CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS

DEL INSTITUTO POLITECNICO NACIONAL

Concurrent Movements Over Rough Surfaces

Student: Farid García Lamont

Advisor: Dr. José Matías Alvarado Mentado

Computation Ph.D.

Index

- Antecedents (RoboCup, Terrain exploration)
- Surface modeling
  - Texture recognition
  - Appearance-Based-Models for Vision
  - **Squaring**
- Rough Surfaces
  - Roughness (friction)
  - Friction Coefficient
- **Movements concurrency on rough surfaces**
  - Concurrent moves on football soccer game
  - Squaring as mean of Concurrency Control
- Future: Motion on irregular surfaces
  - Adaptive velocity = friction(texture)
  - Tests on Bioloid kit

What is RoboCup? (antecedents)

- Tournament with simulation and robotics football players categories.
- Goal: intelligent machines and software deploys to play soccer without human help.

Category: Small Size League

Terrain exploration

Current methods (terrain exploration)

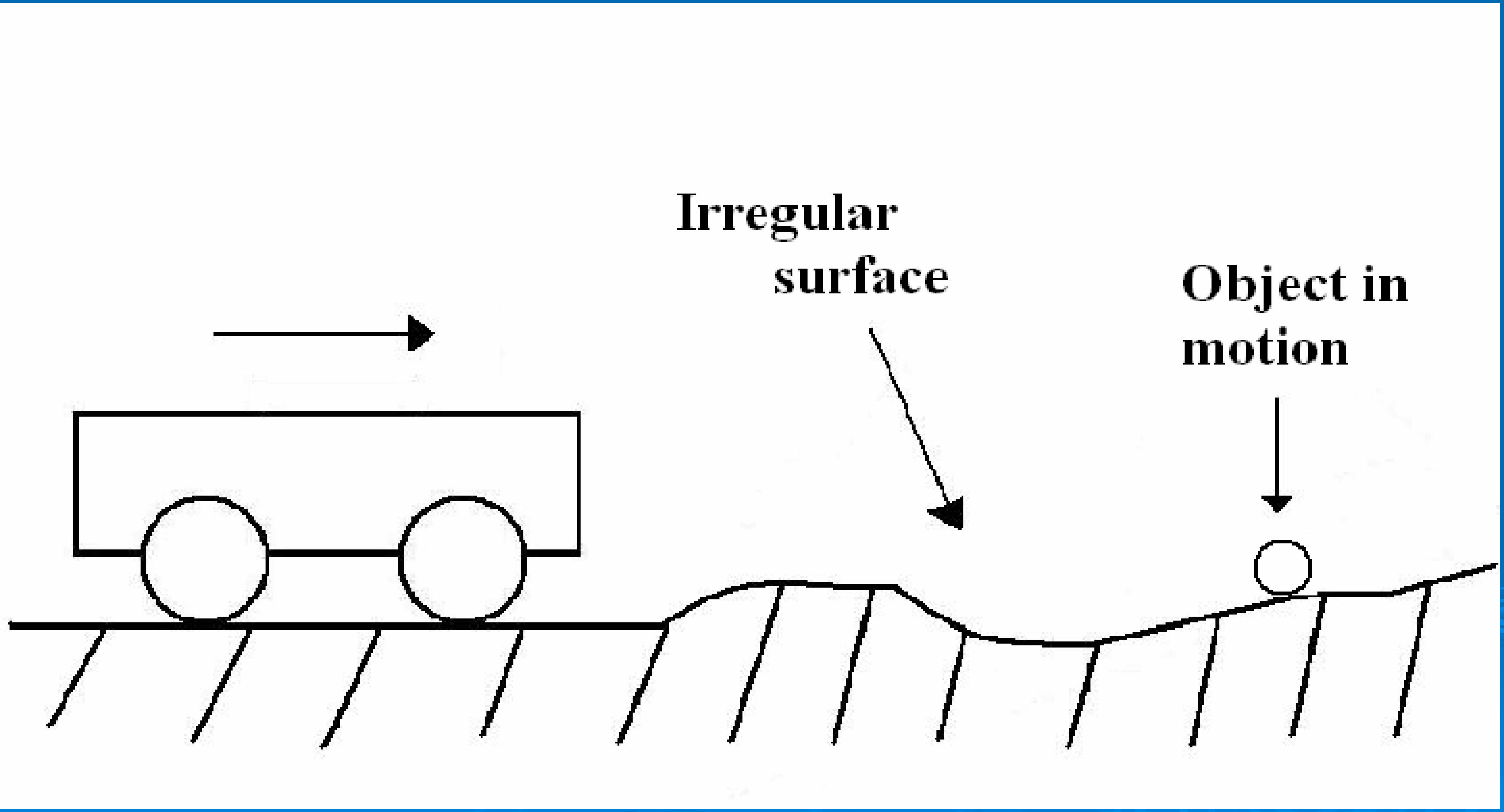
- Laser Methods (2006, 2005, 2004)
  - On-line terrain scanning.
  - Robot's speed can be limited.

- Vibration Methods (2005, 2004, 2003)
  - On-line terrain recognition during robot exploration.
  - Cannot anticipate surface roughness immediately in front of the robot.

Problem overview

- RoboCup players on even surfaces have:
  - ✓ Speed, Agility, Precision.
- **What's on fields with *soft* drops and holes?**
  - **For moving on soft surface:**
    - Neural network training
    - Kalman filter state prediction
  - **Roughness**
    - Irregularities: slopes and holes.
    - Appearance-Based-Modeled (ABM)
  - **Concurrent Environments**
    - Collective games

Problem statement



Surface recognition by texture (roughness) information

- Off-line surface modeling
  - Surface aerial view (satellite images)
  - Training through previous surface images
- Modeling of:
  - Surface irregularities
  - Moving object
- Tracking of moving object

Surface Modeling methodology

Appearance-Based-Model

- Acquire object's images from different perspectives.
- Apply principal component analysis.
- Obtain orthonormal base from eigenvectors.
- Project all images to eigenspace.
- Image recognition.

Eigenspace

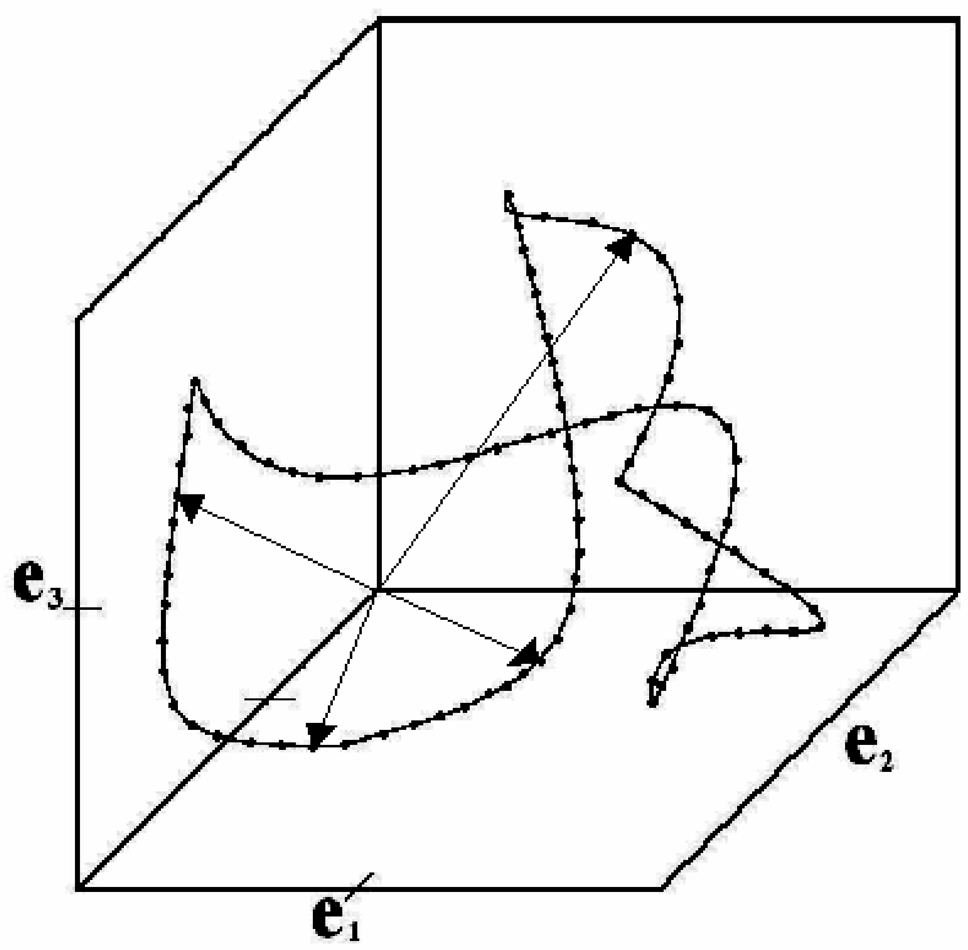
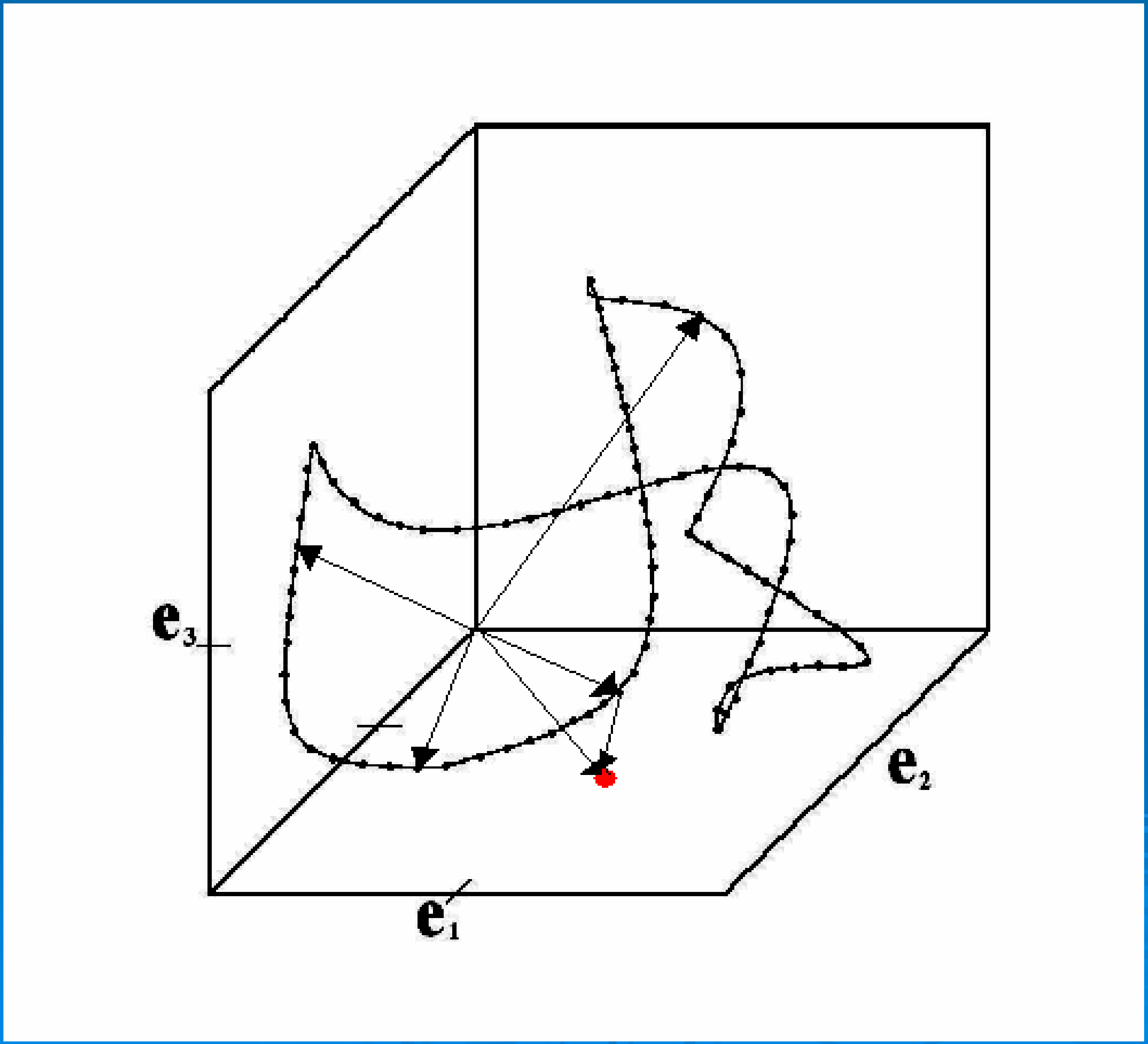


Image Recognition



Advantages:

- Adaptive: the more essential acquired images, the more efficient image recognition is.
- High space dimension, $n \times m$, so expensive computational cost… but eigenspace is reduced with the “Turk and Pentland trick”, $N \times N$ ($N << n \cdot m$).

More advantages:

- Object's integral modeling:
  - Shape, color, texture.
- Common feature to obstacles recognition
- Model soft surfaces irregularities
  - Small drops and holes.
- Slopes minors to 15º
- **Consider surface texture (roughness)**

Disadvantages

- **Light sensitive**

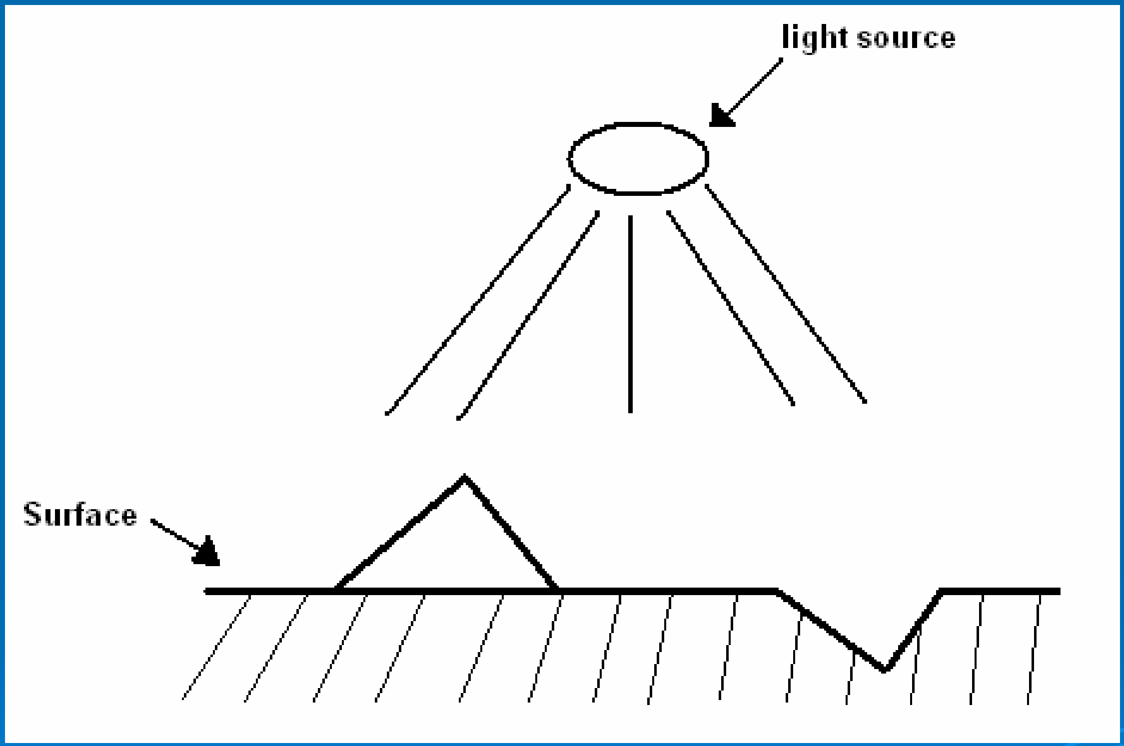
- The number of operations for recognition is $O(k \log_2 n)$, where k is the space dimension and n the quantity of points in the manifold.

Remark: *The illumination problem*

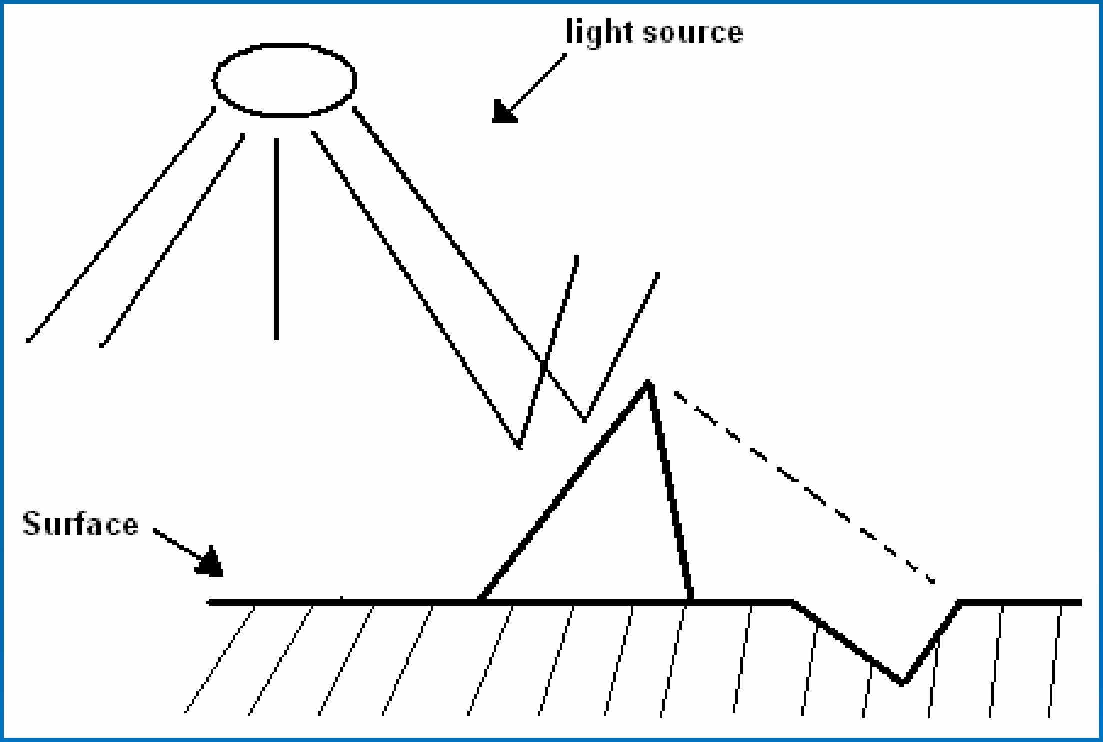
- Slight variations in illumination alters the roughness recognition enormously.

- The light source must be totally perpendicular and in the middle of the surface, at a convenient height.

Correct illumination angle



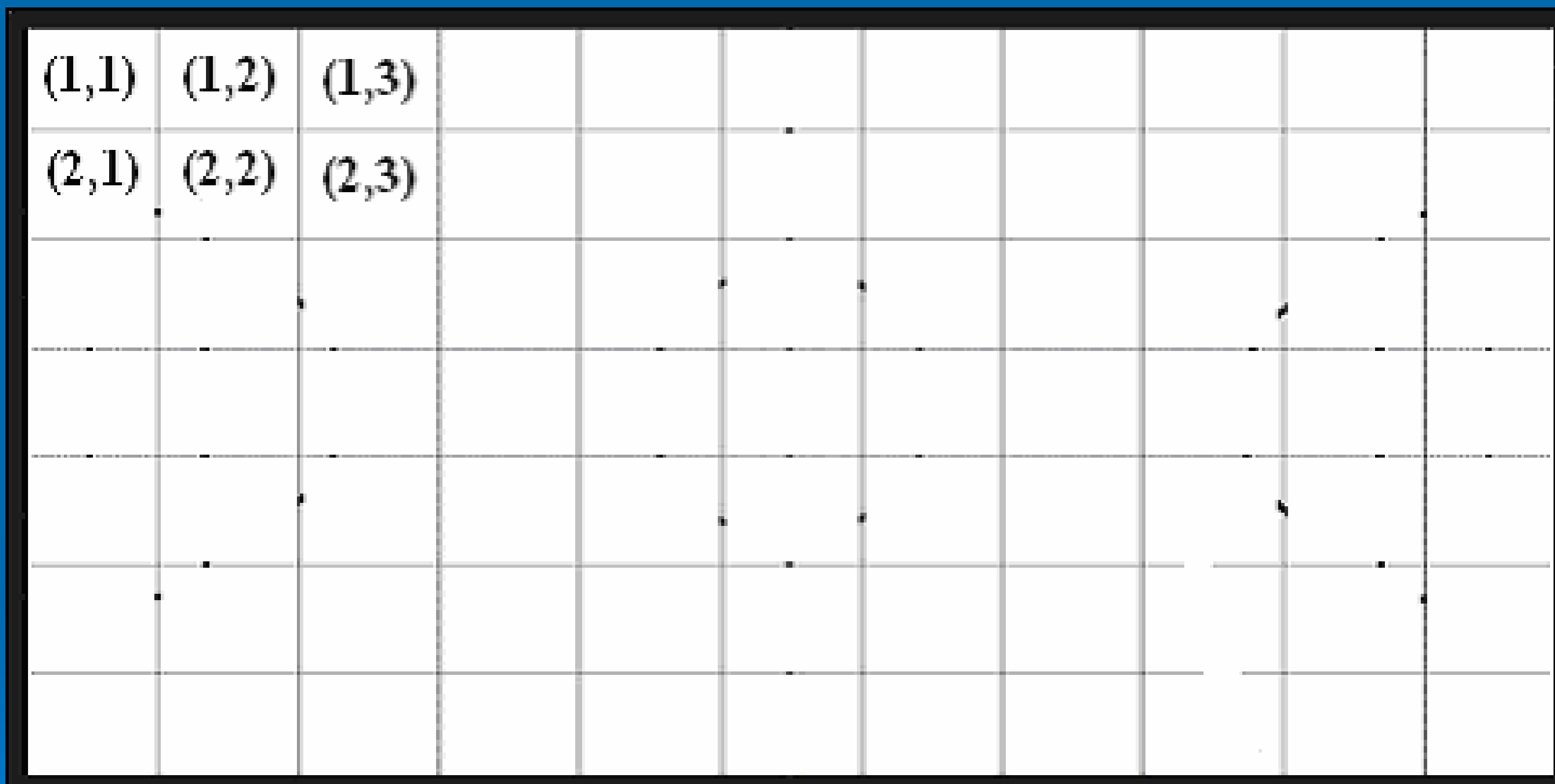
Wrong illumination angle



Surface Squaring:

- Sensitivity to the surfaces' detailed differences: by using ample information from different training images,
- **the roughness differences through the surface are**
  - **Fine modeled**
  - Well integrated in the model.
  - Easy objects location.
- **Facilitates Concurrency Control**

Squaring



Neural network for surfaces recognition

The object recognition as classification problem.

- NN excellent for classification problems.

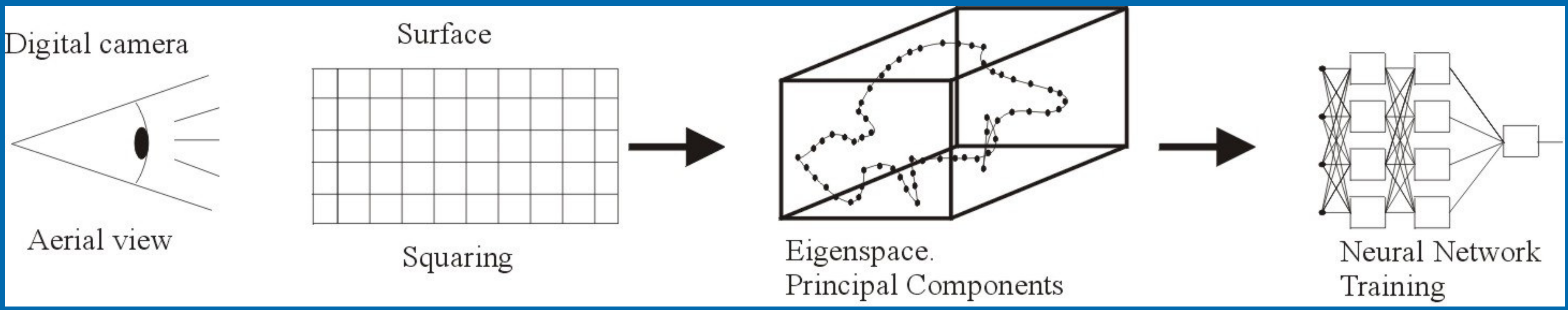
- The principal components and the interpolated points are used as the elements of the NN's training set.

NN Training

- With the feedback – backpropagation algorithm (learning supervised).

- The NN's training and roughness recognition takes a few seconds.

Surface Modeling and NN Training



The NN's training set is $\{(\theta_1, \mu_1), \ldots, (\theta_N, \mu_N)\}$.

μ_i is the friction coefficient of image θ_i.

μ_i integration at the training step

Friction coefficient from roughness information

- Friction coefficient
  - Depends of the surface roughness
  - Mathematical function of the roughness

- NN recognition testing

What is Friction?

- Friction is a tangential force that expresses the opposition of two bodies' surfaces in touch against motion.

- Friction force is expressed as the multiplication of friction coefficient μ and the normal force *R* between both surfaces:

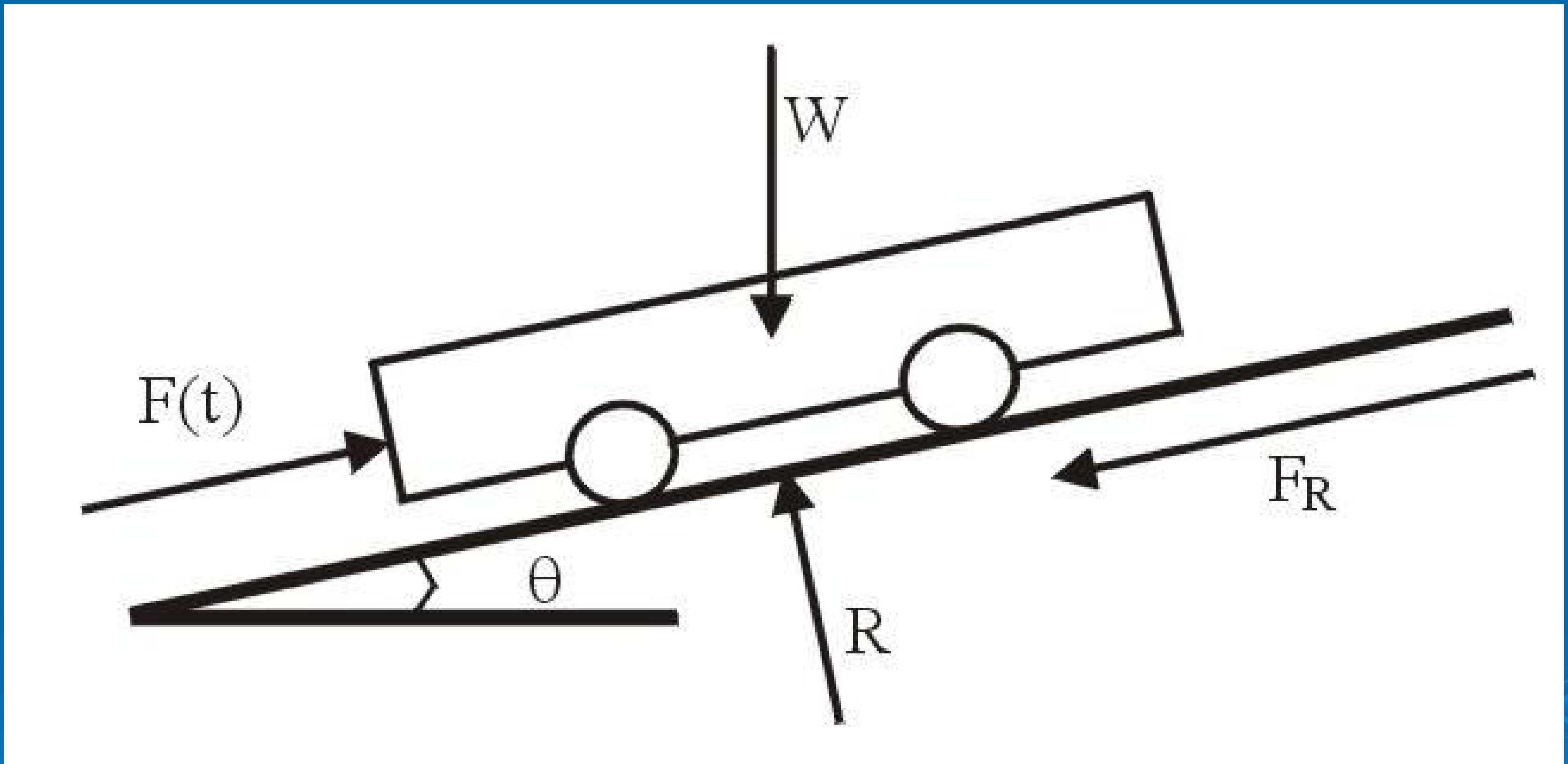
$$F_R = \mu R.$$

Friction Coefficient:

- is a non-unit value μ depending on each pair of touching materials. Real number, $0 \leq \mu \leq 1$.

- Robot has rubber wheels as mean of traction,
  - Necessary to recognize the surface texture, and
  - to look for the corresponding coefficient for rubber on the different materials:
    - Wood, steel, carpet, fabric, ground, grass, mosaic, …

The Physics of friction



$$F(t) - W\sin\theta - F_R = m\ddot{x}(t)$$

Concurrency on Collective Games

- Football soccer
  - Players compete for:
    - Space on the field
    - Ball possessing
- Risk of (situations to avoid):
  - Collisions between players
  - Ball disputing between team partners

Concurrent Control

- Necessary Data
  - Location of:
    - Players (team partners and opponents)
    - Ball
  - Distance between players
  - Opponent's goal
- Squaring helps to draw a map useful:
  - to locate players and the ball,
  - to control the players moves

Concurrent Control

- Each square is occupied by only one player at time (first coming).
- So, before a player walks to a square:
  - it checks if it is occupied, if not,
  - the player locks the square and moves to occupy it.
  - No one else can use that square until the owner releases the lock.

Concurrent Control

- When the owner decides to move,
  - locks the next square where is going to move,
  - then moves to the square and
  - finally releases the lock of the square where it was.

Football Soccer Simulation

- Two teams with five players each one
  - The team with the ball tries to make a goal
  - The other team pursues the opponents in order not to let them make goal
- One coach per team
  - The coach gives instructions of movements to do
  - The coach knows the opponents and partners location

Football Soccer Simulation

… What’s on motion on *irregular surfaces?!*

- Textures on the surface
- Drops and holes on the surface
- Concurrent access to the surface
- Tracking of moving object

My future research **Adaptive Velocity**

- Player's speed cannot be constant due to texture is different throughout the surface!

- Players have to adapt their speed depending on the:
  - Texture
    - Slow if surface is slippery
    - Fast if surface is rough
  - Size of holes or slopes in its trajectory
- Always trying to move as fast as it is possible

Future Tests (Bioloid kit)

Thanks!

Opinions, suggestions or Questions…

Farid García Lamont

email:

farid@computacion.cs.cinvestav.mx

Appearance-Based Model

1) Let $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\} \subset \mathbf{R}^{n \times m}$ the training images.

2) The images are stacked $\{\phi_1, \ldots, \phi_N\} \subset \mathbf{R}^{nm}$.

3) All the vectors are normalized with $\tilde{\phi}_i = \dfrac{\phi_i}{\| \phi_i \|}$,

$\{\tilde{\phi}_1, \ldots, \tilde{\phi}_N\} \subset \mathbf{R}^{nm}$ is the set after this operation.

Appearance-Based Model

4) The average vector is calculated $\vec{\mathbf{C}} = \frac{1}{N}\sum_{i=1}^{N}\tilde{\phi}_i.$

5) The images are centered :

$$\Phi = \left[\tilde{\phi}_1 - \vec{\mathbf{C}} \middle| \cdots \middle| \tilde{\phi}_N - \vec{\mathbf{C}}\right] \in \mathbf{R}^{nm \times N}.$$

6) The covariance matrix is obtained :

$$\Omega = \Phi\Phi^{\mathrm{T}}.$$

Appearance-Based Model

7) The eigenvalues and eigenvectors are calculated with the known equation :

$$\Omega\Psi = \vec{\lambda}\Psi.$$

Where :

$$\Psi = [\mathbf{e}_1 \mid \cdots \mid \mathbf{e}_{nm}],$$

$$\vec{\lambda} = \{\lambda_1, \ldots, \lambda_{nm}\}.$$

Appearance-Based Model

8) All the training images are projected to the eigenspace:

$$\theta_i = \Psi^T \left(\tilde{\phi}_i - \vec{\mathbf{C}} \right) \in \mathbf{R}^{nm}, i = 1, \ldots, N.$$

The images are interpolated using splines, and finally we have a manifold $\theta(\mathrm{q})$.

Image Recognition

Let the testing image $\mathbf{I}_t$, is stacked and normalized $\tilde{\varphi}_t$, then it is projected to the eigenspace:

$$\omega_t = \Psi^T\left(\tilde{\varphi}_t - \vec{\mathbf{C}}\right).$$

The object recognition is reduced to find the closest manifold point, in other words, to find the minimum q such that:

$$\| \omega_t - \theta(q) \| \leq \varepsilon, \text{ for } \varepsilon \geq 0.$$