Conference Paper

The Problem of Human-following for a Mobile Robot

A A Gorbenko

Department of Intelligent Systems and Robotics of the Regional Educational and Scientific Center of Intelligent Systems and Information Security, Institute of Natural Sciences and Mathematics, Ural Federal University, Lenin st., 51, Ekaterinburg, 620083, Russia

Abstract

The problem of human-following for mobile robotic systems have been extensively studied. There are a number of approaches for different types of robots and sensor systems. In particular, different equipment of the environment and sensor-based methods by using a special suit have been applied for solution of the problem of human-following for mobile robots. This paper proposes an algorithm for the problem of human-following in an unequipped indoor environment for a low-cost mobile robot with a single visual sensor. We consider the results of computational experiments. Also, we consider the results of robotic experiments for day and night navigation.

Corresponding Author:
A A Gorbenko
anna.gorbenko@urfu.ru

Received: 10 February 2018
Accepted: 14 April 2018
Published: 7 May 2018

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the RFYS Conference Committee.

1. Introduction

The problem of human-following for mobile robots have attracted considerable attention of researchers in recent years (see e.g. [1-3]). There are a number of different approaches to the problem. In particular, we can mention approaches that are based on the using of laser range scanners [4], sonar sensors [3], visual navigation systems [5], and infrared motion sensing systems [6]. Various systems of sensor integration were also investigated. For instance, an algorithm for a laser range scanner and a visual sensor has been presented in [7].

In recent decades, the major trend in the development of robotic systems was focused on the usage of robots in everyday life. In this case, the most important limitation of sustainable development in such direction is the cost of robotic systems. It is not surprising that researchers pay considerable attention to the development of low-cost robotic systems [8, 9], low-cost and limited-resource control systems [10, 11], and low-cost testbeds [12, 13].

It should be noted that different special equipment of the environment extensively used for solution of the problem of human-following for mobile robotic systems. In

particular, surveillance systems (see e.g. [14]) and sensor-based methods (see e.g. [3]) by using a special suit for motion capture frequently applied in environments, where humans and robots coexist [15]. In this paper, we consider an approach to solving the problem of human-following in an unequipped indoor environment for a low-cost mobile robot with a single visual sensor.

2. Navigation on the base of visual features

Navigation on the base of selection of visual features is extensively used in robotics. Traditionally, the approach is implemented in the following four stages.

- Selection of visual features.
- Calculation of the correlation between visual features.
- Random selection of pairs of visual features.
- Matching of images based on known common points.

Properly defined features allow us to overcome many different problems in visual navigation. There is a large number of different visual feature descriptors that extensively used in the context of robot navigation. Among the most used we can mention SIFT [16] and SURF [17]. It is well known that SIFT is generally too slow to compute for real-time algorithms [18]. However, SIFT usually performs best, SURF provides a good compromise between quality and speed [18].

Frequently, nearest neighbor cluster detection algorithms are used for the calculation of the correlation between visual features. In particular, OpenCV K-means clustering is extensively used for the calculation of the correlation [19].

Typical methods for selection of pairs of visual features include use of FLANN [20], RANSAC [21], and FLANN with a RANSAC post-processing step. However, K-means clustering with a RANSAC post-processing step is the most frequently used approach.

For matching of images based on known common points, we can use one of approaches that are based on affine or projective transformations (see e.g. [22]). Optical flow approaches are also extensively used for matching of images based on known common points. Typical optical flow methods include usage of OpenCV block optical flow, Horn - Schunck optical flow [23], and Lucas - Kanade optical flow [24]. In our case, we have used low-cost cameras with IR illumination that provide low quality images. For such low quality images, Horn - Schunck and Lucas - Kanade methods provide too noisy results that require computationally expensive filtering algorithms.

3. The proposed approach

For solution of the problem of human-following for a low-cost mobile robot, we propose the following approach.

- Selection of visual features.
- Convolutional neural network classification of visual features.
- Calculation of the correlation between visual features.
- Random selection of pairs of visual features.
- Combinatorial refinement of the coordinates of visual features.
- Matching of images based on known common points.

There are a number of different modifications of SURF that are used for the selection of visual features. In our approach, we have considered MDGHM-SURF [25].

It is well known that convolutional neural networks demonstrate high efficiency in the classification of images (see e.g. [26]). Convolutional neural networks employ a hierarchical topology of connections. Such networks are particularly effective in processing raw pixel data. For some image Im, let F(Im) be a set of all visual features that have been extracted from Im. Let H(Im) and E(Im) are sets of all human visual features and all visual features of the environment, respectively. We use a convolutional neural network algorithm to obtain the partition of the set F(Im) into subsets H(Im) and E(Im). Convolutional neural networks demonstrate the ability to fast learning and fast adaptation. We use untrained convolutional neural networks. It should be noted that the robot performs a desired movement task by discrete motions that satisfy movement constraints. When the robot completes each motion command, the camera captures two frames. Such pair of frames reflects only the human motion. We use such pairs to train convolutional neural networks.

A number of different modifications of the K Nearest Neighbors method has been proposed in recent years. Among others, we can mention such methods as kernel K Nearest Neighbors [27], weighted K Nearest Neighbors [28], and mutual K Nearest Neighbors [29]. However, there is no explicit advantage of one method over other. In our approach, we have considered four different methods. We use these methods on a competitive basis. In particular, we have considered OpenCV K-means clustering and the statistics-based nearest neighbor cluster detection algorithm [30]. Also, we have used methods [31, 32] that were optimized based on the approach [33].

In practice, RANSAC not always allows us to achieve high-quality matching even for standard test images. In many cases, we can obtain acceptable quality by repeated application of RANSAC. However, for on-line stream processing of images of poor quality, the non-random behavior of the pseudo-random number generators becomes statistically significant. It is quite natural, since the pseudo-random number generators were originally developed for other purposes and under different operating conditions. In the absence of a specialized pseudo-random number generator, we can try to solve the problem by using several generators. If one of the generators demonstrates the inability to detect a solution of acceptable quality, then instead of repeated restarts it can be replaced by another one. For random selection of pairs of visual features, we use a quite standard implementation of RANSAC with variable pseudo-random number generators. In particular, we have implemented the 32-bit Mersenne Twister for the prime number 2^{19937}-1 [34], AES as a pseudo-random number generator [35], MWC256 [36], 64-bit Xorshift [37], MLCG mod 2^{64} [38], Ran2 [39]. In our approach, the selection of the pseudo-random number generator and its settings is performed under the control of a genetic algorithm. It should be noted that we apply RANSAC separately for H(Im) and E(Im). Therefore, for the sets H(Im) and E(Im), at the same time, the RANSAC transformation can use different pseudo-random number generators.

A number of string processing models has been extensively studied in the context of solution of different robotic problems. In particular, for many robotic methods, images can be considered as strings of features (see e.g. [40]). For such methods, we can use different string matching algorithms to solve various robotic problems. In particular, the longest common subsequence algorithm has been applied to solve the mobile robot localization problem [41]. Also, the algorithm can be used to reduce uncertainty in feature tracking [42]. We can use the longest common subsequence of the state sequences for task generalization (see e.g. [43]). Some variants of the longest common subsequence have been used for the task-level learning from demonstration [44] and trajectory-based representation of human actions (see e.g. [45]).

Following [44], we consider the model of constrained longest common subsequence [46] to reduce uncertainty in feature tracking (see also [42]). Let A={a[1],a[2],...,a[m]} be a fixed alphabet. For any two given strings S and T over A, if the string T can be obtained from the string S by deleting some letters from S, then T is a subsequence of S. It is assumed that the order of the remaining letters of the string S should be preserved. The length of a string S is the number of letters in S. For any given string S, we denote by |S| the length of S. The constrained longest common subsequence

problem (C-LCS) for two strings and arbitrary number of constraints can be formulated as following [44].

C-LCS:

Instance: Two strings U and V over A, a set C={T[1],T[2],...,T[n]} of strings over A.

Task: Find a maximum length string T over A such that

- T is a common subsequence of U and V;
- T[i] is a subsequence of T, for all o<i<n+1.

The string T is called a constrained longest common subsequence of strings U and V and we denote it by C-LCS(U,V). In general case, the constrained longest common subsequence problem is NP-hard [47]. However, in our approach, we consider only some restriction of the problem that can be solved in polynomial time.

For any two given strings U and V over A, the classic longest common subsequence problem asks for a longest string T that is a subsequence of both U and V. The string T is called a longest common subsequence of strings U and V and we denote it by LCS(U,V). It is well known that the longest common subsequence problem can be solved in polynomial time.

We assume that A is the alphabet of visual features. We consider two consecutive images as strings U and V of features. The RANSAC transformation gives us a set

{{(x[1, 1],y[1, 1]),(x[2, 1],y[2, 1])},{(x[1, 2],y[1, 2]),(x[2, 2],y[2, 2])},

{(x[1, 3],y[1, 3]),(x[2, 3],y[2, 3])},{(x[1, 4],y[1, 4]),(x[2, 4],y[2, 4])}}

of pixel pairs. Any pixel (x[i,j],y[i,j]) from R defines some visual feature a[k[i,j]]. It is clear that

U=U[1]a[k[i[1],j[1]]]U[2]a[k[i[2],j[2]]]U[3]a[k[i[3],j[3]]]U[4]a[k[i[4],j[4]]]U[5]

for some strings U[1], U[2], U[3], U[4], U[5] over A. We assume that C={T[1]}, where

T[1]=a[k[i[1],j[1]]]a[k[i[2],j[2]]]a[k[i[3],j[3]]]a[k[i[4],j[4]]].

In this case, it is easy to see that we can find C-LCS(U,V) in polynomial time. In particular, computation of C-LCS(U,V) can be reduced by dynamic programming to computation of LCS(X,Y) for some substrings X and Y of U and V. Positions of features a[k[i[1],j[1]]], a[k[i[2],j[2]]], a[k[i[3],j[3]]], a[k[i[4],j[4]]] in C-LCS(U,V) give us a new set of pixel pairs.

We use the new set of pixel pairs for matching of images based on known common points. For matching of images based on known common points, we use a projective transformation.

4. Experimental setup

To provide an experimental comparison for the proposed approach, we have implemented two algorithms (CC). One of them is based on SURF, OpenCV K-means clustering with a RANSAC post-processing step, and a projective transformation (SR). Another algorithm is based on OpenCV block optical flow (OF).

For computational experiments we have used the Tsotsos Lab dataset (Person Following Robot Using Selected Ada-Boosting with a Stereo Camera, By Bao Xin Chen, Raghavender Sahdev and John K. Tsotsos, In the 14th Conference on Computer and Robot Vision, Edmonton, Alberta, May 16-19, 2017. `http://jtl.lassonde.yorku.ca/2017/02/person-following/`). The Tsotsos Lab dataset was specially designed for experimental studies of the problem of human-following for mobile robots. The dataset consists of four stereo image sequences for person following task by a mobile robot. Each sequence contains from 2096 to 3080 stereo images with ground truth. The dataset was generated by the Pioneer 3AT autonomous robot following a person. All images were captured by a Point Grey Bumblebee stereo camera with a resolution of 640 x 480 pixels. The camera is 88 centimeters above the ground level. The frame rate of the images captured was approximately 14.2 frames per second. In our computational experiments, we have used the left images of the Tsotsos Lab dataset stereo image sequences for the tests. We have used the right images and ground truth for the verification of the results of the experiments. From the Tsotsos Lab dataset we have extracted a number of trajectories with lengths from 10 to 20 meters.

We have used a Logicom Spy-C Tank robot to perform a robotic experiment. The Spy-C Tank robot uses Wi-Fi to connect to laptop Sony VAIO PCG-51111V. The Spy-C Tank robot is equipped with a camera with IR illumination. All images were captured with a resolution of 320 x 240 pixels. The frame rate of the images captured was 25 frames per second. The camera is 7 centimeters above the ground level.

To perform experiments, we have used a natural indoor environment. During the experiments it was assumed that the person is moving along a given trajectory. The length of the trajectory is 14 meters. We have used the Neato Robotics Neato XV-11 robot for the verification of the results of the experiments. The Neato XV-11 robot is equipped with laptop Sony VAIO SVS131A12V and a camera.

5. Experimental results

For robotic experiments, it is assumed that the robot has successfully passed the test, if the robot followed the man from start to finish. Let EP be the maximum error of the person positioning. Let ER be the maximum error of the robot positioning. For computational experiments, it is assumed that the algorithm has successfully passed the test, if EP did not exceed 20 centimeters, ER did not exceed 20 centimeters, and EP+ER did not exceed 30 centimeters. For robotic experiments, we assume that the robot should be located 20 centimeters behind the moving person. For computational experiments, we assume that the robot must maintain a distance. Such assumptions allow us to construct the correct trajectory of motion of the robot and to compute the redundancy of a real trajectory of the robot. In addition, if the robot could not pass the test, then we can use the correct trajectory to compute the part of the traversed path. To evaluate the results of the experiments, we have used the following parameters.

- Success rate of tests (S).
- The average length of the path from start to finish (L).
- The average error of the person positioning (E).
- The redundancy of the robot trajectory (R).

Selected experimental results are given in table 1.

Table 1: Experimental results for algorithms SR, OF, and OA.

	Tsotsos Lab SR	Spy-C Tank SR Day	Spy-C Tank SR Night	Tsotsos Lab OF	Spy-C Tank OF Day	Spy-C Tank OF Night	Tsotsos Lab OA	Spy-C Tank OA Day	Spy-C Tank OA Night
S	100%	88%	7%	93%	81%	59%	100%	100%	100%
L	100%	76%	13%	84%	69%	52%	100%	100%	100%
E	8.7cm	11.2cm	11.4cm	14.8cm	16.6cm	17.1cm	3.4cm	4.1cm	4.2cm
R	3.1%	3.3%	3.3%	5.2%	6.3%	6.5%	1.7%	2.2%	2.3%

The number and quality of extracted visual features for images obtained by the Spy-C Tank robot is significantly lower than the number and quality of extracted visual features for images from the Tsotsos Lab dataset. Furthermore, for images obtained by the Spy-C Tank robot, the number and quality of extracted visual features for night images is significantly lower than the number and quality of extracted visual features for day images. The quality of extracted visual features for night images is not enough

for positioning by the SR algorithm. The quality of the OF algorithm depends essentially on changes in lighting. The OF algorithm showed insufficient quality even for high-quality images of the Tsotsos Lab dataset. However, for sequences with a low level of light variation, the OF algorithm demonstrates better performance than the SR algorithm. Our approach has demonstrated good performance. However, for low quality images, the accuracy of the OA algorithm is significantly lower.

6. Conclusion

In this paper, we have considered the problem of human-following for unequipped indoor environments. We have presented an algorithm for the problem for a low-cost mobile robot with a single visual sensor. We have considered the results of computational experiments. Also, we have considered the results of robotic experiments for day and night navigation.

In comparison with other algorithms, our approach has demonstrated better performance. Moreover, unlike previous studies for a mobile robot with a single camera (see [5]), our experimental results showed that the problem of human-following can be solved with relatively high reliability for a mobile robot with a single camera. However, our approach has demonstrated a significant dependence of the results from the quality of images. This leaves a plenty room for further improvement of the method. Moreover, it is clear that current datasets are not sufficient for a thorough study of the problem of human-following and other problems of human-robot interaction such as human avoidance, interaction-aware navigation, etc.

Acknowledgments

This work is partially supported by the Ministry of Education and Science of the Russian Federation project “Combinatorial models in computer science and their applications”.

References

[1] Bakar M and Amran M 2015 Int. J. Comput. Appl. **125** 27.
[2] Misu K and Miura J 2016 Adv. Intell. Syst. Comput. **302** 705.
[3] Peng W, Wang J and Chen W 2017 Adv. Intell. Syst. Comput. **531** 301.
[4] Kawarazaki N, Kuwae L and Yoshidome T 2015 Procedia Comput. Sci. **76** 455.
[5] Kim J and Do Y 2012 Procedia Eng. **41** 911.

[6] Feng G, Guo X and Wang G 2012 Sens. Actuators A **185** 1.

[7] Alvarez-Santos V, Pardo X, Iglesias R, Canedo-Rodriguez A and Regueiro C 2012 Robot. Auton. Syst. **60** 1021.

[8] Boccanfuso L, Scarborough S, Abramson R, Hall A, Wright H and O'Kane J 2017 Auton. Robot. **41** 637.

[9] Rubenstein M, Ahler C, Hoff N, Cabrera A and Nagpal R 2014 Robot. Auton. Syst. **62** 966.

[10] Santos M, Santana L, Brandao A, Sarcinelli-Filho M and Carelli R 2017 Control Eng. Pract. **61** 93.

[11] Zaidner G and Shapiro A 2016 Biosyst. Eng. **146** 133.

[12] Sanchez-Lopez J, Fu C and Campoy P 2016 Adv. Intell. Syst. Comput. **417** 57.

[13] Tejado I, Serrano J, Perez E, Torres D and Vinagre B 2016 IFAC-PapersOnLine **49** 242.

[14] Jin T, Morioka K and Hashimoto H 2006 Artif. Life Robotics **10** 96.

[15] Wang L, Schmidt B and Nee A 2013 Manuf. Lett.**1** 5.

[16] Lowe D 2004 Int. J. Comput. Vis. **60** 91.

[17] Bay H, Ess A Tuytelaars T and Van Gool L 2008 Comput. Vis. Image Underst. **110** 346.

[18] Schmidt A, Kraft M, Fularz M and Domagala Z 2013 J. Autom. Mob. Robot. Intell. Syst. **7** 11.

[19] Arthur D and Vassilvitskii S 2007 *Proc. 18th Annual ACM-SIAM Symp. on Discrete Algorithms* (Philadelphia: Society for Industrial and Applied Mathematics) p 1027.

[20] Muja M and Lowe D 2009 *Proc. 4th Int. Conf. on Computer Vision Theory and Applications* vol 1 (Lisboa: SciTePRESS) p 331.

[21] Fischler M and Bolles R 1981 Commun. ACM **24** 381.

[22] Hartley R and Zisserman A 2004 *Multiple View Geometry in Computer Vision* (Cambridge: Cambridge University Press).

[23] Horn B and Schunck B 1981 Artif. Intell. **17** 185.

[24] Lucas B and Kanade T 1981 *Proc. 7th Int. Joint Conf. on Artificial Intelligence* (San Francisco: Morgan Kaufmann Publishers Inc.) p 674.

[25] Kang T, Choi I and Lim M 2015 Pattern Recogn. **48** 670.

[26] Krizhevsky A, Sutskever I and Hinton G 2012 *Proc. 25th Int. Conf. on Neural Information Processing Systems* (Lake Tahoe: Curran Associates Inc.) p 1097.

[27] Yu K, Ji L and Zhang X 2002 Neural Process. Lett.**15** 147.

[28] Chernoff K and Nielsen M 2010 *Proc. IEEE 20th Int. Conf. on Pattern Recognition* (Hoboken: Wiley-IEEE Press) p 666.

[29] Liu H and Zhang S 2012 J. Syst. Softw. **85** 1067.
[30] Ritter G, Nieves-Vazquez J and Urcid G 2015 Pattern Recogn. **48** 918.
[31] Ertugrul O and Tagluk M 2017 Appl. Soft Comput. **55** 480.
[32] Zhang X, Li Y, Kotagiri R, Wu L, Tari Z and Cheriet M 2017 Pattern Recogn. **62** 33.
[33] Luo X, Xia Y, Zhu Q and Li Y 2013 Knowl.-Based Syst. **53** 90.
[34] Matsumoto M and Nishimura T 1998 ACM T. Model. Comput. S. **8** 3.
[35] Buchmann J 2004 *Introduction to Cryptography* (New York: Springer).
[36] Couture R and L'Ecuyer P 1997 Math. Comput. **66** 591.
[37] Marsaglia G 2003 J. Stat. Softw. **8** 1.
[38] L'Ecuyer P 1999 Math. Comput. **68** 249.
[39] Press W, Teukolsky S, Vetterling W and Flannery B 2007 *Numerical Recipes: The Art of Scientific Computing* (New York: Cambridge University Press).
[40] Lamon P, Nourbakhsh I, Jensen B and Siegwart R 2001 *Proc. IEEE Int. Conf. on Robotics and Automation* (Piscataway: IEEE Press) p 1609.
[41] Gonzalez-Buesa C and Campos J 2004 *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Piscataway: IEEE Press) p 2475.
[42] Argyros A, Bekris K, Orphanoudakis S and Kavraki L 2005 Auton. Robot. **19** 7.
[43] Nicolescu M and Mataric M 2003 *Proc. 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems* (New York: ACM) p 241.
[44] Gorbenko A 2013 IAENG IJCS **40** 266.
[45] Oikonomopoulos A, Patras I, Pantic M and Paragios N 2007 *Proc. ICMI/ IJCAI Int. Conf. on Artifical Intelligence for Human Computing* (Berlin: Springer) p 133.
[46] Tsai Y 2003 Inform. Process. Lett. **88** 173.
[47] Gotthilf Z, Hermelin D and Lewenstein M 2008 *Combinatorial Pattern Matching* (Berlin: Springer) p 255.