Estimating the geometrical parameters of precariously balanced rocks from unconstrained digital photographs

PBR_slenderness_DH_v1_0

David E. Haddad
Active Tectonics, Quantitative Structural Geology and Geomorphology
School of Earth and Space Exploration
Arizona State University
david.e.haddad@asu.edu
August 2009

Table of Contents

Introduction

An important parameter that is used to model the rocking response of a precariously balanced rock (PBR) to ground motions is slenderness. A PBR's slenderness (α_i in Fig. 1) is defined as the angle made by the vertical (*mg* in Fig. 1) passing through the PBR's center of mass and the lines that connect the PBR's rocking points to its center of mass (R_i in Fig. 1). These values are typically estimated in the field using a plumb bob, measuring tape, and a trained eye.

Recently, photogrammetric and terrestrial laser scanning (TLS) techniques have been employed to capture the 3D geometry of a PBR at decimeter (photogrammetry) and millimeter (TLS) scales (e.g., Anooshehpoor et al., 2007; Haddad and Arrowsmith, 2009; Hudnut et al., 2009). These methods produce spectacular results that are especially useful to modeling the 3D rocking response of a PBR to ground motions (e.g., Hudnut et al., 2009). However, these techniques require a considerable amount of field preparation and data collection time and effort, making them impractical for studies that attempt to survey entire populations of PBRs at the drainage basin scale. For this, we developed a simple tool (*PBR_slenderness_DH_v1_0*) of estimating α_i and R_i from unconstrained digital photographs of PBRs. This approach was first proposed by Purvance (2005), where a photograph of a PBR is taken in the field and the outline of the PBR is digitized to compute its 2D center of mass, α_i and R_i. Theoretically, a PBR can be sectioned along an infinite number of vertical planes that pass through its center of mass along an infinite number of azimuths (Fig. 2). This results in an infinite number of local minimum slenderness values. Therefore, the azimuth at which the photograph is taken is critical to estimating the absolute minimum slenderness value of a PBR.

This document serves as a user manual to the code. The general workflow is as follows: the user digitizes the PBR's rocking points, plumb bob, scale, and the PBR's outline. The code then computes the 2D center of mass and returns α_i and R_i. A more detailed workflow with instructions and a demonstration are presented in this manual.

If you use *PBR_Slenderness_DH_v1_0* in your research, please send me any reprints of publications that used this code (email address provided above). Also, please acknowledge this manual and the software implementation document as follows:

Haddad, D. E., 2009. Estimating the geometrical parameters of precariously balanced rocks from unconstrained digital photographs, 22 p.

Haddad, D. E., 2010. Geologic and geomorphic characterization of precariously balanced rocks, MS thesis, Arizona State University, Tempe, 207 p.

The code and this document are a work in progress. If you experience any bugs, please report them to me. You are welcome to make additions or adjustments that you think will help improve the code, but please send them to me so I can upload them to my website and make them available to everyone.

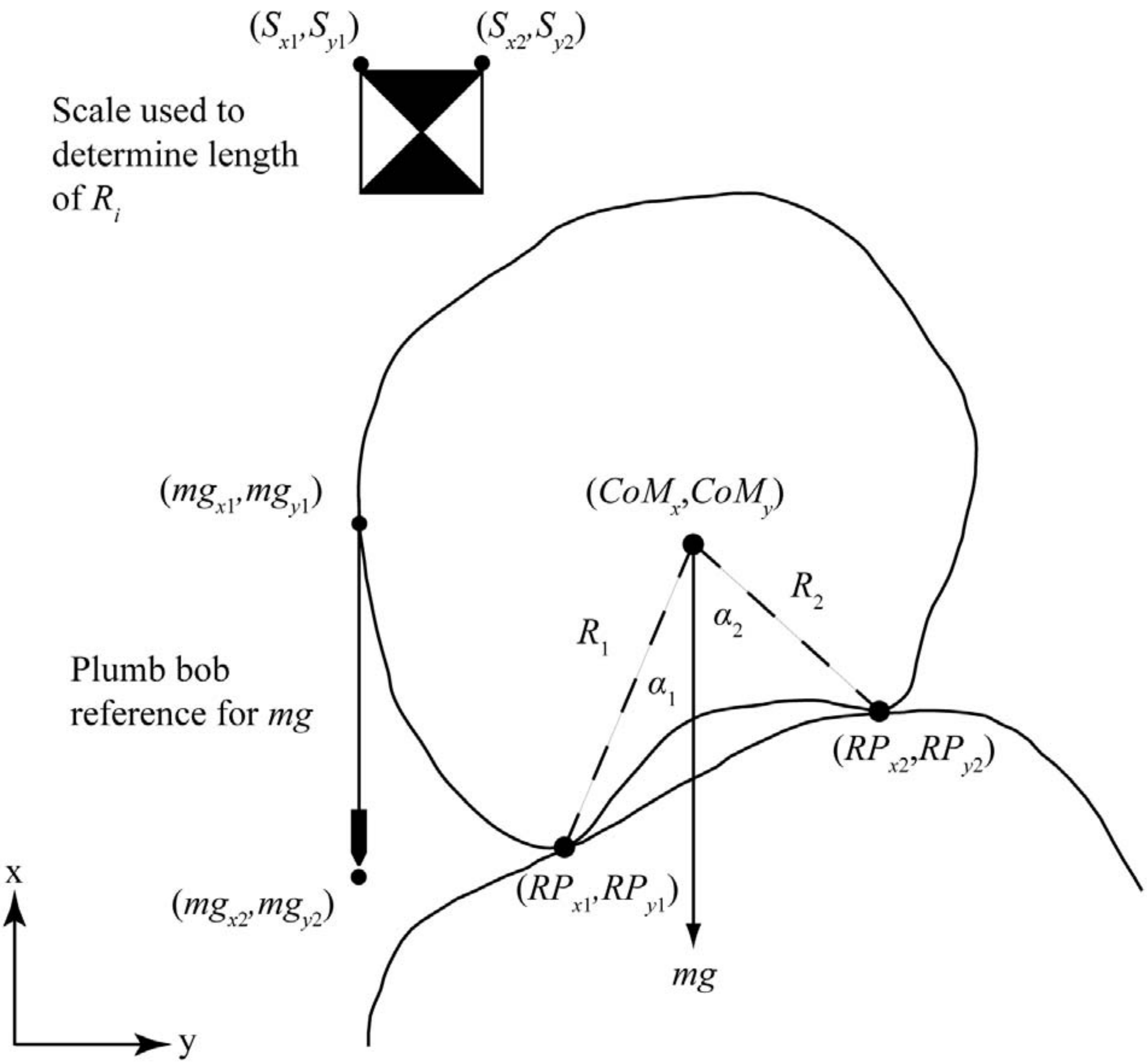


Figure 1. Geometric parameters of a precariously balanced rock (PBR) used in our PBR slenderness estimation method. CoM_x and CoM_y are the coordinates of the PBR's 2D center of mass; RP_{xi} and RP_{yi} are the PBR's rocking points; mg_{xi} and mg_{yi} provide the vertical reference (usually the plumb bob in the PBR's photograph); S_{xi} and S_{yi} are the coordinates of the length scale used to determine the lengths of R_i.

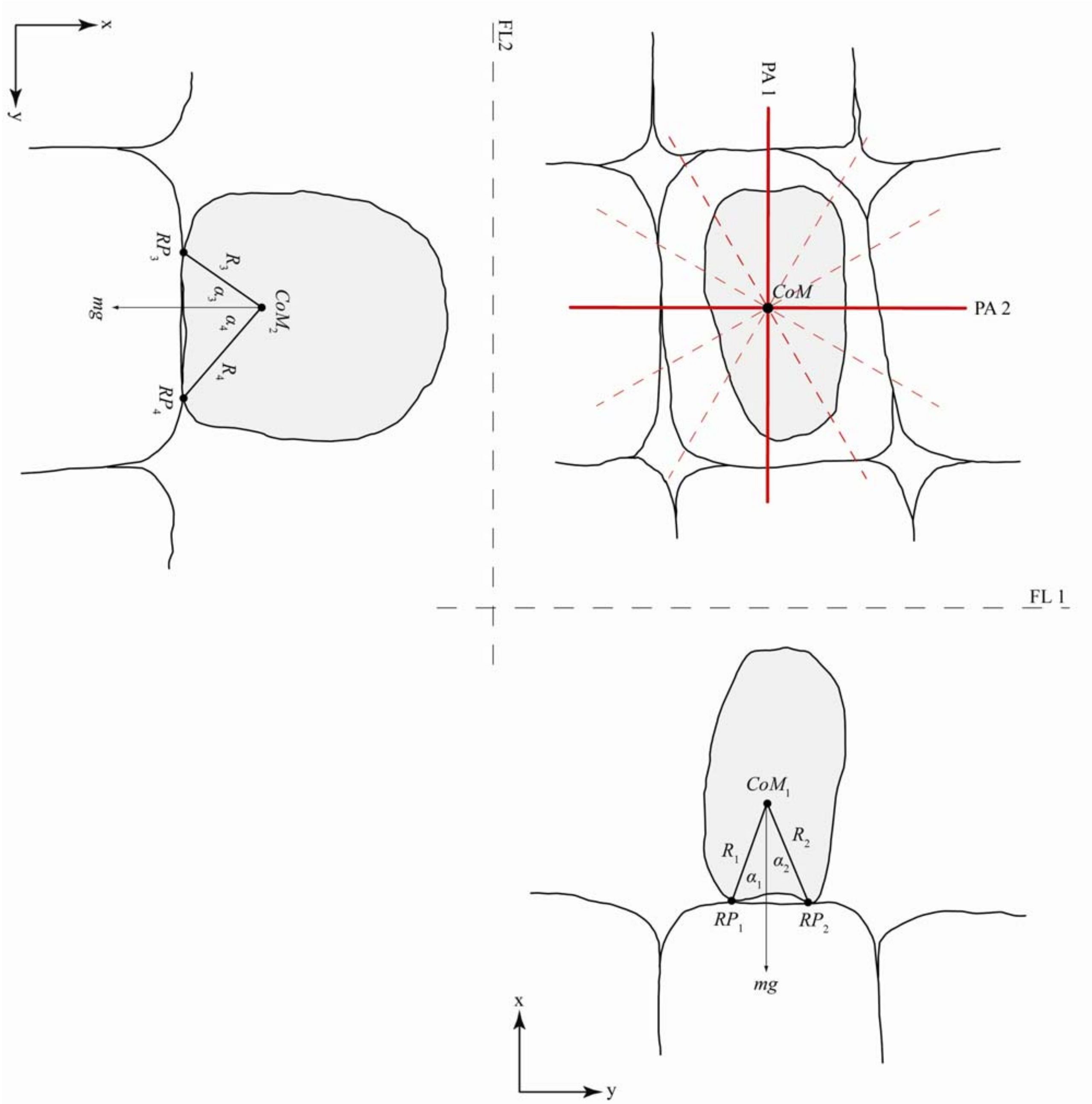


Figure 2. Three-view orthogonal depiction of a precariously balanced rock (PBR) illustrating the derivation of its geometrical framework. A PBR can be sectioned along an infinite number of vertical planes that pass through its center of mass in different azimuths. The azimuthal orientation of the principal axial plane that contains the PBR's greatest slenderness (smallest α_i value) represents the PBR's most likely toppling direction and is thus assigned to the PBR. RP_i = rocking point; CoM_i = center of mass; FL_i = folding line; PA_i = principal axis.

Minimum software requirements

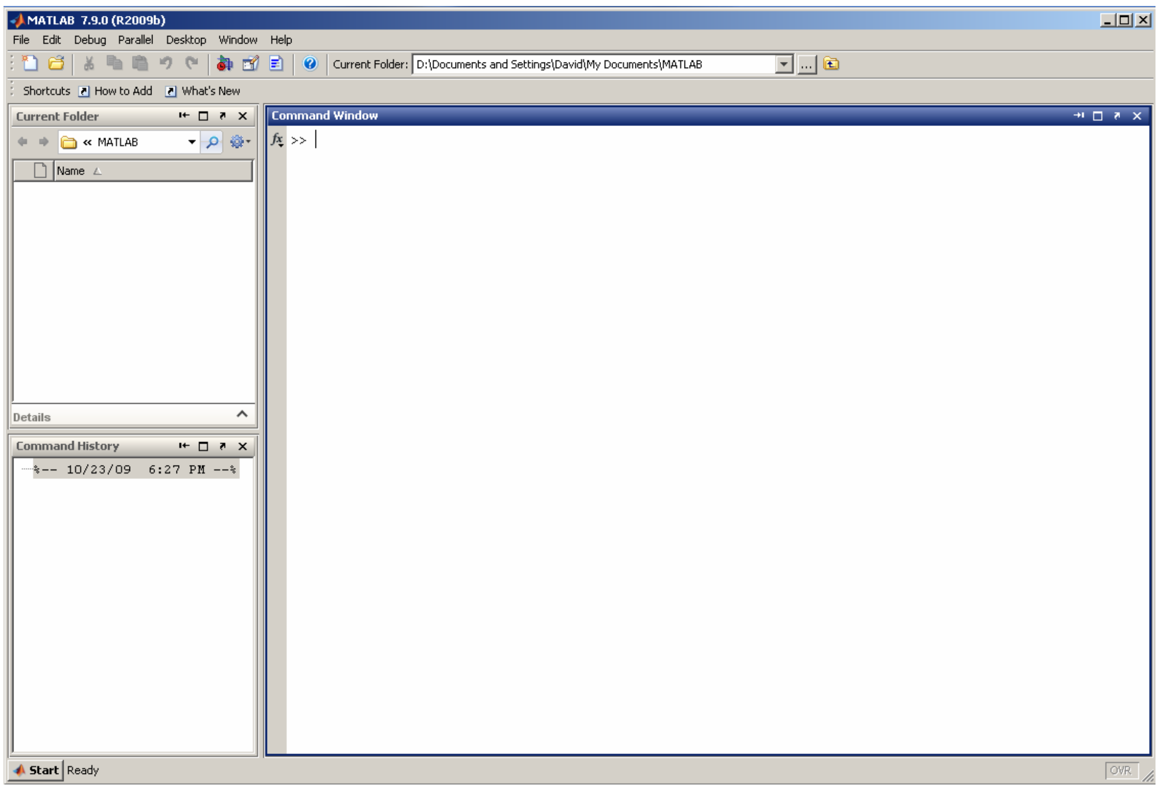
**** IMPORTANT NOTE ****

The code is written in MATLAB and requires the MATLAB Image Processing Toolbox version R2007b or later to run. If MATLAB returns an error using the *impoly.m* function, it is likely that you have an older version of the MATLAB Image Processing Toolbox (or do not have it at all). To accommodate for this, I made some adjustments to the code and functions in an alternative version (*PBR_slenderness_DH_v2_0*). It can be downloaded from the following webpage: http://activetectonics.la.asu.edu/Precarious_Rocks/PBR_Slenderness_Analysis.html.

Operating instructions

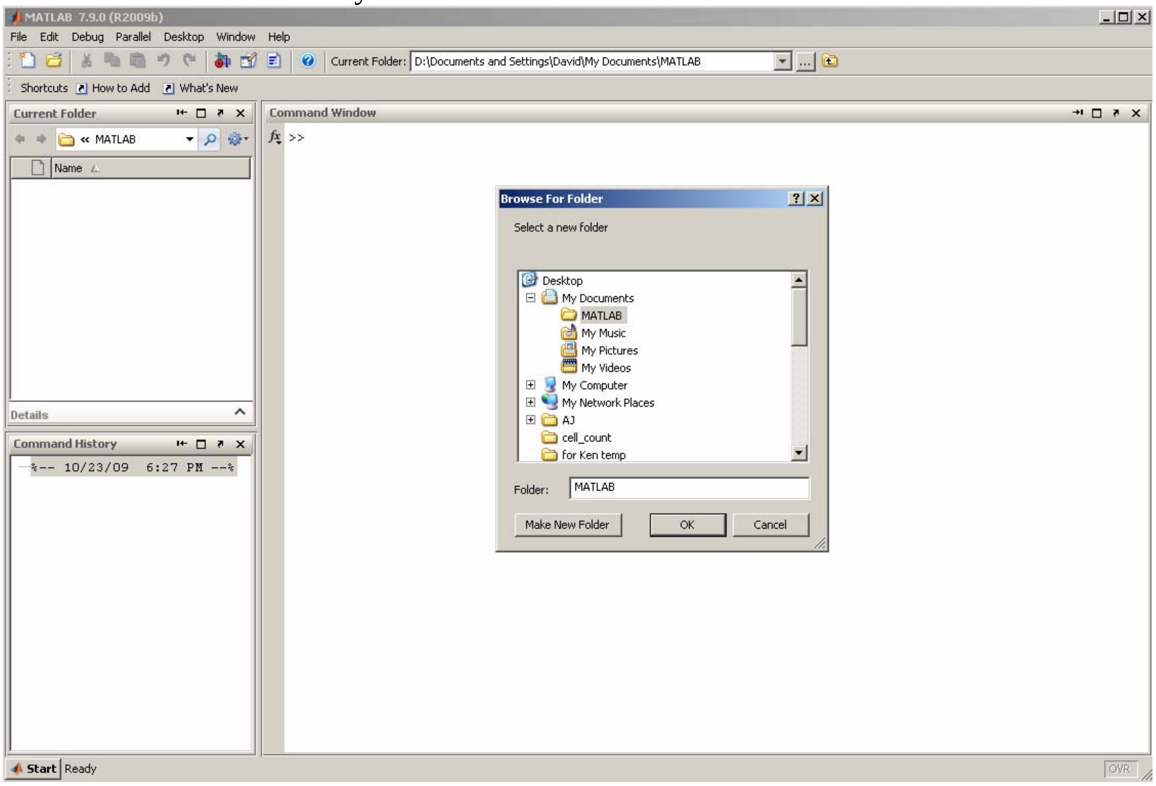
Step 1

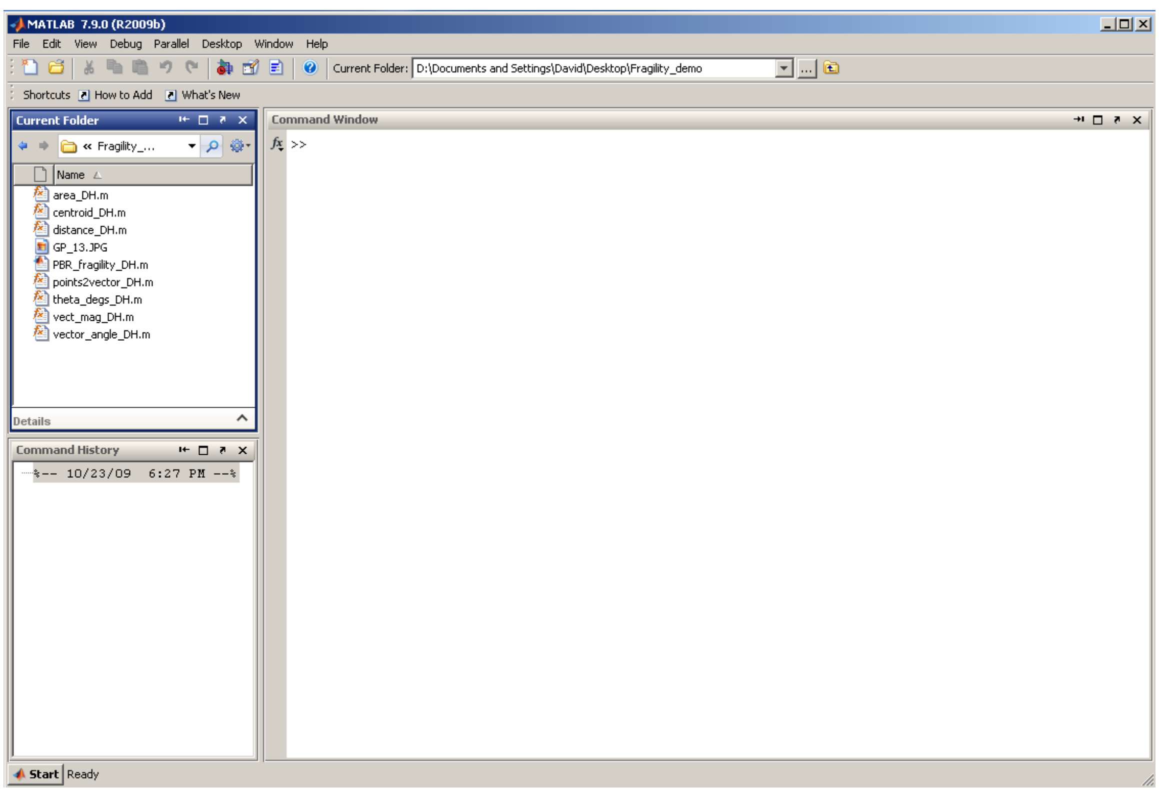
Start MATLAB.



Step 2

Navigate to the directory that contains the scripts. Make sure the scripts and PBR photographs are located in the same directory.

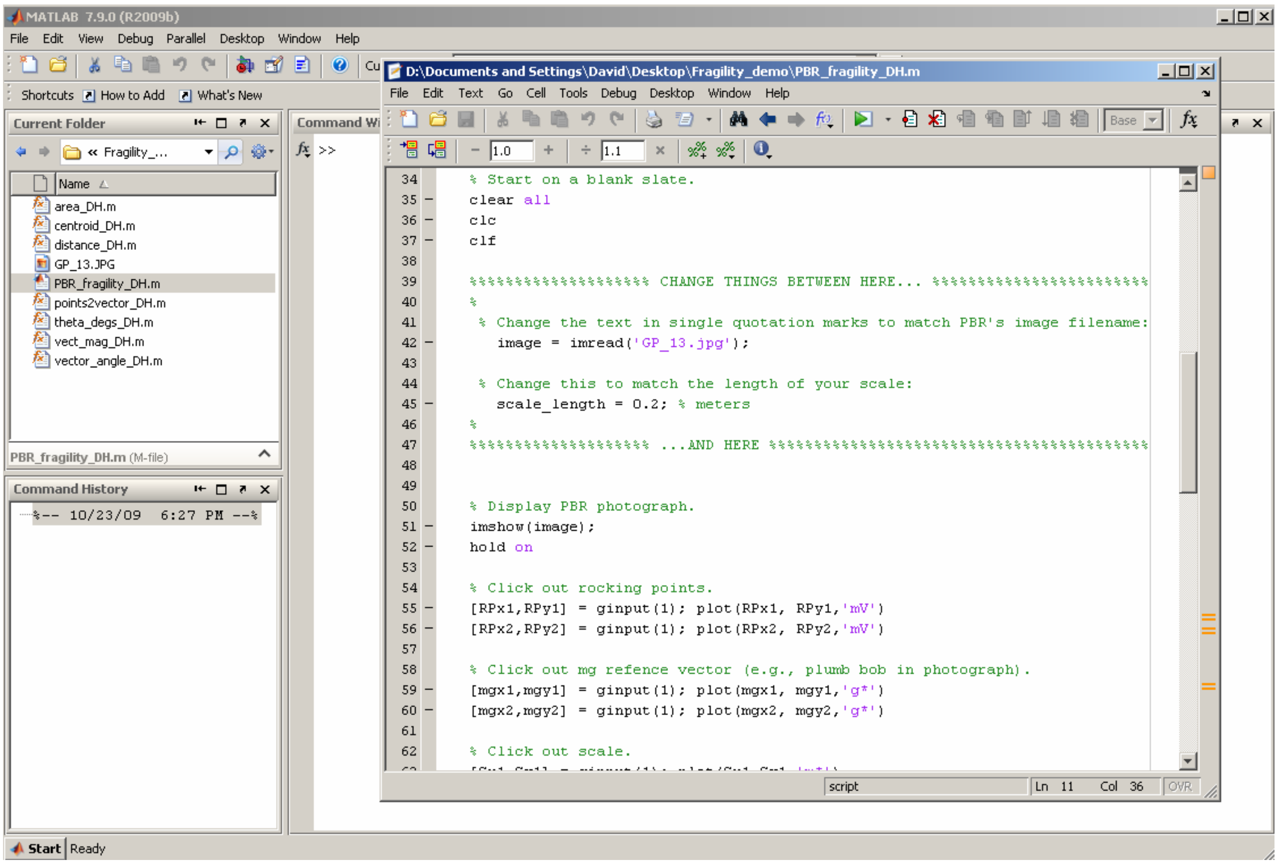




Step 3

Double-click the `PBR_slenderness_DH_v1_0.m` file to open the MATLAB script editor. Scroll down to the section enclosed by percent (%) symbols. There are two parameters that need to be changed:

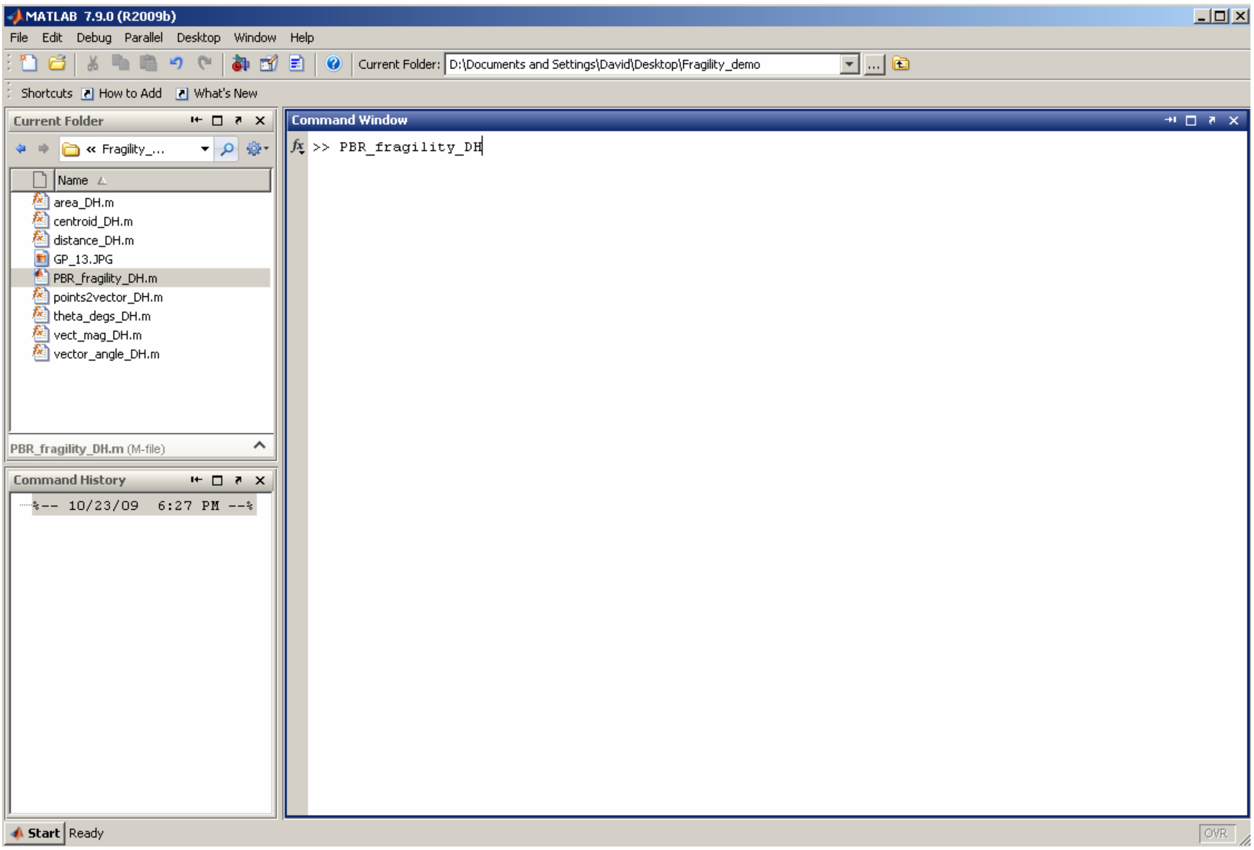
1. The name of the photo file (text inside the single quotation marks, include file extension).
2. Scale length (change this only once if the same length scale is used in all photographs).



Step 4
Save the changes made in the MATLAB script editor. In the Command Window, type:

```
>> PBR_slenderness_DH_v1_0
```

Hit the Enter key to run the main script.



Step 5

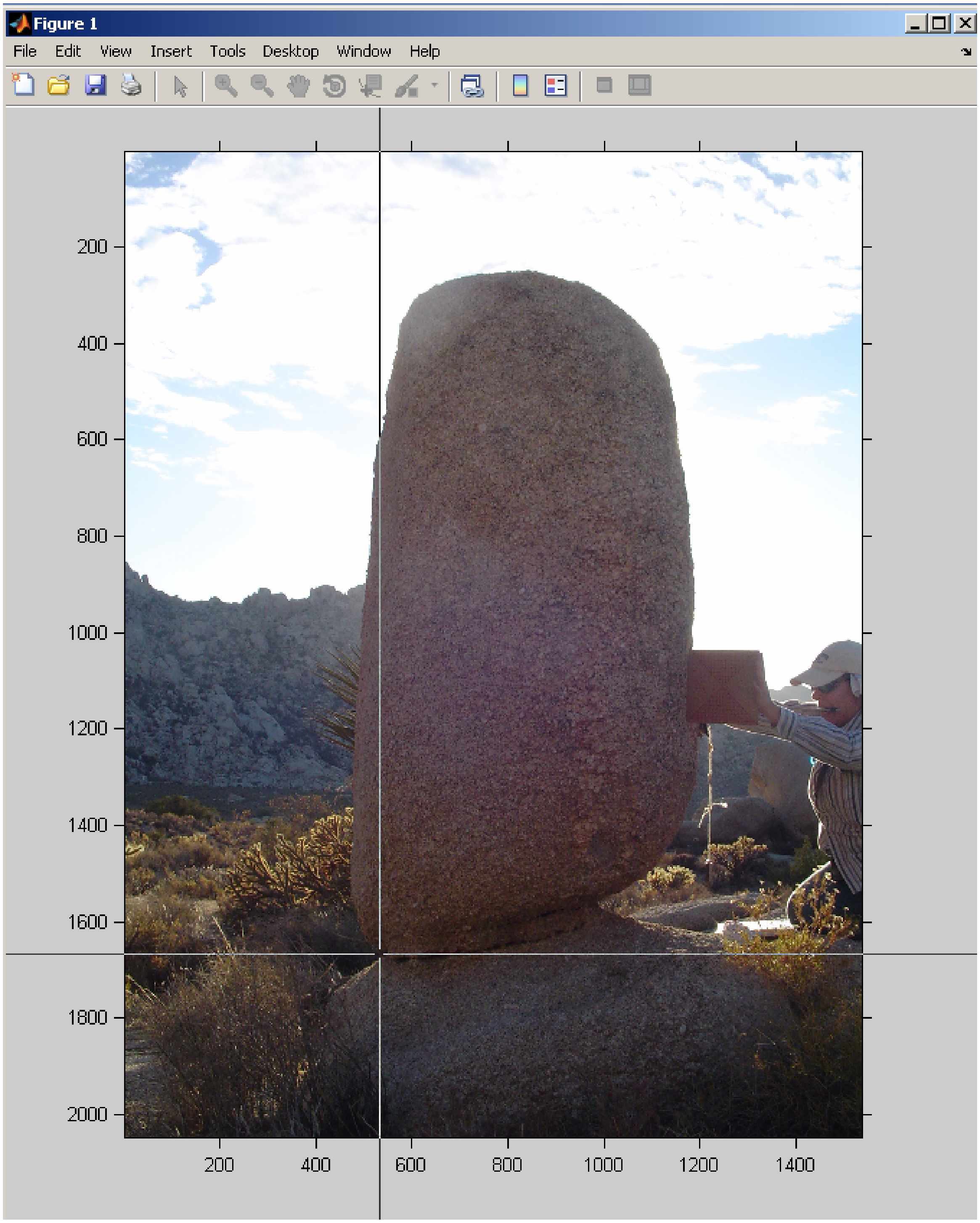
The photograph of the PBR will be loaded as Figure 1. You will notice the cursor change to cross hairs. The intersection of the hairs is where the clicked points are picked and plotted. The following is a breakdown of the demonstration to follow:

- MATLAB at this point is waiting for the following inputs using the **left mouse button in the following order**:
    - Two left clicks to define the PBR's rocking points (one left click for each rocking point).
    - Two left clicks for the *mg* reference vector (i.e. the plumb bob). These points must fall on the plumb bob's line, **regardless of whether the plumb bob is exactly vertical or not**.
    - Two left clicks for the scale.
    - An unlimited number of left clicks that define the PBR's outline. Hover the cursor over the first point to close the PBR's outline. The cursor will turn into a circle, indicating you have reached the first point. Left click once while the cursor is a circle to close the PBR's outline.
- MATLAB will then return α_1, α_2 (in degrees), R_1, and R_2 (in the units of the scale used in the photograph) in the Command Window.

A detailed demonstration with a PBR photograph begins on the following page.

Demonstration

After running the main script, the PBR's photograph will be loaded into Figure 1. Notice how the cursor changes to cross hairs when it hovers over the photograph.



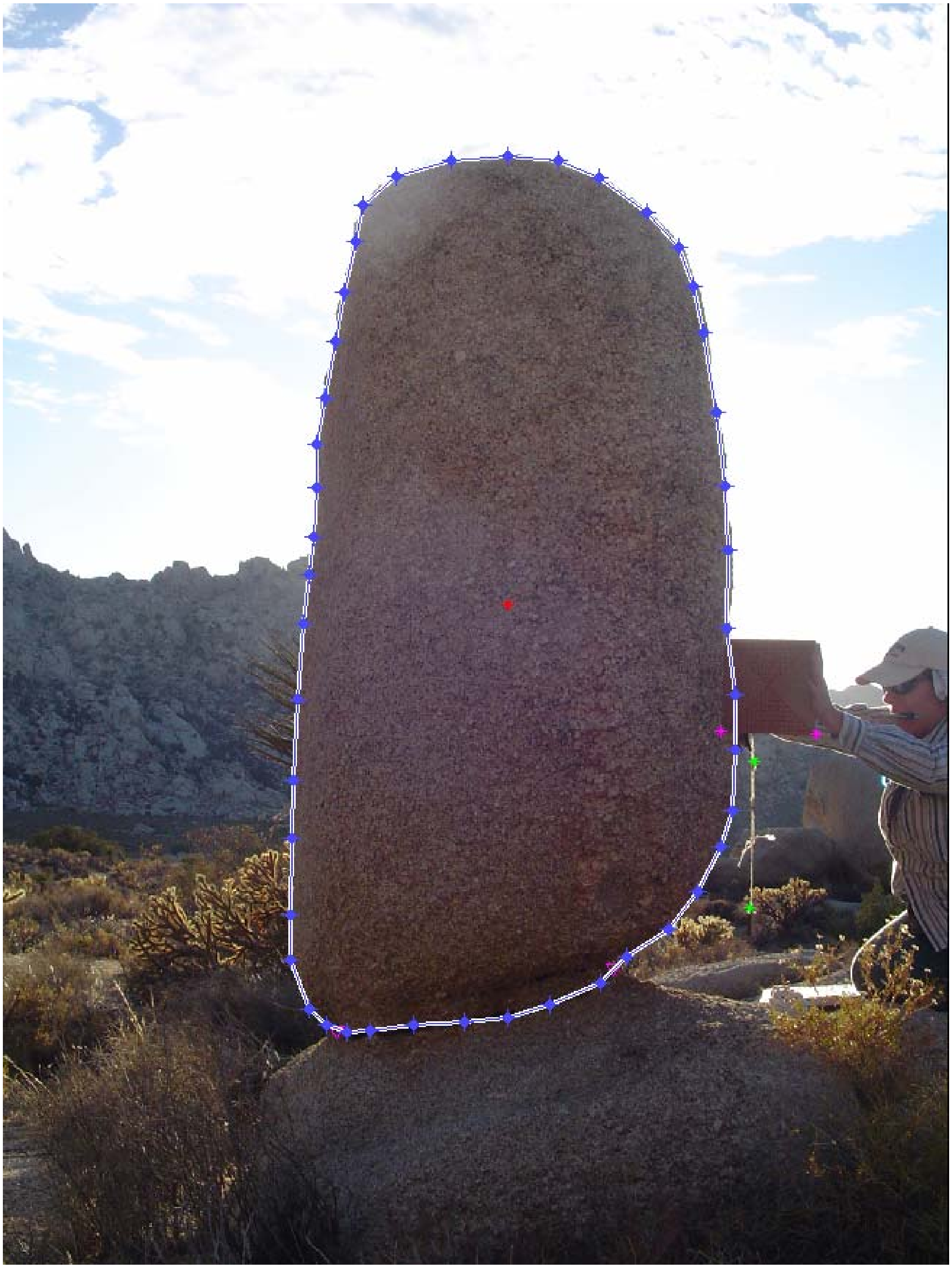
Click the PBR’s rocking points (plotted as downward-pointing purple triangles):

Click the vertical reference (plotted as green stars):

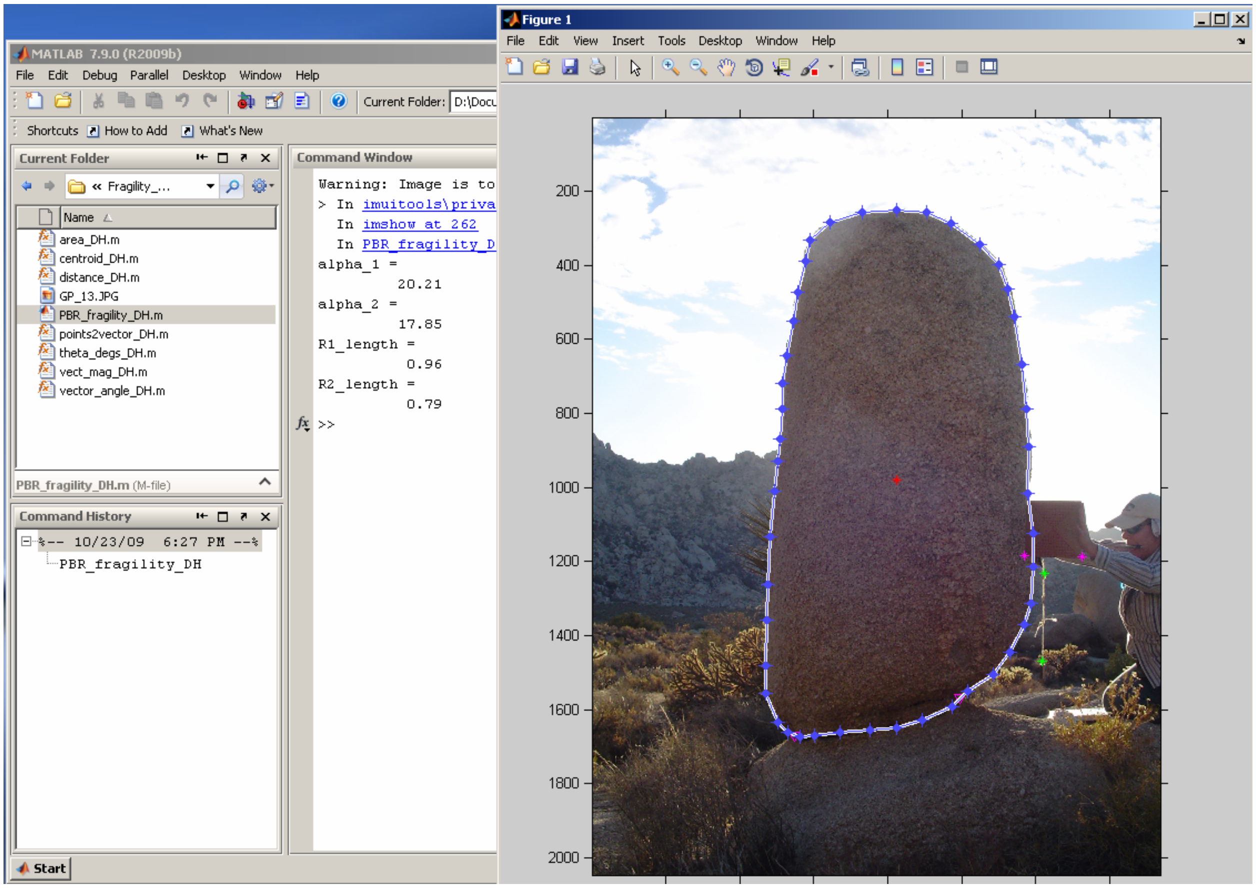
Click the length scale (plotted as purple stars):

Click the outline of the PBR (plotted as blue stars with white connecting lines):

Close the outline of the PBR by hovering the cursor over the first point until it turns to a circle and left clicking once. The 2D center of mass of the PBR in the photograph's plane will then be plotted as a red star:



Alpha 1 and alpha 2 (in degrees) and R_1 and R_2 (in the same units as the scale used in the photograph) will then be returned in the Command Window:



Scripts and functions

The following section lists the scripts and functions used in *PBR_Slenderness_DH_v1_0*. They are also available for download from the following webpage:
http://activetectonics.la.asu.edu/Precarious_Rocks/PBR_Slenderness_Analysis.html

PBR_Slenderness_DH_v1_0.m

This is the main script that calls all auxiliary scripts and returns the PBR's geometric parameters.

```
% David E. Haddad - 08/2009

% PBR_slenderness_DH_v1_0

% This script computes basic parameters of a precariously balanced rock
% (PBR) from an unconstrained digital photograph.

% Script designed by David E. Haddad.
% Method inspired by J Ramon Arrowsmith and Matt Purvance:
% Purvance, M. D., 2005. Overturning of slender blocks: numerical
% investigation and application to precariously balanced rocks in southern
% California, PhD dissertation, University of Nevada, Reno, Reno, Nevada
% (http://www.seismo.unr.edu/gradresearch.html).


% OPERATING INSTRUCTIONS:
% For a complete demonstration, see PBR_slenderness_DH_v1_0_UserManual.pdf,
% downloadable from http://activetectonics.asu.edu/Precarious_Rocks

% (1)  Enter the name of the PBR's image filename.
% (2)  Enter the scale length used in the photograph.
% (3)  Save changes.
% (4)  Run this script to load the PBR's image. MATLAB will be waiting for
%      the following inputs using the left mouse button:
%     (a) Two left clicks to define the PBR's rocking points (one left
%         click for each rocking point).
%     (b) Two left clicks for the mg reference vector (i.e. plumb bob).
%     (c) Two left clicks for the photograph scale.
%     (d) An unlimited number of left clicks that define the PBR's
%         outline. Hover the cursor over the first point to close the
%         outline.  The cursor will turn into a circle, indicating you
%         have reached the first point.  Left click once while the cursor
%         is a circle to close the PBR's outline.  MATLAB will then compute
%         and return alpha_1, alpha_2, R_1, and R_2 in the Command Window.

% Important Note:
% PBR photographs need to be in the same directory as this script and its
% accompanying functions!

% Here we go...

% Start on a blank slate.
clear all
```

```
clc
clf

%%%%%%%%%%%%%%%%%%%% CHANGE THINGS BETWEEN HERE... %%%%%%%%%%%%%%%%%%%%%%%%
%
% Change the text in single quotation marks to match PBR's image filename:
PBR_image = imread('PBR_photograph.JPG');

% Change this to match the length of the scale used in PBR's photograph:
scale_length = 0.2; % meters
%
%%%%%%%%%%%%%%%%%%%% ...AND HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Display PBR photograph.
imshow(PBR_image);
hold on         % hold on to the image so we can start digitizing on it.

% Click out rocking points.
[RPx1,RPy1] = ginput(1); plot(RPx1, RPy1,'mV')
[RPx2,RPy2] = ginput(1); plot(RPx2, RPy2,'mV')

% Click out mg refence vector (e.g., plumb bob in photograph).
[mgx1,mgy1] = ginput(1); plot(mgx1, mgy1,'g*')
[mgx2,mgy2] = ginput(1); plot(mgx2, mgy2,'g*')

% Click out scale.
[Sx1,Sy1] = ginput(1); plot(Sx1,Sy1,'m*')
[Sx2,Sy2] = ginput(1); plot(Sx2,Sy2,'m*')

% Left-click outline of PBR to create a 2D polygonal representation.
h = impoly(gca,[]);

% Get polygon vertices from handle h.
api = iptgetapi(h);
pos = api.getPosition();

% Compute and plot polygon's centroid.
center_of_mass = centroid_DH(pos);
CoMx = center_of_mass(:,1);
CoMy = center_of_mass(:,2);
plot(CoMx, CoMy,'r*')

% Create R1 and R2 vectors using rocking points and centroid.
R1 = points2vector_DH(RPx1,RPy1,CoMx,CoMy);
R2 = points2vector_DH(RPx2,RPy2,CoMx,CoMy);

% Create mg vector from plumb bob reference points.
mg = points2vector_DH(mgx1,mgy1,mgx2,mgy2);

% Compute PBR's slenderness (angles made between mg and R vectors).
alpha_1 = 180 - theta_degs_DH(vector_angle_DH(R1,mg)); % degrees
alpha_2 = 180 - theta_degs_DH(vector_angle_DH(R2,mg)); % degrees
```

```
% Check to make sure it's the acute angle.
if alpha_1 > 90 || alpha_2 > 90
    alpha_1 = 180 - alpha_1
    alpha_2 = 180 - alpha_2
else
    alpha_1
    alpha_2
end

% Compute length scaling factor.
scaling_factor = scale_length / distance_DH(Sx1,Sy1,Sx2,Sy2);

% Compute pixel lengths of R1 and R2 vectors.
R1_pixel_length = vect_mag_DH(R1);
R2_pixel_length = vect_mag_DH(R2);

% Convert R1 and R2 pixel lengths into real lengths using scaling factor.
R1_length = R1_pixel_length * scaling_factor % meters
R2_length = R2_pixel_length * scaling_factor % meters

% End of main script.
```

centroid_DH.m

Function to compute the 2D centroid of a polygon given the polygon's vertices.

```
function centroid = centroid_DH(R)

x = R(:,1);
y = R(:,2);

n = length(R);
cx = 0;
cy = 0;

for i=1:n-1
    cx = cx + (x(i)+x(i+1))*(x(i)*y(i+1) - x(i+1)*y(i));
    cy = cy + (y(i)+y(i+1))*(x(i)*y(i+1) - x(i+1)*y(i));
end

% Include last vertex (because polygon is not closed)
cx = cx + (x(n)+x(1))*(x(n)*y(1) - x(1)*y(n));
cy = cy + (y(n)+y(1))*(x(n)*y(1) - x(1)*y(n));
centroid = [cx cy]/6/area_DH(R);
```

area_DH.m

Function to compute the area of a polygon given the polygon's vertices.

```
function polygon_area = area_DH(polygon)
x = polygon(:,1);
y = polygon(:,2);
n = length(polygon);
sum = 0;
for i=1:n-1
        sum = sum + x(i)*y(i+1) - x(i+1)*y(i);
end
polygon_area = 0.5*(sum + x(n)*y(1) - x(1)*y(n));
```

points2vector_DH.m

Function to create a 2D vector that passes through two points.

```
function R = points2vector_DH(x1,y1,x2,y2)
i = x2 - x1;
j = y2 - y1;
R = [i j];
```

vector_angle_DH.m

Function to compute the angle made between two vectors using the dot product.

```
function alpha = vector_angle_DH(a,b)
alpha = acos(dot(a,b)/(norm(a)*norm(b)));
```

vect_mag_DH.m

Function to compute the magnitude of a 2D vector.

```
function vector_magnitude = vect_mag_DH(V)
vector_magnitude = sqrt(V(:,1)^2 + V(:,2)^2);
```

distance_DH.m

Function to compute the distance between two points.

```
function distance = distance_DH(x1,y1,x2,y2)
distance = sqrt((x2-x1)^2 + (y2-y1)^2);
```

theta_degs_DH.m

Function to convert radians to degrees.

```
function degrees = theta_degs_DH(radians)
degrees = radians.*(180./pi);
```

References

Anooshehpoor, A., Purvance, M., Brune, J., and Rennie, T., 2007, Reduction in the uncertainties in the ground motion constraints by improved field testing techniques of precariously balanced rocks (proceedings and abstracts): SCEC Annual Meeting, Palm Springs, California, September 9-12, 2007, 17.

Haddad, D. E., and Arrowsmith, J. R., 2009, Characterizing the geomorphic situation of preariously balanced rocks in semi-arid to arid landscapes of low-seismicity regions (proceedings and abstracts): SCEC Annual Meeting, Palm Springs, California, September 12-16, 2009, 19.

Hudnut, K., Amidon, W., Bawden, G., Brune, J., Bond, S., Graves, R., Haddad, D. E., Limaye, A., Lynch, D. K., Phillips, D. A., Pounders, E., Rood, D., and Weiser, D., 2009, The Echo Cliffs precariously balanced rock; discovery and description by terrestrial laser scanning (proceedings and abstracts): Southern California Earthquake Center Annual Meeting, Palm Springs, California, September 12-16, 2009, 19.

Purvance, M., 2005, Overturning of slender blocks: numerical investigation and application to precariously balanced rocks in southern California [PhD thesis]: University of Nevada, Reno, 233 p.

Acknowledgements

Thanks to Matthew Purvance and Rasool Anooshehpoor for their insightful discussions about the methods developed here.