line and the right line is limited to be smaller than $1/12\pi$ (clockwise and counterclockwise, respectively). Further, the y -coordinates of the intersection point are limited within $[0, 0.6h]$. As the valid individuals are more likely to represent valid boundaries, such an initialization strategy is expected to contribute to reaching the optimal solution earlier and reducing false detection.

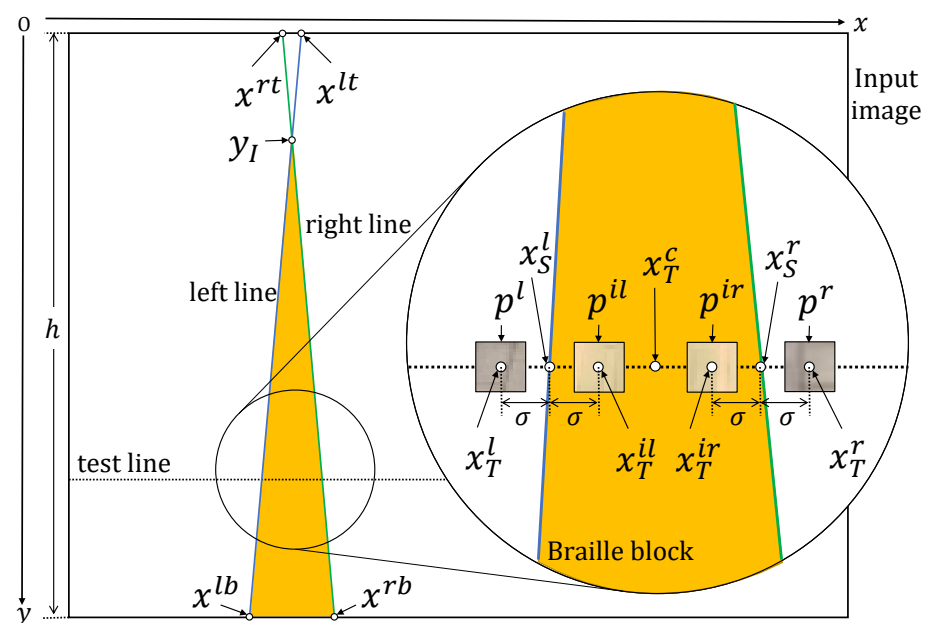


Figure 3. Notation of the variables used for calculating the objective functions.

3.3. Objective Functions

In order to evaluate whether each LP represents a reasonable region of Braille blocks, two objective functions evaluating color and shape features are simultaneously optimized. In the ideal case (no complex background, occlusion or change in appearance), the two objective functions can work collaboratively to locate the Braille block region. However, under the real-world scenarios, as the Braille blocks will show various variations of appearance, rating the individuals in terms of the combination of two objectives will lead to bad solutions with either of the objective values being low (in this paper, the problem is cast as a minimization problem). That is, we aim to obtain a solution that satisfies both objectives to some extent while a solution that satisfies both objectives can hardly exist due to the interference under real-world scenarios.

Figure 3 illustrates the variables used in the objective functions. In objective function 1, following the observation that Braille blocks are yellow, we use color histogram in the HSV color space to assess the following two facts: (1) the color histograms differ between the regions inside and outside the boundary lines; (2) the test pixels sampled from the region inside the boundary lines represent “yellow”. “Yellow” is predefined by an HSV range. The calculation process is summarized in Algorithm 1. Specifically, the HSV image of the input image is denoted by I_{HSV} . As illustrated in Figure 3, the four x -coordinates of the test points on the test line to assess the color histogram are denoted by $x_T = [x_T^l, x_T^{il}, x_T^c, x_T^{ir}, x_T^r]$. With the test points as the centers, the test patches are denoted by $p =$

$[p^l, p^{il}, p^r, p^{ir}]$, and their color histograms are $hist = [hist^l, hist^{il}, hist^r, hist^{ir}]$. In conclusion, x_T is calculated from LP and the position of the test line, p depends on x_T and I_{HSV} , and $hist$ is calculated from p . The y -coordinate of the intersection point of the line pair is y_I . y_I can be calculated from LP . If y_I is on the image, then the test line takes the intersection point as the starting point and moves down pixel by pixel for dense tests because only the Braille block region needs to be tested, else the test line starts moving from the top of the image. Over all the test lines from the intersection point to the bottom of the image, we introduce two counters c_a and c_b to collect summary statistics that contribute to the fitness value with respect to different conditions. For $condition(A)$: neither the similarity between $hist^l$ and $hist^{il}$ nor the similarity between $hist^r$ and $hist^{ir}$ is high. The purpose is to ensure clear boundary lines, which is intuitive. The similarity is calculated by comparing two histograms with respect to Bhattacharyya distance. For $condition(B)$: HSV value of x_T is within the predetermined range for defining “yellow”. The purpose is to ensure the existence of the Braille blocks. Furthermore, as shown in Figure 4, $condition(B)$ has two sub-conditions for fine-grained tests in order to improve the robustness. For $condition(B - 1)$: x_T^{il} and x_T^{ir} are yellow. For $condition(B - 2)$, x_T^{il} and x_T^c are yellow or x_T^{ir} and x_T^c are yellow. Two counters are prepared and their weights are changed according to the test points, in order to improve the noise resistance. The center test point (x_T^c) is used as a remedy in case that x_T^{il} and x_T^{ir} are severely affected by noise. Counter c_a only counts if the condition $(B - 1)$ is met, thus it contributes to position adjustment of the line pair, with low resistance against noise. c_b counts when either condition $(B - 1)$ or $(B - 2)$ is met, thus a line pair can be fitted to the Braille block region allowing a certain level of noise. As the MO problem is set as a minimization optimization problem, the objective values v_1 is set to be negative.

Algorithm 1: Objective Function 1

```

input : An individual  $LP$ ,
        HSV image  $I_{HSV}$ 
output: Objective value 1  $v_1$ 

// initialization
1  $c_a = 0, c_b = 0;$  // counters
2  $y_T = 0;$  //  $y$ -coordinate of the initial test line
3  $y_I = \text{calcIntersectionPoint}(LP);$  //  $y$ -coordinate of the intersection point
4 if  $0 < y_I < h$  then
5 |  $y_T = y_I;$ 
6 end
// calculate objective value
7 for  $y \in [y_T, h]$  do
8 |  $x_T = \text{calcTestPoints}(LP, y);$ 
9 |  $p = \text{makePatches}(x_T, I_{HSV});$ 
10 |  $hist = \text{calcHistogram}(p);$ 
11 | if  $condition(A)$  is met then
12 | | if  $condition(B - 1)$  is met then
13 | | |  $c_a = c_a + 1;$ 
14 | | |  $c_b = c_b + 1;$ 
15 | | else if  $condition(B - 2)$  is met then
16 | | |  $c_b = c_b + 1;$ 
17 | | end
18 | end
19 end
20 return  $v_1 = -(c_a + c_b);$ 

```

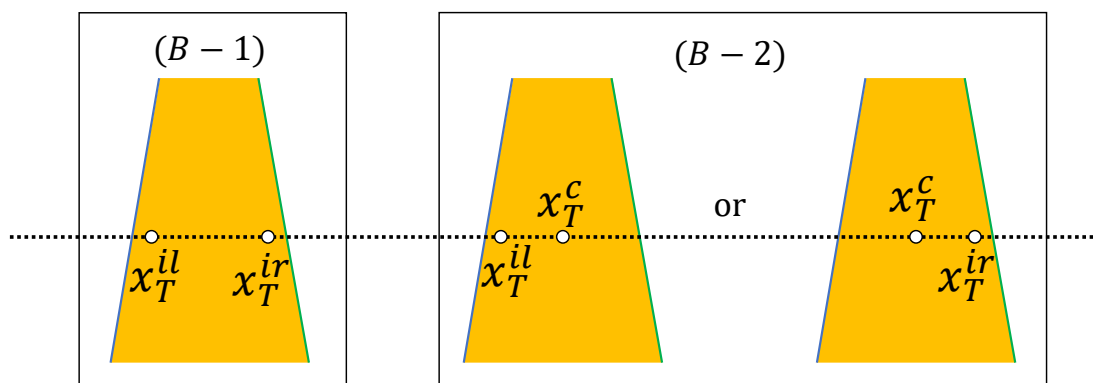


Figure 4. Test points in *condition B*.

Objective function 2 exploits linear shape features as summarized in Algorithm 2. Specifically, given a distance transform image I_{DT} transformed from the edge image, we aim to find the Braille block boundaries by minimizing the sum of distance. In I_{DT} , the value of each pixel $I_{DT}(x, y)$ is the Euclidean distance from the nearest edge as illustrated in Figure 5. Therefore, the sum of the pixel values of the points on the line pair below the intersection point can be treated as the likelihood of an individual representing boundaries. The x -coordinate of the sample points for calculating the distance are denoted by $x_S = [x_S^l, x_S^r]$, which is illustrated in Figure 4.

Algorithm 2: Objective Function 2

input : An individual LP
Distance transform image I_{DT}
output: Objective value v_2

```

// initialization
1  $v_2 = 0$ ; // sum of distance
2  $y_T = 0$ ; // y-coordinate of the initial test line
3  $y_I = \text{calcIntersectionPoint}(LP)$ ;
4 if  $0 < y_I < h$  then
5 |  $y_T = y_I$ ;
6 end
// calculate the sum of distance
7 for  $y \in [y_T, h]$  do
8 |  $x_S = \text{calcSamplePoints}(LP, y)$ ;
9 |  $v_2 = v_2 + I_{DT}(x_S, y)$ ;
10 end
11 return  $v_2$ ;

```



Figure 5. (a) Input image, (b) Edge image, (c) Distance transform image.

3.4. Genetic Operators and Termination Criterion

We adopt NSGA-II [2] as the main algorithm to solve our MO problem described in the previous section. Also, other popular MO algorithms are compared in Section 4. Specifically, three genetic operators including selection, crossover and mutation are used. Crowded binary tournament selection without replacement is used as the selection operator. To propagate the elite individuals found in the previous searches to the next generation, the non-dominant solutions for the parent populations are used based on the non-domination rank and crowding distance. For the crossover operator, the simulated binary crossover (SBX) operator [47] is used. SBX simulates a single-point crossover of binary-encoded real-valued decision variables. For the mutation operator, the polynomial mutation (PM) operator [48] is used. PM simulates binary-encoded bit-flip mutations in a real-valued decision variable. In our experiment, the crossover probability is taken as 1.0 as suggested by [49–51] and the mutation probability is set to 0.25. As to the termination condition, the iteration is run by a determined number of generations.

3.5. Selection of the Final Solution

To generate the final solution based on the solutions on the Pareto front in the last generation, we propose a two-stage strategy: first, remove the individuals that either the y -coordinate of the intersection point belongs to $[0.6h, h]$ or overlapped, which turns out to be able to remove implausible solutions. Second, the average of the remaining individuals is taken as the final solution.

4. Experimental Results

The performance of our proposed method was evaluated by comparing the line pair result with the manually annotated ground truth over our originally collected dataset. Our dataset consists of 50 test images taken by the Vuzix M400 Smart Glasses, which includes five categories in total: illumination change, shadow, deficiency, obstacle and change of view angle. Each category contains 10 images in a size of 320×240 pixels. The MOEAs programs used in the experiment are obtained from MOEA Framework 2.13 (<http://moeaframework.org/>, accessed on 1 December 2020), a java open source library. Each numerical result is averaged by 10 trials with different random seeds. The parameters of NSGA-II and other MOEAs used in the experiments are shown in Table 1, and descriptions of the parameters used are listed in Table 2. For the quantitative evaluation, the evaluation criterion is set as the mean location error, which is calculated by the root mean square error (RMSE) defined as follows,

$$RMSE = \left(\sum_{y=y_G}^h (x_R - x_G)^2 / (h - y) \right)^{1/2}. \quad (1)$$

where x_R and x_G are the x -coordinates of the final solution and the ground truth. y_G is the y -coordinate of the intersection point of the ground truth.

Table 1. Parameter setting in the experiment. Detail description is summarized in Table 2.

MOEA	POP	GEN	SR	PR	DIV	OS	DR	DS	NS	δ	η
NSGA-II [2]	200	50	1.0	0.25	-	-	-	-	-	-	-
SPEA2 [33]	200	50	1.0	0.25	-	200	-	-	-	-	-
IBEA [34]	200	50	1.0	0.25	-	-	-	-	-	-	-
GDE3 [35]	200	50	-	-	-	-	0.1	0.5	-	-	-
MOEA/D [36]	200	50	1.0	0.25	-	-	-	-	20	0.9	2
NSGA-III [37]	200	50	1.0	0.25	4	-	-	-	-	-	-
DBEA [38]	200	50	1.0	0.25	4	-	-	-	-	-	-

Table 2. Description of parameters shown in Table 1.

Par.	Description
POP	Population size.
GEN	Generation size.
SR	Crossover rate of the simulated binary crossover.
PR	Mutation rate of the polynomial mutation.
DIV	Number of divisions.
OS	Number of offspring generated per iteration.
DR	Crossover rate for differential evolution.
DS	Size of each step taken by differential evolution.
NS	Size of the neighborhood for mating.
δ	Probability of mating with an individual from the neighborhood versus the entire population.
η	Maximum number of spots in the population that an offspring can replace.

4.1. Parameter Tuning

First, the results by varying population size and number of generations are shown in Figure 6. It can be observed from Figure 6a that the results of population size = 200, 250 and 300 are very close, which indicates that it is difficult to improve the performance by increasing the population size after 200. On the other hand, with the population size fixed as 200, we can observe from Figure 6b that the number of generations is proportional to the performance and the improvement is trivial after 30. Finally, the optimum setting (population size = 200, generation size = 50) is adopted for the following experiments.

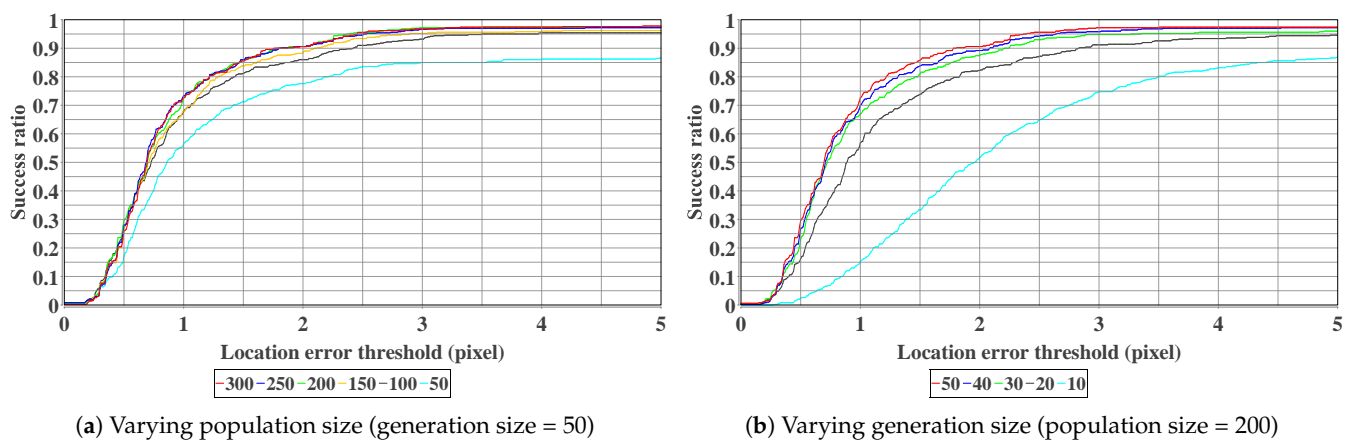


Figure 6. Curves show the average success ratio of 10 trials with different parameter settings. All the test images are used for plotting. The curve closer to the top left represents better performance. Best view in color.

4.2. Performance Evaluation and Limitation Analysis

We present quantitative and qualitative results in this section. Figure 8a shows the overall accuracy with respect to the whole test dataset. As can be observed, a high success ratio (>0.9) can be achieved when the threshold value of the mean location error is larger than two pixels. Success ratio indicates the percentage of the test images that are successfully detected. The success ratio can further increase to 0.95 when three pixels of error are allowed for the detection result.

Our proposed method can also detect Braille blocks under various environments as shown in Figure 7. From the quantitative analysis in Figure 8b, we can observe that our proposed method is especially robust to deal with shadows, deficiency and change of view angle. In the case of obstacles, as Braille blocks are partially covered by obstacles, color and shape features cannot be sufficiently obtained in some test images. In the case of illumination changes, color features are mainly affected, resulting in a decrease in the

success ratio in both cases. False-positive detection is more likely to happen when either of the features is inadequate or one of the features is implausible. As can be observed in Figure 9, our proposed method has limitations especially when it gets dark or obstacles exist on the road. When the color range for defining “yellow” changes significantly, the predetermined HSV range becomes ineffective. Instead of a fixed range, an adaptive color range could probably solve this problem.

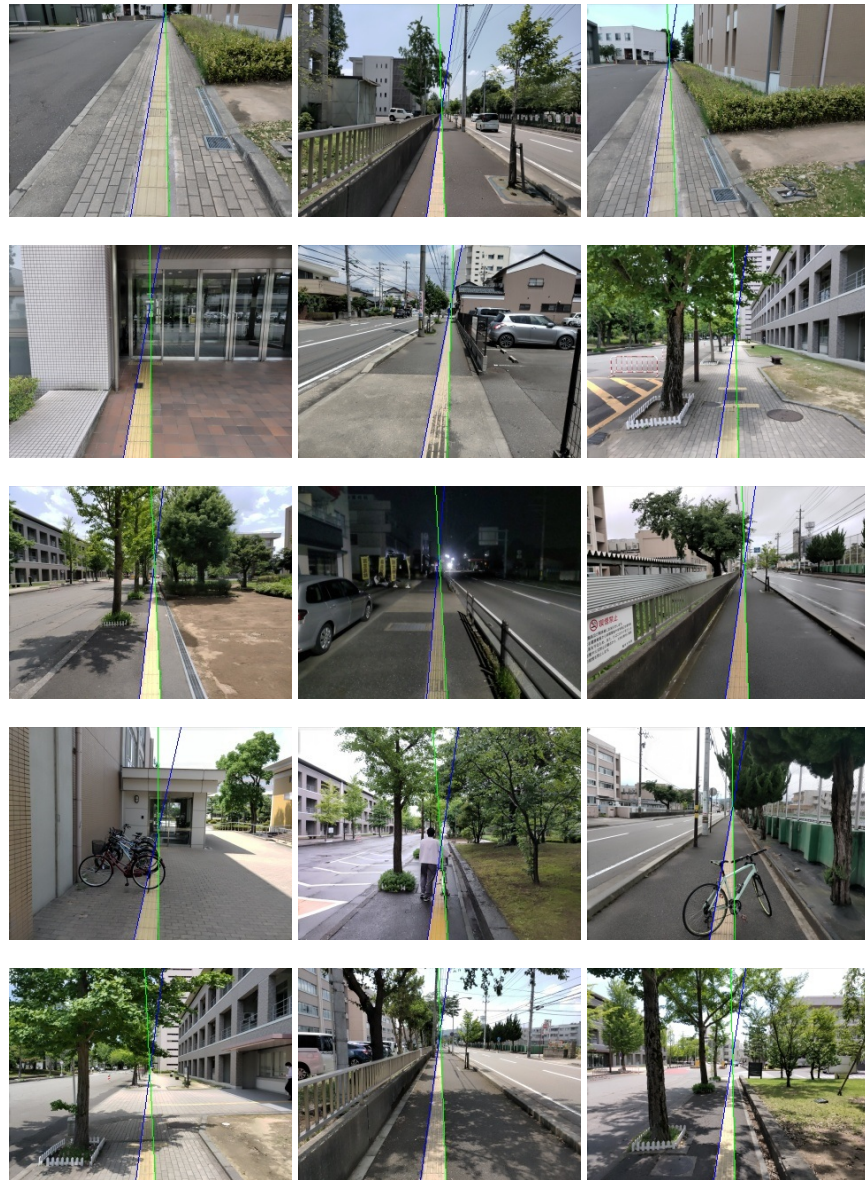


Figure 7. Category-wise qualitative results. A line pair consists of a blue line (left) and a green line (right). Categories from the top row to the bottom row: change of view angle, deficiency, illumination change, obstacle and shadow.

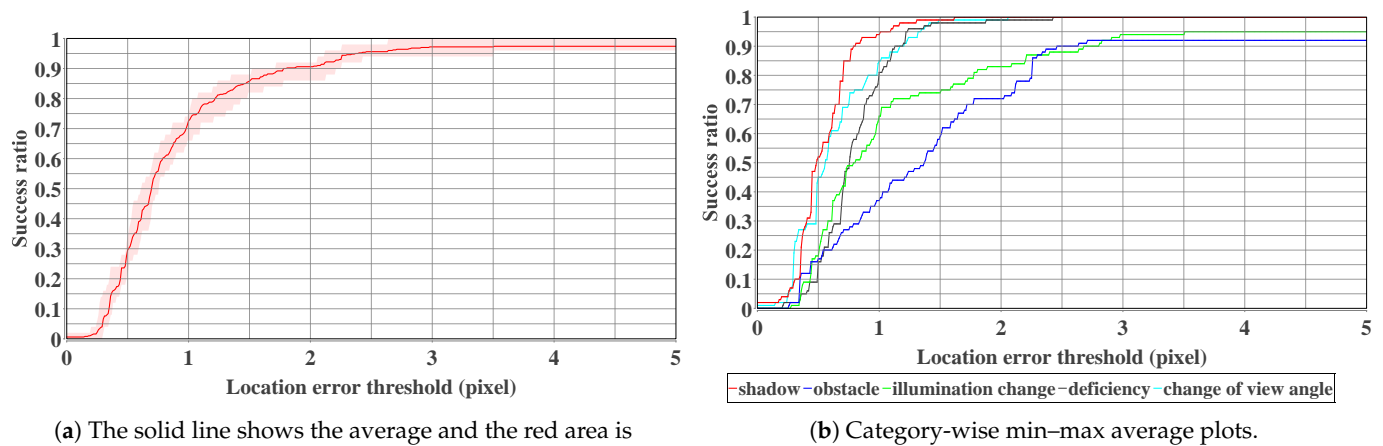


Figure 8. Success ratio plot with respect to the whole (a) and partial (b) dataset.



Figure 9. Examples of failure defections. 1st~3rd examples show the case of illumination change and the last example shows the case of obstacles.

4.3. Comparison over Different MOEAs

Despite NSGA-II, we also test other MOEAs for comparison and provide a reference for future studies. Six MOEAs for multi-objective optimization, namely, SPEA2 [33], IBEA [34], GDE3 [35], MOEA/D [36], NSGA-III [37] and DBEA [38] are compared with the parameter setting summarized in Table 1. As can be seen from Figure 10, NSGA-II, SPEA2, IBEA and NSGA-III show close performance in our Braille block detection problem. Among them, NSGA-II has the highest performance. Also, as NSGA-II has fewer parameters that need to be adjusted compared to SPEA2 and NSGA-III, it is considered the most suitable off-the-shelf MOEA for solving our task in this paper. NSGA-III and SPEA2, which are based on Pareto domination, are considered to be able to perform a similar solution search as NSGA-II. Besides, Indicator-based IBEA has also shown competitive results. GDE3 fails in detecting some difficult test images and is more likely to be trapped by local optima in this task. Additionally, in Figure 11 we show the Pareto front approximation of NSGA-II obtained in the experiment. In the ideal case, as shown in the top image of Figure 11, we can observe a clear trade-off relationship between the two objective functions.

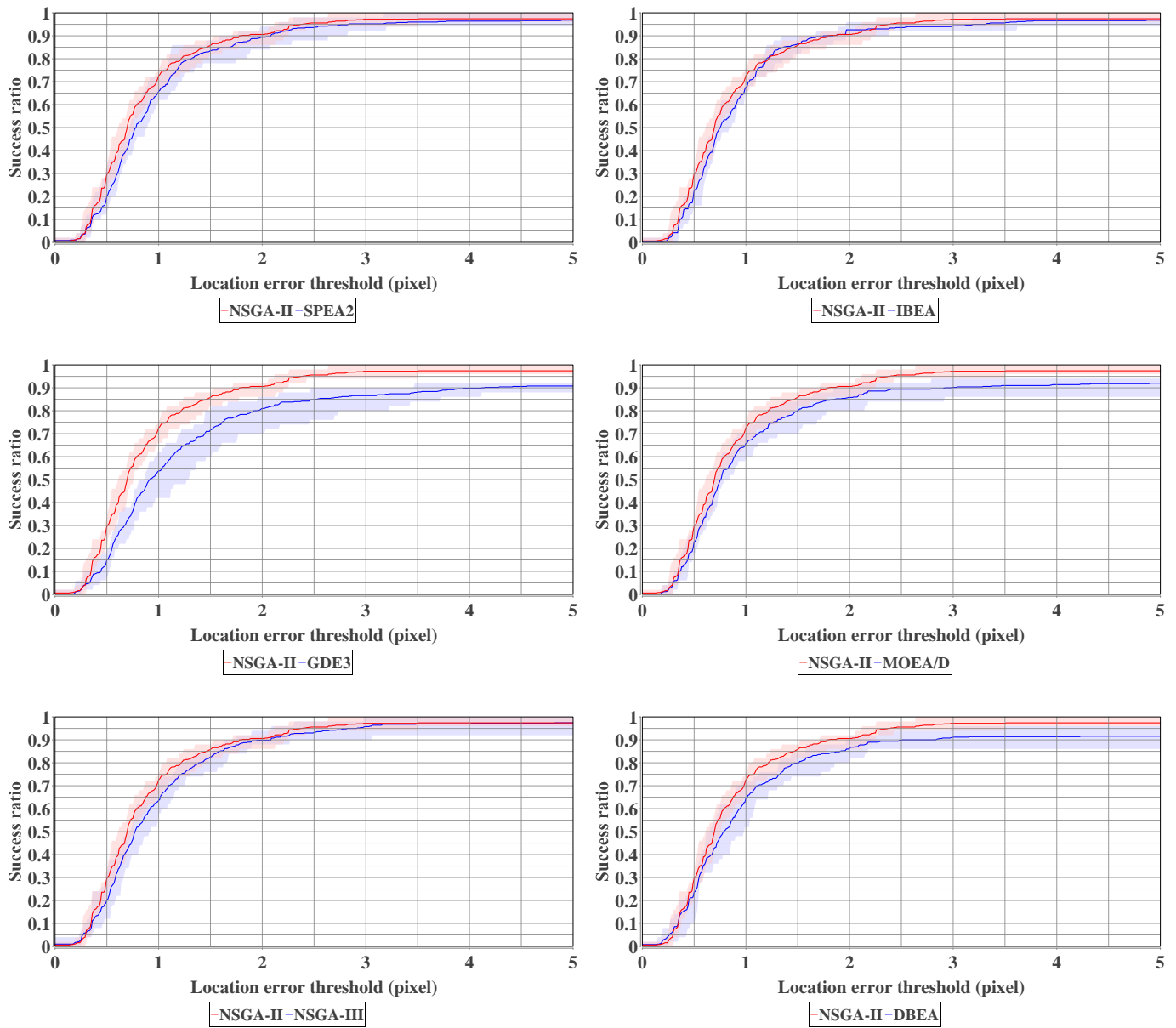


Figure 10. Comparative experiment. All test images are conducted 10 trials for the plots.

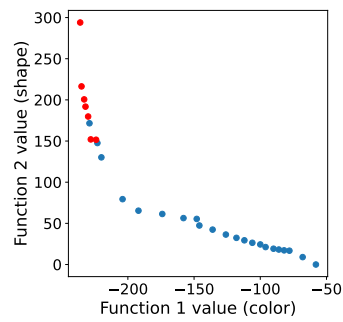


Figure 11. Cont.

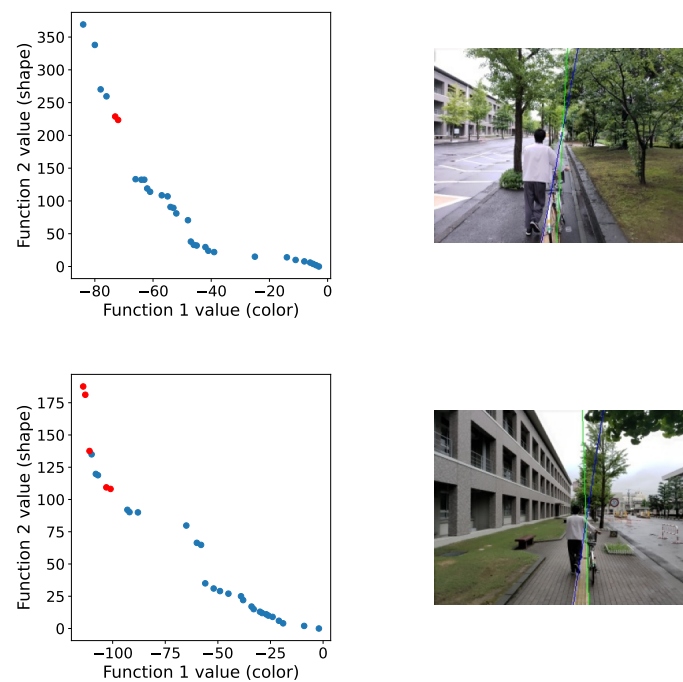


Figure 11. Pareto front approximation of NSGA-II in the final generation. The blue circles show the non-dominated solutions, and the red circles show the selected solution for final decision making (i.e., the average of the red circles is the final result plotted on the right image).

5. Conclusions

In this paper, we presented a method to detect Braille blocks under the framework of multi-objective optimization, which indicates that multi-objective optimization algorithms are potentially useful tools for solving real-world computer vision problems. Besides, we originally built a fully annotated dataset that contains five subcategories for validation. Experimental results show that the proposed method is effective in detecting Braille blocks from an egocentric viewpoint under real scenarios. As a limitation, our method tends to fail when either both of the features (geometric feature and color feature) are inadequate or one of the features are implausible. Nevertheless, in most cases, the algorithm driven by multi-object optimization can select a suitable solution from the solution space even one feature is inadequate due to illumination change, obstacle, deficiency or shadow. As future work, we aim at reducing the computational cost by enlarging the step for sampling test lines and patches for real-time applications, which can further contribute to the walking support for visually impaired people.

Author Contributions: Conceptualization, C.Z.; data curation, T.T. and T.N.; methodology, C.Z., T.T. and T.N.; supervision, C.Z.; writing—original draft, T.T. and C.Z.; writing—review and editing, T.T., T.A. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This article is supported by the Tateisi Science and Technology Foundation and JSPS KAKENHI Grant Number JP20K19568.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhowmick, A.; Hazarika, S.M. An insight into assistive technology for the visually impaired and blind people: State-of-the-art and future trends. *J. Multimodal User Interfaces* **2017**, *11*, 149–172. [[CrossRef](#)]
2. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
3. Ren, X.; Philipose, M. Egocentric recognition of handled objects: Benchmark and analysis. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, USA, 20–25 June 2009; pp. 1–8. [[CrossRef](#)]
4. Ren, X.; Gu, C. Figure-ground segmentation improves handled object recognition in egocentric video. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3137–3144. [[CrossRef](#)]
5. Fathi, A.; Ren, X.; Rehg, J.M. Learning to recognize objects in egocentric activities. In Proceedings of the CVPR 2011, Colorado, CO, USA, 20–25 June 2011; pp. 3281–3288. [[CrossRef](#)]
6. Kang, H.; Hebert, M.; Kanade, T. Discovering object instances from scenes of daily living. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 762–769. [[CrossRef](#)]
7. Liu, Y.; Jang, Y.; Woo, W.; Kim, T.K. Video-based object recognition using novel set-of-sets representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 519–526.
8. Abbas, A.M. *Object Detection on Large-Scale Egocentric Video Dataset*; University of Stirling: Stirling, UK, 2018.
9. Yoshida, T.; Ohya, A. Autonomous mobile robot navigation using braille blocks in outdoor environment. *J. Robot. Soc. Jpn.* **2004**, *22*, 469–477. [[CrossRef](#)]
10. Okamoto, T.; Shimono, T.; Tsuboi, Y.; Izumi, M.; Takano, Y. Braille Block Recognition Using Convolutional Neural Network and Guide for Visually Impaired People. In Proceedings of the 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, The Netherlands, 17–19 June 2020; pp. 487–492. [[CrossRef](#)]
11. Roth, G.; Levine, M.D. Extracting geometric primitives. *CVGIP Image Underst.* **1993**, *58*, 1–22. [[CrossRef](#)]
12. Roth, G.; Levine, M.D. Geometric primitive extraction using a genetic algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 901–905. [[CrossRef](#)]
13. Chai, J.; Jiang, T.; De Ma, S. Evolutionary tabu search for geometric primitive extraction. In *Soft Computing in Engineering Design and Manufacturing*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 190–198. [[CrossRef](#)]
14. Yao, J.; Kharna, N.; Grogono, P. Fast robust GA-based ellipse detection. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26–26 August 2004; Volume 2, pp. 859–862. [[CrossRef](#)]
15. Yao, J.; Kharna, N.; Grogono, P. A multi-population genetic algorithm for robust and fast ellipse detection. *Pattern Anal. Appl.* **2005**, *8*, 149–162. [[CrossRef](#)]
16. Ayala-Ramirez, V.; Garcia-Capulin, C.H.; Perez-Garcia, A.; Sanchez-Yanez, R.E. Circle detection on images using genetic algorithms. *Pattern Recognit. Lett.* **2006**, *27*, 652–657. [[CrossRef](#)]
17. Değirmenci, M. Complex geometric primitive extraction on graphics processing unit. *J. WSCG* **2010**, *18*, 129–134.
18. Raja, M.M.R.; Ganesan, R. VLSI Implementation of Robust Circle Detection On Image Using Genetic Optimization Technique. *Asian J. Sci. Appl. Technol. (AJSAT)* **2014**, *2*, 17–20.
19. Lutton, E.; Martinez, P. A genetic algorithm with sharing for the detection of 2D geometric primitives in images. In *European Conference on Artificial Evolution*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 287–303. [[CrossRef](#)]
20. Mirmehdi, M.; Palmer, P.; Kittler, J. Robust line segment extraction using genetic algorithms. In Proceedings of the 1997 Sixth International Conference on Image Processing and Its Applications, IET, Dublin, Ireland, 14–17 July 1997; Volume 1, pp. 141–145. [[CrossRef](#)]
21. Kahlouche, S.; Achour, K.; Djekoune, O. Extracting geometric primitives from images compared with genetic algorithms approach. *Comput. Eng. Syst. Appl.* **2003**.
22. Dong, N.; Wu, C.H.; Ip, W.H.; Chen, Z.Q.; Chan, C.Y.; Yung, K.L. An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection. *Comput. Math. Appl.* **2012**, *64*, 1886–1902. [[CrossRef](#)]
23. Dasgupta, S.; Das, S.; Biswas, A.; Abraham, A. Automatic circle detection on digital images with an adaptive bacterial foraging algorithm. *Soft Comput.* **2010**, *14*, 1151–1164. [[CrossRef](#)]
24. Cuevas, E.; Sención-Echauri, F.; Zaldivar, D.; Pérez-Cisneros, M. Multi-circle detection on images using artificial bee colony (ABC) optimization. *Soft Comput.* **2012**, *16*, 281–296. [[CrossRef](#)]
25. Das, S.; Dasgupta, S.; Biswas, A.; Abraham, A. Automatic circle detection on images with annealed differential evolution. In Proceedings of the 2008 Eighth International Conference on Hybrid Intelligent Systems IEEE, Barcelona, Spain, 10–12 September 2008; pp. 684–689. [[CrossRef](#)]
26. Cuevas, E.; Zaldivar, D.; Pérez-Cisneros, M.; Ramírez-Ortegón, M. Circle detection using discrete differential evolution optimization. *Pattern Anal. Appl.* **2011**, *14*, 93–107. [[CrossRef](#)]
27. Cuevas, E.; Santuario, E.L.; Zaldivar, D.; Perez-Cisneros, M. Automatic circle detection on images based on an evolutionary algorithm that reduces the number of function evaluations. *Math. Probl. Eng.* **2013**, *2013*. [[CrossRef](#)]
28. Fourie, J. Robust circle detection using harmony search. *J. Optim.* **2017**, *2017*. [[CrossRef](#)]

29. Kharma, N.; Moghnieh, H.; Yao, J.; Guo, Y.P.; Abu-Baker, A.; Laganieri, J.; Rouleau, G.; Cheriet, M. Automatic segmentation of cells from microscopic imagery using ellipse detection. *IET Image Process.* **2007**, *1*, 39–47. [[CrossRef](#)]
30. Soetedjo, A.; Yamada, K. Fast and robust traffic sign detection. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics IEEE, Waikoloa, HI, USA, 12 October 2005; Volume 2, pp. 1341–1346. [[CrossRef](#)]
31. Cuevas, E.; Díaz, M.; Manzanares, M.; Zaldivar, D.; Perez-Cisneros, M. An improved computer vision method for white blood cells detection. *Comput. Math. Methods Med.* **2013**, *2013*. [[CrossRef](#)]
32. Alwan, S.; Caillec, J.M.; Meur, G. Detection of Primitives in Engineering Drawing using Genetic Algorithm. In Proceedings of the ICPRAM 2019, Prague, Czech Republic, 9–21 February 2019; pp. 277–282. [[CrossRef](#)]
33. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report* **2001**, *103*. [[CrossRef](#)]
34. Zitzler, E.; Künzli, S. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 832–842. [[CrossRef](#)]
35. Kukkonen, S.; Lampinen, J. GDE3: The third evolution step of generalized differential evolution. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, IEEE, Edinburgh, UK, 2–5 September 2005; Volume 1, pp. 443–450. [[CrossRef](#)]
36. Li, H.; Zhang, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2008**, *13*, 284–302. [[CrossRef](#)]
37. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [[CrossRef](#)]
38. Asafuddoula, M.; Ray, T.; Sarker, R. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 445–460. [[CrossRef](#)]
39. Nakane, T.; Bold, N.; Sun, H.; Lu, X.; Akashi, T.; Zhang, C. Application of evolutionary and swarm optimization in computer vision: A literature survey. *IPSJ Trans. Comput. Vis. Appl.* **2020**, *12*, 1–34. [[CrossRef](#)]
40. Bandyopadhyay, S.; Maulik, U.; Mukhopadhyay, A. Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1506–1511. [[CrossRef](#)]
41. Mukhopadhyay, A.; Maulik, U.; Bandyopadhyay, S. Multiobjective Genetic Clustering with Ensemble Among Pareto Front Solutions: Application to MRI Brain Image Segmentation. In Proceedings of the 2009 Seventh International Conference on Advances in Pattern Recognition, Kolkata, India, 4–6 February 2009; pp. 236–239. [[CrossRef](#)]
42. Nakib, A.; Oulhadj, H.; Siarry, P. Image thresholding based on Pareto multiobjective optimization. *Eng. Appl. Artif. Intell.* **2010**, *23*, 313–320. [[CrossRef](#)]
43. Shanmugavadivu, P.; Balasubramanian, K. Particle swarm optimized multi-objective histogram equalization for image enhancement. *Opt. Laser Technol.* **2014**, *57*, 243–251. [[CrossRef](#)]
44. De, S.; Bhattacharyya, S.; Chakraborty, S. Color Image Segmentation by NSGA-II Based ParaOptiMUSIG Activation Function. In Proceedings of the 2013 International Conference on Machine Intelligence and Research Advancement, Katra, India, 21–23 December 2013; pp. 105–109. [[CrossRef](#)]
45. Zhang, M.; Jiao, L.; Ma, W.; Ma, J.; Gong, M. Multi-objective evolutionary fuzzy clustering for image segmentation with MOEA/D. *Appl. Soft Comput.* **2016**, *48*, 621–637. [[CrossRef](#)]
46. Wang, J.; Peng, H.; Shi, P. An optimal image watermarking approach based on a multi-objective genetic algorithm. *Inf. Sci.* **2011**, *181*, 5501–5514. [[CrossRef](#)]
47. Deb, K.; Agrawal, R. Simulated Binary Crossover for Continuous Search Space. *Complex Syst.* **1994**, *9*, 115–148.
48. Deb, K.; Goyal, M. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Comput. Sci. Inform.* **1996**, *26*, 30–45.
49. Bandyopadhyay, S.; Bhattacharya, R. Applying modified NSGA-II for bi-objective supply chain problem. *J. Intell. Manuf.* **2013**, *24*, 707–716. [[CrossRef](#)]
50. Kalaivani, L.; Subburaj, P.; Willjuice Iruthayarajan, M. Speed control of switched reluctance motor with torque ripple reduction using non-dominated sorting genetic algorithm (NSGA-II). *Int. J. Electr. Power Energy Syst.* **2013**, *53*, 69–77. [[CrossRef](#)]
51. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 3045–3052. [[CrossRef](#)]