

Making Industrial Robots Smarter with Adaptive Reasoning and Autonomous Thinking for Real-Time Tasks in Dynamic Environments: A Case Study

Jaime Zabalza¹, Zixiang Fei², Cuebong Wong², Yijun Yan¹, Carmelo Mineo¹, Erfu Yang^{2*}, Tony Rodden³, Jorn Mehnert², Quang-Cuong Pham⁴ and Jinchang Ren¹

¹ Department of Electronic and Electrical Engineering,

² Department of Design, Manufacture and Engineering Management,

³ Advanced Forming Research Centre,

University of Strathclyde, Glasgow, G1 1XJ, UK

⁴ School of Mechanical and Aerospace Engineering,

Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

* erfuyang@strath.ac.uk

Abstract. In order to extend the abilities of current robots in industrial applications towards more autonomous and flexible manufacturing, this work presents an integrated system comprising real-time sensing, path-planning and control of industrial robots to provide them with adaptive reasoning, autonomous thinking and environment interaction under dynamic and challenging conditions. The developed system consists of an intelligent motion planner for a 6 degrees-of-freedom robotic manipulator, which performs pick-and-place tasks according to an optimized path computed in real-time while avoiding a moving obstacle in the workspace. This moving obstacle is tracked by a sensing strategy based on machine vision, working on the HSV space for color detection in order to deal with changing conditions including non-uniform background, lighting reflections and shadows projection. The proposed machine vision is implemented by an off-board scheme with two low-cost cameras, where the second camera is aimed at solving the problem of vision obstruction when the robot invades the field of view of the main sensor. Real-time performance of the overall system has been experimentally tested, using a KUKA KR90 R3100 robot.

Keywords: Machine Vision, Path Planning, Robot Control, Adaptive Reasoning, Dynamic Environment.

1 Introduction

The integration of robotic systems in human life has become a reality thanks to the major advancements in the field of robotics [1]. Comprising of many applications [2-3], smart robotic systems are not only able to perform faster and more accurately but are also to learn, adapt to the environment and make decisions in several degrees of intelligence [4], achieving collaborative systems able to interact with humans [5]. Most real-world systems are based on predefined series of tasks planned offline [6-7], and

have the advantage of being highly effective for traditional mass-production processes based on repetition. However, predefined planning lacks the capability for adapting to changes, which becomes a huge shortcoming in industrial systems. Hence, some barriers need to be removed before robots and humans can share the same workspace [8].

In order to provide robotic systems with autonomous adaptability and smart interaction with the environment [9], new capabilities linked to sensory attributes are required. Although there is a number of potential sensing approaches, including ultrasonic [8] or laser, machine vision [9] is widely used not only in robotics [10] but also in many other applications [11-12], due to its satisfactory performance and/or affordable cost. Therefore, the use of cameras, in conjunction with effective image processing, is a feasible approach to dealing with non-uniform lighting conditions and shadows [13] and other problem requirements such as adaptability and robustness [14]. In the same terms, multiple cameras can be used to overcome problems such as vision obstruction [15-17].

In this work, a real-time approach for smart path planning [18] and control of robotic arms is developed. The integration of machine vision provides the whole system with sensory capabilities. Although it is possible to find some research literature evaluating real-time path planning or related adaptability in robot manipulators [7,18-19], such previous works possess a number of drawbacks including the use of expensive sensors, non-integrated implementations and lack of modularity. In this work, the KUKA QUANTEC KR90 R3100 [20] is selected as a platform for testing an intelligent system, able to adapt to a dynamically changing environment. This is achieved through a modular machine vision software platform, based on low-cost cameras running in parallel.

The robot performs pick-and-place tasks in a challenging scenario, where a moving obstacle can intercept any planned path between the start (pick) and goal (place) locations. Therefore, collision with the moving obstacle has to be avoided, adapting the robot path to the workspace conditions in real-time. With a biological inspired integration of vision, reasoned planning and motion control, including cognitive structures that can be identified as offline training and calibration, the main objective of this work is to achieve an adaptive intelligent system with environmental awareness. Whenever a potential collision is detected, the robot has to divert towards an alternative effective collision-free path, in a prompt fashion. The aim is to have a better reaction time than the average human reaction time, whose average is equal to 180 milliseconds [18].

This manuscript is organized as follows. Section 2 gives a brief overview of the path planning and robot control implementation. Section 3 presents and develops the sensing strategy adopted to provide the whole system with sensory capabilities. The experimental setup for a physical demonstrator is detailed in Section 4, while Section 5 evaluates the achieved performance. Final remarks are drawn in Section 6.

2 Path Planning and Robot Control

2.1 Overview

Pick and place tasks are traditionally automated through offline programming, by defining predefined paths between the two points. While this approach is suitable for

structured and static environments, this study addresses dynamic, changing environments, where an obstacle, moving within the workspace, can invalidate the currently active robot path at any point in time. Thus an additional requirement of the path planning module is to find a feasible (collision-free) path in real-time, such that the robot trajectory can be updated quickly enough, when an interfering obstacle is detected.

This real-time path planning problem is solved using the dynamic roadmaps method, which is a real-time variant of the configuration space (C-space) sampling-based probabilistic road maps technique [21]. The dynamic roadmap method has been proven effective for motion planning in changing environments due to its high-speed online computation. This speed is achieved through an offline pre-processing phase, which is required to be computed only once for any particular robot.

After the path planning procedure, the robot is controlled through a novel approach based on a KUKA software add-on, known as Robot-Sensor Interface (RSI). This allows sending command positional packets at a rate of 250 Hz. With this method, motion paths can be updated rapidly in response to changes in the environment, as determined by the machine vision module. The path planning stages are explained below.

2.2 Offline Stage

During offline pre-processing, mapping between a C-space roadmap and a discretized Cartesian (geometric) workspace is created such that an entirely feasible roadmap can be quickly obtained when the geometric information about the obstacles in the environment is known. The offline pre-processing stage comprises: (i) robot C-space sampling, (ii) workspace discretization and (iii) mapping between C-space and Cartesian space.

The C-space sampling involves randomly sampling the entire C-space to obtain nodes of the roadmap. This is performed assuming a completely obstacle-free space. Then, pairs of neighbouring nodes are connected to form the edges of the roadmap. Next, the workspace is discretized into uniform cells. The spatial resolution available is obviously dependent on this size, with a subsequent trade-off between finer resolution and faster computation. Increasing the number of cells increases the computation time of the third stage exponentially. Finally, the mapping between C-Space and workspace is obtained by iteratively checking every robot configuration associated with all sampled nodes and along each edge of the roadmap. All workspace cells that collide with the robot at these configurations are mapped to the associated nodes and edges. Hence during online execution, when the cells that are pre-occupied by an obstacle is known, the roadmap can be “trimmed” to leave only feasible segments.

2.3 Online Stage

In real-time operation, the algorithm combines the vision information about the position of a moving obstacle and the offline-computed mapping to create a graph representation of the collision-free regions of the C-space. The desired start and goal configuration (which may differ in each instance of path planning) is connected to the nearest node in the roadmap. Then, the A* search algorithm [22], which is an extension of the Dijkstra’s algorithm, is used to search for the shortest route within the graph in order to find

a path between start and goal that guarantees no collision with the moving obstacle. Afterwards, B-splines [23] are used to smooth the obtained path and provide continuous smooth motion from any start to goal configuration. A diagram is given in Fig. 1.

As the robot executes the initially planned path, the path planner continues to monitor the obstacles in the environment. When a change results in a collision with the previously planned path, a new updated path is planned using the same approach described above. This is made possible by the low computation time (< 50 milliseconds) achieved using the dynamic roadmaps approach.

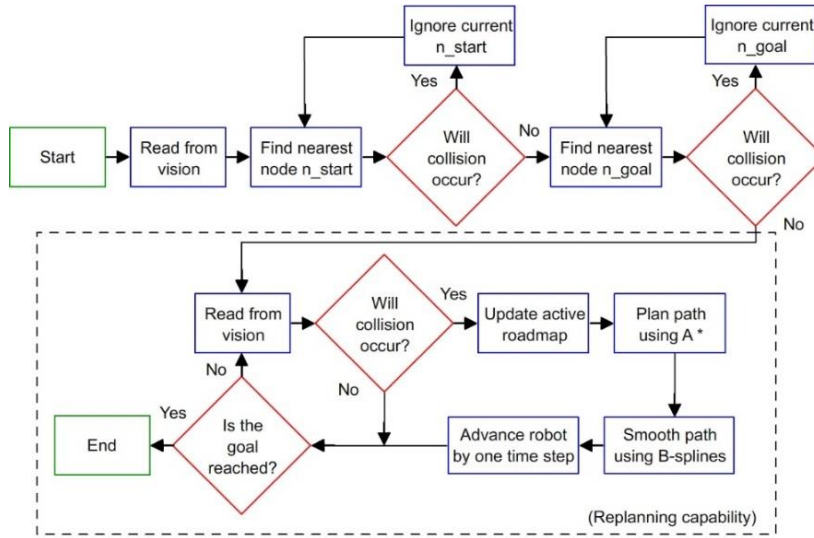


Fig. 1. Flowchart for dynamic path planning. Dashed box indicates replanning capability.

3 Machine Vision

3.1 Overview

In order to provide the robotic system with vision capabilities, a sensing strategy based on machine vision was developed. It uses optical cameras and runs independently but simultaneously to the rest of the system, comprising path planning and robot control.

The machine vision code architecture is made up of three independent stages, which also run in parallel by the implementation of three threads: (i) frames acquisition, (ii) image processing and (iii) communication. First, the acquisition thread manages the use of cameras for video stream acquisition. Second, the obtained frames are treated by the image processing stage, which performs object detection based on color and additional filtering. Third, the communication thread stores the detection information in a package, to be sent directly to the path planner by TCP/IP communication sockets [24]. This enables the main control module and the machine vision module to communicate with each other in real-time. The computation time for acquisition, processing and communication do not cumulate.

3.2 Camera Acquisition

The implemented machine vision strategy uses low-cost webcams. These cameras are not mounted on the robot as in many other applications [25], but placed at external fixed locations. In other words, they are off-board cameras. Fixed off-board cameras facilitate the computation of spatial coordinates, at the cost of having robot intrusion into the cameras' field of view. The robot can hide the targeted object while moving. In order to solve this intrusion, two cameras are used, denoted as Cam-1 and Cam-2.

Cam-1, the main camera, is placed overhead to get a top view of the whole robot workspace. The second camera, Cam-2, is introduced as an auxiliary sensor, to make sure the targeted object can be detected at all times and solve the robot intrusion issue into the field of view of Cam-1. The disposition of the two cameras is discussed in Section 4.

An important consideration related to the machine vision is how to solve the sensor fusion problem, as two different cameras are used. The proposed strategy addresses sensor fusion at the output level: the algorithm prioritizes the main camera (Cam-1) and, as long as this camera is able to detect the moving obstacle, its information is used for path planning. In the case that the main camera is not able to detect the object, then the machine vision opts to use the auxiliary camera (Cam-2), using its information. A related diagram is shown in Fig. 2.

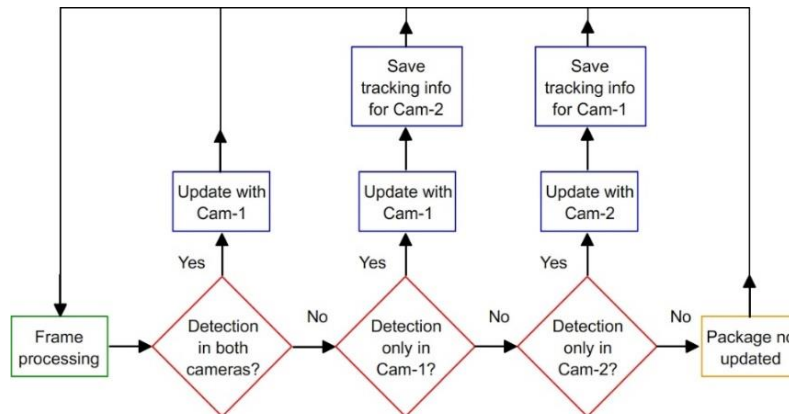


Fig. 2. Flowchart for sensor fusion in real-time.

3.3 Image Processing

Object detection is based on color discrimination, implemented in the hue-saturation-value (HSV) color space. This space, which differs from others such as the red-green-blue (RGB) [26-27], comprises parameters related not only to the true color targeted (hue), but also the color depth (saturation) and color darkness (value). A good selection of allowed parameter ranges in the HSV color space can deal with reflections, shadows and other related issues, achieving satisfactory object detection performance.

In a similar way to the path planning implementation, there are two stages for the image processing: (i) offline calibration and (ii) online workflow.

Offline implementation involves some calibration procedures. First, a contour mask is defined for masking the frames. Second, some calibration points are measured in the workspace and translated into pixel coordinates in the images. These calibration points are used for training a projection algorithm, which will translate coordinate pixels to 2D real-world coordinates. Finally, the selection of range values to target in the HSV color space make up the last part of the calibration procedures.

The online workflow involves the following steps: (i) transformation from RGB image to HSV image, (ii) transformation from HSV image to binary image, by means of applying a selected HSV color range, and (iii) posterior treatment of the binary image, including size and tracking filtering, avoiding the detection of undesired objects.

3.4 Communication

While running in parallel, the machine vision can communicate in real-time with the main control software by means of TCP/IP sockets communication [24]. In the TCP/IP configuration, the machine vision software acts as a server (slave), while the main control software acts as a client (master). Therefore, the machine vision waits for any request and replies immediately with the latest information available. In order to do so, an information package is defined, which is made up of several bytes containing the coordinates (x, y) and approximated dimensions (bounding box) of the detected object.

4 Experimental Setup

4.1 Robotic Manipulator

The implementation of the integrated system is based on the KR90 R3100 from KUKA [20], a robotic arm usually found in many factories. This robot is a 6-axis jointed-arm manipulator, with volume of 66 m³ and 50 kg weight. In order to control this robot, dynamic link libraries (DLLs) were developed in C++ language.

The KR90 R3100 has been selected for experiments because of its wide use in industry, however, the implemented approach is easily transferable to other robots and experimental conditions.

4.2 Workspace and Moving Obstacle

In the experiments, a table with dimensions 160x110 cm was selected as the robot workspace (Fig. 3). The table is covered by a recycled tablecloth containing different textile traces with the purpose of simulating a dirty, noisy environment.

The moving obstacle is implemented by a remote controlled car (20x10 cm) covered by cardboard that is yellow in colour (intending to reproduce a similar safety jacket tonality). Being remotely controlled, the car can be driven by an operator behind safety barriers, complying with all safety regulations.



Fig. 3. Workspace table (left) with pick-and-place box (top) and moving obstacle (bottom).

4.3 Cameras and Location

Two Advent HD Pro (AWCAMHD15) were selected as sensors to prove that an inexpensive machine vision implementation is possible. These are low-cost cameras able to capture images with an original resolution of 640x480 pixels and frame rate of 30 fps.

The main camera, Cam-1, was placed in an overhead location approximately 3 m high over the ground, capturing images of the workspace table in landscape mode. The auxiliary Cam-2 was placed at a much lower height of 1.5 m, and provided a side view of the workspace in relation to Cam-1. The selected disposition for Cam-2 covered all blind points in Cam-1 when the robotic arm intruded its field of view.

4.4 Host Computer and Related Software

A laptop was used to run all software modules in parallel. The path planning module was implemented in Matlab. Robot control and machine vision were implemented in C++ language and called from Matlab through DLLs and executable files.

The laptop was based on a quad-core Intel i7 CPU with 16GB RAM, running the 64-bit Windows 10 operating system. Matlab R2016b and the OpenCV 3.1 library (for machine vision) were used. Two USB ports (one for each camera) and an Ethernet port for connecting to the robot were required.

5 Performance Evaluations

The performed experimental tests proved that the proposed machine vision works effectively and robustly, as it was able to continuously track the moving obstacle under changing conditions including reflections and shadows. Additionally, the intrusion of the robot into the field of view of Cam-1 was successfully solved by the introduction of the auxiliary Cam-2.

The precision (resolution) of the vision system has been measured to be approximately 0.3 cm per pixel, with an estimated positional accuracy of approximately ± 3 cm for the obstacle, which are acceptable figures for the performed tasks.

Thanks to the machine vision, the path planning module was able to detect potential collisions in the predefined paths between start and goal locations, allowing the robotic manipulator to trace alternative trajectories to avoid the moving obstacle in real-time during pick-and-place tasks. Fig. 4 shows a frame from a video acquired during demonstrations. In conclusion, the integrated system performed satisfactorily.

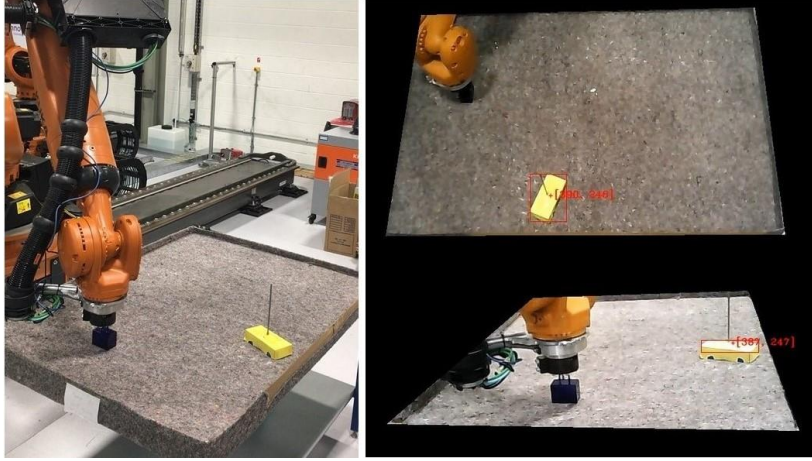


Fig. 4. A real video frame (left) with related captures by Cam-1 (top) and Cam-2 (bottom).

In relation to the computation time (Table 1), the main bottleneck lies in the image acquisition, due to the cameras' limited frame rate. However, the parallel execution of machine vision and path planning avoids accumulation of the computation times. Therefore, the overall time for the integrated system (around 60 milliseconds) is faster than the one in [18] (around 90 milliseconds) and much lower than that of the average human reaction time, estimated as 180 milliseconds.

Table 1. Estimated computation time for machine vision, path planning and overall system.

Machine vision		Path planning		Overall time (ms)
Stage	time (ms)	Stage	time (ms)	
Camera acquisition	45	Read package	12	Proposed ~ 60
Image processing	14	Check collision	15	Reference [18] ~ 90
Communication	< 1	Replanning	30	Human reaction ~ 180

6 Conclusions

In this work, a sensing strategy based on machine vision using low-cost optical cameras was developed for a robot manipulator system with real-time path planning and dynamic collision-avoidance capabilities. An integrated system was developed and tested with the KUKA KR90 R3100, a traditional industrial robot with 6 degrees of freedom.

The machine vision is based on two cameras placed off-board in fixed locations. The detection is performed by color in the HSV space to deal with a challenging environment. Computed in parallel with the main control software, they communicate in real-time for efficient pick-and-place planning tasks. The path planner was implemented using dynamic roadmaps, with the well-known A* algorithm for graph search and B-splines for smoothing, while robot motion control was achieved by means of DLLs.

The motivation of this work was to build a flexible and autonomous smart system with environmental awareness through a biological inspired integration of sensory attributes (machine vision), reasoning (path planning), coordinated movement (robot control) and cognitive learning (offline training and calibration). Performance in a physical demonstrator showed that intelligent behavior of robots for challenging environments in the manufacturing industry is possible, with a global system frequency well below 180 milliseconds, the estimated human reaction time.

Acknowledgements

This research is supported by the Advanced Forming Research Centre (AFRC) under its Route to Impact Funding Programme 2017-2018. We would also like to thank Dr Remi Christopher Zante, Dr Wenjuan Wang (AFRC, University of Strathclyde), Prof Stephen Gareth Pierce, Dr Gordon Dobie, and Dr Charles MacLeod (Centre for Ultrasonic Engineering, University of Strathclyde), and Dr Francisco Suarez-Ruiz (Nanyang Technological University, Singapore) for their contribution and support towards the completion of this project.

References

1. Zeng, Y., Zhao, Y., Bai, J., Bo, X.: Toward robot self-consciousness (II): Brain-inspired robot bodily self-model for self-recognition. *Cognitive Computation* 10(2), 307-320 (2018).
2. Liu, Y., Tian, Z., Liu, Y., Li, J., Fu, F., Bian, J.: Cognitive modeling for robotic assembly/maintenance task in space exploration. In: Baldwin C. (eds) *Advances in Intelligent Systems and Computing*, vol 586. Springer (2017).
3. Mineo, C., Pierce, S.G., Nicholson, P.I., Cooper, I.: Robotic path planning for non-destructive testing—A custom MATLAB toolbox approach. *Robotics and Computer-Integrated Manufacturing*, 37, 1-12 (2016).
4. Chella, A.: A robot architecture based on higher order perception loop. In: *Brain Inspired Cognitive Systems 2008*, Springer (2008).
5. Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraise, P.: Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing* 40, 1-13 (2016).
6. Ajwad, S.A., Ullah, M.I., Khelifa, B., Iqbal, J.: A comprehensive state-of-the-art on control of industrial articulated robots. *Journal of Balkan Tribological Association* 20(4), 499-521 (2014).
7. Lopez-Juarez, I.: Skill acquisition for industrial robots: From stand-alone to distributed learning. In: *The 2016 IEEE International Conference on Automatica*, pp. 1-5 (2016).

8. Anand, G., Rahul, E.S., Bhavani, R.R.: A sensor framework for human-robot collaboration in industrial robot work-cell. In: International Conference on Intelligent Computing, Instrumentation and Control Technologies, Kerala State, Kannur, India, pp. 715-720 (2017).
9. Perez, L., Rodriguez, I., Rodriguez, N., Usamentiaga, R., Garcia, D.F.: Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors* 16(3), 335 (2016).
10. Feng, Y., Ren, J., Jiang, J., Halvey, M., Jose, J.M.: Effective venue image retrieval using robust feature extraction and model constrained matching for mobile robot localization. *Machine Vision and Applications* 23, 1011-1027 (2012).
11. Wang, Z., Ren, J., Zhang, D., Sun, M., Jiang, J.: A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos. *Neurocomputing* 287, (2018).
12. Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J.: Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing* 53(6), 3325-3337 (2015).
13. Yan, Y., Ren, J., Zhao, H., Sun, G., Wang, Z., Zheng, J., Marshall, S., Soraghan, J.: Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos. *Cognitive Computation* 10, 94-104 (2018).
14. Liu, Q., Wang, Y., Minghao, Y., Ren, J., Li, R.: Decontaminate feature for tracking: adaptive tracking via evolutionary feature subset. *Journal of Electronic Imaging* 26(6) (2017).
15. Ren, J., Orwell, J., Jones, G.A., Xu, M.: Real-time modeling of 3-D soccer ball trajectories from multiple fixed cameras. *IEEE Transactions on Circuits and Systems for Video Technology* 18(3), 350-362 (2008).
16. Ren, J., Orwell, J., Jones, G.A., Xu, M.: Tracking the soccer ball using multiple fixed cameras. *Computer Vision and Image Understanding* 113(5), 633-642 (2009).
17. Ren, J., Xu, M., Orwell, J., Jones, G.A.: Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. *Machine Vision and Applications* 21, (2010).
18. Kunz, T., Reiser, U., Stilman, M., Verl, A.: Real-time path planning for a robot arm in changing environments. In: The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan (2010).
19. Galvao-Wall D., Economou, J., Goyder, H., Knowles, K., Silson, P., Lawrance, M.: Mobile robot arm trajectory generation for operation in confined environments. *Journal of Systems and Control Engineering* 229(3), 215-234 (2015).
20. KR QUANTEC extra HA Specifications (2013), <https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-quantec-extra>, last accessed 2018/05.
21. Leven, P., Hutchinson, S.: A framework for real-time path planning in changing environments. *International Journal of Robotics and Research* 21(12), 999-1030 (2002).
22. Cui, S.G., Wang, H., Yang, L.: A simulation study of A-star algorithm for robot path planning. In: 16th International Conference on Mechatronics Technology, pp. 506-510 (2012).
23. De Boor, C.: A practical guide to splines. *Applied Mathematical Sciences* (1978).
24. Donahoo, M.J., Calvert, K.L.: TCP/IP Sockets in C Practical Guide for Programmers (2009).
25. Mutlu, M., Melo, K., Vespignani, M., Bernardino, A., Ijspeert, A.J.: Where to place cameras on a snake robot: Focus on camera trajectory and motion blur. In: 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) pp. 1-8 (2015).
26. Abu, P.A., Fernandez, P.: Performance comparison of the Teknomo-Fernandez algorithm on the RGB and HSV color spaces. In: Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, Palawan, pp. 1-6 (2014).
27. Zhou, Y., Zeng, F.Z., Zhao, H., Murray, P., Ren, J.: Hierarchical visual perception and two-dimensional compressive sensing for effective content-based color image retrieval. *Cognitive Computation* 8, 877-889 (2016).