Concurrent movements over rough surfaces

Farid García Lamont & Matías Alvarado Mentado
farid@computacion.cs.cinvestav.mx, matias@cs.cinvestav.mx
Departamento de Computación, Centro de Investigación y de Estudios Avanzados del IPN

Navigation of autonomous vehicles and motion of robots have been of interest for space exploration [1], navigation missions through unknown terrain [2], or collective games [6]. Many techniques consider the texture or roughness –synonym words here- of the terrain for navigation. The collective game of robots playing football soccer implies to compete for both field game spaces and the ball. It must control the player's access to each part of the game field and the ball in order to avoid collisions; it can be done by applying concurrent control techniques [5]. In [1] uses a vibration-based method for terrain classification for planet reconnaissance. Vibrations are measured using an accelerometer mounted on the robot structure. The classifier is trained using labeled vibration data during an offline learning phase. The algorithm uses signal processing techniques, principal component analysis and linear discriminant analysis. The algorithms of obstacle detection and local map building [2], are enough for cross-country navigation. In architecture based on behavior, these algorithms control the vehicle's speed considerably higher than a system that plans an optimal path through a high resolution terrain's map. In [3] presents a method that enables a vehicle to acquire roughness estimation so as to navigate at high velocities. It uses supervised self-learning, that enables the vehicle to learn to detect rough terrains while it is in motion and without human help. The training data is obtained from a filter analysis of inertial data acquired in the centre of the vehicle. Such data is used to train a classifier that predicts the terrain roughness from a laser ray.

We propose to use Appearance-Based Vision (ABV) for roughness analysis and recognition, in order to calculate the friction of robots motion on different textures. The ABV calculates the principal components of the distributed objects, in other words, the eigenvectors of covariance matrix of a set of objects images. An image is a point in an hiperdimentional space. For texture recognition it is used a Neural Network (NN) because of its efficiency, generalization and pattern recognition capacity. The set of image training projected to the eigenspace, principal components, is used to train the NN. These vectors are orthogonal and linearly independent; in consequence the NN training lasts just a few seconds, since principal components can be separated easily.

In order to calculate the textures model of the surface, the images of the surface are used but, if the surface image is taken as an only image. Then it is obtained the surface model as an object, and the data located in the middle of the surface is ignored, in other words, it cannot acquire important data about roughness. In order to solve this problem, the surface is divided in squares, squaring. Squaring enables to both model and recognize textures characteristics of the surface. Squaring offers the next advantages:

- The roughness details can be well modeled using data from different training images, such that surface details are integrated to the model.
- The same last criterion applies by modeling surface's irregularities.
- The more divisions are made the more exact the surface's model is.
- The NN training is robust and roughness recognition is agile and precise.

The roughness recognition is a classification problem, where the training set is made of all the eigenvectors. In order to solve the classification problem a NN is used. The NN works as a function mapping the discrete points, i.e. the textures to identify, to their corresponding friction coefficient. The input vector is the image projection to the eigenspace, while the friction coefficient is the output. An important feature for both texture modeling and texture recognition to point is illumination. Slight variations alter the roughness recognition enormously. The light source must be totally perpendicular and in the middle of the surface, at a convenient height, such to illuminate uniformly throughout. If the illumination angle is different or the illumination height is not adequate, the irregularities may generate shadows.

Friction plays a role in the simplest actions of living, such as walking, grasping and stacking. Friction is a tangential force that expresses the opposition of two bodies' surfaces in touch against motion. The friction coefficient is a non-unit value that depends on the contacting materials, as wood on steel, steel on concrete, etc. The roughness value is useful in order to calculate the necessary strength to move a body. In many cases the friction force is not small. Surface roughness is directly related to surface friction coefficient.

Surface roughness data is useful to establish the proper speed a robot can move without slide risk and, in consequence, to fall because of it, either going up or going down slopes, or by doing sudden movements like a hard breaking. Firstly, as training phase, it is made a sampling of the surface from a superior perspective (aerial view). Later, it is calculated the mathematical model of roughness with eigenspaces, and then several tests are made with different textures [4].

Concurrency of movements: In football soccer game, the players compete for the ball and the position on the field. A player can take the ball and occupy any part of the field. This obligates to have a concurrency control (CC) so as to synchronize two or more processes that cooperate to realize a task; the processes are coordinated by shared sources. The shared source can be modified only one process at time [5]. In this case the task is the soccer game and each player is a process, the CC is necessary in order to avoid the next situations:

- Collisions between players when they compete for a position in the field.
- Ball disputes between team partners, when somebody is playing with it.

In order to implement CC it is necessary to know the location of the players, as team partners as opponent, the distribution of them over the surface, and to determine distances between them, the ball and the opponent's goal. With a coordinate system the field game can be divided and elaborate a map so as to locate the players easily and to proceed to control their movements. The surface is divided and it is establish that each square can be occupy only by one player at time, the players that arrives first, this guarantees that the players do not crash.

In football soccer simulations the team with the ball looks for an advantage position in order to make a goal, avoiding the rival's mark. Each defender pursues an attacking player previously assigned. The coach of the attacking team indicates each player which part of the field has to move. The coach of the opponent team locates the position of each attacking player and notifies to its players the current location of attacking players. Although each coach is a monitor of the player's movements, they do not notice the collision risk between them. Each player locks and releases the squares necessary to complete its trajectory to the assigned position by the coach. Before a player walks, the player verifies if the square, where is going to move, is occupied; if it is not occupied, the players locks the square and then it occupies it. When a square is locked, no one else can move to that square until the owner decides to move and then releases the square when the player has already left it. If the square is occupied, the player must round it, if it is possible, or to wait for instructions from the coach. From simulations of two teams with five players per team, it concludes that the CC proposed is successful by avoiding the players' crash and the game's development is normal on surfaces totally plain and smooth.

Using ABV, it must build a model able to recognize different kind of textures on the terrain. Future work consists on, to control the player's velocity considering friction; it depends on surface roughness, usually different throughout the field. By adding the roughness variable to CC mechanism, the robots would play football soccer with adaptable mobility (adaptive), depending on the texture of the field. The vision system must incorporate the field's aerial view so as to model the surface and to recognize its textures. With this data the simulated game becomes more real world-based, on surfaces with irregularities, holes and/or slopes. As an immediate future work, irregularities less than 15 degrees, enough to alter the player's dynamics. The idea is to include the friction of the corresponding field's texture at the moment when the velocity is calculated, such that to avoid slide or fall risks.

References

[1] Brooks, A.C. 2005, Vibration-Based Terrain Classification for Planetary Exploration Rovers. *IEEE Transactions on Robotics 21(6)*, 1185-1191.

[2] Langer, D., Rosenblatt, J. K., Herbert, M., 1994. A Behavior-Based System for Off-Road Navigation, *IEEE Transactions on Robotics and Automation 10(6)*, 776-783.

[3] Stavens, D., Thrun, S., 2006. A Self-Supervised Terrain Roughness Estimator for Off-Road Autonomous Driving. *22nd Conference on Uncertainty in artificial Intelligence*, Cambridge, MA, USA.

[4] García, F., Alvarado, M., 2007. Surface Recognition using Roughness Information. *Pattern Recognition Letters*. Under review, sent in November 2007.

[5] Andrews, G., Concurrent Programming, Principles and Practice, *The Benjamin/Cummings Publishing Company*, 1991.

[6] www.robocup.org