Antonio Miguel Batista DOURADO[1,2], Emerson Carlos PEDRINO[2]

Federal Institute of São Paulo (1), Federal University of São Carlos (2), Brazil

doi:10.15199/48.2020.01.25

Automatic generation of high performance morphological filters to fix missing data in depth images on real-time embedded systems for visually impaired people

Abstract. *The paper presents an evolutionary multi-objective approach to automatically generate morphological filters to solve unknown distances areas, found in depth images used by real-time embedded systems for visually impaired people, and to prevent accidents. It was used Cartesian Genetic Programming as base for the NSGAII multi-objective optimization algorithm proposed to optimize two objectives: low error rates for quality x low complexity for speed. Results showed this approach was able to deliver feasible solutions with good quality and speed to be used in real-time systems.*

Streszczenie. *W artykule zaprezentowano metodę ewolucyjną do automatycznego generowania morfologicznego filtru do określania brakujących danych w obrazach ludzi otrzymywanych on-line. Użyto programu Cartesian Genetic do optymalizacji algorytmu. Zastosowane rozwiązanie umożliwiało dostarczanie poprawę szybkości o dokładności przetwarzania obrazu.**(Automatyczne generowanie filtru morfologicznego umożliwiającego odzyskiwanie brakujących danych w systemie wizualizacji i identyfikacji obrazu człowieka)***

Keywords: Image Processing, Real-Time Embedded Systems, Cartesian Genetic Programming, Multi-Objective Optimization, Assistive Technologies.
Słowa kluczowe: przetwarzanie obrazu, systemy wbudowane, identyfikacja pobrazu

Introduction

Real-time embedded systems are used in many areas, ranging from illumination control to airplane critical systems [1, 2, 3]. These systems are built on top of a set of constraints of one or more tasks with varying levels of tolerance regarding required times to accomplish each task within the resource limitations presented by certain hardware (i.e.: low memory, processor speed,etc.) [4, 5].

Navigation systems for visually impaired people (VIP) can also be embedded in portable hardware for assisting that kind of user on many daily tasks (such as avoiding and/or identifying obstacles, walking around with safety, etc.) by acquiring data, processing them and delivering important information in real-time [6, 7]. Most of the required data might be acquired from various devices like RFID tags, infrared lasers (IR), inertial measurement units (IMUs), cameras, ultrasonic sensors, and others, being processed by specific algorithms for each type of input data and the results are sent to VIP as feedback [8, 9, 10, 11]. There are two main methods of feedback for navigation systems: tactile and acoustic. Tactile feedback uses physical stimuli to pass information (i.e.: vibration motors on arms) [12]. Acoustic or sound feedback use music, tones and/or voices sending them through earphones or speakers to VIPs [13].

Vision-based systems use cameras as main input devices, specifically RGB-D ones which also retrieve depth data from environment and this process depends on the technique being used by that kind of camera [14]. Stereoscopic, structured light, passive infrared, and time-of-flight are the most popular techniques with their pros and cons [6, 15, 16]. However, all RGB-D cameras share a common problem regarding areas with unknown depth estimation. This is noticed when depth data are visualized as gray-scale level images and black pixels indicating unknown areas [17], leading those navigation systems to output wrong information to VIPs and possibly jeopardize their safety. Fig. 1 shows depth images from structured light and stereoscopic cameras.

Unknown distances in depth images for VIP navigation systems can be fixed with the mathematical morphology operators [18]. One could manually design morphological filters, in this context, by defining each operation and the structuring elements, however, this might be an exhaustive and very slow process as each change must be evaluated across multiple scenarios. This can be automated by Cartesian Genetic Programming (CGP) [19] to generate and evolve low error filters based upon ideal results, also known as ground truth or target images [20]. Although desirable, low error rate of evolved filters for depth images, used by real-time embedded navigation systems for VIP, is not the sole objective. Also, the complexity of those filters might compromise the feedback throughput, jeopardizing the whole system and making it unreliable due to the time spent [21, 22]. Thus, a good high performance filter must have the lowest complexity as much as possible to be executed without causing impact on performance and still being able to fix unknown distance areas, turning this problem into a two-objective scenario, thereby being addressable by Multi-Objective Optimization (MOO) algorithms aiming at a feasible solution that meets both objectives within an acceptable trade-off margin between them [23].

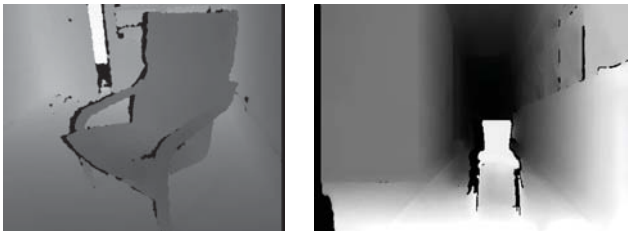

Fig. 1. Depth images captured by RGB-D cameras: structured-light (left) and stereoscopic(right)

The generation and evolution of morphological filters using CGP was presented previously by the author's research group. That approach allowed to generate complex and high-quality filters that fit in generic scenarios with a single objective [20]. This work presents an evolutionary multi-objective approach to optimize high performance morphological filters for depth images used by an ongoing project involving a real-time embedded navigation system for VIP [6]. Also, this approach was based on CGP to generate and evolve filters, and NSGAII to optimize them by focusing on low error and low complexity objectives. The results showed that the filters evolved and optimized achieved good quality while keeping low complexity, affecting positively the performance of the filter and might reflect in less power consumption as well.

Cartesian Genetic Programming

Cartesian Genetic Programming is a sub-area of Genetic Programming in which the population is a group of individuals (chromosomes) represented as directed graphs ar-

ranged in two-dimensional arrays of genes. Each chromosome has a predefined number of rows, columns and levels-back, which indicates up to how many columns back a gene can receive input from [19].

The evolutionary process of CGP chromosomes takes place mostly through mutations as in Evolutionary Strategy. Besides, in the literature is shown that point mutation is the most popular genetic operator in CGP that uses the $(1+\lambda = 4)$ strategy which selects the best $(1+\lambda)$ as parents (often $\lambda = 4$) and mutates them to generate offsprings[24, 25].

Nondominated Sorting Genetic Algorithm

In the early 2000s, Deb and colleagues introduced a novel evolutionary multiobjective optimization algorithm called Nondominated Sorting Genetic Algorithm II (NSGAII) [26]. This algorithm employs an elitist approach which splits a population into groups named "fronts", according to the same dominance level of individuals, meaning that an individual x_a dominates x_b if the objectives of x_a are equal to the x_b and at least one of them is better [23]. Each front is also internally sorted by a crowding-distance value defined by how dense it is the region of an individual [27].

NSGAII uses the $2N$ elitist approach such that, for a N-sized population, N new offsprings are generated through recombination/mutation process and added to the current population, making it $2N$-sized. Afterward, the double-sized population is sorted in fronts and by crowding-distance, and the N best individuals are selected for the next generation and so on [26].

Proposed Approach

In this approach, NSGAII was adapted and implemented on the top of the CGP base, by adjusting only a few and crucial details to guarantee the optimal utilization of both techniques. The whole process is guided by a set of execution parameters with all the information needed for the training. Fig. 2 illustrates the proposed approach. The population now is composed of individuals (chromosomes) with $NR * NC + OUT$ $(NR, NC \geq 1)$ genes of LB $(1 \leq LB \leq NC)$ levels-back, where NR and NC are rows and columns, respectively. OUT is the output gene and LB is the number of columns back a gene can receive input from. They share an input set with multiple structuring elements (SE) and one training image, which is the raw depth image to be fixed containing unknown distance areas. Every gene receives two inputs (images or structuring elements), process them using a function operator and produces one output which can be an image or even a new structuring element if both inputs are SEs. Fig. 3 exemplifies an individual composed by one input image and three structuring elements with different shapes. Chromosomes have three columns, two rows and levels-back = NC (genes can receive input from any previous column).

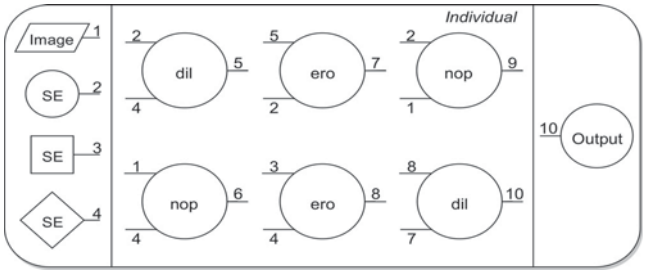

Fig. 3. Example of CGP individual with three columns, two rows, levels-back = NC and input set with one image and three structuring elements.

Since CGP and NSGAII algorithms differ in the evolution strategies, it was needed to define which strategy would be better when dealing with two-objective problems. An elitist form of $(1+\lambda)$ was designed to select the best fronts instead individuals but it was outperformed by the original $2N$ strategy, which randomly selects N (population size) individuals and mutates a predefined percentage of them to generate new offsprings until the population reaches $2N$ size. Thus, parents and offsprings are ranked by fronts and distances, and the best N ones are turned into the new population for the next generation. Recombination techniques are not used here. Algorithm 1 shows the implemented approach as pseudo-code.

Still according to Algorithm 1, a set of parameters must be set up to guide the execution and provide resources used through the evolution and optimization process. Table 1 lists all required parameters to run the proposed approach.

Six shapes were used as structuring elements: circle, square, octagon, diamond, rectangle, and 1-D line with dimensions varying between 1 and 7, by considering only odd numbers. Additionally, every line was rotated to 45, 90 and 135 degrees for each dimension. Due to empirical analysis, it was observed no leverage with dimensions higher than 7 for any shape. Thus, 40 SEs resulted from these shapes and dimensions.

Inputs received by genes are processed by one of the six functions defined in Table 2, two morphological operators, three logical operators and a bypass. Depending on whether the inputs are pairs of images, pairs of SEs, or whatever,

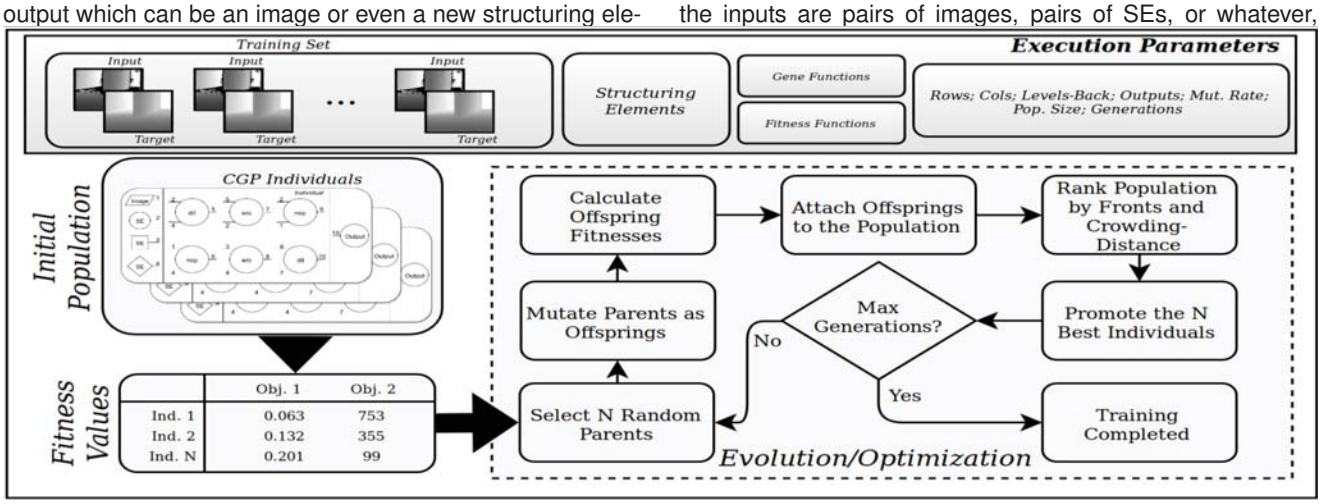

Fig. 2. Proposed approach

Algorithm 1: Pseudo-code of implemented approach

Input: $PARAMS$
Result: $\{POP, RANK\}$
POP ← *genRandomPopulation*(PARAMS)
FITARR ← *calcFitness*(POP,PARAMS)
PARAMS.get(NGEN,NPOP,TMUT)
GENS ← 0
while $GENS < NGEN$ **do**
  for $i = 1$ **to** $NPOP$ **do**
    PARENT ← *selRandParent*(POP)
    OFFSPR ← *mutate*($PARENT, TMUT$)
    OFFSFIT ← *calculateFitness*($OFFSPR$)
    POP[$NPOP + i$] ← OFFSPR
    FITARR[$NPOP + i$] ← OFFSFIT
  end
  FRONTS ← *calcFronts*(POP,FITARR)
  RANK ← *calcCrowdingDist*(POP,FRONTS)
  POP ← POP[RANK[1 to NPOP]]]
  FITARR ← FITARR[RANK[1 to NPOP]]
end

Table 1. Required parameters.

Parameter	Used	Description
IM_SET	10 pairs	Set of training image pairs {input,target}
SE_SET	40 SE	Set of structuring elements
NGEN	100	Number of training generations
NPOP	1000	Population size
NC	500	Number of columns
NR	1	Number of rows
LB	500	Levels-back
OUT	1	Number of outputs
TMUT	25%	Mutation rate percentage
FITFNS	@ssim @cmplx	Objective functions
GENFNS	Table 2	Gene functions

these functions should operate accordingly. The training set is composed of ten image pairs: one raw input image and a target image of ideal output for comparison. These images were extracted from rooms and hallways of real scenarios where author's VIP navigation system should be used.

Table 2. List of operators used in this approach

Fn.	Description
dil, ero	Morphological operator. *If two images, output one of them.
and, or	Logical operator. *Process inputs with the same dimension.
not	Logical operator. Complements inputs and outputs one of them.
nop	Bypass operator. Outputs first input if gene is odd, second otherwise.

**Specific cases. For all the other possible combinations, the functions operate as expected.*

This approach aims to optimize two conflicting objectives: low error X low complexity. Thus, the best individual should present a feasible trade-off between them, providing a low error to ensure quality and low complexity to maintain good speed as well as low power consumption. The first objective is evaluated using structural similarity error of target images x and images processed by the evolved filters y (Eq. 1). C_1 and C_2 are normalization constants[28].

$$SSIM_{x,y} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1)$$

In order to evaluate the complexity of the evolved filters, the author's approach sums the sizes of all the structuring elements used by an individual (Eq. 2), and considers that high complexity values represent more operations which demands additional time and power. The equation defines S as the total number of SEs used by an individual and SE_List is the set of the SEs used by the user.

$$Complexity = \sum_{i=1}^{S} count(SE\_List(i)) \quad (2)$$

Other execution parameters as NGEN, NPOP, TMUT, NR, NC, and LB were set up after empirical tests, which showed to the authors that training sessions with big populations (1000) and a few generations (100) yield better individuals due to the high diversity. The levels-back and number of columns were set up to 500, being the optimal values since higher values (i.e.: 750 and 1000) didn't prove better, as lower values resulted in worse results. It was observed a considerable increase in training times if levels-back was set too low compared to columns (i.e.: levels-back 5 for 500 columns), without any improvement to results. The mutation rate of 25% was the best value rate in its range during preliminary results.

Results and Discussion

To evaluate the results, the authors picked up two solutions (individuals) generated and optimized by the presented approach: the best quality solution overall, which often implies in very high complexity values and a feasible solution that solves unknown distance areas of depth images keeping an acceptable level of complexity. Table 3 presents the results of both solutions for each image used to evolve both solutions minimizing error (O_{err}) and complexity (O_{cpl}) as well as execution time in milliseconds (ET_{ms}).

Table 3. Results of best quality solution and feasible solution.

	Best Quality (BQ)			Feasible Choice (FC)		
Image	O_{err}	O_{cpl}	ET_{ms}	O_{err}	O_{cpl}	ET_{ms}
$Room_1$	0.0271		15.5	0.1069		5.3
$Hall_1$	0.0176		15.0	0.0346		4.8
$Room_2$	0.0900		15.8	0.0740		5.0
$Corridor_1$	0.0729		15.9	0.0605		4.9
$Hall_2$	0.0356	3200	15.1	0.0343	410	4.8
$Room_3$	0.0119		15.0	0.0305		4.8
$Kitchen_1$	0.0389		15.1	0.0656		4.9
$Hall_3$	0.0111		14.9	0.0741		5.1
$Office_1$	0.0723		15.5	0.0772		5.1
$Office_2$	0.0240		15.1	0.0477		4.9

Based on the results of Table 3 is possible to notice some major differences between the two filters. BQ filter had a huge complexity value as expected compared to FC filter, which presented 87.2% less complexity. This difference is reflected in the execution times of the filters, where FC was up to 3.24 times faster than BQ for the $Corridor_1$ training. As for error rates, BQ filter had great overall rates with the majority of the errors below 0.04 mark and FC filter surprisingly managed to beat BQ filter in three training errors ($Room_2$,

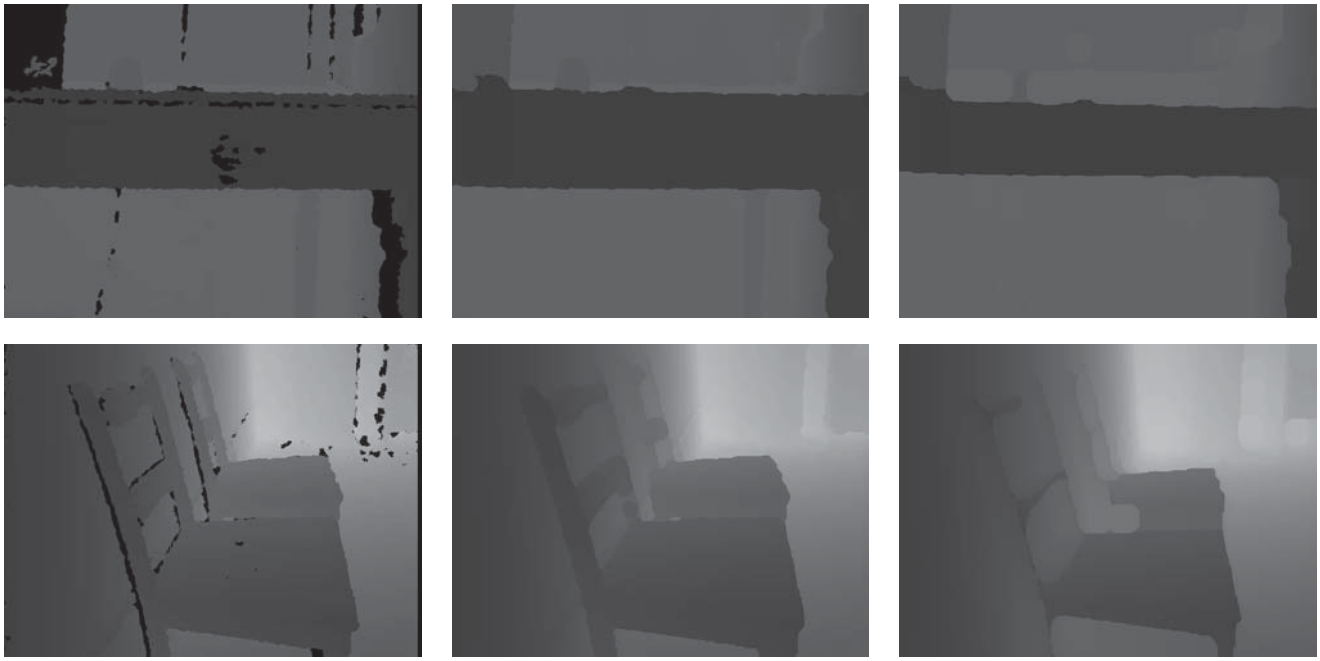

Fig. 4. Unused raw depth images (left) processed by BQ filter (center) and FC filter (right).

Corridor$_1$ and Hall$_2$). The authors will investigate which image features might be responsible for this behavior for further improvements. All the other seven images presented higher errors for FC as expected, but still being able to solve unknown areas in the depth images used. The FC filter is shown in Table 4.

Table 4. Feasible Choice filter.

	No.	Desc.	Dimension	
Inputs	**1**	Image	640x480	
	4	Diamond	2x2	
	6	Rectangle	3x1	
	12	Square	3x3	
	18	Line	3, 0°	
	22	Square	5x5	
	25	Rectangle	5x7	
	30	Line	5, 90°	
	34	Diamond	8x8	
	36	Rectangle	7x3	
	No.	**Operation**	**In1**	**In2**
Genes	**42**	ero	1	12
	44	nop	22	**6**
	45	nop	**42**	18
	46	nop	45	**4**
	47	dil	44	45
	49	nop	**25**	30
	51	nop	**36**	12
	54	dil	51	47
	60	nop	45	**49**
	83	dil	54	60
	90	dil	51	83
	93	dil	90	44
	143	nop	**46**	4
	236	dil	34	93
Output	**253**	dil	236	143

One might observe that no logical operators were used. In fact, even during preliminary tests **AND**, **OR** and **NOT** operators ended up left aside by the evolution and optimization process. Thus, for this specific scenario, logical operators didn't seem to have advantages in terms of quality or complexity.

Finally, Fig. 4 shows the visual results from both the BQ and FC filters compared to the raw inputs. These images are not part of the training set, and the evolved filters fulfilled their roles as expected. It can be noticed that the images processed by the FC filter have less quality compared to BQ's. However, the depth images didn't lose any important feature that could jeopardize the operation of a real-time embedded navigation system for visually impaired people, thereby the FC filter is a good feasible solution that guarantees high performance for its purposes.

Conclusion

This work presented an evolutionary multi-objective approach to automatically generate morphological filters of high performance to fix unknown distance areas in depth images. By using CGP along with the NSGAII multi-objective technique, the authors were able to develop a flexible approach to minimize simultaneously "structural similarity error x filter complexity" to optimize quality and performance. Most of the details of this approach were introduced as pseudo-code and the list of execution parameters/operators allows others to replicate this work.

Results were based upon the comparison of two generated filters after training with ten image pairs: one high quality filter with the best error rate (BQ) and one feasible filter (FC) that the authors used to accomplish both quality and speed requirements. It has been observed that the complexity gap between them showed up to 3.24x less time to apply the FC filter. Both the visual quality and the error rate differences didn't show enough evidences to overcome the FC speed gains considering real-time embedded navigation domain. Thus, the proposed approach managed to fix missing areas in depth images by generating feasible low complexity and low error filter. However, it is important to state that this approach might not be ideal for scenarios involving filtering to keep small details in depth images as shoes, door knobs, etc.

Currently, the authors are using this approach to develop morphological filters in an ongoing project for VIP navigation system. Further steps might include: to expand and propose new error formulas for improving the current results; to build a formal method for this approach with better applicability and to encourage others to contribute; and, ultimately, to develop a C++ library aiming at generating, presenting and exporting filters.

Acknowledgments

This study was financed by the Coordination for the Improvement of Higher Education Personnel (CAPES) – Finance Code 001, and by the São Paulo Research Foundation (FAPESP) – Project 2017/26421-3. We also thank the Federal Institute of São Paulo and the Para-D.V. nongovernmental organization, whose contribution made this work possible.

***Authors**: Prof. Antonio Miguel Batista Dourado, M. Sc. Federal Institute of São Paulo, Av. Zélia de Lima Rosa - 100, CEP: 18550-000, Boituva-SP, Brazil, email: dourado@ifsp.edu.br.*

Prof. Emerson Carlos Pedrino, Ph.D., Department of Computing, Federal University of São Carlos, Rod. Washington Luís - km 235, CEP: 13565-905, São Carlos-SP, Brazil, email: emerson@dc.ufscar.br.

REFERENCES

[1] Kazimierz KRZYWICKI. SoC Research and Development Platform for Distributed Embedded Systems. *Przegląd Elektrotechniczny*, 1(10):264–267, 2016.

[2] Jean Liénard, Andre Vogs, Demetrios Gatziolis, and Nikolay Strigul. Embedded, real-time UAV control for improved, image-based 3D scene reconstruction. *Measurement*, 81:264–269, mar 2016.

[3] Przemysław PTAK. Embedded system to control lighting of the office workplace. *Przegląd Elektrotechniczny*, 1(11):78–81, 2018.

[4] Xiaocong Fan. *Real-Time Embedded Systems: Design Principles and Engineering Practices*. 2015.

[5] Alexander Barkalov, Larysa Titarenko, and Małgorzata Mazurkiewicz. *Foundations of Embedded Systems*, volume 1 of *Studies in Systems, Decision and Control*. Springer International Publishing, Cham, 2019.

[6] Antonio Miguel Batista Dourado and Emerson Carlos Pedrino. Embedded Navigation and Classification System for Assisting Visually Impaired People, 2018.

[7] Matteo Poggi and Stefano Mattoccia. A Wearable Mobility Aid for the Visually Impaired based on embedded 3D Vision and Deep Learning. In *First IEEE Workshop on ICT Solutions for eHealth (IEEE ICTS4eHealth 2016) in conjunction with the Twenty-First IEEE Symposium on Computers and Communications*, 2016.

[8] Diego López-de Ipiña, Tania Lorido, and Unai López. Indoor Navigation and Product Recognition for Blind People Assisted Shopping. In *Lecture Notes in Computer Science*, volume 6693 LNCS, pages 33–40. 2011.

[9] J A Hesch and S I Roumeliotis. Design and Analysis of a Portable Indoor Localization Aid for the Visually Impaired. *The International Journal of Robotics Research*, 29(11):1400–1415, jan 2010.

[10] Xiaochen Zhang, Bing Li, Samleo L. Joseph, Jizhong Xiao, Yi Sun, Yingli Tian, J. Pablo Munoz, and Chucai Yi. A SLAM Based Semantic Indoor Navigation System for Visually Impaired Users. *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pages 1458–1463, 2016.

[11] Ruxandra Tapu, Bogdan Mocanu, and Titus Zaharia. A computer vision-based perception system for visually impaired. *Multimedia Tools and Applications*, pages 1–37, may 2016.

[12] Limin Zeng. Non-visual 2D representation of obstacles. *ACM SIGACCESS Accessibility and Computing*, (102):49–54, jan 2012.

[13] Boris Schauerte, Manel Martinez, Angela Constantinescu, and Rainer Stiefelhagen. An assistive vision system for the blind that helps find lost things. *Computational Science and Its Applications–ICCSA 2007*, 7383 LNCS(Chapter 83):566–572, jan 2012.

[14] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M. Cortelazzo. TOF Cameras and Stereo Systems: Comparison and Data Fusion. In *TOF Range-Imaging Cameras*, volume 9783642275, pages 177–202. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[15] Stefano Mattoccia and Matteo Poggi. A passive RGBD sensor for accurate and real-time depth sensing self-contained into an FPGA. *Proceedings of the 9th International Conference on Distributed Smart Camera - ICDSC '15*, pages 146–151, 2015.

[16] Larisa Dunai, Beatriz Defez Garcia, Ismael Lengua, and Guillermo Peris-Fajarnes. 3D CMOS sensor based acoustic object detection and navigation system for blind people. In *IECON Proceedings (Industrial Electronics Conference)*, pages 4208–4215, jan 2012.

[17] Fei Qi, Junyu Han, Pengjin Wang, Guangming Shi, and Fu Li. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, 34(1):70–76, jan 2013.

[18] Emerson Carlos Pedrino and Valentin Obac Roda. Real-time morphological pipeline architecture using high-capacity programmable logical devices. *Journal of Electronic Imaging*, 16(2):023002, apr 2007.

[19] Julian F Miller and Peter Thomson. Cartesian Genetic Programming. In *Genetic Programming*, volume 1802, pages 121–132. Springer Berlin Heidelberg, Berlin, Heidelberg, jan 2000.

[20] P.C.D. Paris, E.C. Pedrino, and M.C. Nicoletti. Automatic learning of image filters using Cartesian genetic programming. *Integrated Computer-Aided Engineering*, 22(2):135–151, feb 2015.

[21] Emerson Carlos Pedrino, Jose Hiroki Saito, Edilson R R Kato, Orides Morandin, Luis Mariano Del Val Cura, Valentin Obac Roda, Mario L Tronco, and Roberto H Tsunaki. Automatic construction of image operators using a genetic programming approach. In *2011 11th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 636–641. IEEE, aug 2015.

[22] Hugo Hedberg, Fredrik Kristensen, and Viktor Owall. Low-complexity binary morphology architectures with flat rectangular structuring elements. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(8):2216–2225, 2008.

[23] Deb Kalyanmoy. *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, 1 edition, 2001.

[24] J.F. Miller and S.L. Smith. Redundancy and computational efficiency in Cartesian genetic programming. *IEEE Transactions on Evolutionary Computation*, 10(2):167–174, apr 2006.

[25] Simon Harding, Vincent Graziano, Jürgen Leitner, and Jürgen Schmidhuber. MT-CGP: Mixed Type Cartesian Genetic Programming. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12*, page 751, New York, New York, USA, 2012. ACM Press.

[26] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[27] Carlo R. Raquel and Prospero C. Naval. An effective use of crowding distance in multiobjective particle swarm optimization. *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, page 257, 2005.

[28] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, apr 2004.