MOBILE ROBOT VISION SYSTEM FOR OBJECT COLOR TRACKING

Mladen Crneković, Zoran Kunica, Davor Zorc

Prof. dr.sc. Mladen Crneković, University of Zagreb, FSB, I. Lučića 5, 10002 Zagreb
Prof. dr.sc. Zoran Kunica, University of Zagreb, FSB, I. Lučića 5, 10002 Zagreb
Prof. dr.sc. Davor Zorc, University of Zagreb, FSB, I. Lučića 5, 10002 Zagreb

Keywords: mobile robot, vision, docking, color tracking

Abstract

By using a vision system mobile robot autonomy increases significantly. Instead of a robot motion programming, robot task programming is possible. In the paper a vision system of the eMIR mobile robot is presented that gives the robot ability to recognize objects by their color. For the object recognition HSV color system is used where a picture divides into cells with dimensions of 9x9 pixels. After determining the cell similarity with the reference sample, identified cells are grouped in clusters that are a basis for the object recognition. For each object, size and center of gravity are determined. The whole process of recognition is performed in a real time, with own developed algorithms for image processing. Recognition process is not carried out in the whole image, but only around the area where the object is detected in the previous recognition step. This allows fast search and recognition of not only static objects but also moving objects. The motion of the robot towards recognized object takes place in three stages: 1. robot's rotation, to allow bringing of the object being searched for in the robot's visual field, 2. aligning of the robot, by its rotation, until the found object comes into the center of the image that is in front of the robot, and 3. movement of the robot (simultaneous translation and rotation) to a distance of 10 cm.

1. MOTIVATION

Researches in the field of mobile robotics are carried out in at least two directions. First is the robot mobility, i.e. the possibility of motion in a specific field. The second research direction is to navigate a robot in an environment to independently reach the goal. To achieve the second job, researchers usually take simple design robot with three wheels, and its motion is limited to a bounded flat surface filled with obstacles. The task of the robot is to reach the goal and accomplish the task by using just simple motions. In achieving this goal robot should not collide with any obstacle. Additionally, it may be requested that the task execution time is shortest or is given by some other condition. Robot motions are not directly defined, but just ways of behavior that the robot selects from the currently understood situation. This means that the same task can be done with a variety of movements, and that the program itself (in this case, a set of rules of behavior) remains the same. To successfully execute the task, robot must be equipped with appropriate sensors. It is the vision system that gives the robot the most information about the environment, but also requires the most intensive information processing. In this paper a system with eMIR mobile robot [1] is described that has the camera and infrared rangefinders. The task of the robot is to find the target mark and move closer to the mark up to the given distance.

2. PREVIOUS WORK

One of the earlier works [2] is where the robot on the mobile platform must approach the table on which is an object (target) that robot should take. For objects identification two cameras are used, and the decision about the motion of the robot is left to the neural network. Although the authors write that the work is successful, they do not provide sufficient evidence for this.

The majority of papers on this topic were done when the wireless cameras become cheaper and personal computers fast enough to handle a real-time picture of VGA resolution. Tasks that are put to the robot movement are: coming to a default position, moving across a pre-programmed path, mapping the environment, or following a moving object (usually another robot).

In the paper [3] authors used a commercially available eROSI mobile robot with the aim to arrive at a target location. To achieve the goal, the robot used a camera, and the markers are placed so that they form four squares of known geometry as a basis for distance determination. The robot is performing the operation of approaching to default markers only, without search for them in the space.

In the paper [4] the camera is placed above the mobile robot's workspace, and the primary goal of the robot is to achieve accurately defined path. Because of the major disturbances on the camera, to determine the position of the mobile robot, Kalman filter was used. The application of such vision systems achieved good results in the straight ahead and circle robot movement.

Work [5] deals with positions determination of the robots that play soccer, and the position of the ball. Rules about colors determine FIRA associations. As robots move very quickly, priority was to determine the favorable number of frames per second (80 fps), and for colors definition the Color Filter Array (CFA) has been used. Due to the incomplete transfer of colors, it was necessary to incorporate color interpolator. To calculate the position and orientation of the robot, extended Kalman filter is used.

In the paper [6] the problem of the so-called *Docking Station* is solved by the vision system with three reference points, while the control algorithm is derived by using the *finite state machine* that has four states and six transitions. The default outputs are translational and rotational speed of the robot. Authors proved the solution convergence, and the experiment was made with the Pioneer 3AT mobile robot. The work is tightly based on the behavioral theory [7] that is well-elaborated in [8].

3. eMIR MOBILE ROBOT

To locate and access the given markers eMIR mobile robots are used (figure 1) as described in [1]. These robots are fully developed and manufactured at the Faculty of Mechanical Engineering and Naval Architecture University of Zagreb. The robot has well known differential structure, by size of 300x250 mm, equipped with a battery power source and managed with Atmel microcontroller.

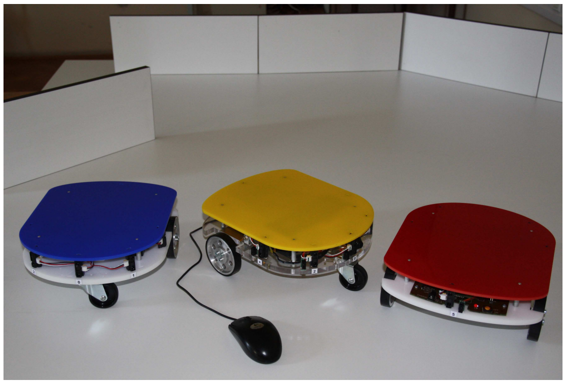


Figure 1. eMIR mobile robots

On the front side of the robot there is a camera that sends a color image of 720x576 resolution to a video receiver connected to a personal computer. Besides camera, the robot is equipped with six infrared range finders deployed on the robot in order to cover possible directions of motion of the robot (figure 2). Range finders measure distances up to 80 cm. In addition, battery voltage and the wheel speed are measured. All these data are sent to computer using the Bluetooth communication. After processing of the received data and image, the appropriate command is sent to the robot for its movement. Command for the robot motion has the following form:

# 01 vv vv CS /

where 01 is the code for the robot speed definition, vv sets translational (± 100 %), and rr sets rotational robot speed (± 100 %), while the CS is checksum that the recipient uses to verify correctness of the received packet.

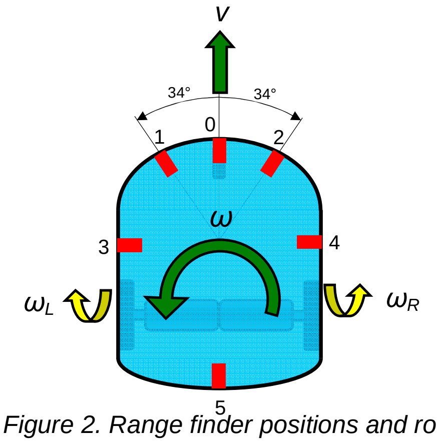


Figure 2. Range finder positions and robot kinematics

Motion commands are sent to the robot 4 to 10 times per second. When the robot receives a command for the motion, it passed it to the control system that solves the inverse kinematics. Nonlinear PI controller with feed forward action is used to achieve given robot speed. Robot kinematics equations are:

$$\begin{aligned} v &= D(\omega_R + \omega_L)/4 \\ \omega &= D(\omega_R - \omega_L)/2L \end{aligned} \qquad (1)$$

where *D* is the diameter of the wheels, and *L* is the axial spacing of the wheel.

If within one second the robot has not received a new order for the motion, it stops. If, within 120 seconds the robot has not received any correct command, the robot will shut off. Also, if the state of its battery is 10,5 V or less, and the motors are not moving, the robot will switch off (with a three beep) and thus protect its power. Information on measured distances, actual speeds and battery condition robot sends in the following format:

# aa bb cc dd ee ff uu vv vv mm CS /

where:

aa .. ff - measured distances of the infrared sensors in range of 0 to 80 cm,
uu - battery voltage and state of charge,
vv, rr - measured translational and rotational speed in %
mm - working mode,
CS - checksum of the received packets.

Packages of the robot state are sent 10 times per second.

The robot work space is determined by polygon with dimensions 4x2 m. Work space is in white color to help all signs and barriers stood out well. A polygon can be configured as required, using additional boards.

4. VISION SYSTEM AND IMAGE PROCESSING

Image processing was done in the Delphi programming language using VideoLab components [9]: VLDSCapture component is used for the images reception, VLGenericFilter to merge live image and results of processing, and VLDSVideoLogger for the motion recording. It is not necessary to process each incoming image. This strategy avoids computer overloading. Variable *Capture* (figure 3) determines the frame that will be processed; therefore it is possible to adjust processing to computer speed. Figure 3 shows processing, display and send commands to the robot.

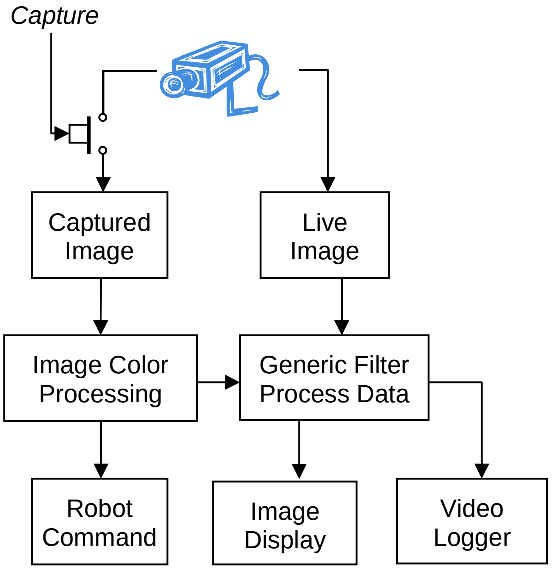


Figure 3. Signal flow and image processing

Image acquisition is done in multithread technique, so there is no interruption in the image acquisition during its processing. At the same time the data of the internal states of the sensors are accepted. The result of the captured image processing is the image (Image Display) that shows recognized parameters and corresponding command to the robot (Robot Command). Whole process as seen by the robot can be recorded in a video file compressed by adequate codec that is installed in the PC computer (Video Logger).

Although the image from the camera has a resolution of 720x576 pixels, image processing unit uses VGA format 640x480 with 32 bit resolution. As recognition by color is selected, the first color system that might be used is RGB color system, because the color components may simply be obtained from the video card. However, the RGB color system is suitable for color reproduction on the monitor, but is not suitable for identifying the color as seen by the human eye. A far better system to display colors is HSV (Hue, Saturation, Value) color system in which the first value (Hue) can to some extent be identified with color. In this paper, the transition from RGB to HSV color system used an algorithm in [10]. In doing so, the Hue parameter ranged from 0 to 360, Saturation from 0 to 100 %, and Value from 0 to 100 %, which avoids the floating point computation.

The color of an object may not been concluded on the basis of one pixel only. That is why the basic elements of the color recognition are not selected pixels but selected cell. The cell is a square of size 9x9 pixels in which a center is the selected pixel. In this way, the image is preformatted on 54x72 cells that is significantly smaller number of elements compared with the VGA image. For now, the cell size is constant, but it would be interesting to vary its size. Larger cells would provide somewhat less noise, but also rougher recognition with greater uncertainty in recognition of individual cells. Smaller cells reveal the finer details, but rise noise and increase processing time.

For each pixel in the cell HSV parameters are calculated, and then for the entire cell average values of *H**, *S** and *V** are calculated:

$$H^* = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9} H_{ij}$$

$$S^* = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9} S_{ij}$$

$$V^* = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9} V_{ij} \quad . \qquad (2)$$

As the value of H has not Euclidean properties (ranging from 0 to 360), for computing its average value, non-Euclidean procedure should be applied. Average values of HSV parameters would not give enough information for color identification. Additional necessary information is deviations from the average value as:

$$r_H = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9}\left|H^* - H_{ij}\right|$$
$$r_S = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9}\left|S^* - S_{ij}\right|$$
$$r_V = \frac{1}{9^2}\sum_{i=1}^{9}\sum_{j=1}^{9}\left|V^* - V_{ij}\right| \quad . \qquad (3)$$

If the deviation of the cell from the average value of any HSV parameter is too high, such cell is not suitable for recognition. This happens at the edge of the color transitions. Also, if the value of the cell *S* parameter is too low, it means that the color flux is insufficient for color identification.

If the value of *V* is too small, it means that there is not enough light for recognition.

Hue value is divided into segments, with meaning of assigned color (table 1). Each color has assigned center and range. If the color range is greater, then the color will be recognized with greater certainty. For the case $S < S_{min}$ a cell color is declared as white (no color), and for the case $V < V_{min}$ a cell color is black (no light). From table 1 it can be seen that the best chances to be recognized have colors red (purple), green, and blue color, while the yellow will be difficult to identify.

Table 1. Colors of the HSV system

Num	Hue	color	range	center
1	320 – 20	red	60	350
2	20 – 40	purple	20	30
3	40 – 70	yellow	30	55
4	70 – 170	green	100	120
5	170 – 190	turquoise	20	180
6	190 – 260	blue	70	225
7	260 – 320	purple	60	290
8	$S < S_{min}$	white	-	-
9	$V < V_{min}$	black	-	-

Recognition process begins by applying the reference sample. It is done by the mouse click on the color that wants to be recognized within the live image. Selected point becomes the center of the cell that will calculate HSV parameters. If the selected cell has sufficient flux dye ($S > S_{min}$), sufficient light ($V > V_{min}$), and dissipations are within the defined limits, according to table 1 color is assigned and this color becomes a reference color.

Search process starts with calculation of the HSV parameters and their deviations for each cell in the image. If all of the requirements for recognition (comparison) with the reference sample are fulfilled, the cell is declared as recognized, i.e. similar to the reference, otherwise it is not recognized.

After identifying the individual cells, recognized cells must be linked into groups (clusters). Algorithm for clustering starts with searching first recognized cell and put it in the cluster array. This cell also becomes *central* cell. By the eight directions searching around the central cell, cluster algorithm is looking for all those cells that are recognized and connected with the central one. When it finds a recognized cell, cluster algorithm put that cell in the cluster array (if it is not already there). When all eight directions are examined, the next central cell from cluster array is selected. When searching around central cell is no longer possible to find any new cell which corresponds to the sample, a cluster array with coordinates of corresponding cells is obtained. If the number of cells in the cluster array is less than the specified minimum, the cluster is rejected. To identify the cluster boundary, edges of the cluster and center of gravity are calculated. Then the algorithm proceeds to search for the next cluster.

When the algorithm searched the whole picture a certain number of clusters with a defined size and focus are obtained. It is necessary to identify the cluster that corresponds to the desired object that was clicked by the mouse. The first will be to identify the cluster that is closest to position where the reference sample is taken. After the cluster (or an object) has been identified, the following searches no longer have to count and recognize all of the cells in the image, but only those found in the neighborhood of the cluster recognized as object. This greatly speeds image processing and the object finding. For the case that an object unexpectedly disappears from the image, the scope of the search expands to the entire image. The scope of the search can be further improved by taking into account the motion of the robot (e.g., the robot turns to the right, then an object will move in the picture to the left).

5. ROBOT CONTROL

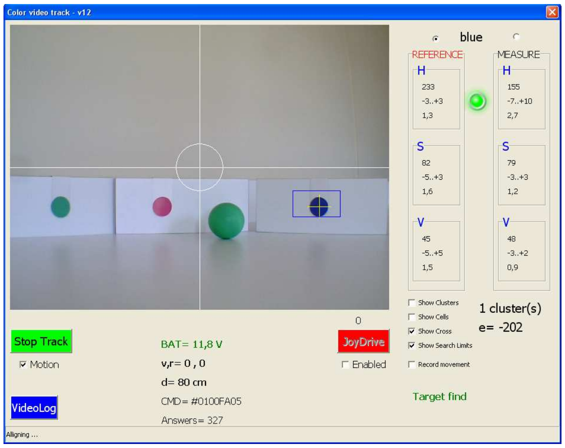
By defining the reference sample the robot task is to locate and move closer to designate object up to the given distance. For achieving this goal, there are three modes of the robot behavior:

1) Search is activated if the selected object is not in the robot visual field. Strategy for object searching is by robot rotation, so long as the object is not found in the visual field, or until the specified time has elapsed. Rotation speed should not be too high because it can happen that although requested object is observed, it can get out of the camera field of view because the robot has not reached stop.

2) Align is triggered when the selected object is in the visual field of the camera, and previously the search behavior was active. At this stage, the robot rotates until it finds the selected object close to the horizontal center of the visual field.
3) Approach is triggered when the selected object is found around the middle of the visual field. Approaching to the selected object begins with translation speed setting and correcting the direction of rotation. Front rangefinder provides information about the distance to the object, and when the distance is equal to or less than the default value, the robot stops.

If, for example, the approach behavior of the robot results with the removal of the selected object too far from the middle of the optical axis of the camera, the robot will return to align behavior. If during the approach behavior of the robot the selected object completely disappears from camera view, search behavior is activated and then again the align behavior. As the robot movement is partially nonlinear, robot velocities are restricted in order to avoid instability and twitch.

During each phase of the robot task execution it is possible to record robot kinematics; therefore the robot behavior can be properly quantified, not only visually monitored. In addition, video from the robot camera can be recorded with elements of recognition.



● blue ○

REFERENCE

H
233
-3..+3
1,3

S
82
-5..+3
1,6

V
45
-5..+5
1,5

MEASURE

H
155
-7..+10
2,7

S
79
-3..+3
1,2

V
48
-3..+2
0,9

□ Show Clusters
□ Show Cells
☑ Show Cross
☑ Show Search Limits
□ Record movement

1 cluster(s)
e= -202

0

☑ Motion

BAT= 11,8 V
v,r= 0 , 0
d= 80 cm
CMD= #0100FA05
Answers= 327

□ Enabled

Alligning ...

Figure 4. Interface to eMIR robot

All these functions are integrated into the program named ColorVideoTrack, whose interface is shown in the figure 4. Clicking on the live picture HSV parameters can be obtained, as well as borders and deviations of the selected cell. This cell can be a reference cell or just measured cell. Diode symbol will show whether the measured sample is similar to the reference (green) or is different (red), according to set recognition conditions.

The program also allows the operator to move the robot manually by joystick and it shows the basic feedback information from the robot: battery voltage, translational and rotational movement speed of the robot and measurement values of the front range finder. By activating of certain program options, it is possible to display some sub score processing results on the live image. The program also monitors the number of detected clusters as well as the recognized object position with regard to the middle axis of the camera.

6. EXAMPLE

Robot is set to recognize green marker as a reference and the task is to approach the marker to a distance of 20 cm (figure 5).

Hue value of the reference sample, based on which the sample was identified as green, was *H** = 163, with a deviation of 1,5. The maximum rotation speed of the robot is limited to 10%, and the translation speed to 20 %. In order for a sample to be recognized (declared as a similar to the reference parameter *H**), it should: not be different from the reference sample by more than 20, no deviation parameter can be greater than 10, the color flux must be greater than 30, and the light parameter *V* must exceed 20. With other parameters of the reference sample: *S** = 78, *V** = 51, *rH** = 1,5, *rS** = 2,7, *rV** = 0,9, it can be concluded that the selected sample is good and can be used for recognition. A valid cluster must have at least two cells. Kinematics of the robot movement is shown in the figure 6, and the video recording can be uploaded from the web page karmela.fsb.hr/eMIR. In the initial moment, the requested object was not in the camera's field of view − therefore the searching algorithm accomplished all three phases described in the section 5.

Towards the end of the robot motion it is noticed relatively large oscillations of the robot direction towards the target marker, in spite that the guidance controller gain value towards the goal are small, and the robot has its own kinematics controller. Unwanted jumps are caused by construction of the front robot wheel, mostly because of the influence of the dry friction during frequent changes in the direction of rotation of the robot, becoming the dominant reason for the twitch in the robot motion. In addition, approaching the selected object, the impact of turning direction becomes larger, so the approach should reduce the gain, i.e., slowing the process. But because the whole process stops at some distance from the goal, it can be concluded that despite the process almost entering the unstable area of control, the task is carried out successfully.

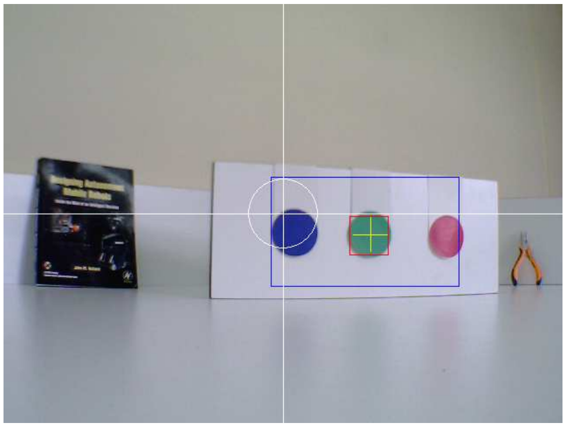


Figure 5. Green color as a reference

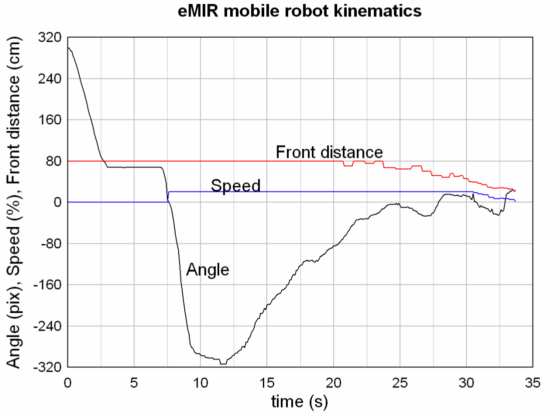


Figure 6. Robot kinematics during the task achievement

7. CONCLUSION

Described algorithm works well under the controlled environmental of the test site. Detection success mostly depends on the amount of light and light change in the space. More intensive light reduces noise and increases color saturation, therefore the recognition will be better.

If the searching algorithm finds two labels (clusters) of the same characteristics, the algorithm can make mistakes in identification. Introduction of additional recognition parameters (size, shape, etc.), would significantly increase the probability of correct recognition. Also, the robot motion strategies can be significantly improved by introducing additional forms of behavior (e.g., avoid obstacles, placed vertically on the obstacle, etc.). Additional forms of strategy would result in much more effective robots behavior (motion). Algorithms for image processing, robot movement strategy and communication with the robot should be structured in modules, and probably distribute to multiple programs linked by virtual communications.

If it is necessary to record processed images using compression, the entire task would be impossible for one computer, because of its limited resources. Therefore it would be necessary to distribute the entire processing on at least two computers – one for image processing and another for robot control.

8. ACKNOWLEDGMENT

The paper is an outcome of the scientific project “Modeling of mechanical behavior for assembly, packaging and disassembly”, supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

9. LITERATURE

[1] Crneković, M., Zorc, D., Kunica, Z., 2012, Research of Mobile Robot Behavior with eMIR, International Conference on Innovative Technologies, Rijeka, pp. 463-467

[2] Cooperstock, J.R., Milios, E.E., 1992, A neural Network Operated Vision-Guided Mobile Robot Arm for Docking and Reaching

[3] Min, H. J., Drenner, A., Papanikolopoulos, N., 2007, Autonomous Docking for an eROSI Robot Based on a Vision System with Point Clustering, *Proc. of the 2007 Mediterranean Conference on Control & Automation*, Athens, Greece

[4] Yang, J. L., Su, D. T., Shiao, Y. S., Chang, K. Y., 2010, Path-tracking controller design and implementation of a vision-based wheeled mobile robot, *Proc. of the IMechE*, Vol. 223, pp. 847-862.

[5] Klančar, G., Brezak, M., Matko, D., Petrović, I., 2005, Mobile Robots Tracking using Computer Vision, *Proc. of 13th International Conference on Electrical Drives and Power Electronics*, Dubrovnik, E05-20.pdf

[6] Amarashinge, D., Man, G.K., Gosine, R.G., 2005, Vision-Based Hybrid Control Strategy for Autonomous Docking of a Mobile Robot, 2005 IEEE Conference on Control Applications Toronto, Canada

[7] Arkin, R. C., 1998, *Behavior Based Robotics*. MIT Press, Cambridge, MA

[8] Braitenberg, V., 1986, Vehicles – Experiments in Synthetic Psychology, The MIT Press

[9] www.mitov.com, Accessed: 20120509

[10] Foley, J. D., van Dam, Feiner, Hughes, 1990, Computer Graphics: Principles and Practice in C, Addison-Wesley