Improving the Robustness of a Direct Visual Odometry Algorithm for Planetary Rovers

Geovanni Martínez

Image Processing and Computer Vision Research Laboratory (IPCV-LAB)
School of Electrical Engineering, University of Costa Rica

September 6th, 2018

IEEE CCE-2018, September 5-7, 2018, Mexico City, Mexico

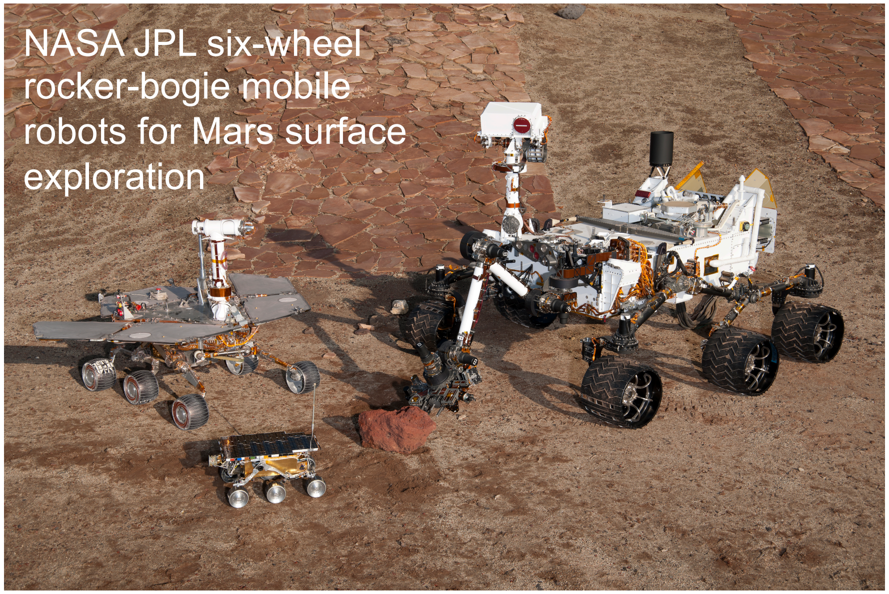
Overview

- Introduction
- Monocular visual odometry algorithm
- Problem
- Approach
- Outlier detection algorithm
- Results
- Summary

IPCV-LAB robot VISION Universidad de Costa Rica

Introduction

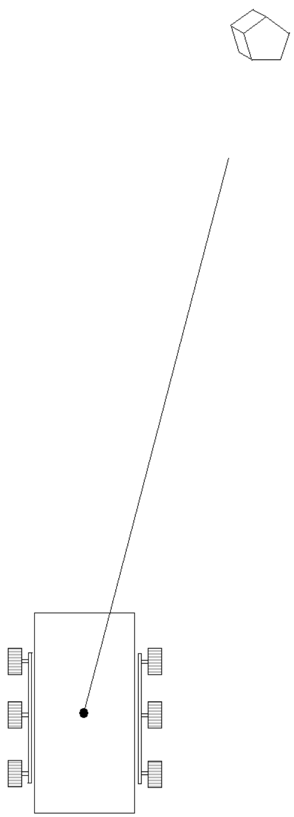
- One important feature of these rovers for Mars exploration is their ability to navigate and perform activities autonomosly



Courtesy NASA/JPL-Caltech

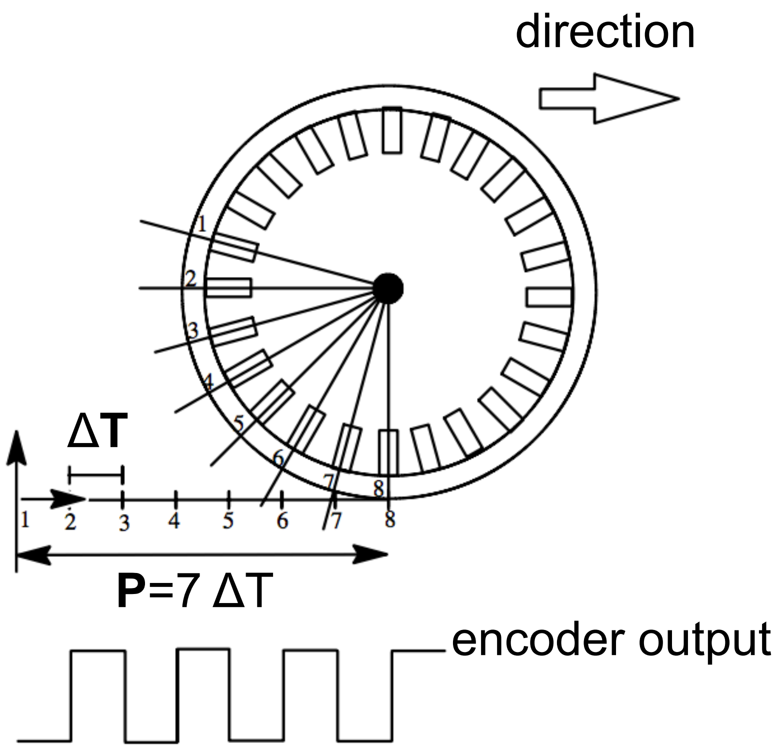
Introduction

- For safe and precise autonomous navigation of a rover along a predefine path, the rover needs to know its exact position and orientation at any time



Introduction

- The rover´s position **P** is obtained by integrating its translation **ΔT** over time
- **ΔT** is estimated from encoder readings of how much the wheels turned (wheel odometry)



IPCV-LAB robot VISION Universidad de Costa Rica

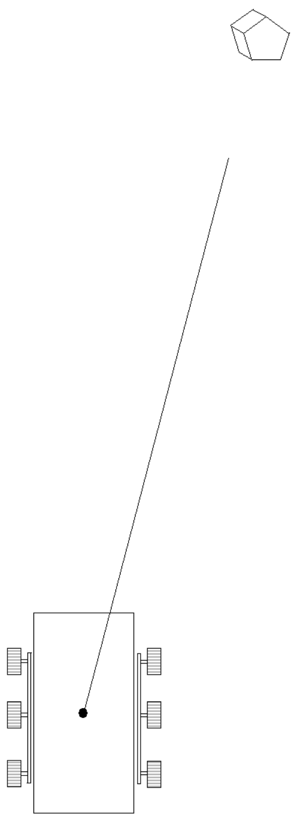
Introduction

- Wheel odometry fails on slippery environments, such as steep slopes and sandy terrain, because the wheels slip due to the loss of traction

Courtesy NASA/JPL-Caltech

Introduction

- This could cause the rover to deviate from its desired path and this in turn could cause the loss of an entire day of scientific activity, trap the vehicle in hazardous terrain, or even damage the hardware

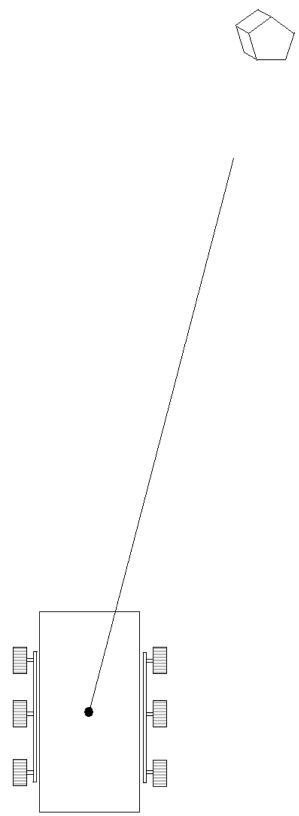


Introduction

IPCV-LAB robot VISION Universidad de Costa Rica

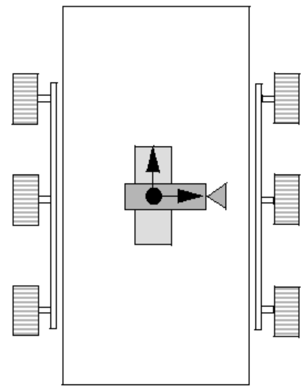
Slip detection and compensation

- Estimate the rover´s motion **B** from the video signal delivered by a single camera mounted on the rover by applying a direct **monocular visual odometry algorithm**
- Compute the rover’s position by accumulating the motion estimates $\hat{\mathbf{B}}$ over time
- Detect and compensate any slip that may occur by using the computed rover’s position



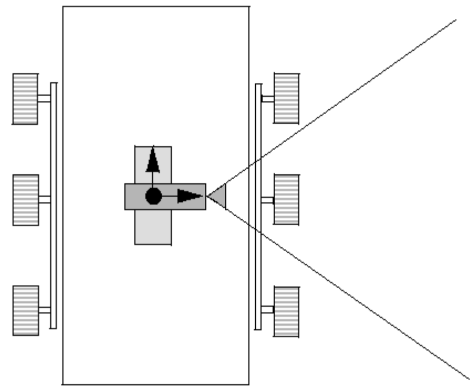
Monocular Visual Odometry

Algorithm



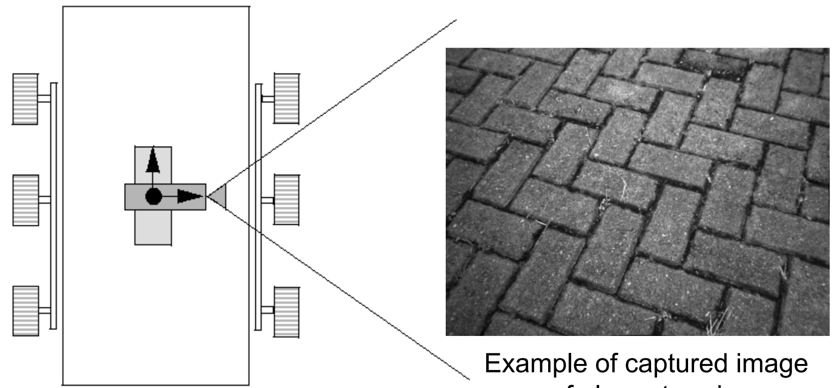
Monocular Visual Odometry

1. Capture a first intensity image before the rover's motion



Monocular Visual Odometry

1. Capture a first intensity image before the rover's motion

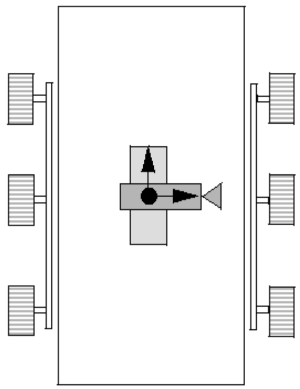


Example of captured image of planar terrain

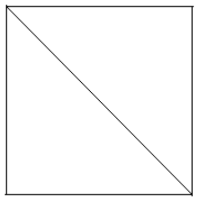
IPCV-LAB robot VISION Universidad de Costa Rica

Monocular Visual Odometry

2. Adapt the size, position and orientation of a generic surface model to the content of the first intensity image



Adapted surface model

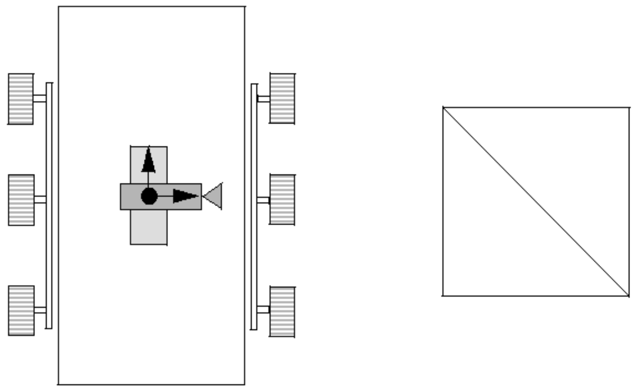


This assumption affects the performance of the algorithm in irregular terrain

This model assumes that the surface is flat and rigid, and describes it by a rigid and flat mesh of triangles consisting of only two triangles

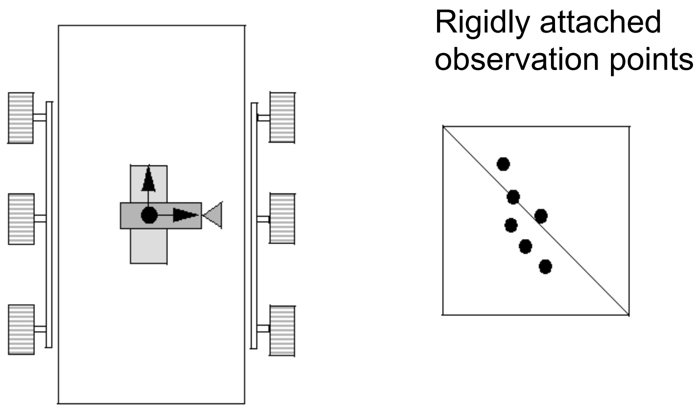
Monocular Visual Odometry

3. Select as observation points those image points in the first intensity image with high linear intensity gradients



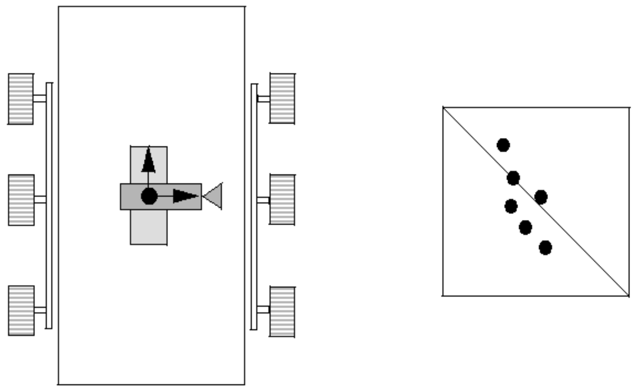
Monocular Visual Odometry

3. Select as observation points those image points in the first intensity image with high linear intensity gradients and attach them (together with their intensity values) rigidly to the surface model



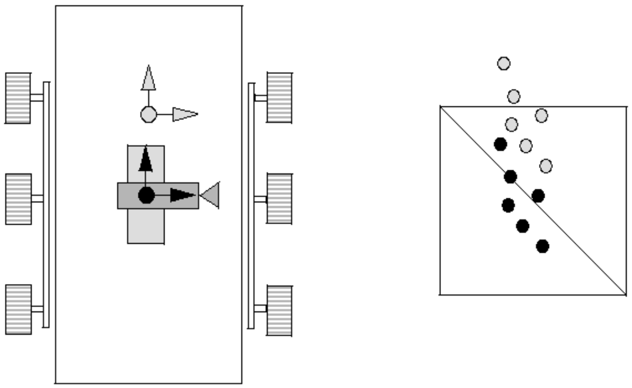
Monocular Visual Odometry

4. Allow the rover to move



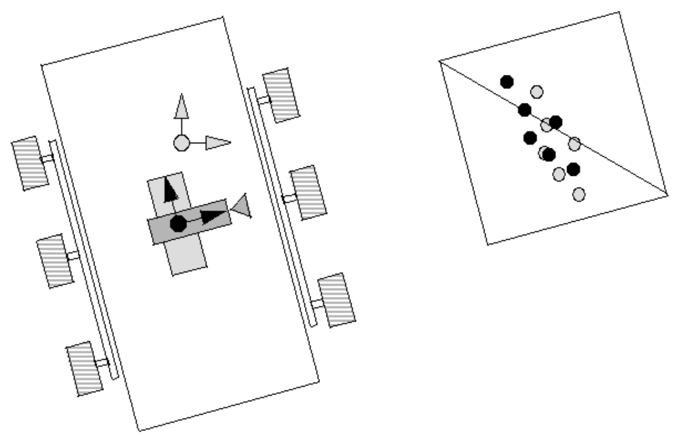
Monocular Visual Odometry

4. Allow the rover to move



Monocular Visual Odometry

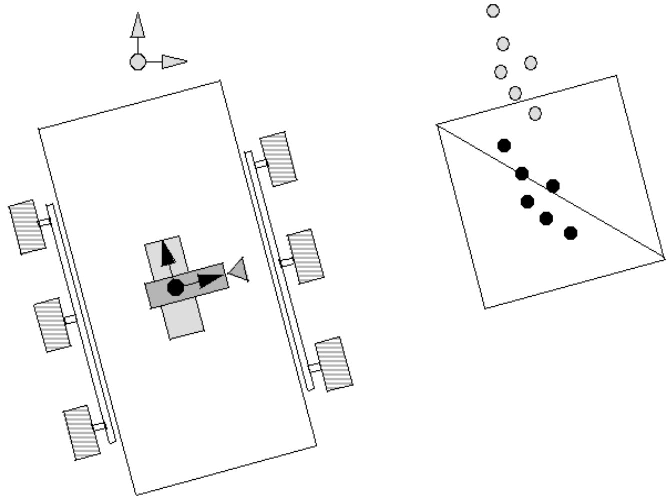
4. Allow the rover to move



IPCV-LAB robot VISION Universidad de Costa Rica

Monocular Visual Odometry

4. Allow the rover to move



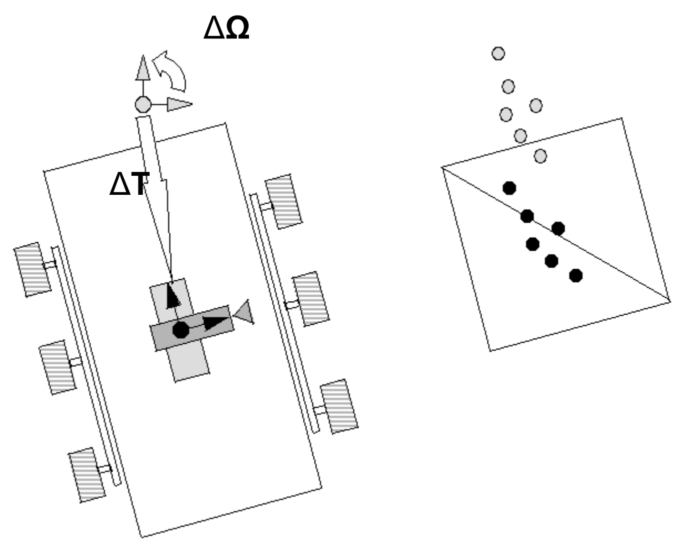
Monocular Visual Odometry

IPCV-LAB robot VISION Universidad de Costa Rica

4. Allow the rover to move

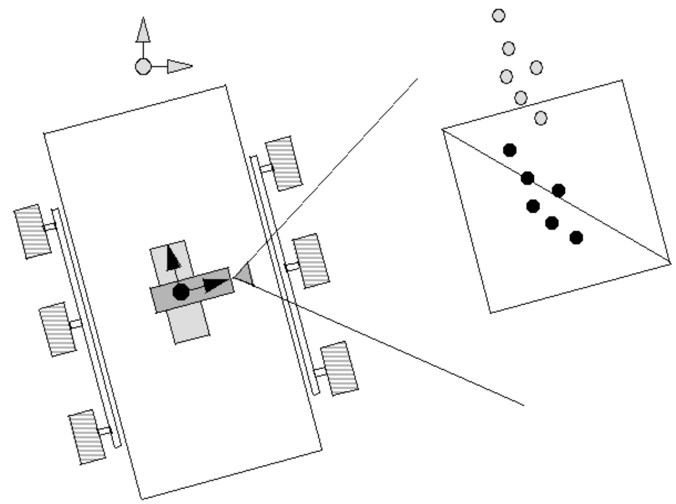
6 motion parameters:

B=(**ΔT**, **ΔΩ**)^T



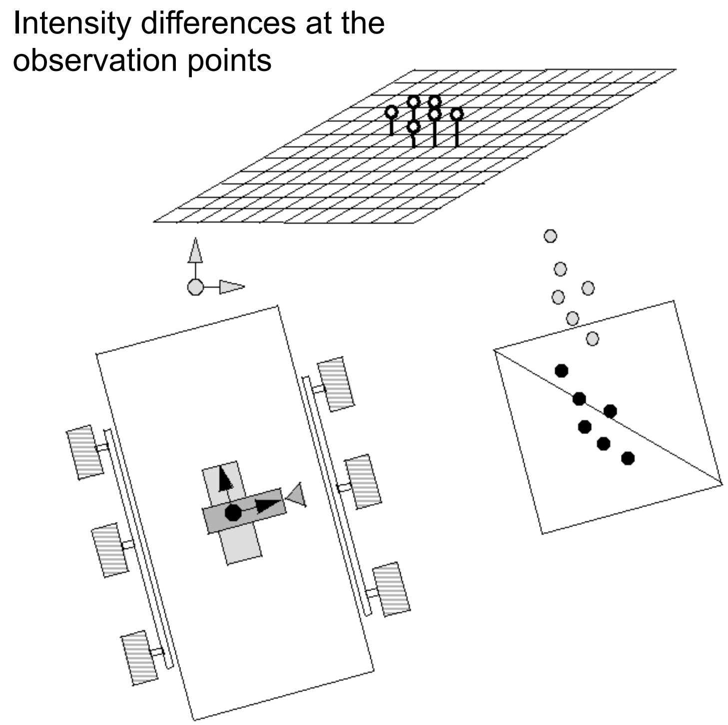
Monocular Visual Odometry

5. Capture a second intensity image after rover´s motion



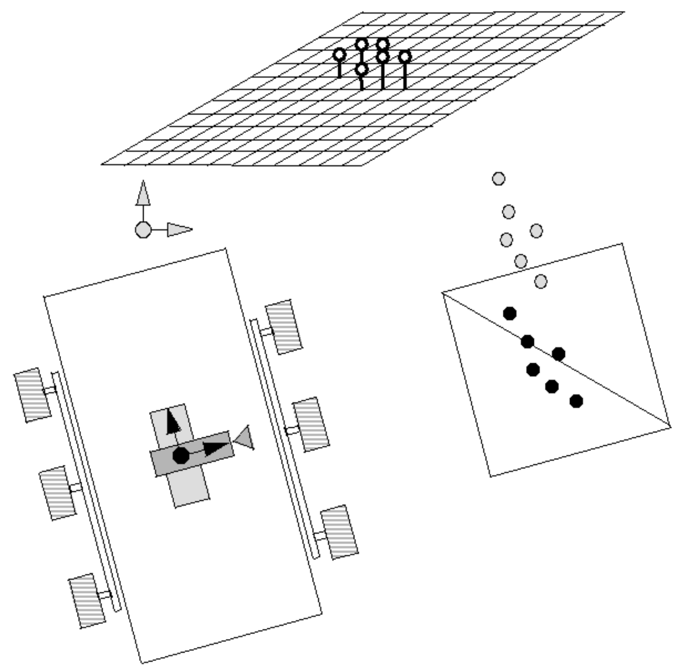
Monocular Visual Odometry

6. Project the observation points into the image plane and compute the intensity differences between their intensity values and the second intensity image



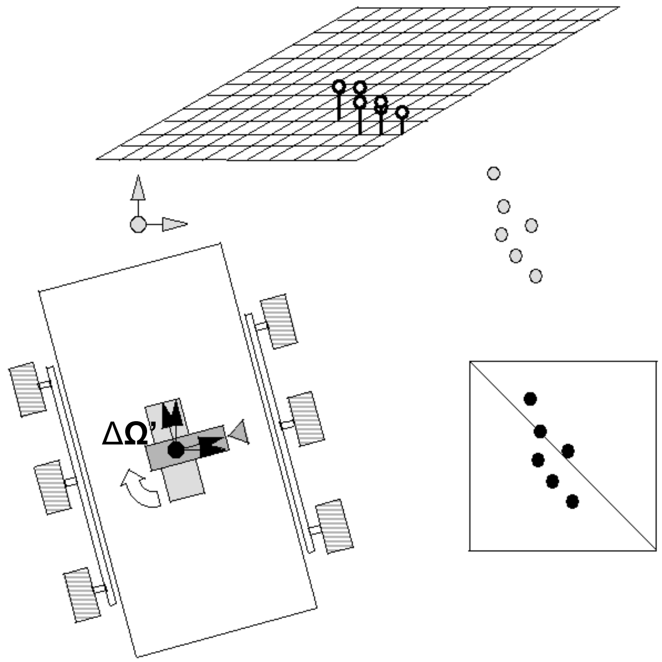
Monocular Visual Odometry

7. Search for those parameters **B'** that move the surface model (and therefore the rigidly attached observation points) to that place where the intensity differences become as small as possible



Monocular Visual Odometry

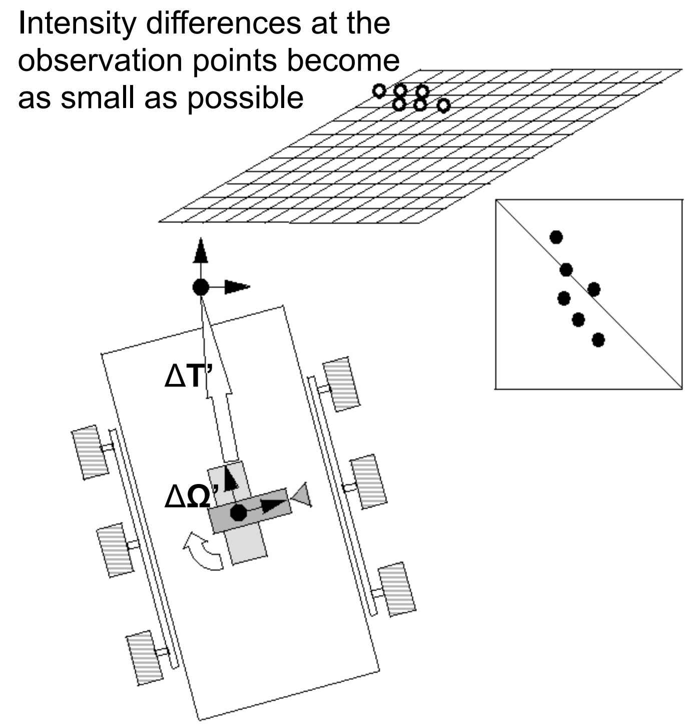
7. Search for those parameters **B’** that move the surface model (and therefore the rigidly attached observation points) to that place where the intensity differences become as small as possible



IPCV-LAB robot VISION Universidad de Costa Rica

Monocular Visual Odometry

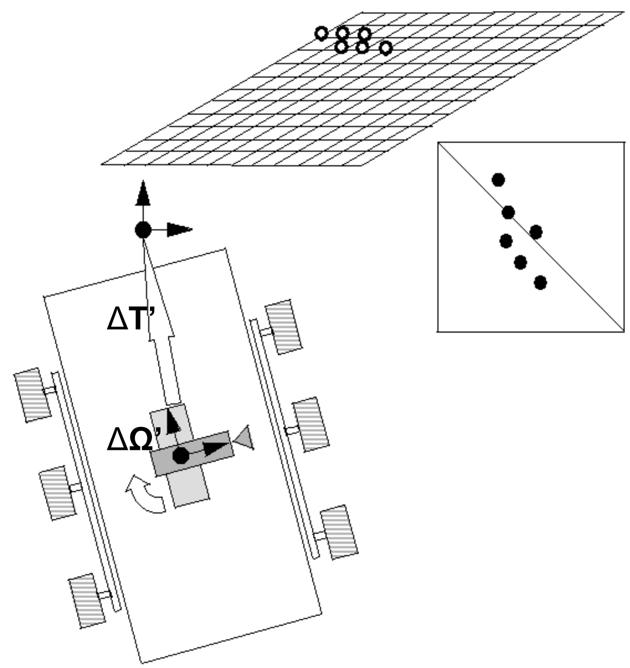
7. Search for those parameters **B’** that move the surface model (and therefore the rigidly attached observation points) to that place where the intensity differences become as small as possible



The parameters **B’** are searched by maximizing a likelihood function of the intensity differences at the observation points

Monocular Visual Odometry

7. Search for those parameters **B’** that move the surface model (and therefore the rigidly attached observation points) to that place where the intensity differences become as small as possible

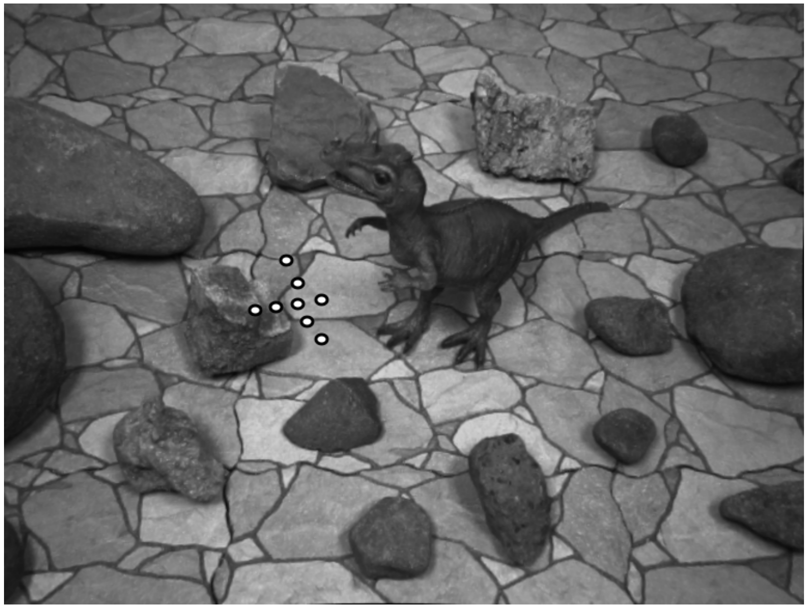


The rover’s motion estimates are $\hat{\mathbf{B}}$**=-B’**

Problem

Assumption of flat terrain

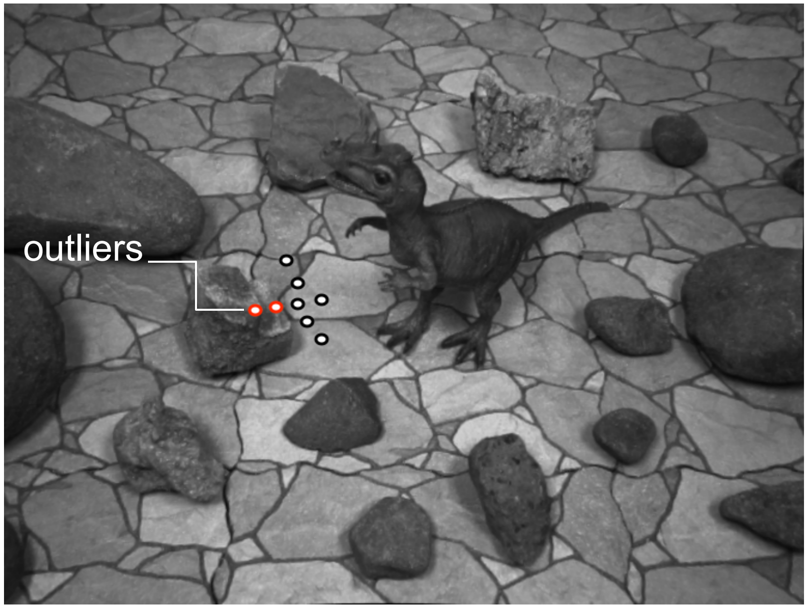
- The observation points over non-flat terrain parts such as stones generate intensity differences, which affect rover's 3D motion estimation
- This in turn deteriorates the positioning reliability performed by the algorithm



Example of captured image of terrain with non-planar parts such as stones

Approach

Detect the observation points over non-flat terrain parts (outliers) and exclude them from rover's 3D motion estimation

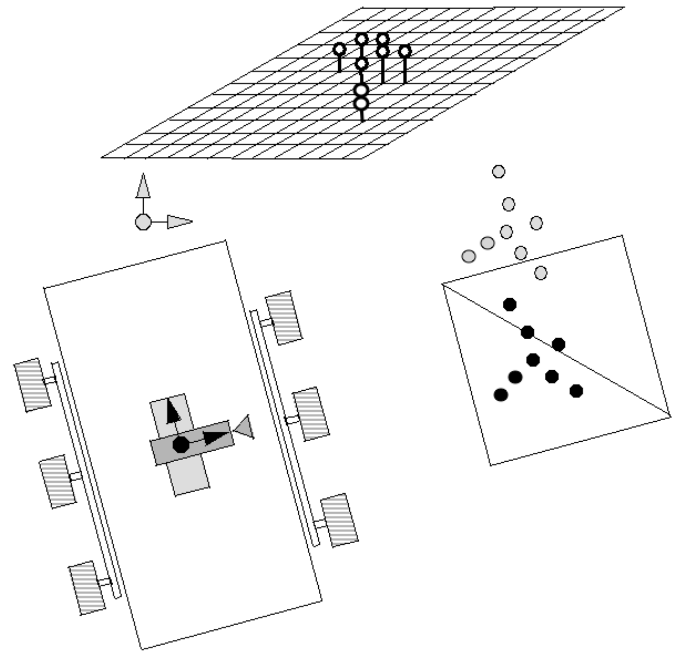


Example of captured image of terrain with non-planar parts such as stones

Approach

For outlier detection:

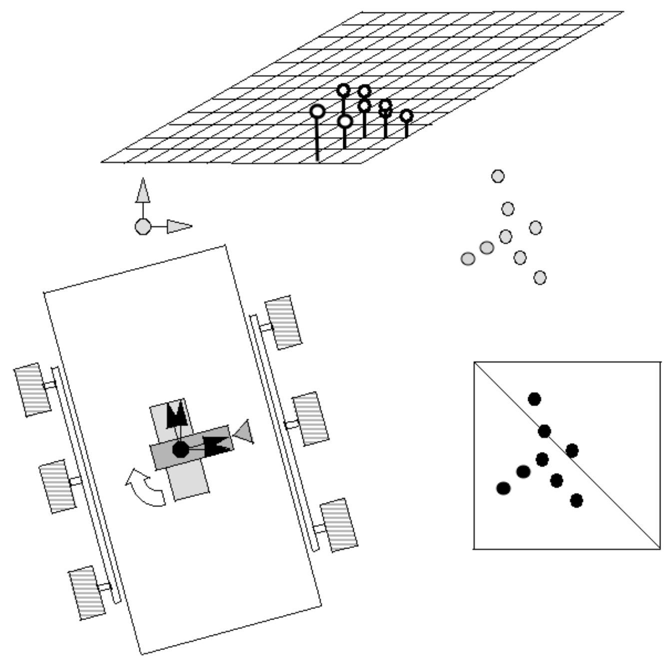
- Take into account that the intensity differences over non-flat terrain parts can not be described (reduced to zero) by motion compensation with the estimated rover`s 3D motion estimates



Approach

For outlier detection:

- Take into account that the intensity differences over non-flat terrain parts can not be described (reduced to zero) by motion compensation with the estimated rover`s 3D motion estimates

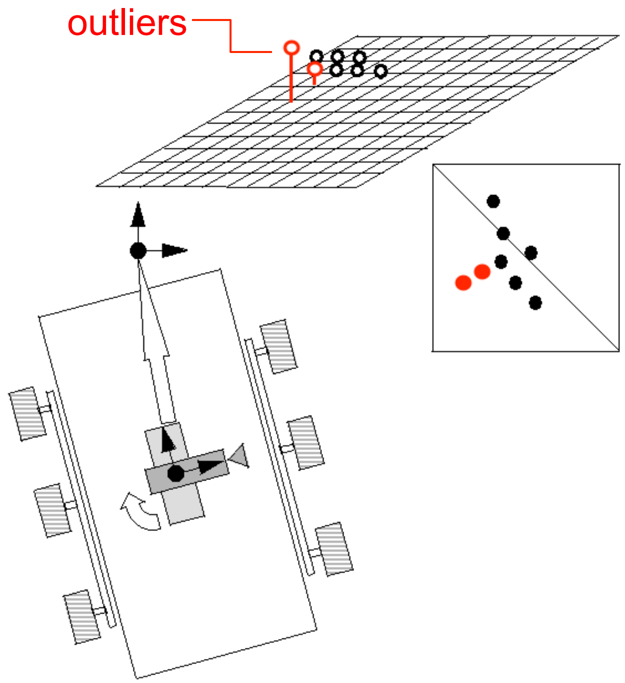


IPCV-LAB robot VISION Universidad de Costa Rica

Approach

For outlier detection:

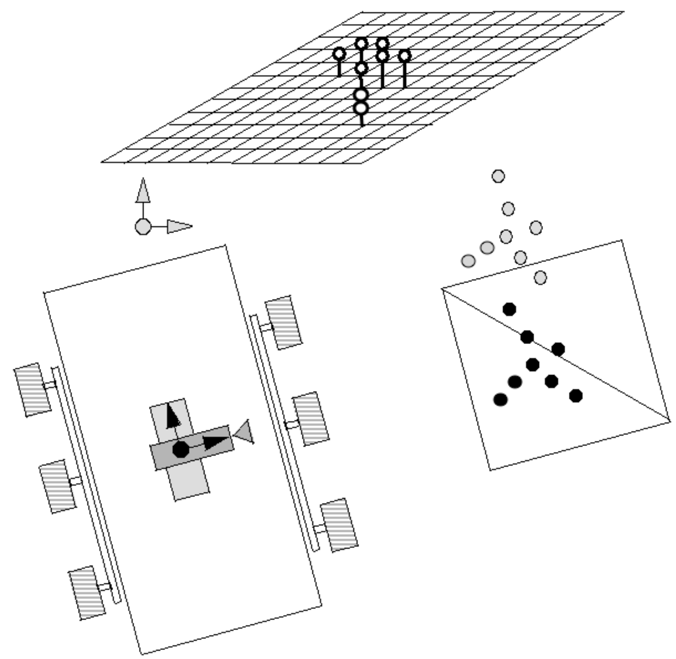
- Take into account that the intensity differences over non-flat terrain parts can not be described (reduced to zero) by motion compensation with the estimated rover`s 3D motion estimates



Example of captured image of terrain with non-planar parts such as stones

Outlier Detection Algorithm

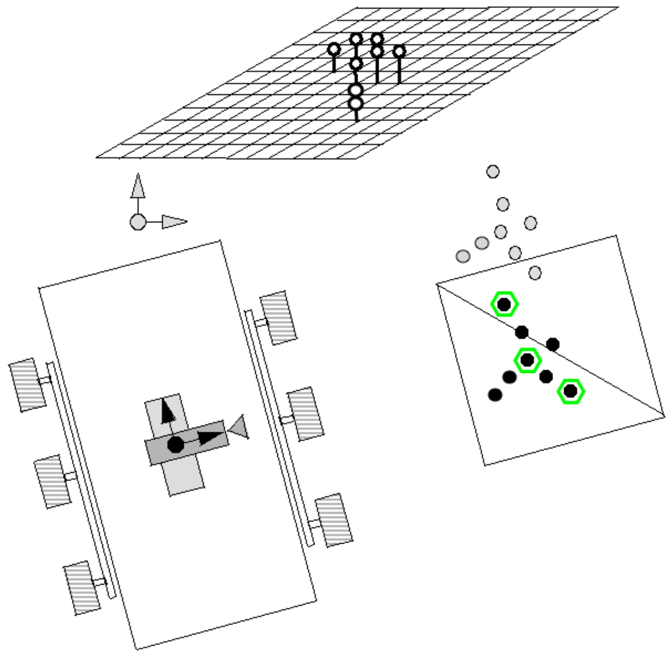
1. Randomly select a reduce amount of observation points



Based on RANSAC

Outlier Detection Algorithm

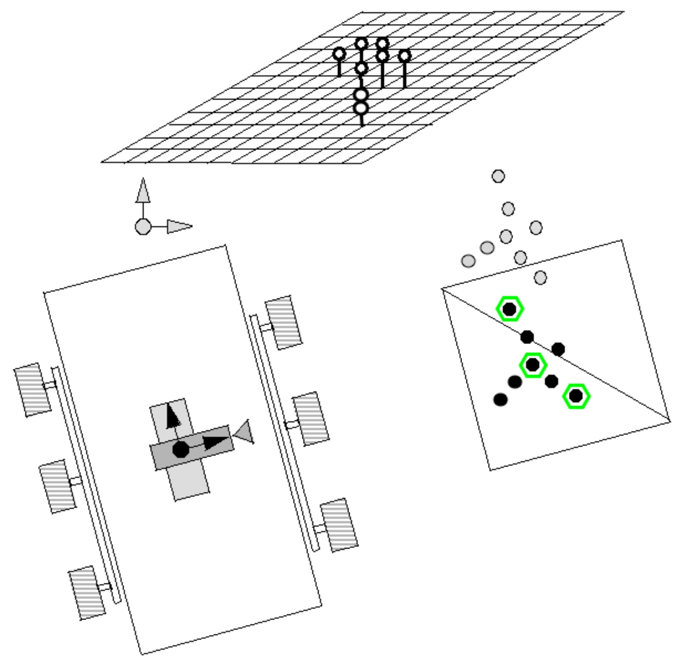
1. Randomly select a reduce amount of observation points



Based on RANSAC

Outlier Detection Algorithm

2. Estimate the rover´s 3D motion with randomly selected subset of observation points

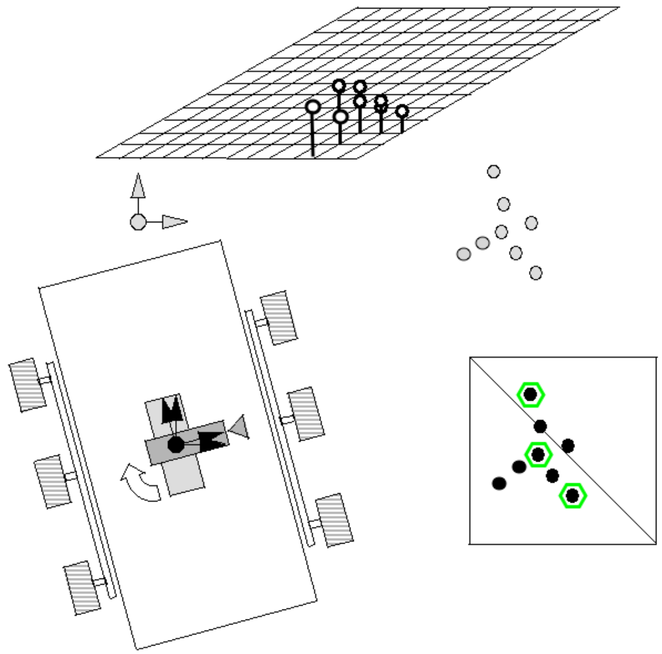


Based on RANSAC

IPCV-LAB robot VISION Universidad de Costa Rica

Outlier Detection Algorithm

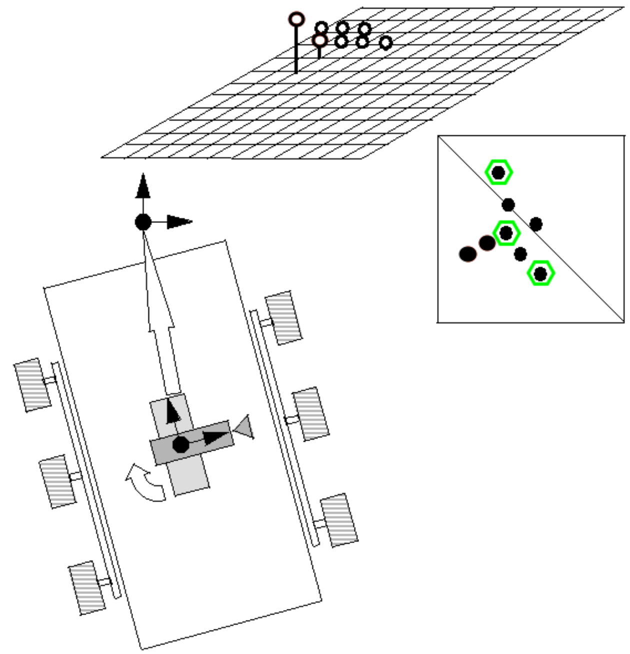
2. Estimate the rover´s 3D motion with randomly selected subset of observation points



Based on RANSAC

Outlier Detection Algorithm

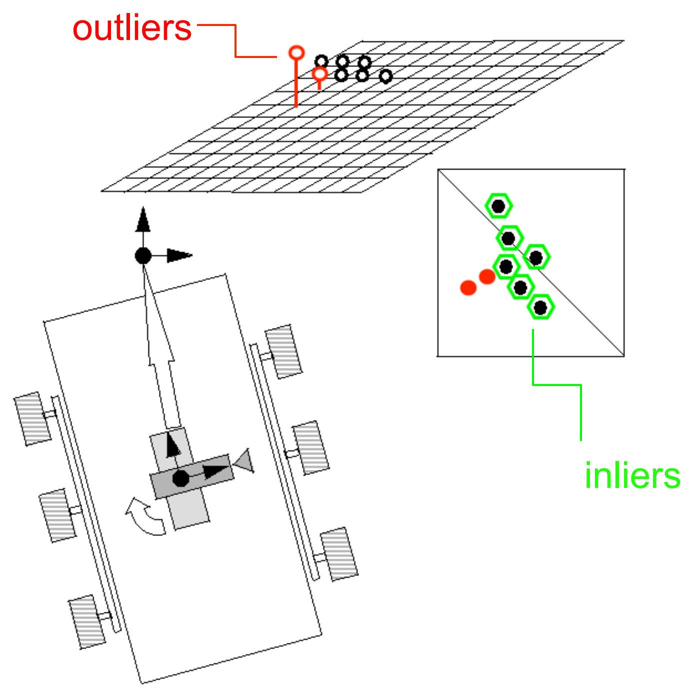
2. Estimate the rover´s 3D motion with randomly selected subset of observation points



Based on RANSAC

Outlier Detection Algorithm

3. Detect as outlier those observation points whose intensity difference could not be compensated (reduced to zero) with the estimated rover's 3D motion. The rest represent the inliers.

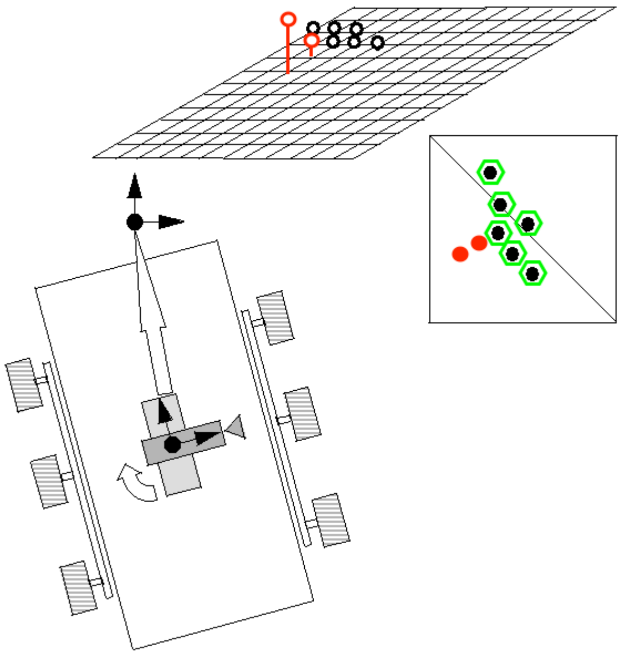


Example of captured image of terrain with non-planar parts such as stones

Based on RANSAC

Outlier Detection Algorithm

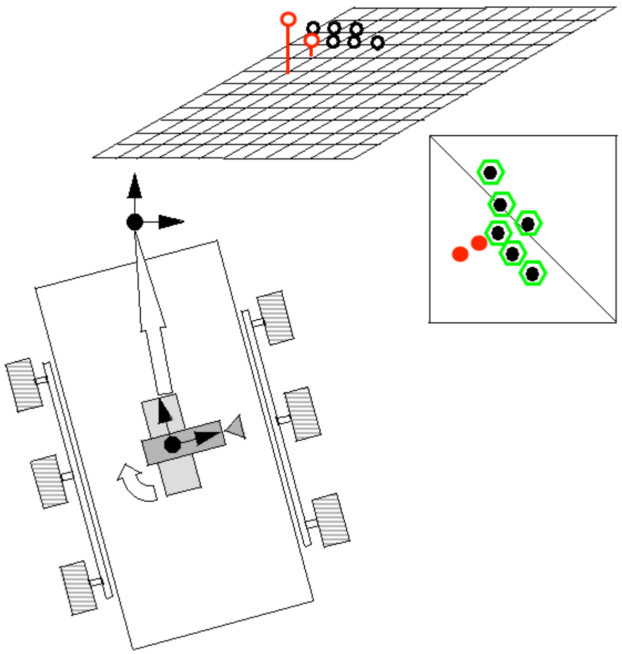
4. End the algorithm if the number of inliers is more than 90% of the total number of observation points; otherwise iterate it with a different subset of randomly selected observation points.



Based on RANSAC

Outlier Detection Algorithm

If the maximum number of iterations is reached (3 in our experiments), use the iteration subset with the largest amount of inliers.



Based on RANSAC

Experimental Results

- Implemented in C
- Tested in a real rover platform (Husky A200)
- 50 experiments
  - Over flat paver sidewalks with stones
- Performance measure
  - Absolute position error of distance traveled

Clearpath Robotics Husky A200 rover platform and Trimble S3 robotic total station used for the experiments.

Experimental Results

- Paths and velocity
  - Straight paths
  - 2 cm/sec constant velocity
- Video Camera
  - Rigidly attached to the rover
  - 15 fps
  - 77 cm above the ground
  - Looking to the left side of the rover tilted downwards 37°
  - 640x480 pixel^2, 43° FOV

Captured intensity image

IPCV-LAB robot VISION Universidad de Costa Rica

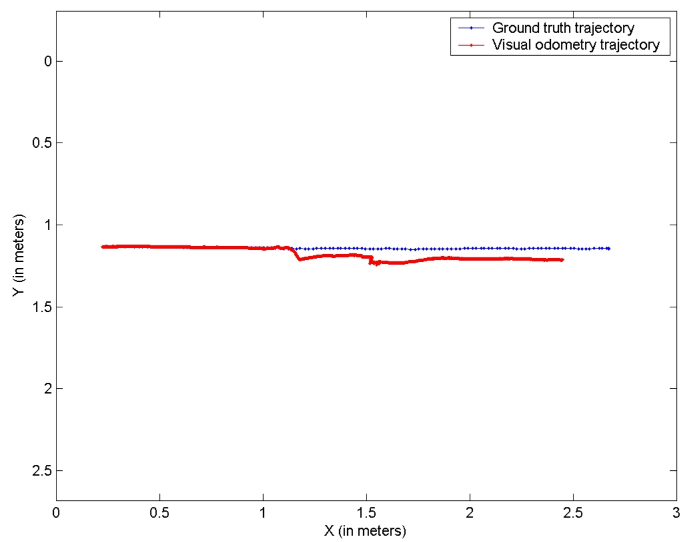
Experimental Results

- Ground truth
  - Robotic total station
  - Tracks a prism rigidly attached to the rover
  - Delivers its 3D position with high precision (<5 mm) every second
- Comparison of ground truth trajectory and visual odometry trajectories with and without outlier detection

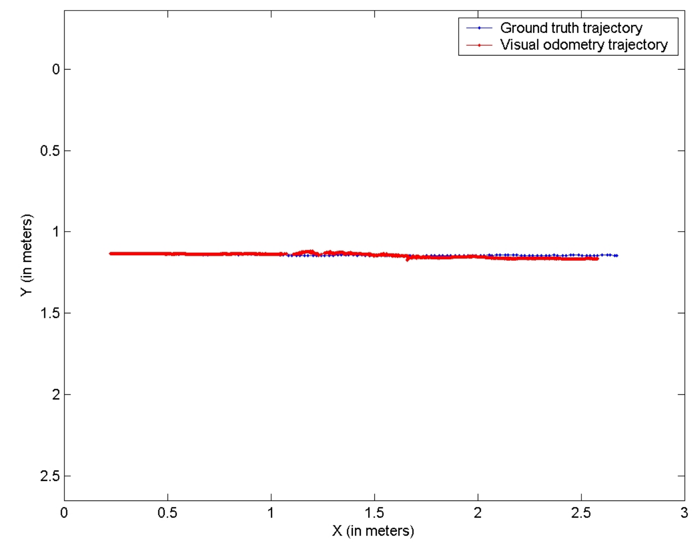
IPCV-LAB robot VISION Universidad de Costa Rica

Experimental Results

14785 observation points	Reduction by a factor of 10 →	1401 observation points
4.7% of distance traveled	Reduction by a factor of 3 →	1.8% of distance traveled
0.08 sec/image	Increased by a factor of 3 → no real time operation possible	0.21 sec/image



Without outlier detection



With outlier detection

Experimental Results

Outliers in red and inliers in green

Summary

- To improve robustness in scenes of irregular terrain, outliers are detected in the observation points, and excluded from rover's 3D motion estimation.

- Outlier detection based on RANSAC.

- Observation points detected as outliers if their intensity differences can not be described by motion compensation.

- Number of observation points reduced by a factor of 10.
- Positioning error reduced by a factor of 3.
- Processing time increased by a factor of 3 (no real time operation possible).

IPCV-LAB robot VISION Universidad de Costa Rica

Future work

- The algorithm will continue to be tested with other types of irregular surfaces. Our final goal is to make it work quite well no matter what the irregular surface type of the scene.
- Efforts will be made to get it to operate again in real time.
- New surface model will be proposed, where the surface may be irregular, described by a mesh of more than 2 triangles, and non-coplanar vertices

Thanks!

Any question?