Energy-Efficient Scheduling for Autonomous Mobile Robots

Jeff Brateman, *Changjiu Xian, and Yung-Hsiang Lu
School of Electrical and Computer Engineering, *Department of Computer Science
Purdue University, West Lafayette, IN 47907
brateman, cjx, yunglu@purdue.edu

***Abstract*— An autonomous robot has to sense its environment and prevent collision. Stereovision is a widely used technology for distance calculation and collision prevention. While a robot is moving, it has to detect an obstacle before a collision. This results in a real-time constraint and it may be adjusted by slowing down the motor. Since mobile robots usually carry limited energy, energy conservation is crucial. Stereovision requires substantial amounts of computation and causes much energy consumption.**

This paper presents a new approach for energy conservation for a mobile robot. We consider the energy consumed by both the on-board processor and the motor. Our method controls both the processor's frequency and the motor's speed to reduce energy while preventing collision. We formulate the problem as non-linear optimization and demonstrate that more energy can be saved by adjusting both the frequency and the speed simultaneously.

I. INTRODUCTION

Autonomous mobile robots provide great potential in transportation, entertainment, environment sensing, search, rescue, reconnaissance, hazard detection, and carpet cleaning [4], [5]. Mobile robots usually carry limited energy, such as rechargeable batteries, so energy conservation is crucial. Makimoto et al. [11] predicted that robots would be a major challenge for future low-power designs. A robot requires many different sensors to detect the environment. Among all sensing technology, stereovision is widely used for determining the distances of obstacles [8], [14], [20]. In a mobile robot, the processor and the motor are two major energy consumers [12]. Even though a display (including backlight) is a major energy consumer in a laptop computer [9], most robots do not carry such a display. In this paper, we consider a robot with only one motor, but the method can be generalized to multiple motors.

A robot is a real-time system. The processor has to determine the distance of an obstacle before the robot collides with the obstacle. The robot could stop during distance calculation, but to conserve energy the robot should be moving while performing this calculation. Many studies have been conducted on energy conservation for real-time systems [7], [15], [22], [24]. Existing studies assume that the deadlines are *externally* determined. For example, a video player has to provide 30 frames per second to prevent jitters. This 33 ms deadline for each frame is given by human's visual perception and cannot be changed by the video player. In contrast, in a mobile robot the deadline is not pre-determined for vision. If the obstacle is static, the robot can slow down or even stop to postpone the deadline before an impending collision. Hence, the deadline is determined by the interaction between the robot's processor and its motor. We believe this paper is the first study on energy conservation in a real-time system whose deadlines can be *internally* adjusted.

This paper presents a probabilistic approach for energy reduction in a mobile robot moving across an environment with static (i.e. not moving) obstacles. The robot uses stereovision to calculate the distance to each obstacle. We assume that each obstacle represents a pass/stop signal, and the minimum distance between signals is a known constant. The robot must recognize the actual distance to the signal before crossing the minimum distance to avoid collision. The computation cycles needed to calculate the distance follow a known probability distribution. Our method controls both the robot's processor frequency (and voltage) and the motor's speed to reduce the total energy consumption. Our method can save up to 15% additional energy when it is compared with existing solutions that adjust the frequencies only and use constant motor's speeds.

II. RELATED WORK

A. Probability-Based Voltage Scaling

Some studies have been conducted for dynamic voltage scaling (DVS) by considering the probability distributions of tasks' cycle demands. When the execution cycles of different task instances follow a known probabil-

3-901882-19-7 2006 IFIP

ity distribution, the processor can start at a low frequency (and voltage). If one instance requires fewer cycles, energy is saved because of the lower voltage. If the instance requires more cycles, the processor's frequency gradually rises to ensure that the instance can finish before the deadline. This approach is called *accelerating frequencies*. Lorch et al. [10] and Gruian [7] first use accelerating frequencies for intra-task scheduling. Xu et al. [22] consider bounded discrete frequencies. Yuan et al. [24] combine accelerating frequencies with soft real-time constraints for multimedia applications. Xian et al. [21] use accelerating frequencies for multiple tasks by integrating intra-task and inter-task voltage scheduling.

Suppose a task demands at most W cycles and the distribution of the cycles is expressed by the cumulative distribution function (CDF). The probability that the i^{th} cycle is needed is $P(i) = 1 - CDF(i-1)$. Note that P is non-increasing because CDF is non-decreasing. Since a task may demand millions of cycles, it is impractical to store the distribution in individual cycles. Thus, we partition $[0, W]$ into M bins and each bin contains b cycles ($b = \lceil \frac{W}{M} \rceil$). The CDF is then a function of the bins. The probability that the j^{th} bin is needed is $P(j) = 1 - CDF(j-1)$. The frequency assigned to the j^{th} bin is f_j and the execution time for this bin is $\frac{b}{f_j}$. The processor's power is proportional to $v^2 f$ and $v \propto f$ (here v is the voltage). The energy for this bin is $(v_j^2 f_j) \times \frac{b}{f_j} \propto b f_j^2$. The expected energy consumption for this bin is proportional to the product of the energy and the probability: $b f_j^2 P(j)$. Suppose the task is released at time zero and the deadline is t. The goal is to find a schedule $\{f_1, f_2, ..., f_M\}$ to minimize the total expected energy. This is formulated as follows.

$$\text{minimize} \sum_{1 \le j \le M} b f_j^2 P(j) \tag{1}$$

$$\text{subject to} \sum_{1 \le j \le M} \frac{b}{f_j} \le t \tag{2}$$

Based on earlier studies [10], [22], [24], the optimal solutions can be obtained by assigning f_j:

$$f_j = \frac{\sum_{i=1}^{M} b \sqrt[3]{P(i)}}{t \sqrt[3]{P(j)}} \tag{3}$$

B. Energy Conservation for Mobile Robots

Batteries are often used to provide power for mobile robots; however, they are heavy to carry and have limited energy capacity. A Honda humanoid robot can walk for only 30 minutes [1]. Rybski et al. [17] show that power consumption is one of the major issues in robot design. Sun et al. [19] present an algorithm for finding energy efficient paths. Yamasaki et al. [23] present an energy-efficient walk generation algorithm for a humanoid robot. A case study [12] shows that motor power is less than 50% of the total power in a mobile robot. Hence, the power for electronic components cannot be ignored. In recent years, small robots have been studied for sensing [2], [3], [5], [18]. Energy-efficiency is also a concern for these robots.

C. Contributions

Even though DVS and energy conservation for mobile robots have been studied, the close interaction between computation and motion remains unexplored. Specifically, we make the following contributions in this paper: (a) We consider a real-time system in which the deadline is determined by the interaction between two components: processor and motor. (b) We formulate an optimization problem to reduce the overall energy consumption. (c) We present a probabilistic solution to find the processor's frequency and the motor's speed.

III. Frequency and Speed Scheduling

This section presents our approach for energy conservation of a mobile robot by adjusting the robot's processor frequency and the motor's speed. We use a motivating example to illustrate the important concept and then formulate the problem as probabilistic non-linear optimization.

A. Motivating Example

As a simple example, suppose the total power of a robot's motor is $s^2 + s + 1$ at speed s meters per second. Here, the constant 1 is used to model the DC loss of the motor. The processor's power consumption is $f^3 + 1$ at frequency f MHz and a constant leakage power of 1. Suppose the robot has to travel along a road. The road contains signs indicating whether the robot can pass or has to stop. The signs do not change (unlike traffic lights) and the minimum distance between two adjacent signs is 100 meters. Even though the distance between signs may be larger than 100 meters, the robot must recognize the distance to the sign by the time it has traveled the minimum distance to prevent crossing a stop sign. If the robot does not to recognize the sign in time, the robot may collide with the sign and fail.

We define the optimal speed as the speed to consume the minimum energy per unit distance. Suppose the minimum distance between signs is d. The time to cross this distance is $\frac{d}{s}$. The total energy consumption

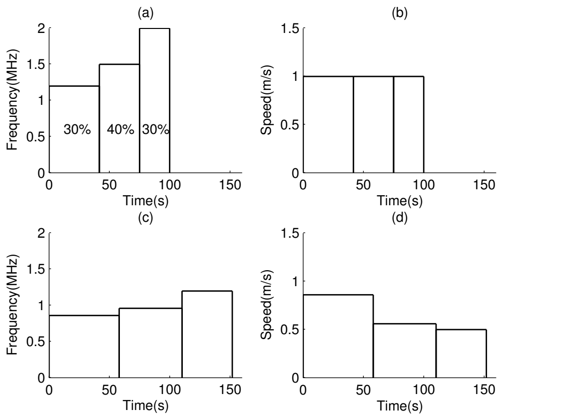

Fig. 1. Processor and motor scaling schedule assuming a constant motor speed in (a) and (b). Processor and motor scaling schedule if the motor speed is allowed to change in (c) and (d).

is $(s^2 + s + 1)\frac{d}{s}$ and the energy per unit distance is $\frac{s^2+s+1}{s} = s + 1 + \frac{1}{s}$. Thus, the optimal speed is 1 meter per second. If the robot moves at this speed, it takes 100 seconds to reach the next sign. The worst-case execution cycle is 150 million cycles so the processor has to operate at 1.5 MHz to ensure recognizing every sign in time. The total energy consumed by the motor at $s = 1$ is 3 × 100 = 300 J. The total energy consumed by the processor at $f = 1.5$ MHz is $(f^3+1)\frac{cycles}{f} = 4.38\times 100$ = 438 J. The overall energy is 738 J to cross the minimum distance between two signs.

If we consider the power of the motor and the processor simultaneously, we can reformulate the problem as follows. The time to cross the distance is $\frac{100}{s}$ at speed s. The processor has to operate at $1.5s$ MHz to meet the deadline. The total energy is $\frac{100}{s} \times \{(s^2 + s + 1) + [(1.5s)^3 + 1]\}$. The minimum energy occurs when $s \approx 0.62$ and the overall energy consumption is 614 J, or a 17% reduction from 738 J. This shows the importance of considering both frequency and speed simultaneously.

We consider a further extension of this example. The computation cycles vary due to the scene complexity surrounding the signs: Among all signs, 30% require only 50 million cycles, 40% for 100 million cycles, and the remaining 30% for 150 million cycles. The probability can be expressed in the following way. The first 50 million cycles are always needed so the probability is 100%. The second 50 million cycles are needed with probability 70%. Finally, the last 50 million cycles are needed only 30%. With this additional information, we can compute the *expected*, rather than the worst-case, energy consumption and still detect the sign in the worst case. If the motor's speed is a constant at 1 m/s, the deadline is 100 seconds. We can adopt the strategy with accelerating frequencies explained in Section II-A as shown in Figures 1 (a) and (b). The overall system saves energy in average cases because most tasks need only 50 or 100 million cycles. Meanwhile, the system still meets the deadline in the worst cases by using a higher frequency when needed. This requires expected energy consumption of 611 J, less than 1% reduction from 614 J. We can accelerate the processor's frequency and *simultaneously decelerate the motor's speed* and save more energy, as shown in Figures 1 (c) and (d). By decreasing the motor's speed, the processor's frequency does not have to rise significantly, and system's expected energy is reduced. This approach can further reduce the expected energy to 529 J, or 14%. The following sections will explain how to determine the frequency and the speed simultaneously to achieve better energy savings.

B. Problem Formulation

The minimum distance between two signs is a known constant, D. The maximum number of cycles needed for recognition is W and is divided into n bins. Each bin has $b = \lceil \frac{W}{n} \rceil$ cycles. We use $P(i)$ to represent the probability that the i^{th} $(1 \le i \le n)$ bin of cycles is needed. As defined in Section II-A, $P(i) = 1 - CDF(i-1)$ and $P(i) \ge P(i+1)$. The processor operates at frequency f_i for the i^{th} bin. When the processor is computing for the i^{th} bin, the robot moves at speed s_i. The execution time for the i^{th} bin is $\frac{b}{f_i}$. The distance traveled during this time is $d_i = s_i \frac{b}{f_i}$. The timing constraint is that the processor has to finish the computation of all bins before the robot crosses the distance of D. In other words, the sum of d_i cannot exceed D:

$$\sum_{i=1}^{n} d_i \le D \Rightarrow \sum_{i=1}^{n} \frac{bs_i}{f_i} \le D \qquad (4)$$

Let $\alpha(f_i)$ be the power consumption of the processor at frequency f_i when voltage scaling is also applied. When the processor finishes the task, the processor's frequency can be reduced to zero. In this case, the processor consumes static power $\alpha(0)$. Let $\beta(s_i)$ be the power consumption of the motor at speed s_i. The *expected* energy for crossing the distance is the sum of the processor's energy and the motor's energy over all bins. The energy consumed can be divided into two parts: (i) when the processor is still computing, and (ii) when all computation has finished.

When the i^{th} bin is being computed, the processor consumes power $\alpha(f_i)$ and the motor consume power

$\beta(s_i)$. The duration of this bin is $\frac{b}{f_i}$, and this occurs with probability $P(i)$. Therefore, the expected energy is

$$\sum_{i=1}^{n} \frac{P(i)b}{f_i}(\alpha(f_i) + \beta(s_i)) \tag{5}$$

To compute the energy in (ii), we first determine the distance the robot has traveled while the processor is computing. The total expected distance is $\sum_{i=1}^{n} \frac{bP(i)s_i}{f_i}$ and the remaining distance is $D - \sum_{i=1}^{n} \frac{bP(i)s_i}{f_i}$. When the robot is traveling through this remaining distance, the processor is turned off and consumes power $\alpha(0)$. Let s_o be the speed for the remaining distance. The time to cross this distance is $\frac{1}{s_o}(D - \sum_{i=1}^{n} \frac{bP(i)s_i}{f_i})$. Hence, the total expected energy is

$$\frac{1}{s_o}(D - \sum_{i=1}^{n} \frac{bP(i)s_i}{f_i})[\alpha(0) + \beta(s_o)] \tag{6}$$

The optimization problem is to find the values of f_i, s_i, and s_o for minimizing the sum of (5) and (6), with the constraint in (4).

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \tfrac{P(i)b}{f_i}(\alpha(f_i) + \beta(s_i)) + \\ & \tfrac{1}{s_o}(D - \sum_{i=1}^{n} \tfrac{bP(i)s_i}{f_i})[\alpha(0) + \beta(s_o)] \end{aligned} \tag{7}$$

C. Frequency and Speed Assignment

Our solution uses accelerating frequencies ($f_i \leq f_{i+1}$, $1 \leq i \leq n-1$) and decelerating speeds ($s_i \geq s_{i+1}$). To find the initial values for f_1 and s_1, we examine the schedulability of the problem using the constraint of inequality (4). The initial value of f_1 is the lowest frequency to satisfy (4) when all s_i's are assigned the lowest speed. Similarly, the initial value of s_1 is the highest speed to satisfy (4) when all f_i's are assigned the highest frequency. If f_1 exceeds the highest available frequency or s_1 is below the minimum available speed, no solution can be found. After finding the initial values for f_1 and s_1, we enumerate all feasible solutions and find the schedule that provides the minimum expected energy and meets the constraint in (4). Experimental data indicate that energy savings saturate when the number of bins (n) exceeds four. For a small value of n, it takes only several minutes on a modern computer to find the optimal schedule. Since the schedule can be calculated off-line, the time to find the optimal schedule is not a concern.

TABLE I
XSCALE'S FREQUENCY/VOLTAGE AND POWER.

Frequency(MHz)	150	400	600	800	1000
Voltage(V)	0.75	1.0	1.3	1.6	1.8
Power(mW)	80	170	400	900	1600

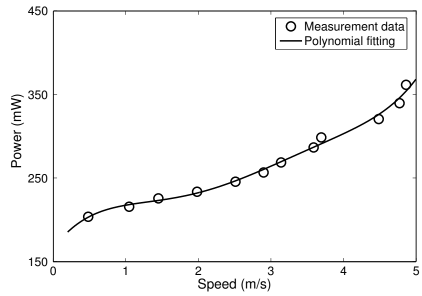

Fig. 2. Power efficiency of a robot at different speeds.

IV. EXPERIMENTS

A. Setup

We use the voltage/frequency settings of the Intel XScale processor [22] with five discrete frequency settings. Their associated power consumption is shown in Table I. The motor power is from the measurements performed by Mei et al. [13] shown in Figure 2. We limit the motor's speed between 0.5 m/s and 5 m/s with 0.5 m/s as the step size, and assume the minimum distance to travel is 500 meters.

Four benchmarks are used. The first three benchmarks are "synthetic" benchmarks with distributions of uniform, Gaussian, and exponential functions. These synthetic workloads have worst-case execution cycles (WCEC) of 100 billion cycles. For the uniform distribution, the actual number of needed cycles is between 0 and WCEC. For the Gaussian distribution, the mean is half WCEC and the standard deviation is a quarter WCEC. For the exponential distribution, the mean is a quarter WCEC. The distributions are normalized after removing the negative cycles and the cycles above WCEC.

The last benchmark is image correspondence used for stereovision [16]. Pairs of stereovision images are taken from the image database of the city of West Lafayette and Indianapolis in the state of Indiana [6]. Figure 3 shows the distribution of the needed cycles for running the correspondence programs on 700 pairs of images. Note that there is great potential for energy savings as the probability of the WCEC is only 0.14%.

We compare our approach with three other methods. The first uses a constant frequency and a constant speed.

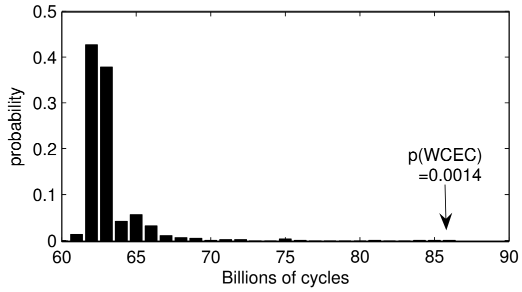

Fig. 3. Probability distribution of stereovision computations.

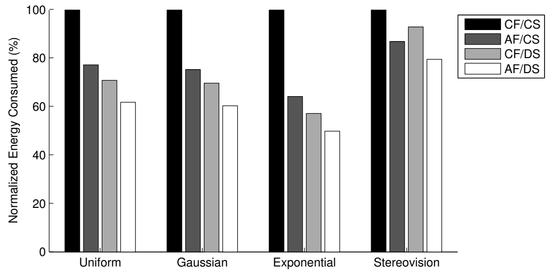

Fig. 4. Normalized energy consumption from the set of the three synthetic tasks with uniform, Gaussian, and exponential distributions, and the stereovision benchmark, respectively. The four methods use either constant frequencies (CF) or accelerating frequencies (AF), and constant speeds (CS) or decelerating speeds (DS).

The frequency and the speed are selected from the discrete settings such that they minimize the total energy consumption and satisfy the constraint. In the synthetic distributions, a search results in the optimal energy consumption solution which meets the constraint is to set the frequency to 400 MHz and the speed to 1.5 m/s. The second method uses a constant speed and accelerating frequencies. The third method uses a constant frequency and decelerating motor speeds. The processor frequency is set to the middle frequency setting of the processor, 600 MHz. The fourth uses both accelerating frequencies and decelerating speeds; this is the method proposed in this paper.

B. Experimental Results

Figure 4 shows the relative energy consumption of the four methods for the four benchmarks. All numbers are normalized related to the first method with a constant frequency and a constant speed. As can be seen in this figure, our method can save 20% to 50% energy compared with the first method in the four benchmarks. Compared with the second and the third methods, our method can save an additional 7% to 15% energy. These results are generated using 10 bins.

The exponential distributions show the greatest potential for savings as compared with the constant frequency and the constant motor speed schedule. In an exponential distribution, the task finishes quickly more often, and has a low probability of finishing near the WCEC. We can see the potential for reducing the expected energy as opposed to WCEC scheduling. In the stereovision distribution, the energy savings is not as high as the exponential distribution because no task finishes before $\frac{2}{3} * WCEC$ cycles in the 700 pairs of images. The algorithm requires certain processing steps to compute regardless of the complexity of the images. Our method still saves energy over the constant frequency and constant motor speed scheduling.

Figure 5 shows the energy consumption for a growing number of bins. This figure indicates that energy consumption decreases as the number of bins grows because more frequencies and speeds can be used. The figure also indicates that the energy consumption begins to stabilize as the number of bins exceeds four. In other words, a large number of bins may not provide a significant amount of additional savings. For the simulated distributions, the percent change from four bins to five bins drops below 2.5%, and continues to decrease as the number of bins increases. It should be noted that with the stereovision distribution, the energy actually increases in some cases. This is due to the division of the PDF into a relatively small number of bins. Some of the areas with high probability are divided in some sizes of n, resulting in an increased expected energy. However, energy is still reduced from the extreme case of one bin. Because of the small number of bins used to compute the frequency and the speed schedule, our method can be applied to practical systems. Even though our method may have exponential computation time, our method computes the schedule only once offline. Therefore, the overhead is only experienced once prior to runtime.

V. CONCLUSION

We present a method to simultaneously scale processor frequencies and motor speeds for autonomous robots with hard deadlines. We formulate the problem as an optimization problem, and we present a solution in this paper. A probability distribution of the number of cycles required for stereovision distance calculation is used for our simulations. Our experimental results show that we achieve energy savings from 7% to 15% more than only scaling processor frequency. These results can be achieved through the calculation of an optimal schedule off-line.

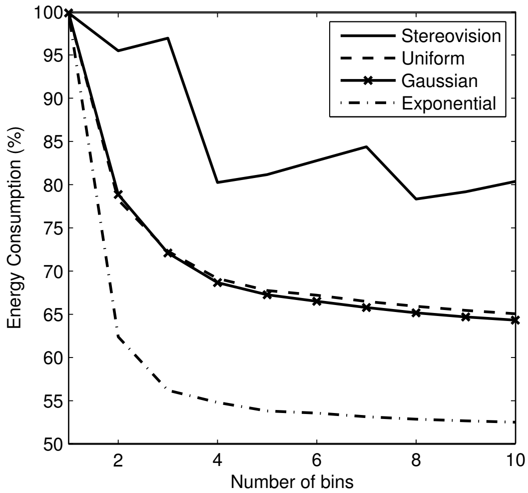

Fig. 5. Energy consumption of each benchmark over different number of bins.

VI. Acknowledgments

This work is supported in part by National Science Foundation IIS-0329061, CNS-0347466, and Purdue Research Foundation. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] R. Aylett. *Robots: Bringing Intelligent Machines To Life*. Barrons, 2002.

[2] M. C. Birch, R. D. Quinn, G. Hahm, S. M. Phillips, B. Drennan, A. Fife, H. Verma, and R. D. Beer. Design of a Cricket Microrobot. In *International Conference on Robotics and Automation*, pages 1109–1114, 2000.

[3] A. Colot, G. Caprari, and R. Siegwart. InsBot: Design of An Autonomous Mini Mobile Robot Able To Interact With Cockroaches. In *International Conference on Robotics and Automation*, pages 2418–2423, 2004.

[4] A. Davids. Urban Search and Rescue Robots: From Tragedy To Technology. *IEEE Intelligent Systems*, 17(2):81–83, March 2002.

[5] A. Drenner, I. Burt, T. Dahlin, B. Kratochvil, C. McMillen, B. Nelson, N. Papanikolopoulos, P. E. Rybski, K. Stubbs, D. Waletzko, and K. B. Yesin. Mobility Enhancements to the Scout Robot Platform. In *International Conference on Robotics and Automation*, pages 1069–1074, 2002.

[6] S. Gautam, G. Sarkis, E. Tjandranegara, E. Zelkowitz, Y.-H. Lu, and E. J. Delp. Multimedia for Mobile Users: Image Enhanced Navigation. In *Multimedia Content Analysis, Management, and Retrieval, IST/SPIE Symposium on Electronic Imaging*, 2006.

[7] F. Gruian. Hard Real-Time Scheduling for Low-Energy Using Stochastic Data and DVS Processors. In *International Symposium on Low Power Electronics and Design*, pages 46–51, 2001.

[8] R. K. Lenz and R. Y. Tsai. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology. *IEEE Transactions onPattern Analysis and Machine Intelligence*, 10(5):713–720, September 1988.

[9] J. R. Lorch and A. J. Smith. Apple Macintosh's Energy Consumption. *IEEE Micro*, 18(6):54–63, November 1998.

[10] J. R. Lorch and A. J. Smith. Improving Dynamic Voltage Scaling Algorithms with PACE. In *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 50–61, 2001.

[11] T. Makimoto and Y. Sakai. Evolution of Low Power Electronics and Its Future Applications. In *International Symposium on Low Power Electronics and Design*, pages 2–5, 2003.

[12] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee. A Case Study of Mobile Robot's Energy Consumption and Conservation Techniques. In *International Conference on Advanced Robotics*, pages 492–497, 2005.

[13] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee. Energy-Efficient Motion Planning for Mobile Robots. In *International Conference on Robotics and Automation*, pages 4344–4349, 2004.

[14] D. Murray and C. Jennings. Stereo Vision Based Mapping and Navigation for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1694–1699, 1997.

[15] P. Pillai and K. G. Shin. Real-time Dynamic Voltage Scaling for Low-power Embedded Operating Systems. In *ACM Symposium on Operating Systems Principles*, pages 89–102, 2001.

[16] A. Redert, E. Hendriks, and J. Biemond. Correspondence Estimation in Image Pairs. *IEEE Signal Processing Magazine*, 16(3):29–46, May 1999.

[17] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson, and M. D. Erickson. Enlisting Rangers and Scouts for Reconnaissance and Surveillance. *IEEE Robotics and Automation Magazine*, 7(4):14–24, December 2000.

[18] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-scale Ad-hoc Sensor Networks. In *International Conference on Robotics and Automation*, pages 1143–1148, 2002.

[19] Z. Sun and J. Reif. On Energy-Minimizing Paths on Terrains for A Mobile Robot. In *International Conference on Robotics and Automation*, pages 3782–3788, 2003.

[20] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

[21] C. Xian and Y.-H. Lu. Dynamic Voltage Scaling for Multitasking Real-Time Systems with Uncertain Execution Time. In *GLSVLSI*, pages 392–397, 2006.

[22] R. Xu, C. Xi, R. Mehlem, and D. Moss. Practical PACE for Embedded Systems. In *ACM International Conference On Embedded Software*, pages 54–63, 2004.

[23] F. Yamasaki, K. Hosoda, and M. Asada. An Energy Consumption Based Control for Humanoid Walking. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 2473–2477, 2002.

[24] W. Yuan and K. Nahrstedt. Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems. In *ACM Symposium on Operating Systems Principles*, pages 149–163, 2003.