Advanced Robotics, Vol. 14, No. 1, pp. 7–26 (2000)

Full paper

Integration of vision and walking skills together with high-level strategies for quadruped robots to play soccer

VINCENT HUGEL, PATRICK BONNIN, JEAN CHRISTOPHE BOURAMOUÉ and PIERRE BLAZEVIC

Laboratoire de Robotique de Paris, 10/12 avenue de l'Europe, 78140 Vélizy, France

Received 15 December 1998; revised 4 May 1999; accepted 31 October 1999

Abstract—Legged robots taking part in real multi-agent activities represent a very innovative challenge. This domain of research requires developments in three main areas. First, without any feedback information from the environment, there is no way for robots to achieve some tasks autonomously. Fortunately, the quadruped 'Sony' prototypes on which all experiments are carried out are equipped with an enhanced vision system; thanks to its CCD camera located in its head, the robot can obtain color images of the scene around it. Extracting relevant information from the images captured is not easy since it must be done onboard in real time. Moreover, image treatment procedures should have high process rates for the robot to react quickly in front of unexpected events. A special vision module composed of three parts has been designed for these purposes. The second point to focus on is the walking ability of the robot. Quadrupeds are designed to move efficiently and rapidly on flat ground. The objective of the walking module is to generate appropriate walking patterns allowing the machine to walk in the desired direction. Walking gaits are produced like reflexes by the robot itself to adapt to the situation. With regard to the design of these gaits, emphasis has been put on increasing speed and mastering transitions. Finally, the machine should be given a minimum of intelligence since it has to manage vision information and its walking gaits by itself. When involved in situations of cooperation or competition or both, like in a soccer game, a high-level supervision task is welcome. This paper presents detailed developments of these three points and describes how they are implemented on a real robot.

Keywords: Robotic soccer; quadruped locomotion; walking patterns; vision recognition; behavior strategies.

1. INTRODUCTION

The LRP legged machines team was given the opportunity of participating in the RoboCup competition which was held in July 1998 in Paris. Sony Corporation and, specifically, the D21 Laboratory in Tokyo very kindly lent three pairs of 'pet robots' to three teams throughout the world. These prototypes represent a high-level development platform on which to put into practice optimized algorithms of vision and locomotion [1]. Behavior strategies can also be included after testing on simulation.

Bearing in mind that three robots must cooperate to score a goal in the opponent's goal, two kinds of strategies are considered. The first one has to answer the following question: how can we incorporate the three functions mentioned above, i.e. locomotion, vision and supervision, and how are they going to interact with each other in an efficient manner? This strategy of implementation is very important since all tasks are to be performed on-board in real time. The second strategy is the decision-level strategy to adopt on the soccer field to win the game.

Section 2 of this paper describes the structure of the onboard control program which manages the three functions. In Section 3, the different steps of vision recognition and localization procedures are explained. Section 4 is dedicated to the walking patterns used to make the robot move quickly in every direction. Section 5 is devoted to the strategy employed for the robots to play soccer. Finally, results are presented and conclusions are drawn from this experiment.

2. ONBOARD CONTROL PROGRAM STRUCTURE

The onboard program is divided into three 'objects' that can be viewed as three execution threads (see Fig. 1). The central one is the 'locomotion object', the one responsible for obtaining information from the environment is called the vision object and the third is dedicated to achieving a recovery procedure if the robot has fallen down.

The 'locomotion object' has three roles. First, it generates appropriate gaits according to the direction planned and sends the corresponding input signals to

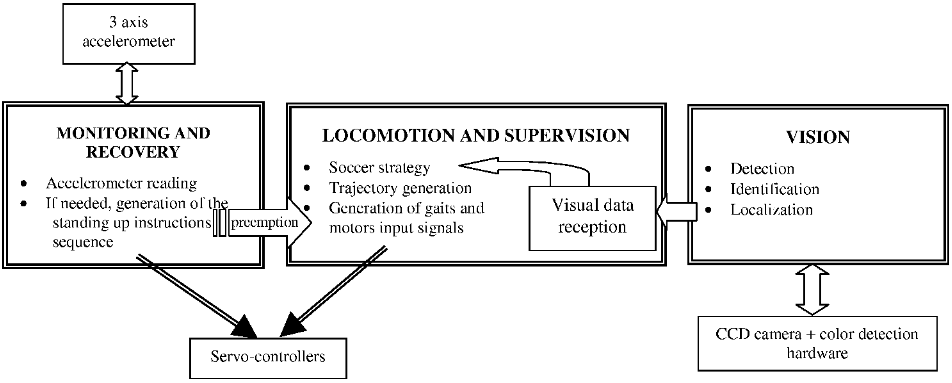

Figure 1. Architecture of the on-board software program. It is composed of three objects that can be seen as execution threads, namely 'monitoring and recovery', 'locomotion and supervision' and 'vision'. The object 'monitoring and recovery' reads outputs from accelerometer and checks if the robots has fallen down. 'Locomotion and supervision' is in charge of generating gaits and strategies to play soccer. The 'vision' object processes image treatment to obtain information about objects located in the scene.

the servo-controllers of the leg joints. The second function deals with receiving visual information from the 'vision object'. The latter captures images through the color detection hardware driver, processes them and transmits useful data to the central task; these high-level visual data are the presence and the location of the elements in sight on the soccer field. The vision thread is given a lower priority than the central one so that it does not disturb the motion. The central 'locomotion object' does not request visual information from the 'vision object'. It is the 'visual object' that puts them at disposal. The task of receiving these data runs in parallel to the task generating walking patterns in the 'locomotion object'. The third function performed by the 'locomotion object' refers to the strategy used to play soccer. The different rules governing decision strategies are updated within a timer routine.

The last object, called 'monitoring and recovery object', appears necessary to make the robot stand up again in the case during the game where it falls down after slipping or collision. This thread runs continuously with the others and monitors the horizontality of the body platform thanks to the onboard three-axis accelerometer. When a fall is detected, the 'standing up object' pre-empts the central task and starts the recovery procedure. Once the robot is standing up again, the object wakes up the central one for it to resume the control of the system.

The next sections detail each of the functions mentioned earlier, i.e. vision, locomotion and behavior strategy.

3. VISION SYSTEM MODULE

To extract relevant information from images captured, a special vision module composed of three parts has been designed. The goal of the vision system is to detect, to identify and to spot the different elements constituting the scene during the play. 'Detecting' means extracting all connected components belonging to the scene elements from the color images. 'Identifying' means finding the one or several connected components constituting an object in the scene and giving it a symbolic label such as ball, beacon, own or opponent goals, partner or opponent player, edge of the soccer field. 'Spotting' the ball, beacons or goals means determining the view angle in azimuth with respect to the head direction, and a 'rough' evaluation of the distance between the head and target.

To accomplish its task, the vision system must deal with three main problems:

- The lighting conditions.
- The needed computing power, compared to the one available.
- Its integration in a real-time system in a walking robot context.

Moreover, the challenge between several teams having the same hardware at their disposal leads us to look for the best adequacy between the developed algorithms and the available architecture, and for the best trade-off between the quality of the obtained results and the algorithmic complexity. In the remainder of this section these problems are first described and then the three parts of the proposed method are presented.

3.1. Three main problems

3.1.1. Lighting conditions. Lighting conditions for the indoor soccer field are not best suited for the vision system. Even if the measurements of the lighting conditions and the color temperature given respectively by an illuminometer (580 lux) and a chromaticitymeter (4500 K) are always the same throughout the soccer field, the appearance of an object in the color image (i.e. Y, U and V values) depends in a major part on its location and the location of the camera. In fact, lighting conditions are produced by four spotlights oriented according to an oblique incidence, they generate shadows (players and beacons) and reflections (on the ball and on the goals) which sometimes saturate the CCD. Curiously, these problems disappear for the most part if a single spotlight, oriented like a shower, produces the lighting conditions, as on a soccer training field. The measurements of the illuminometer and the chromaticitymeter are certainly less constant, but the appearance of the objects does not change.

This problem is crucial because the color processing hardware, specially designed to detect color in real time, does it using thresholding procedures acting directly on Y, U and V values. We pay special attention to determining threshold values using a visual graphic interface.

3.1.2. The needed computing power, compared to that available. A lot of image processing algorithms (e.g. edge detection followed by a connected component extraction, region growing, cooperative edge and region-based segmentation [2], multi-spectral segmentation [3], etc.) are able to detect connected components belonging to scene elements. Some of them are able to adapt themselves (i.e. through their control parameters) to illuminance variations. However, even when implemented on powerful computers like the connection machine CM5, these complex algorithms do not run in real time.

Moreover, if the goal is to extract visual information at a rate of at least 10 images per second, only fast algorithms must be taken into account. This rate seems to be reasonable for continuous walking of the robot and a good perception of the dynamic environment.

3.1.3. The integration of the vision system in a real-time system in a walking robot context. This problem relates to the previous problem. To be of interest, information must be extracted in a short time compatible with the robot displacement (latency time), otherwise decision making with regard to action resulting from this information may not be appropriate. Moreover, for safety reasons, the vision system, unlike the walking module, has no priority over other tasks. For example, for a legged robot it is more important to put down its leg rather than watch the scene.

Thus, the delay between two image acquisitions and the latency time (i.e. the delay between image acquisition and the moment when the information is available) are not constant.

In addition, it is more difficult for the vision system to operate in a walking robot context, because the movement of the scene in the image sequence is jerky, due to the successive take-offs and landings of the legs. The two previous reasons make the temporal tracking of all scene objects in an image sequence very difficult to perform.

3.2. Details of the proposed method

This extraction technique is divided into three parts, i.e. detection, identification and localization steps.

3.2.1. The detection step. The detection step is composed of the five following algorithms:

(1) *Color detection*, performed by Sony-specific hardware using threshold values as input controlled parameters and providing an output in the form of 8 bit-planes, each corresponding to a color template.

(2) *Opening* performed simultaneously on each of the 8 bit-planes, using an isotropic 3×3 centered neighborhood. It cleans the '8-binary' image as it removes pixels resulting from color detection noise.

(3) *Connected component extraction* [4], which requires a single scan of the image, detects equivalent labels and computes the connected component attributes: gravity center, surface, bounding box.

(4) *Filtering* on the surface attribute of the connected components to remove small components due to bad lighting conditions.

(5) *Merging of connected components*, to deal with bad lighting conditions responsible for the decomposition of an object into several connected components. To be merged, two connected components must have their bounding boxes close to each other. An 'image object' resulting from the merging of several connected components is not a connected component. However, its parameters such as surface, gravity center and bounding box are computed directly from the parameters of the merged connected components.

Steps (1), (2) and (3) are low-level image processing applied to image data. Steps (4) and (5) are intermediate-level image processing applied to feature attributes, and they are fast to perform.

3.2.2. The identification step. The ball is a small scene object which may be partly or wholly occluded, and generally it does not produce more than a single image object. Thus, the ball identified in the image is the image object corresponding to the orange color template, whose surface is maximum.

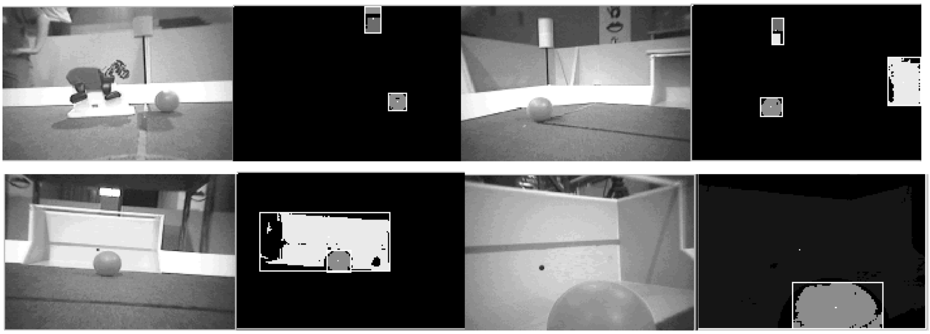

Figure 2. Results of the Image Processing Identification: ball, goals and beacons. Four images captured by the robot are pictured. For each image, the objects detected are represented on the right. Each object has a specific color and is contained inside its bounding box. On the first image, the vision system can spot a landmark and the ball. On the top right image, the system can detect a landmark, the ball and a goal. On the bottom left, a goal and the ball in front of it are detected. This is the same for the last image.

Beacons are also small scene objects, but they are localized in such a way that they cannot be occluded. They are composed of two colors: one pink, the other yellow or blue or green. Taking into account the geometry of the camera and the soccer field, two beacons can be viewed at most. An identified beacon is composed of two image objects, the first is necessarily pink, the second is yellow, blue or green, and the two image objects are located one above the other.

Goals are yellow or blue, but they have the same colors as the beacons. The goals are seldom viewed entirely. Either the robot is far from the goal and the goal is occluded by one or more players or the robot is close and the goal comes out of its field of view. Thus, the identified goals are the image objects (blue or yellow) which are not identified as beacons.

Players are either dark blue or red. However, depending on their direction, they appear constituted by one or two image objects.

The edges of the soccer field are white. It is the last available template color! See Fig. 2.

3.2.3. The localization step. The angle in azimuth with respect to the head direction is computed from the x-component of the center of gravity of the identified object, knowing the image resolution on the x-axis and the horizontal field of view of the video camera (52°), assuming a pin hole model for the camera.

A 'rough' evaluation of the distance between the head and a scene objet is given by a look-up table based on the surface attributes for the ball and the beacons and the height of the bounding box for the goals. However, it appears impossible to give a reliable rough measurement of the distance from another player, because all of the feature attributes change for a great part when a player rotates on itself at the same distance! An 'alert' is generated if the surface of the image object of a player increases beyond a threshold value in order to stop the robot and consequently avoid collisions.

A 'rough' measurement of the distance from the robot to the edge of the soccer field is geometrically computed.

In conclusion, the rate of image processing measured by two different benchmarks is about 15 frames per second while the robot is walking. Under these conditions, the head can rotate smoothly to track the ball, even during walking.

4. WALKING PATTERNS

After mastering visual recognition, the second point to focus on is the walking ability of the robot. Quadrupeds are designed to move efficiently and rapidly on flat ground. The objective of the walking module is to generate appropriate walking patterns allowing the machine to move in the required direction.

4.1. Forward gait

4.1.1. Simple crawl gait. The first stage of the walking pattern study consisted in testing the simple crawl gait on the experimental Sony quadruped machine. This helps us to become aware of the real problems of implementation. As defined by McGhee [5, 6], the crawl gait is a regular symmetric gait, the term 'regular' meaning that each leg reproduces the same trajectory cycle with a phase difference. A symmetric gait is defined by a phase shift of $1/2$ of the cycle between opposite legs, knowing that the body has a rectangular shape (Fig. 3).

Hence the phase difference between consecutive legs on one side determines entirely the sequence of the legs in the creeping gait. In Fig. 4, the general crawl gait is represented using leg state diagrams, greater values refer to the air phase of the legs when they are moving forward to reach their next foothold to begin a new traction phase.

Figure 5a and b represents the successive transitions in the crawl gait. During the cycle three legs always remain on the ground. The period of time called the 'duty factor' β is the fraction of cycle period relating to the support phase. For the crawl the maximum value for β is $3/4$, see Fig. 4.

If up and down moving times of the legs are neglected, the body velocity v_{G} can be expressed as:

$$v_{\mathrm{G}} = \frac{1-\beta}{\beta}\, v, \tag{1}$$

where v is the leg swinging velocity.

Higher values of β involve a reduced speed of the robot if we assume that the mean velocity of actuators in the swinging phase is the same. Indeed, the period of time of the air phase does not change but its corresponding fraction time in the cycle is reduced, hence the period of the complete cycle is increased and, as

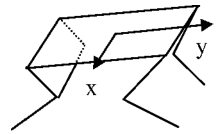

Figure 3. Representation of the quadruped machine. The y-axis of the body coordinate system indicates the direction of motion.

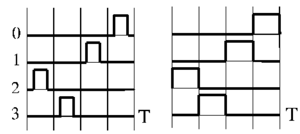

Figure 4. Leg state diagram for the crawl gait with $\beta > 0.75$ and $\beta = 0.75$. Leg 0 is the fore-left leg with respect to a top view. Legs are numbered 0 to 3 in the counterclockwise direction. T is the cycle period. State 0 refers to the stance phase and state 1 to the transfer phase. For duty factor of 0.75, a take-off of a leg occurs simultaneously with the touchdown of another leg.

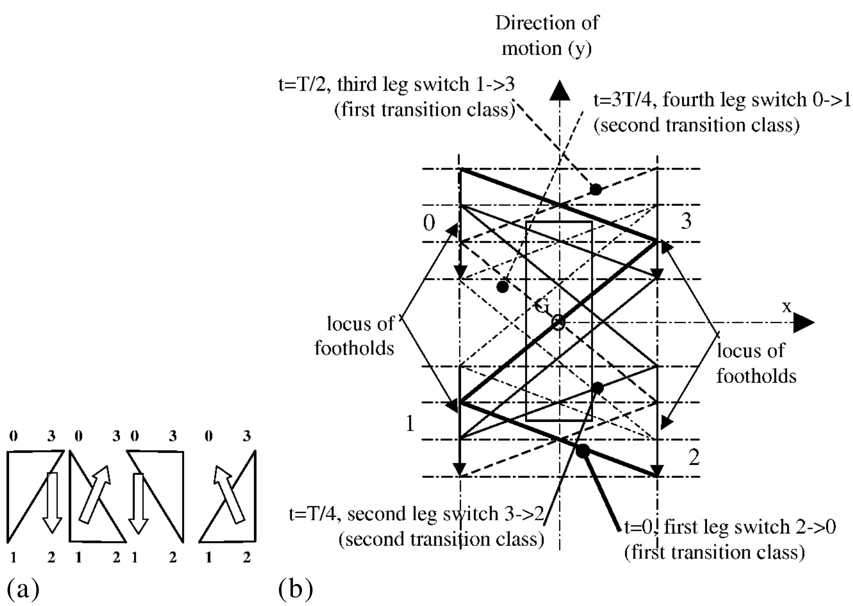

Figure 5. (a) Sequence of supporting polygons with circulation of events of take-off and landing of legs; (b) successive support polygons for the forward crawl gait at fractions of cycle time of 0, $T/4$, $T/2$ and $3T/4$. Duty factor is set to 3/4. Time $t = 0$ refers to the time when leg 0 touches ground.

the distance traveled over the cycle does not vary, speed is reduced. Increasing β without a reduction of speed would mean that it would be possible to increase actuator velocity during the air phase. Therefore, the choice of $\beta = 3/4$ with maximum actuator velocity in the air phase should provide the highest motion speed while keeping three legs on the ground.

Lower values of β give periods of time when only two legs are in the support phase, in these cases problems of stability become more complex. In this paper the study of the crawl focuses on gaits with $\beta = 0.75$.

To study static stability, different positions of the legs on the ground at times of zero (0), a quarter ($T/4$), a half ($T/2$) and three-quarters ($3T/4$) of the cycle period T are represented in the body reference frame. In Fig. 5b footholds are joined to form polygons. Static stability of the crawl depends on the vertical projection of the position of the center of gravity (normally equal to the projection of the geometric center of the rectangular platform joining the four shoulder points) with respect to the edges of these stability polygons. The four changes in the set of the three supporting legs are pictured in Fig. 5a and b. These changes can be partitioned into two pairs or classes. The first class regroups the two transitions where the take-off of a rear leg occurs immediately after the landing of the diagonally opposite front leg. The other class contains the remaining leg changes where the takeoff of a front leg occurs immediately after the landing of the rear leg located on the same side. It is clear that the first class can suffer losses of balance as the center of mass is situated just at the frontier of the stability polygon, see Fig. 5b. However the center must absolutely remain just above this frontier line, e.g. if there is some deviation between the center of gravity and the geometric center, the risk for the robot tipping over is increased as the center of gravity can go out of the current stability polygon some time.

4.1.2. Forward crawl gait with sideways motion. The implementation of the simple crawl on the real robot shows poor results, with frequent losses of balance; the quadruped often tips over the diagonal supporting line and falls backwards. To avoid these problems, it has been decided to improve the gait by adding sideways motion of the body center of mass. The idea of sideways motion is not new, it was introduced by Hirose [7] to study continuous generation of gaits from crawling to troting. In troting, sideways motion is determined according to dynamics considerations aimed at neutralizing the overturning moment due to gravity around the diagonal supporting line [8]. However, the strategy of the trot gait cannot be applied to this case since there is theoretically no overturning gravity moment. The crawl with $\beta = 0.75$ cannot be termed a quasi-static gait either since there are two transitions where the vertical projection of the center of mass is theoretically on the diagonal edge of the support polygon. Considering a sine sideways motion of G, the problem is to determine its amplitude a.

$$a(t) = a \cdot \sin(2\pi t/T). \tag{2}$$

Figure 6 represents the successive trajectory points of G on the x-axis in the body reference frame. In practice there are always differences between the theoretical and the real positions of the center of mass at the instants of transitions of class 1 and these differences can lead to instabilities when the projection of the center of gravity lies outside the support polygon. The duration of such an instability

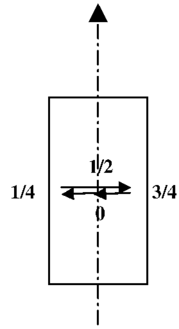

Figure 6. Periodic sideways motion of G in the body reference frame. At times 0 and $T/2$, the amplitude of sideways motion is zero. At $T/4$ and $3T/4$ amplitude is maximal.

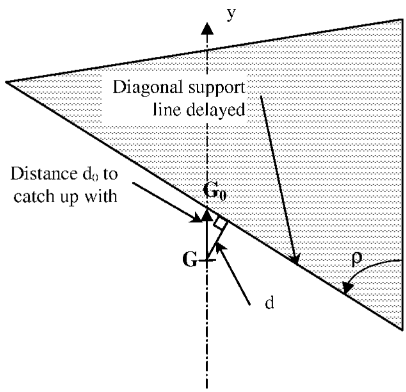

Figure 7. Study of instability, G out of the support pattern. The diagonal line of the support pattern is delayed. The robot must catch up with a certain distance d_0 to avoid falling down. The body must have sufficient kinetic energy to counterbalance the effect of the overturning gravitational moment.

should be kept to a minimum. To achieve this the machine must have sufficient kinetic energy at the time of the transition in order that the center of gravity joins the moving diagonal support line without yielding to the overturning gravitational moment (see Fig. 7). Here a deviation along the y-axis, d_0, between the real and theoretical centers of mass is simulated as first class transition happens. If we call t_0 the instant of transition time and t_1 the instant when G joins the diagonal support line in G_0, the maximal kinetic energy between these two moments of time is:

$$E_c = \frac{1}{2} M v^2(t_0), \tag{3}$$

with $v(t_1) = 0$ in the worst case. M is the mass of the body.

The work performed by the overturning gravity momentum is approximately equal to:

$$W = \frac{1}{2} M g \frac{d^2}{h}, \tag{4}$$

where d is the distance to the diagonal support line at t_0 and h is the height of G.

If kinetic energy is spent to resist the work developed by gravity, we have:

$$v^2(t_0) = (a\omega)^2 + v_{\mathrm{G}y}^2 = g\,\frac{d^2}{h}$$
$$\Rightarrow a = \frac{T}{2\pi}\sqrt{g\,\frac{(d_0 \tan\rho)^2}{h} - v_{\mathrm{G}y}^2}, \tag{5}$$

where ρ is the angle of incline of the diagonal support line with respect to the longitudinal axis of the robot, d_0 is the deviation between current position of G and G_0, and $\omega = 2\pi/T$.

This formula cannot been utilized to determine exact practical values of a. In fact, d_0, v_G and ρ can only be approximated in practice as they are subjected to disturbances. However, this equation shows that if ρ is increased, amplitude must also be increased. If ρ' and a' refer to another type of gait with different parameters, we have:

$$\frac{(a'\omega)^2 + v_{\mathrm{G}y}^2}{(a_0\omega)^2 + v_{\mathrm{G}y}^2} = \frac{\tan^2\rho'}{\tan^2\rho_0}, \tag{6}$$

assuming the same conditions of instability.

Therefore the procedure consists of experimentally estimating the appropriate amplitude a_0 that gives satisfactory results for the forward motion whose diagonal support line incline is ρ_0. Then, as a function of the motion required, ρ' is computed geometrically to get the new sideways motion amplitude a' thanks to relationship (6).

4.2. Turning motions

Walking patterns for straight line and turning motions have been designed specifically in order to walk efficiently. Procedures of switching from one gait to another have also been developed to allow the robot to follow a changing directions. The capabilities of turning with a small turning circle and rotating round its center of gravity appear very useful in the case where the robot has to locate the ball it has just lost.

4.2.1. Turning gaits. The design approach consists in defining turning motions whose center of curvature is located on the transverse axis of the robot. This enables the machine to turn right or left with a varying turning circle. The idea lies in adapting the forward sideways crawl. Since it has been decided to exclude prediction features in the definition of the gait, such as predicted duty times or predicted footholds, it seems difficult to master transitions at any moment of time. To avoid this problem turning gaits are designed in such a manner that they share a common set of contact points with the forward walking pattern at a particular instant of cycle time. This brings advantages in terms of speed and drawbacks due to transition delays.

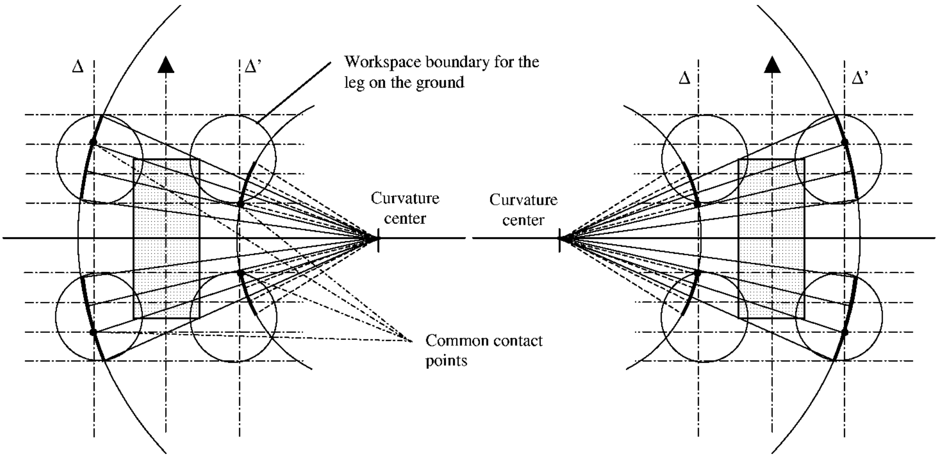

Figure 8. Common set of footholds for right and left turns. For a specific time, footholds are the same in forward walking pattern and in turns. They are situated on lines Δ and Δ' that are the traces on the ground of the vertical planes where the leg trajectories take place during forward motion. Each foothold must be located inside the workspace of the corresponding leg.

Figure 8 shows a common set of contact points for right turns on lines Δ and Δ'. The configuration for right turns refers to the foothold positions of the legs in the forward motion at instant $T/4$ and for left turns it corresponds to positions of leg tips at $3T/4$. The exact distance traveled by the leg in the traction phase with respect to the body reference frame is computed so that the tangential speed of G remains the same, this is to guarantee continuity of speed. The sequence of legs is the same as in the crawl leg state diagram in Fig. 4.

However, the center of gravity must be shifted so as to be on the diagonal supporting line when transitions of class 1 occur (see Fig. 9). Moreover, the sharper the turning circle, the larger the amplitude of the sideways motion. When the turning circle is shorter, the incline ρ of the diagonal supporting line increases and the amplitude must be increased according to (6).

In addition, as the magnitude of lateral oscillations is limited, there is a threshold turning circle under which it is not possible to use the same technique any more. This threshold is approximately equal to the length of the machine, which is about 15 cm. Finally, it must be noticed that right and left turns converge towards the forward crawl gait when the turning circle tends towards infinity.

4.2.2. Particular turn around the center of gravity. Since the goal is to design efficient motion of the robot in every direction, it is interesting to define a rotation motion around the center of gravity. The first step is to find a common set of leg contact points for the transitions. The configuration is given by the intersection between the trajectories in straight-line motion and the circle spotted in Fig. 10.

Transitions can occur at $T/4$ or $3T/4$, depending on the initial and final motions.

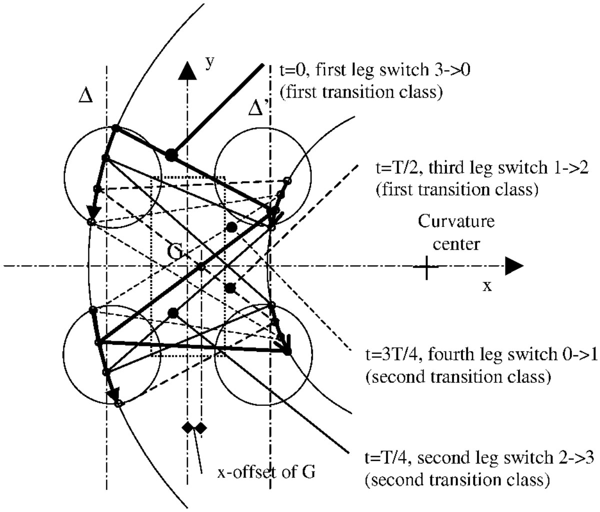

Figure 9. Successive support polygons for right turning gaits at fractions of cycle time of 0, $1/4$, $1/2$ and $3/4$. The reference position of G must be shifted in order that the center of gravity be situated above the diagonal support line when a critical transition occurs. Time $t = 0$ refers to the touchdown of leg 0.

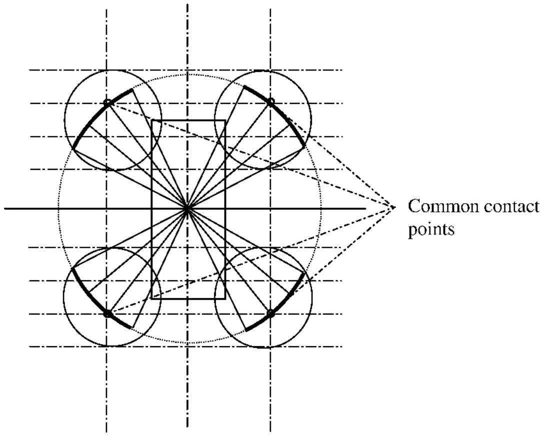

Figure 10. Common set of footholds for right and left turns. The common contact points are situated at the intersection of lines Δ and Δ' with the circular path corresponding to rotation modes.

One important point in rotation motion is that phase differences between legs are changed. It is not a symmetric gait any more, i.e. opposite legs are no longer half a cycle period out of phase. The corresponding leg state diagram is featured on Fig. 11. The successive support polygons are plotted in Fig. 12. Obviously there is no need for sideways displacement of the center of gravity. Rotations in clockwise and counterclockwise directions are straightforward, and transitions between them can be triggered at any time.

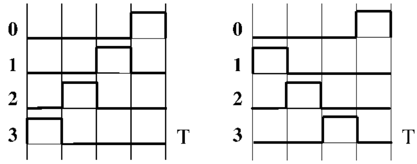

Figure 11. Leg state diagram for right and left rotation motions. Phases between trajectories of legs are not the same as in forward and turning motions. In case of rotation, the event of take-off propagates clockwise or counterclockwise, depending on the direction of rotation.

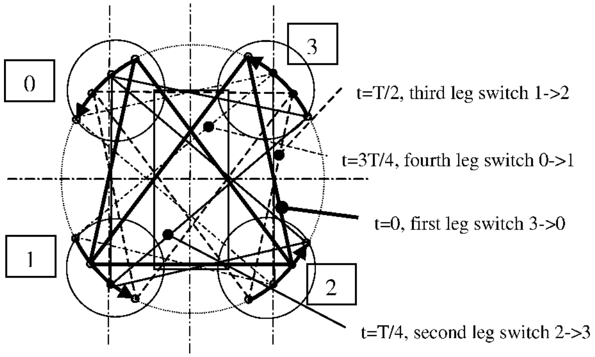

Figure 12. Successive support polygons for right rotation at fractions of cycle time of 0, 1/4, 1/2 and 3/4. Each support pattern rotates as the body turns in place. The center of gravity does not move.

5. BEHAVIOR AND STRATEGY OF QUADRUPED ROBOTS

Vision and walking skills on their own are not sufficient to play soccer. Each robot must take up an appropriate game behavior. Currently robot prototypes are not able to exchange information and therefore a strategy based essentially upon visual information is needed. This is all the more difficult because behaviors have to be developed in real time; the decision module, which often consumes a lot of CPU time, must be carefully implemented so that it does not cause the images processing rate of 15 per second to fall. In the case of the soccer game, behavior strategy modules should provide robot agents with the ability to deal with cooperation and competition activities.

This section focuses on the strategy for playing soccer games which has been tested in simulation and partly implemented on the prototype.

5.1. Strategy

Having two teams of three robots allows us to considere the whole game as a multi-agents system. The main difficulty encountered is the communication between agents (i.e. robots). As a matter of fact, viewing using an embedded vision system is the only way for robots to exchange information by spotting each other on the field.

The idea is to allow the quadruped to quickly swap between planning and reactivity, i.e. that a robot should be able to generate a trajectory, for instance, to kick the ball, or protect its own goal, and give up all its plans at the next decision cycle if the situation has changed.

With regard to strategy, a 'Buckett Brigade' type algorithm has been used. Each agent has a rule system and a weight is associated to each rule. Applying a rule means, for an agent, choosing a role, i.e. a specific behavior such as a role as a Kicker or a Defender. During the game, weights are dynamically modified: the weight of a rule is increased if applying this rule allows the robot to kick the ball or even score a goal; otherwise, the weight is decreased.

Each agent can behave according to the following different roles:

- Move towards the ball and kick it.
- Kick pointing towards the opponent goal (Kicker).
- Defend its own side by clearing the ball (Defender).
- Intercept the player with the ball.
- Look for the ball when its position is not up to date any more.
- Withdraw to the initial position.
- Follow the player with the ball and try to intercept a pass (Interceptor).
- Defend the goal (Goalie).

The basic structure of the rules used in this system is defined as:

$$\text{Rule (Ri): IF} < \text{condition} > \text{THEN Role} = \text{Rk; Weight} = \text{Pi.}$$

About twenty rules have been defined. The triggering condition is defined as the conjunction, the disjunction or the negation of the following predicates:

- I see the ball.
- I see an opponent player.
- I see a partner.
- I see the opponent's goal.
- I see my own goal.
- I am in my own field.
- I am in the opponent's side.
- I am at a distance away from the ball, belonging to an interval I.
- I am at a distance away from one partner, belonging to an interval I.
- I am at a distance away from one opponent, belonging to an interval I.
- I am the closest to the ball.
- One partner is the closest to the ball.
- One opponent is the closest to the ball.

Some comments need to be made here. All predicates involving vision recognition are considered untrue if objects to be seen are located out of the field view. For predicates that compare distances, the agent can use the information of the last known position of the object provided it is known with sufficient certainty. As for the rule weights, they are initially given the value of 500. At every timer cycle (200 ms) the weight of a rule which has not been triggered is decreased:

$$\mathrm{Pi}(t+1) = (1-\varepsilon)\mathrm{Pi}(t).$$

When the robot is touching the ball, the weight of the rule currently selected is increased by an amount C. If the quadruped scores a goal, an amount B is added.

5.2. Simulation results

To get an idea of the different emerging behaviors, several soccer games have been simulated using $\varepsilon = 0.003$. We built a simulator for Windows 95 which uses the real robot's odometry parameters. Playing several games in this simulator allows the quadrupeds to learn how to select the right role for a given game situation by adjusting the rules' weights.

Every agent of the 'red team' was equipped with the weighted rules system described earlier. The other team 'in blue' was given a simple predefined decision model composed of four points:

(1) 'If the ball is located in my field side, I position myself between it and my goal'.

(2) 'If the ball is in the opposite side, I position myself so as to see the ball in front of me in the direction of the opponents goal, then I kick'.

(3) 'If I have lost the ball, I start to look for it'.

(4) 'If the ball is on the right side and I am playing left wing, I wait (and vice versa)'.

The value of B is set to 3000. Results are divided in three games series according to the value of C. In the first series, C is set to 150. The first games are won by the blue team. However, the red team is developing a behavior that, even though it does not allow their agents to score a goal, often puts them in the way of the blue ones. Red agents react very quickly when the ball is close. The most selected role (more than 90% of the cases) is the one that brings the agent towards the ball to kick it. In this phase, very few goals are scored.

In the second series, C is equal to 120. In this case, contact with the ball rewarded less and this allows the red robots to select other behaviors. The role as a kicker is often chosen and this is good to score a goal if another agent is not encountered, which happens frequently. Roles of interceptor and defender are less often selected. However, the 'reds' keep their ability to undermine the planned strategy of the 'blues'. The later games are mostly won by the red team, even if there are few goals. The reds move more quickly, and show more flexibility and reflex capacity. Sometimes combinations emerge, such as intercepting the opponent player that has the ball, followed by moving towards the ball and kicking it. In this case it can result in the partner player, after looking for the ball, kicking it towards the goal.

The last games series features a value of C equal to 90. The behavior as an interceptor is selected most of the time when the agent and one partner are close to the ball. The role as a kicker appears when no opponent is in the surroundings. There are more situations of real cooperation; even if the reactivity of the reds is less noticeable, it remains highly superior to that of the blues. The last soccer games show the clear domination of the red team over the blue one.

These results are of course dependent on the strategies adopted by the different teams on the field. In practice, robots do not know the strategy employed by the opponents. Since they cannot exchange information between them, the only way for a team to win the game is to move quicker than the other or try to guess the strategy chosen by opponents and adapt to it. However, this is not an easy matter!

5.3. Experimental results and future improvements

The main objective of this research is the integration of legged locomotion and vision recognition on a real system with limited processing power.

The strategy adopted for locomotion proves very efficient. Ball tracking allows us to test the motion reactivity of the robot. The machine should not only follow the ball with the head but also move its body in the right direction to get close to it. The quadruped succeeds in tracking the ball in every direction, even when the user moves it around its body. The robot reacts so well that it looks like a living quadruped, playing around with the ball. In fact, this result is due to the design of transitions between the different locomotion modes and to the special design of the turn-in-place modes. However, since walking pattern modes are not continuous at every point in the locomotion cycle, transitions can be delayed. In the worst case, the duration of such a delay is equal to the time spent to walk a step, i.e. the cycle period, which is 1 s for a duty factor set to $3/4$. The maximal speed reached in straight-line motion is 7 cm/s. Even if the system is not perfect, it can represent a good solution to make a quadruped robot move efficiently on level terrain. The main advantage is that it incorporates changes in direction. Another advantage is that the computer power needed is reduced. From a vision point of view, there is no innovation in terms of recognition algorithms. Here the challenge consists of choosing good trade-offs between the processing power required and the goal to reach. One proof of the relevance of the choices made is that the robot is able to walk while getting visual information at the video hardware rate. It means that image processing algorithms and walking pattern generators do not reduce the visual refreshment rate. This is a big advantage since the robot can spot the objects in the scene in real time while walking (in RoboCup-98, objects were the ball and the goals, other objects such as landmarks were very difficult to capture because of bad lighting conditions). By spotting we mean that the machine knows every 65 ms the angle of the object viewed. It is then possible for the robot to center its head to the target and to use the angle of its head to evaluate the turning circle of the next piece of trajectory. By these means, it is able to track a desired object despite some bouncing images due to legged locomotion. However, the solution implemented for vision recognition remains very sensitive to lighting conditions. If the latter are slightly altered, then the machine may no longer distinguish between two different colors. In addition, the distance threshold under which it can identify the object color can decrease dramatically. To avoid color confusion, one can add some filtering but this kind of image processing takes a lot of time and some visual information is lost in the process. That is why future studies will focus on adaptive vision. One axis of research is the combination of shape recognition and color detection.

Regarding the behavioral strategy, further work can be explored in the two following domains:

- *Pattern recognition and learning*. In more complex and less friendly worlds than the soccer field of RoboCup, teaching the robot how to build a model of its environment should help it localize itself and move around. The robot must also be reactive enough to avoid unexpected obstacles which have recently moved in the room. At the same time, it must be able to reach its goal in an efficient way by planing its trajectories and actions.
- *Cooperation between robots to achieve a common task*. Developments in this domain could improve the whole behavior of the robots belonging to the same soccer team and therefore increase their capacity to win the match, e.g. we can think of two robots covering each other when the one or the other has the ball, here the common task is ‘keeping the ball’. In addition, other applications can be imagined, like cleaning a room by pushing objects too heavy for only one robot. However, that domain is strongly dependent on vision and walking: to cooperate usefully, robots need to improve their abilities to communicate and move.

6. CONCLUSION

This paper has described the onboard control structure, vision and walking skills, and behavior strategies implemented on the Sony robots for RoboCup-98. Thanks to the vision module the robot can access the visual data of the positions of the different elements on the soccer field at a rate of 15 images per second. Visual information is processed thanks to a high-level strategy module that generates appropriate gaits towards the updated goal. Walking patterns have been specially designed to make the quadruped walk and change direction as quickly as possible. Forward, backward, turning and revolution gaits are derived from the crawl, which is the best walking gait with three legs on the ground used by mammals to move on flat terrain. In addition, several simulations of RoboCup-98 involving different strategies for each team have been tested. During the competition in Paris our prototypes showed good abilities in moving on the soccer field and tracking the ball. Further developments are under way to improve the whole behavior of our quadruped team. These improvements include the positioning accuracy of the robot with respect to the ball and the goals, the pattern recognition system, the statistical management of the previous visual information, and adaptive behavior strategies.

Acknowledgements

Special thanks are given to Sony Corp. and, in particular, to the D21 Laboratory in Tokyo, Japan for lending us six prototypes of the 'Sony Pet Robot' and helping us with the RoboCup.

REFERENCES

1. H. Kitano, M. Fujita, S. Zrehen and H. Kitano, Sony legged robot for RoboCup challenge, in: *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, Leuwen, Belgium, pp. 2605–2612 (1998).
2. P. Bonnin, Systematic method of design and realization of vision applications, PhD Thesis, University of PARIS VII, France (1991) (in French).
3. P. Bonnin, B. Hoeltzener and E. Pissaloux, A new way of image data fusion: the multi-spectral image cooperative segmentation, in: *Proc. IEEE Int. Conf. on Image Processing ICIP95*, Washington, DC, Vol. III, pp. 572–575 (1995).
4. A. Rosenfeld and J. L. Pfalz, Sequential operations in digital picture processing, *J. ACM* **13**, 471–494 (1966).
5. R. B. McGhee, Some finite state aspects of legged locomotion, *Mathematical Biosci.* **2**, 67–84 (1968).
6. R. B. McGhee and A. A. Frank, On the stability properties of quadruped creeping gaits, *Mathematical Biosci.* **3**, 331–351 (1968).
7. K. Inagaki and H. Kobayashi, Dynamical motion control for quadruped walking with autonomous distributed system, in: *IROS, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Munich, Germany, pp. 1004–1010 (1994).
8. S. Hirose, K. Yoneda, R. Furuya and T. Takagi, Dynamic and static fusion control of quadruped walking vehicle, in: *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, pp. 199–204 (1989).

ABOUT THE AUTHORS

Vincent Hugel graduated from the 'Ecole des Mines de Nancy', one of the leading French engineering schools in 1994. He received his specialization degree in Robotics at INSTN (CEA, FRANCE) in 1995. He has been involved in the design and implementation of walking algorithms on hexapods and quadrupeds. This work is part of the development of his PhD thesis, which he received in November 1999.

Patrick Bonnin was born in 1961. He graduated from ENS de CACHAN, France, and received his Agregation of Applied Physics in 1985. He received his Masters degree in 1986 in Remote Sensing and his PhD on Image Processing in 1991, both from Paris VII University. Since 1992 he has been an Assistant Professor at Paris Nord University – IUT de Villetaneuse. From 1985 to 1996, he worked at ETCA (French Defense Laboratory) in the field of image processing, data fusion and parallel processing. He joined the LRP in 1998. His current research interests are real-time vision systems for robotics, segmentation of color images and stereoscopic vision.

Jean-Christophe Bouramoué is currently working as an AI Engineer in a French software company. He received his Master degree in Artificial Intelligence in 1998 (DEA IARFA – Paris VI). His domains of interests are object-oriented design and modeling, human – machine interface design, and multi-agent systems.

Pierre Blazevic was born in 1962. He graduated from ENS de CACHAN, France, and received his Agregation of Applied Physics in 1985. He received his Masters degree in Robotics in 1986 and his PhD in 1991, both from Paris VI University. Since 1992 he has been an Assistant Professor at Versailles University. He joined the LRP at its creation in 1987. His current research interests are wheeled and legged locomotion, and force feedback devices for teleoperation.