Extending the Measurement Error Model of a Direct Visual Odometry Algorithm to Improve its Accuracy for Planetary Rover Navigation

Geovanni Martínez

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

November 28th, 2019

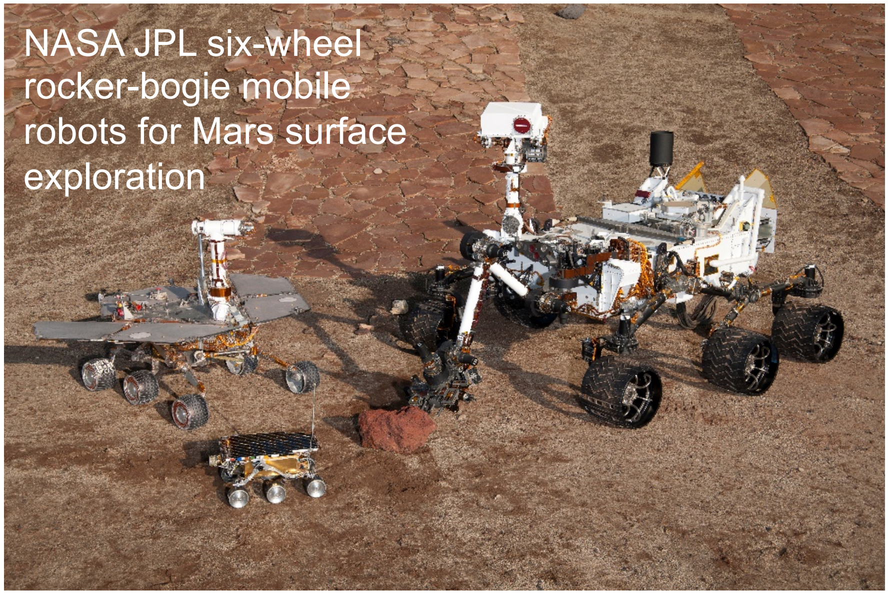
IEEE iCASAT-2019, November 27-29, 2019, Queretaro, Mexico

Overview

- Introduction
- Monocular visual odometry algorithm
- Problem
- Approach
- Extended measurement error model
- Results
- Summary

Introduction

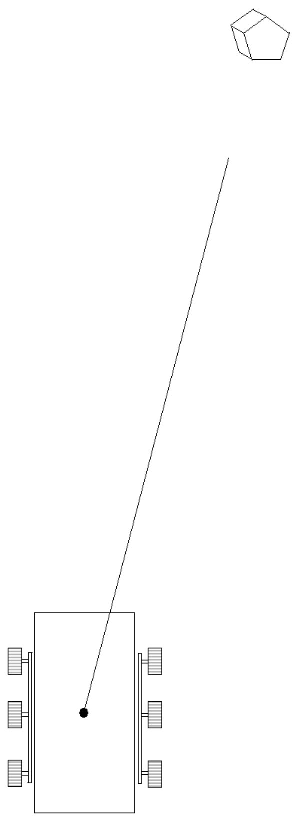
- One important feature of these planetary exploration rovers is their ability to navigate autonomosly



Courtesy NASA/JPL-Caltech

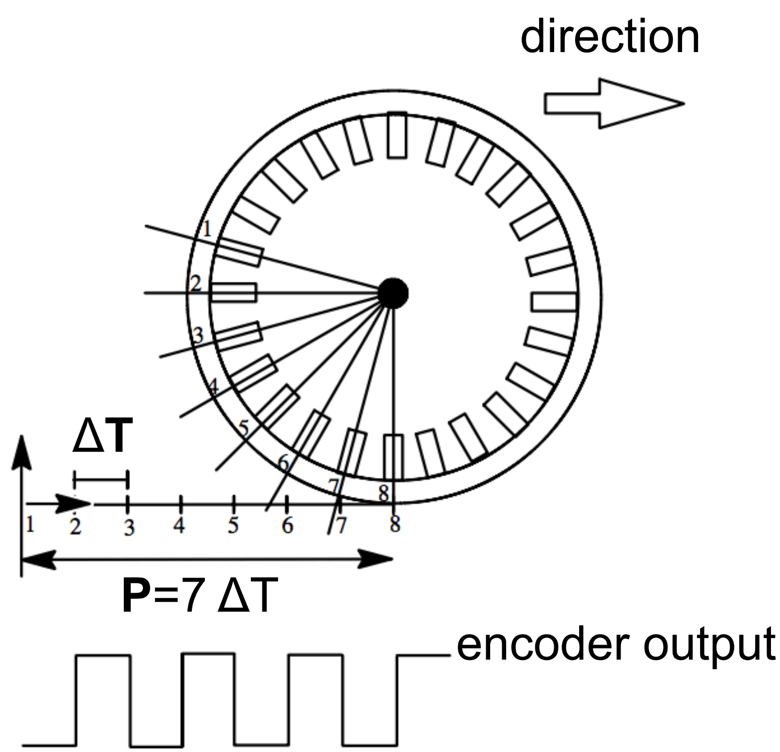
Introduction

- If you want a rover to navigate autonomously in a precise way, then the rover must know its 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)



Introduction

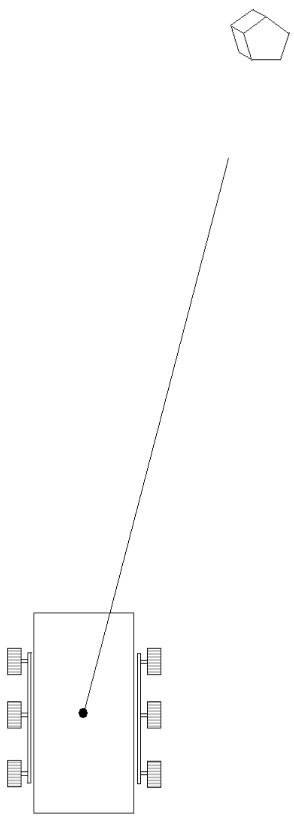
IPCV-LAB robot VISION Universidad de Costa Rica

- Wheel odometry fails on slippery environments, such as sandy terrain, due to the loss of traction

Courtesy NASA/JPL-Caltech

Introduction

- This could cause the rover to deviate from its desired path

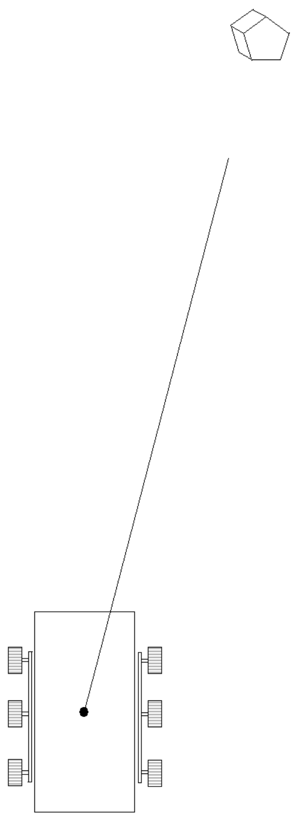


Introduction

The problem is solved by detecting and compensating any slip that may occur

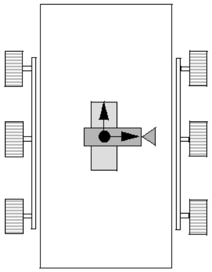
- Compute the rover's position and orientation by applying a **monocular visual odometry algorithm**

- Knowing the rover's position and orientation, detect and compensate any slip that may occur



Monocular Visual Odometry

Algorithm



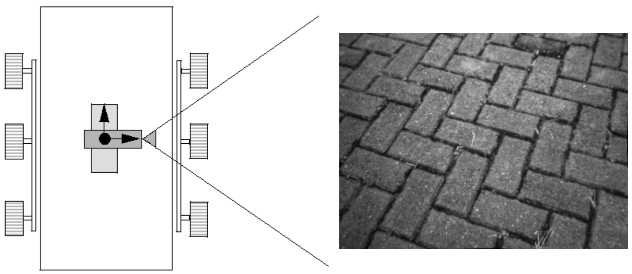
- Estimate the rover´s motion **B** from the video signal delivered by a single camera mounted on the rover

- Compute the rover’s position and orientation by accumulating the motion estimates **B** over time

…in following detailed stepwise description

Monocular Visual Odometry

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

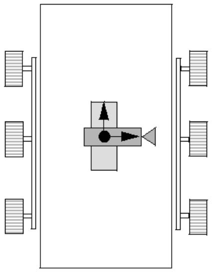


Example of captured image of planar terrain

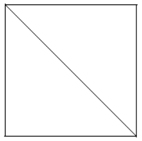
IPCV-LAB robot VISION Universidad de Costa Rica

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

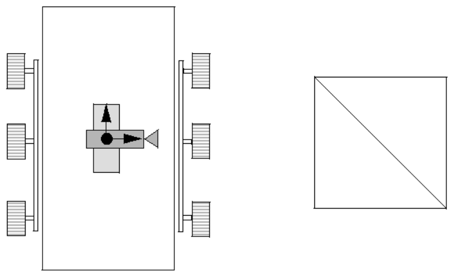


This assumption affects the performance of the algorithm in irregular terrain

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

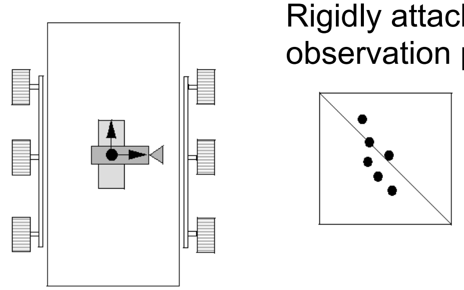
Monocular Visual Odometry

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



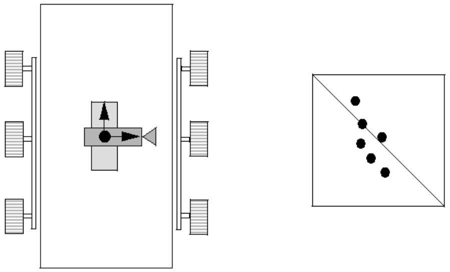
Monocular Visual Odometry

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

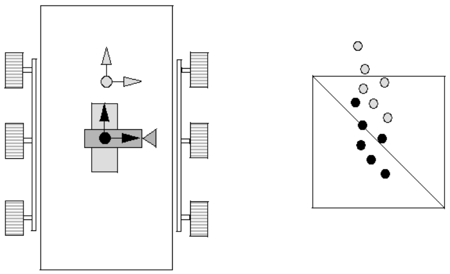
4. Allow the rover to move



IPCV-LAB robot VISION Universidad de Costa Rica

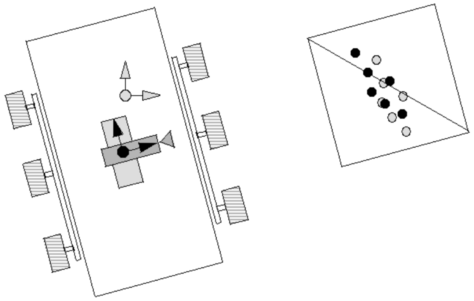
Monocular Visual Odometry

4. Allow the rover to move



Monocular Visual Odometry

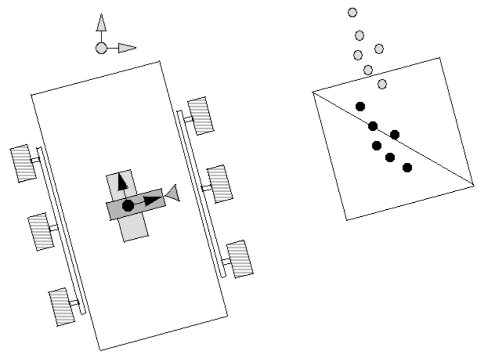
4. Allow the rover to move



IPCV-LAB robot VISION Universidad de Costa Rica

Monocular Visual Odometry

4. Allow the rover to move

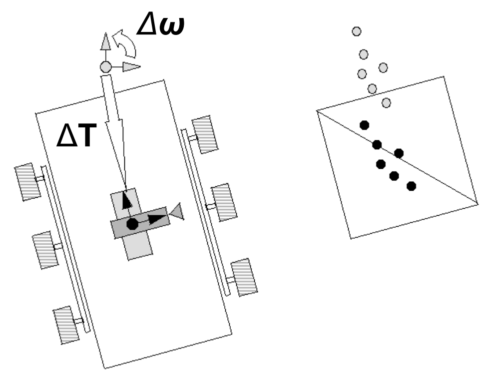


Monocular Visual Odometry

4. Allow the rover to move

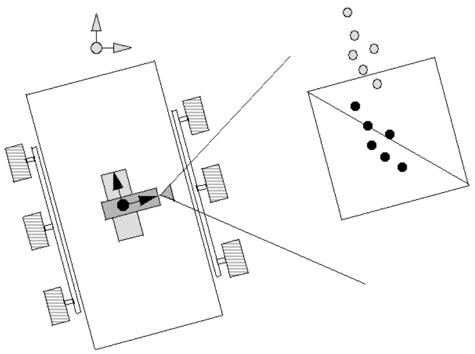
   6 motion parameters:

   $\boldsymbol{B}=(\Delta \boldsymbol{T}, \Delta \boldsymbol{\omega})^T$



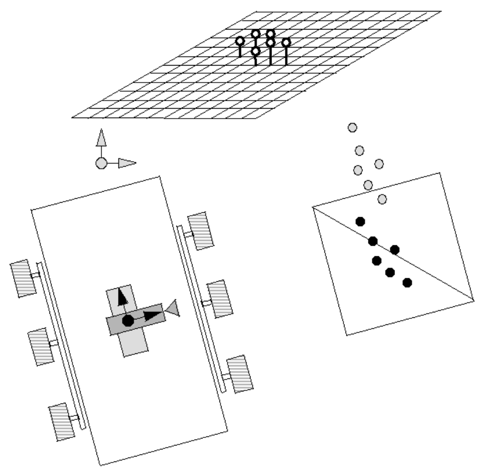
Monocular Visual Odometry

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



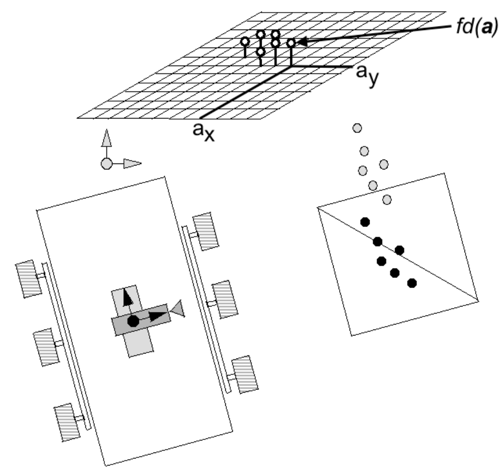
Monocular Visual Odometry

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



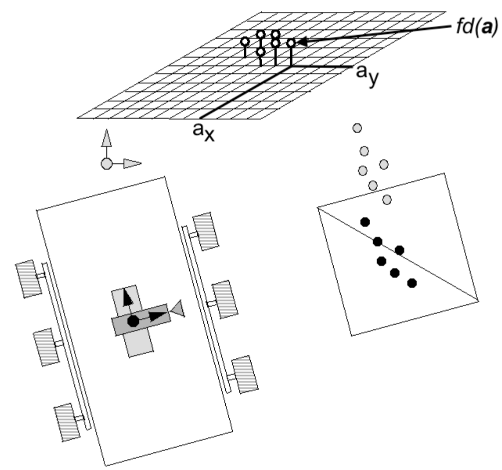
Monocular Visual Odometry

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. Evaluate the linear relationship between intensity difference and rover motion parameters at *N>>6* observation points

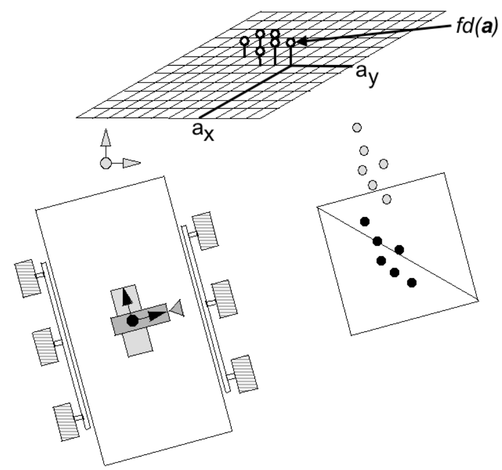


$$fd(\boldsymbol{a}_{N-1}) = \boldsymbol{o}_{N-1}^T \boldsymbol{B} + \Delta I_{N-1}$$
$$\vdots$$
$$fd(\boldsymbol{a}_n) = \boldsymbol{o}_n^T \boldsymbol{B} + \Delta I_n$$
$$\vdots$$
$$fd(\boldsymbol{a}_0) = \boldsymbol{o}_0^T \boldsymbol{B} + \Delta I_0$$

$$\boldsymbol{FD} = \boldsymbol{O} \cdot \boldsymbol{B} + \boldsymbol{V}$$

Monocular Visual Odometry

8 Solve the over determined system of linear equations by applying a Maximum Likelihood estimator, where ***U*** is the covariance matrix of the measurement error at the observation points



$$\hat{\boldsymbol{B}} = [\boldsymbol{O}^T\boldsymbol{U}^{-1}\boldsymbol{O}]^{-1}\boldsymbol{O}^T\boldsymbol{U}^{-1}\boldsymbol{F}\boldsymbol{D}$$

where

$$\boldsymbol{U} = E[\boldsymbol{V}\,\boldsymbol{V}^T]$$

Monocular Visual Odometry

Measurement error model

ΔI_n is assumed to be due to the camera noise only and Gaussian distributed with

$$\mu_n = 0 \quad \sigma_n^2 = \sigma_{noise}^2 = 1$$

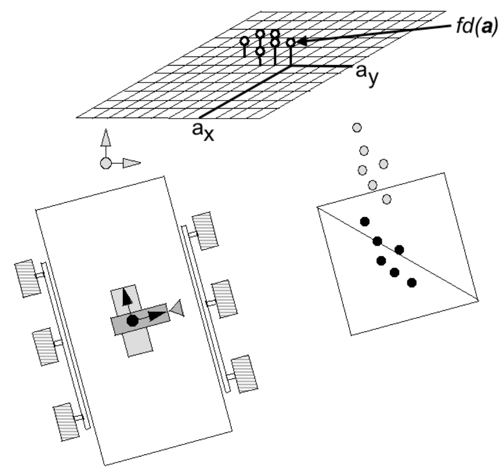
In addition, $\Delta I_1, \Delta I_2, \dots, \Delta I_n, \dots, \Delta I_N$ are supposed to be statistically independent

then

$$\boldsymbol{U} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = \boldsymbol{I}$$

Monocular Visual Odometry

Set ***U=I*** in the estimator to finally get the motion estimates $\hat{\mathbf{B}}$



$$\hat{\boldsymbol{B}} = [\boldsymbol{O}^T \boldsymbol{I}^{-1} \boldsymbol{O}]^{-1} \boldsymbol{O}^T \boldsymbol{I}^{-1} \boldsymbol{F}\boldsymbol{D}$$

$$\hat{\boldsymbol{B}} = [\boldsymbol{O}^T \boldsymbol{O}]^{-1} \boldsymbol{O}^T \boldsymbol{F}\boldsymbol{D}$$

Note: due to the linearizations needed to obtained $fd(a)=\mathbf{o}^T\mathbf{B}+\Delta I$, the estimator must be applied iteratively

IPCV-LAB robot VISION Universidad de Costa Rica

Monocular Visual Odometry

9. Compute the rover's position and orientation by accumulating the motion estimates $\widehat{\mathbf{B}}$ over time

$$\boldsymbol{P}_k = \sum_{\kappa=1}^{k} \Delta\widehat{\boldsymbol{T}}_{(\kappa-1)\to\kappa}$$

$$\boldsymbol{R}_k = \Delta\boldsymbol{R}\left|\begin{matrix} \Delta\widehat{\omega}_{X,(k-1)\to k} \\ \Delta\widehat{\omega}_{Y,(k-1)\to k} \\ \Delta\widehat{\omega}_{Z,(k-1)\to k} \end{matrix}\right. \cdots \Delta\boldsymbol{R}\left|\begin{matrix} \Delta\widehat{\omega}_{X,0\to 1} \\ \Delta\widehat{\omega}_{Y,0\to 1} \\ \Delta\widehat{\omega}_{Z,0\to 1} \end{matrix}\right.$$

Problem

- Due to the assumption of flat terrain the intensity difference measurement error also depends on measurement error due to the 3D shape error between the assumed flat surface shape and the true planetary surface shape
- Its variance is not longer constant but variable at each observation point

$$\Delta I_n \neq \Delta I_{noise}$$

Approach

To extend the measurement error model to also consider the measurement error due to the 3D shape error

This will improve the accuracy of rover's motion estimation

$$\Delta I_n = \Delta I_{noise}^{(n)} + \Delta I_{\Delta a_n}$$

$$\mu_n = 0 \quad \sigma_n^2 = 1 + \sigma_{\Delta I_{\Delta a_n}}^2$$

$$\boldsymbol{U}_{new} = \begin{bmatrix} 1 + \sigma_{\Delta I_{\Delta a_{N-1}}}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 + \sigma_{\Delta I_{\Delta a_0}}^2 \end{bmatrix}$$

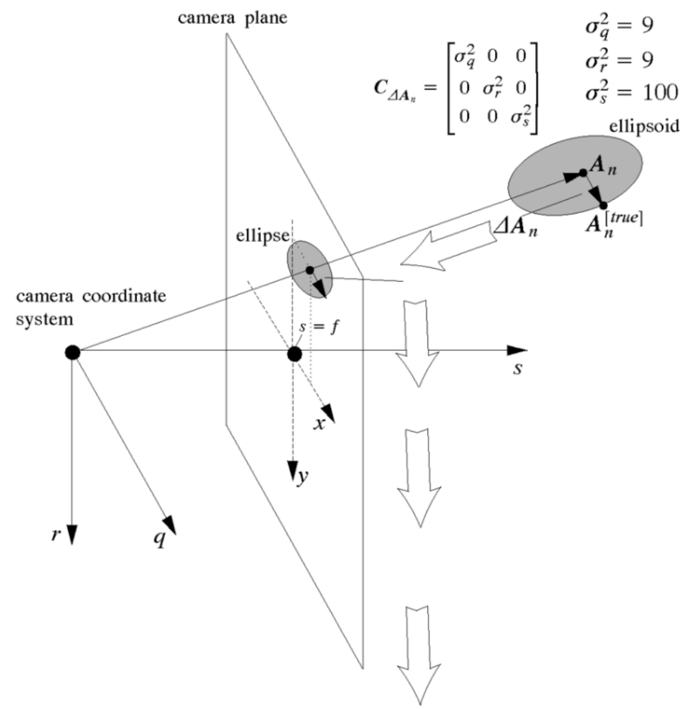
In this paper a method to compute $\sigma_{\Delta I_{\Delta a_n}}^2$ at each observation point *n* is presented

For 3D motion estimation instead of $\widehat{\boldsymbol{B}} = [\boldsymbol{O}^T \boldsymbol{O}]^{-1} \boldsymbol{O}^T \boldsymbol{F} \boldsymbol{D}$ the following will be used

$$\widehat{\boldsymbol{B}} = [\boldsymbol{O}^T \boldsymbol{U}_{new}^{-1} \boldsymbol{O}]^{-1} \boldsymbol{O}^T \boldsymbol{U}_{new}^{-1} \boldsymbol{F} \boldsymbol{D}$$

New covariance matrix U_{new}

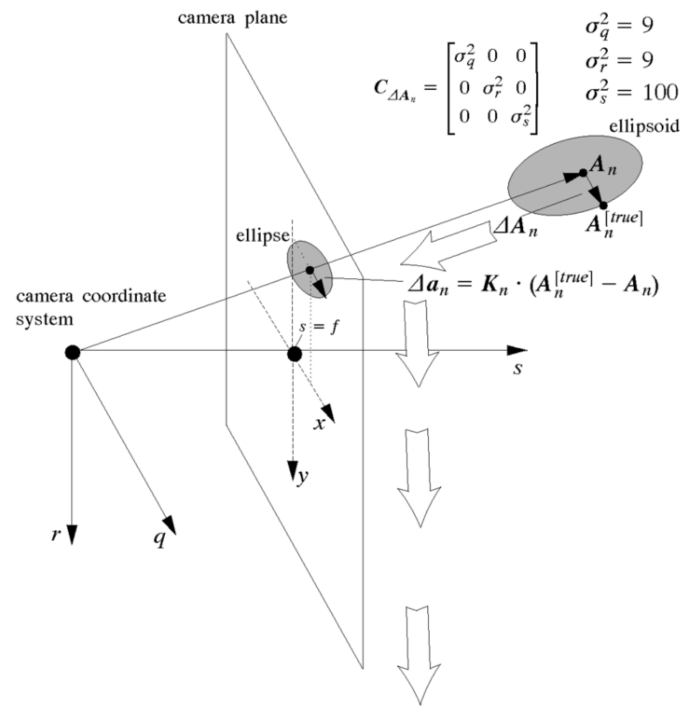
Steps for computing U_{new}



New covariance matrix U_{new}

Steps for computing U_{new}

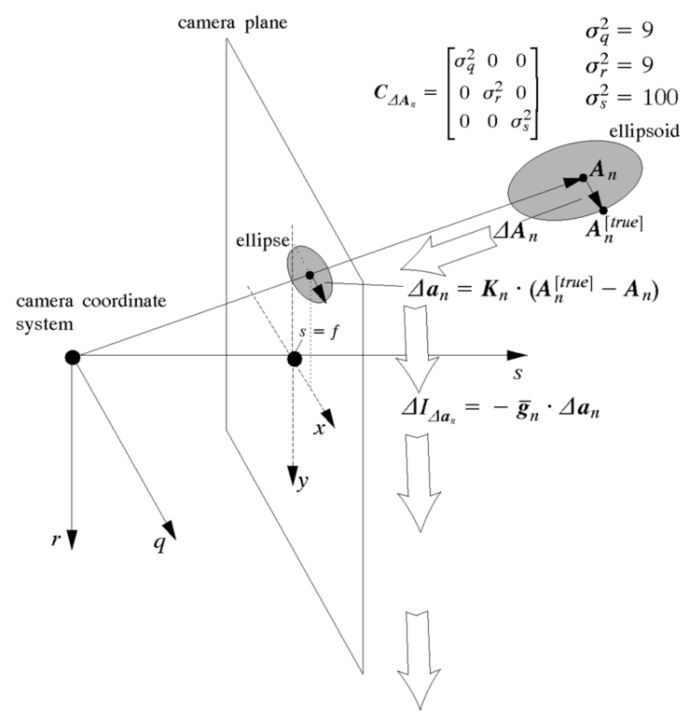
1. Transform the 3D shape error into a 2D position error on the camera plane



New covariance matrix U_{new}

Steps for computing U_{new}

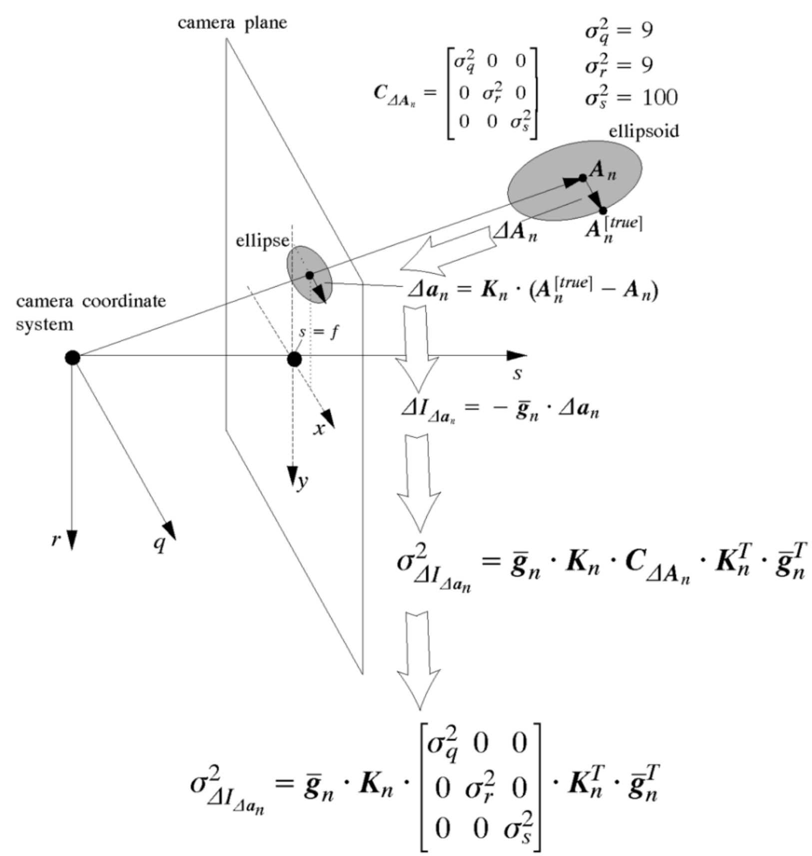
1. Transform the 3D shape error into a 2D position error on the camera plane
2. Transform the 2D position error into a intensity difference measurement error



New covariance matrix $\boldsymbol{U}_{new}$

Steps for computing $\boldsymbol{U}_{new}$

1. Transform the 3D shape error into a 2D position error on the camera plane
2. Transform the 2D position error into a intensity difference measurement error
3. Get the corresponding variance



New covariance matrix U_{new}

Steps for computing U_{new}

1. Transform the 3D shape error into a 2D position error on the camera plane
2. Transform the 2D position error into a intensity difference measurement error
3. Get the corresponding variance and covariance matrix

$$\sigma^2_{\Delta I_{\Delta a_n}} = \bar{\boldsymbol{g}}_n \boldsymbol{K}_n \boldsymbol{C}_{\Delta A_n} \boldsymbol{K}_n^T \bar{\boldsymbol{g}}_n^T$$

$$\boldsymbol{U}_{new} = \begin{bmatrix} 1+\sigma^2_{\Delta I_{\Delta a_{N-1}}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1+\sigma^2_{\Delta I_{\Delta a_0}} \end{bmatrix}$$

New covariance matrix U_{new}

Steps for computing U_{new}

1. Transform the 3D shape error into a 2D position error on the camera plane
2. Transform the 2D position error into a intensity difference measurement error
3. Get the corresponding variance and covariance matrix
4. Evaluate the U_{new} en el estimador

$$\sigma^2_{\Delta I_{\Delta a_n}} = \bar{\boldsymbol{g}}_n \boldsymbol{K}_n \boldsymbol{C}_{\Delta A_n} \boldsymbol{K}_n^T \bar{\boldsymbol{g}}_n^T$$

$$\boldsymbol{U}_{new} = \begin{bmatrix} 1+\sigma^2_{\Delta I_{\Delta a_{N-1}}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1+\sigma^2_{\Delta I_{\Delta a_0}} \end{bmatrix}$$

$$\widehat{\boldsymbol{B}} = [\boldsymbol{O}^T \boldsymbol{U}_{new}^{-1} \boldsymbol{O}]^{-1} \boldsymbol{O}^T \boldsymbol{U}_{new}^{-1} \boldsymbol{F} \boldsymbol{D}$$

New covariance matrix U_{new}

Steps for computing U_{new}

1. Transform the 3D shape error into a 2D position error on the camera plane
2. Transform the 2D position error into a intensity difference measurement error
3. Get the corresponding variance and covariance matrix
4. Evaluate the U_{new} en el estimador

$$\sigma^2_{\Delta I_{\Delta a_n}} = \bar{g}_n K_n C_{\Delta A_n} K_n^T \bar{g}_n^T$$

$$U_{new} = \begin{bmatrix} 1+\sigma^2_{\Delta I_{\Delta a_{N-1}}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1+\sigma^2_{\Delta I_{\Delta a_0}} \end{bmatrix}$$

$$\hat{B} = [O^T U_{new}^{-1} O]^{-1} O^T U_{new}^{-1} FD$$

Original

$$U = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = I \rightarrow U = I$$

$$\hat{B} = [O^T U^{-1} O]^{-1} O^T U^{-1} FD$$

IPCV-LAB robot VISION Universidad de Costa Rica

Experimental Results

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

Experiment setup.

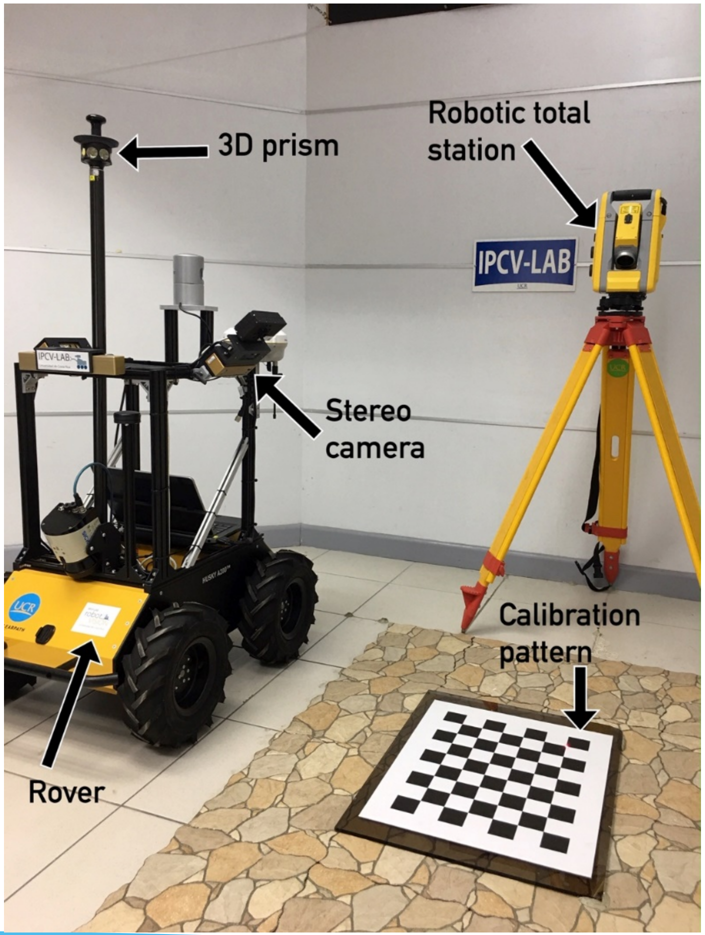
Experimental Results

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

Captured intensity image

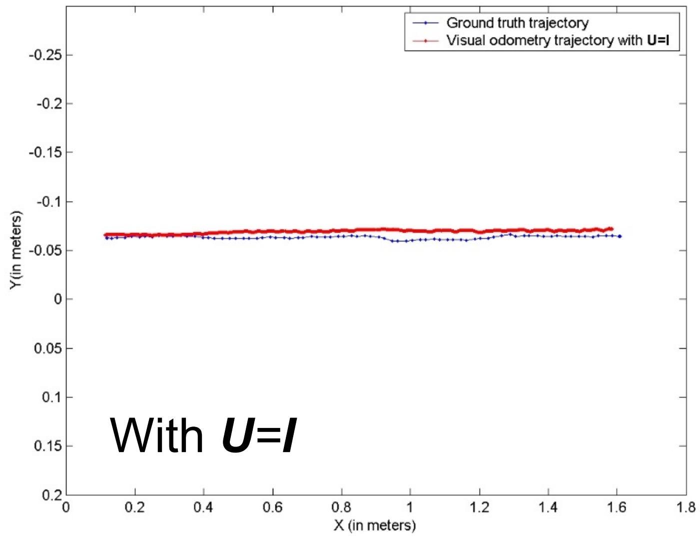
Experimental Results

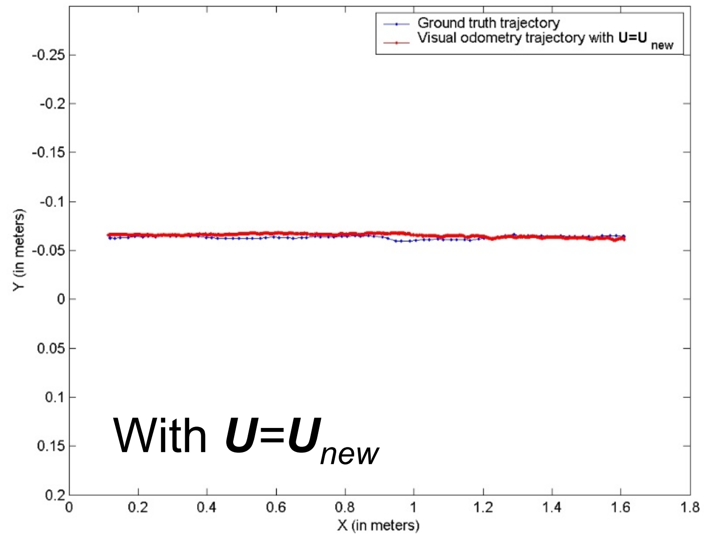
- Ground truth
  - Robotic total station
  - Tracks a 3D prism rigidly attached to the rover
  - Delivers its 3D position with high precision (<5 mm) every second
- Comparison of ground truth trajectory and visual odometry trajectories with $\boldsymbol{U}=\boldsymbol{I}$ and with $\boldsymbol{U}=\boldsymbol{U}_{new}$



Calibration setup.

Experimental Results

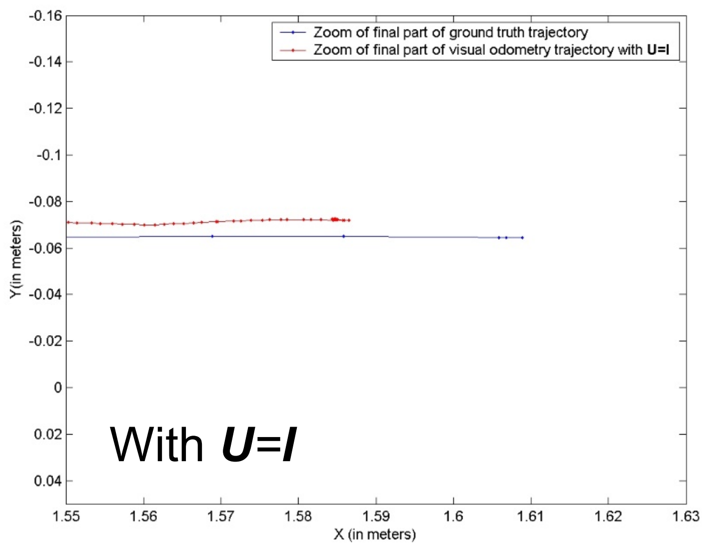


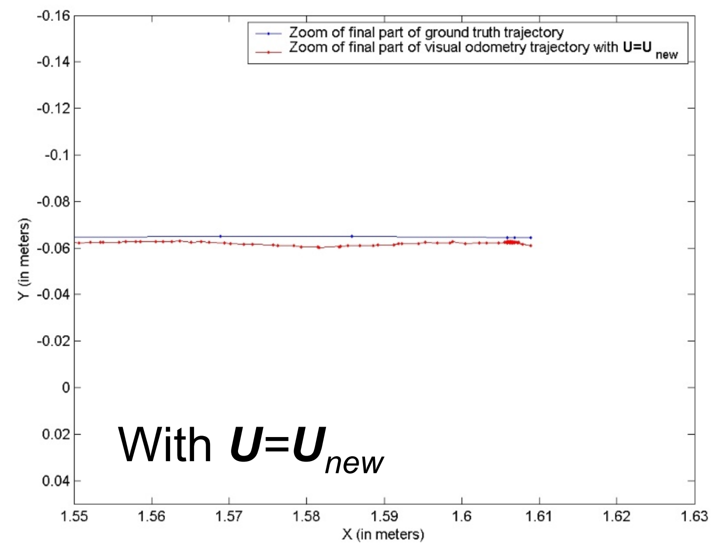


Ground truth versus visual odometry trajectories

Experimental Results

The accuracy has improved as expected





Ground truth versus visual odometry trajectories
(closer look)

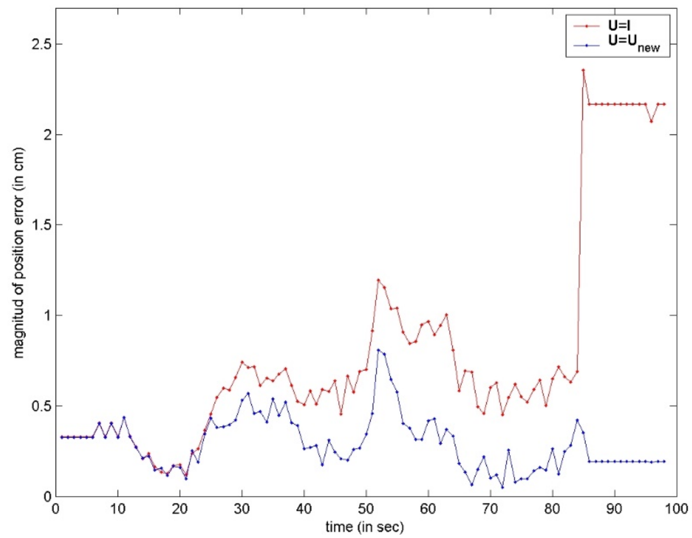
Experimental Results

on average

12 iterations	Increased by a factor of 2 →	23 iterations
0.066 sec/image	Increased by a factor of 2 →	0.1429 sec/image
1.4911% of distance traveled	Improved by a factor of 2 →	0.7295% of distance traveled

Experimental Results

The accuracy has improved as expected



Magnitude of position error

IPCV-LAB robot VISION Universidad de Costa Rica

Experimental Results

Animated surface model with negative estimated motion parameters
(It should be seen glut to the surface)

Summary

- In this paper, the accuracy of a monocular visual odometry algorithm is improved
- The improvement was achieved by extending the measurement error model to consider also the measurement error due to the 3D shape error between the assumed and the true planetary surface shape
- Positioning error decreased by a factor of 2
- Processing time increased by a factor of 2

Future work

- The experiments will continue until a version can be obtained, capable of operating in real time, with great reliability for longer trajectories, regardless of the climate and terrain

Thanks!

Any question?