TEMPLATE MATCHING USED FOR SMALL BODY OPTICAL NAVIGATION WITH POORLY DETAILED OBJECTS

Joshua R. Lyzhoft[1*], Andrew J. Liounis[1], and Dante S. Lauretta[2]; [1]Goddard Space Flight Center - 8800 Greenbelt Rd, Greenbelt, MD 20771 , [2]Lunar and Planetary Laboratory, University of Arizona, 1415 N 6th Ave, Tucson, AZ 85705 , *[joshua.r.lyzhoft@nasa.gov]

Abstract. *Object and template matching becomes difficult when an image lacks detail. This is particularly worrisome when typical matching techniques, cross-correlation, log-polar mapping, and key point matching fail. Work herein describes a formulation that identifies objects of interest, estimates the affine transformation between a template object and scene using Principal Component Analysis (PCA), and provides a fit value for the objects and template incorporating Hu's Moments. The algorithm presented is tested on synthetic images and images obtained from the OSIRIS-REx mission while the spacecraft was approaching its target, Bennu. Results for the current formulation show that, with the presence of large-scale variations and rotation, the fitting scheme performs well when compared with other techniques.*

Introduction. In certain areas of digital image processing, it is beneficial to match an image or object template to a given scene. Typically this is done by algorithms such as a standard cross-correlation technique, log-polar mapping, or key point matching using SIFT or SURF.[1–5] However, all these forms typically breakdown when images and templates lack sufficient object features, details, or have large-scale differences. Such scenarios occur in space missions where the spacecraft is arriving at a target body. These scenarios may be a flyby or rendezvous required for planetary defense missions or science missions, such as the OSIRIS-REx mission to the asteroid Bennu. There is a time during approach, when capturing images, where the target body is an extended source on the image but only appears as a fuzzy shape, lacking sufficient surface features or details.

During approach, along with proximity operations, it is advantageous to have the knowledge of the target's size and relative orientation with the observing device (rotation about boresight specifically). However, there may be situations where the scale, camera orientation, and object's position are not well known. In the case of the OSIRIS-REx mission, an initial shape model and state of Bennu were generated by information from the Arecibo observatory, lightcurve data, and the Goldstone tracking station.[6] Currently, it is not published how well the initial shape model fit with the actual shape of Bennu. Work herein describes a new process using Principal Component Analysis (PCA), which determines the scale, rotation, and translation of a template object compared to a scene. Additionally, a fitness value is developed, which indicates the match of the template to the scene target.

Methodology. Since both the template and the scene object lack sufficient detail, a new process is created, incorporating ideas from object silhouettes and blob detection. Scenarios presented in this paper assume that a predicted template is available. A multi-level threshold algorithm is applied to both the template and image to create binary blobs. Each blob is then analyzed to find its centroid, eigen axes, and edge interest points (EIP)s. After a blob has been analyzed, it then undergoes a minimization, calculating the best fit affine transformation for centroid and EIP matching between the scene and template. Once the transformation is estimated, the template is mapped to the scene and a fitness value is calculated. An overall workflow can be seen in Fig. 1.

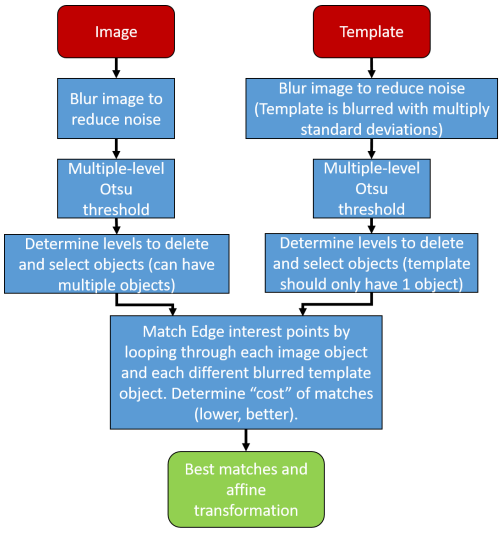

Figure 1. Overall workflow of algorithm

Blob Detection. As is done in traditional scale-invariant feature descriptors, the algorithm begins by building an image pyramid for both the template and the image by convolving Gaussian kernels of varying standard deviations. At each stage, a multi-level threshold is completed on the template and image. This is done by implementing Otsu's method.[7] Once each level is determined (generally a starting number of levels is six) a process of merging levels is completed. Doing so involves calculating outliers, using median formulation, of the number of pixels in each level. In general, the background makes up the majority of the pixels in the first threshold level, while the top level (6) and surrounding levels (4 and 5) hold objects of interest. If a lower amount of

levels is desired, for example four, then the top two to three levels hold objects of interest, usually. However, if the image is "noisy", only the top level should be used. The noise can create false objects of interest. After the image is split up into background levels and objects of interest levels, the levels that contain background values are merged, and the threshold value is used create a binary image. Each image is then converted into a binary image after the suitable level threshold is determined. Thereafter, image segmentation is completed to identify objects in the scene and template. A sub-image is created for each binary blob, creating a "local" image to be compared with the template.

Principal Component Analysis. Eigen axes are computed for each object in the scene and template.[8] However, the eigenvectors, mathematically, can have a positive or negative direction depending on the formulation. This direction ambiguity of the axis is remedied by incorporating data analysis. An inner product of the points within the binary object and the eigen axes is computed.

$$s_i = \sum_{k=1}^{n} \mathbf{u}_i \mathbf{x}_k \, |\mathbf{u}_i \mathbf{x}_k| \tag{1}$$

where s is the the scalar value to be used to determine the eigenvector sign, i is the i$^{\text{th}}$ eigenvector, n is number of interest points, $\mathbf{u}_i$ is the i$^{\text{th}}$ 2x1 eigenvector, and $\mathbf{x}_k$ is the k$^{\text{th}}$ 1x2 interest point coordinates. If $s < 0$, then the i$^{\text{th}}$ eigenvector is multiplied by -1. Otherwise, the eigenvector's direction does not need to be reversed. Similar techniques have been investigated for single value decomposition.[9]

An EIP is found by determining a point in the blob that is farthest away from the center-of-mass along a given eigen axis or rotated eigen axis. This point does not need to have integer coordinates, which pertains to an image pixel location. If the EIP locations were to be integer values, they might result is suitable locations for EIP point matching between the template and scene object.

Interest Point Determination. After the eigen axes are computed, a set of axes is also determined by rotating the computed eigen axes. These rotated axes can be at any angle or be of any number. An example of the eigen axis and subdivisions can be seen in Figs. 2 and 3. Along each axis, an EIP is found. Fig. 3 shows binary objects, both from the template and image, showing the eigen axes, rotated axes (set of 2), centroid, and EIPs.

The collection of EIPs is important to the overall shape and orientation of an object. In essence, the EIPs create boundary points that have an orientation referenced to an eigen axis. This reference of the points gives the initial guess for scale and orientation. If the incorrect eigen axes are matched, it could result in an object that has incorrect rotation, scale, and translation. Simply obtaining the boundary of an object will not suffice. Boundary points are not referenced to an eigen axis, which causes the inability to provide an initial scale and rotation estimate.

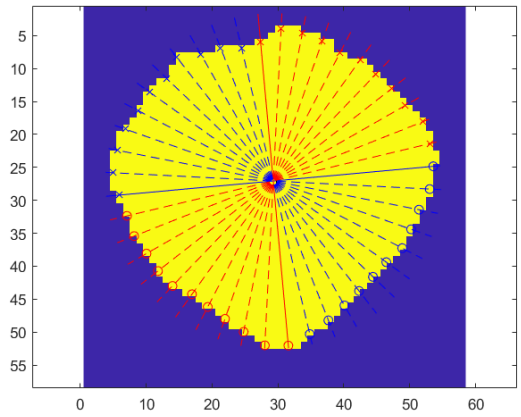

Figure 2. Binary object with EIPs, eigen axes, and 10 rotated axis sets (eigen axes are solid lines)

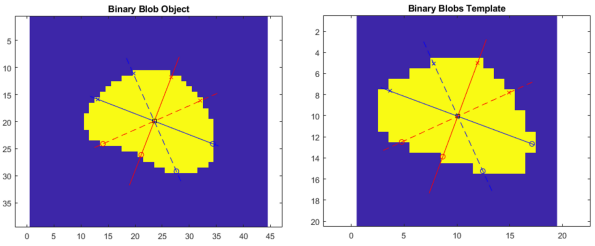

Figure 3. Binary objects (scene object left and template right) with EIPs along the eigen axes and rotated axes (eigen axes are solid lines)

Affine Transformation Estimation. Since EIPs and the centroid have been determined, the affine transformation between the template and scene object points can be estimated. This is done by minimizing the squared differences of the transformed template EIPs and the scene object EIPs. The expression for minimization is given by the following equation

$$G = \sum_{i=1}^{n} [\mathbf{T}\mathbf{P}_{T_i} - \mathbf{P}_{S_i}]^T_{[1\text{x}2]} \, [\mathbf{T}\mathbf{P}_{T_i} - \mathbf{P}_{S_i}]_{[2\text{x}1]} \tag{2}$$

where G is the value to minimize, n is the number of EIPs, i is the index of the EIPs, $\mathbf{P}_T$ is a matrix (3 x n) containing the x and y pairs for the EIPs corresponding to the template with the third row being unity, and $\mathbf{P}_S$ is a matrix (2 x n) containing the x and y pairs for the EIPs corresponding the scene. The matrix $\mathbf{T}$ is the transformation (2 x 3 matrix) given by

$$\mathbf{T} = \begin{bmatrix} K\cos(\theta) & -K\sin(\theta) & T_X \\ K\sin(\theta) & K\cos(\theta) & T_Y \end{bmatrix} \tag{3}$$

where K is the scale, θ is the rotation about the center-of-brightness, T_X is the horizontal translation, and T_Y is the vertical translation. These four variables are what change in the minimization scheme. Initial conditions can be computed by using a comparison between scene and template centroids as well as one EIP. The translation initial condition is found by taking the difference between the template and scene object centroid; the initial scale is found by the ratio of the principle axis EIP magnitudes corresponding to the template and scene object; the initial rotation is determined by calculating the angle between the principal axes of the template and scene object.

After $\mathbf{T}$ is estimated, the scene with respect to the object is reconstructed from the template, and another minimization is completed to better refine the transformation matrix. The minimization attempts to minimize the following function

$$J = \sum_{ij} \frac{-1}{1 + \left(R_{ij} - L_{ij}\right)^2} \tag{4}$$

where i and j are the columns and rows of the cut template, R is the i^{th} and j^{th} element of the template sub-image, and L is the interpolated value from the image at the i^{th} and j^{th} element of the template sub-image.

Transformation Fit Value. To determine if the object and image reconstruction are correctly matched, a fitness value is computed based on cross-correlation, normalized summed square differences, and normalized absolute differences of the image and template's Hu's moments.[10,11] The normalized summed square difference is computed by

$$c_d = \frac{\sum \left((\mathbf{M} - \mathbf{R}) \circ (\mathbf{M} - \mathbf{R})\right)}{n} \tag{5}$$

where $\circ$ is the Hadamard product (matrix or vector element-wise multiplication), $\mathbf{M}$ is the image resulting from the Hadamard product of the object sub-image and object blob sub-image, and $\mathbf{R}$ is the template resulting from the Hadamard product of the template sub-image and template blob sub-image. The blob sub-images, for both object and template, are represented by zeros and ones. Creating $\mathbf{M}$ and $\mathbf{R}$ through this multiplication eliminates any noise that is outside the object ("cookie cutting" the object based on its blob representation) and helps in the correlation coefficient calculation.

The cross-correlation if calculated by

$$c_r = \frac{\sum \left(\mathbf{M}' \circ \mathbf{R}'\right)}{\sqrt{\left(\sum \left(\mathbf{M}' \circ \mathbf{M}'\right)\right)\left(\sum \left(\mathbf{R}' \circ \mathbf{R}'\right)\right)}} \tag{6}$$

where $\mathbf{M}'$ is the sub-image, $\mathbf{M}$, subtracted by its own mean value, and $\mathbf{R}'$ is the template sub-image, $\mathbf{R}$, subtracted by its own mean value.

A third variable required by the overall fit value involves the computation of Hu's moments. Once the object moments are calculated, the final component required for the fit value can be computed as

$$c_H = \sum \left(|\mathbf{H}_M - \mathbf{H}_R| \oslash |\mathbf{H}_M|\right) \tag{7}$$

where $||$ denotes absolute value of each vector component, $\oslash$ is the Hadamard division (matrix or vector element-wise division), $\mathbf{H}_M$ is the Hu's moments vector (1 x 7) of the scene sub-image $\mathbf{M}$, and $\mathbf{H}_R$ is the Hu's moments vector (1 x 7) of the template sub-image $\mathbf{R}$.

Combining the three calculated components allows for a final fitness value. This value is given by

$$c = 100 \; c_H \; c_d \; \left(1 - c_r\right)^2 \tag{8}$$

The closer the fitness value is to zero, the more likely the template affine transformation and template match the scene object.

Preliminary Results. By following the above process, a collection of scenarios is analyzed. These scenarios are based on images generated for and taken from the OSIRIS-REx mission. To understand the presented results, a few definitions and clarifications must be made.

The scale value indicates how much larger the actual object is compared to the template object (a value of 2.0 indicates that the scene object is 2 times larger than the template or that the template is 0.5 times as large as the scene). Furthermore, the vertical and horizontal translation for the actual value is estimated based on center-of-brightness differences between the template and scene object, much like how cross-correlation works. When the rotation is different than zero degrees, the method developed herein does not match with the actual value of translation. This is due to the formulation and representation of the affine transformation, a translation from the scaled and rotation template. Additionally, rotation is defined by how many degrees the template must be rotated to be aligned with the scene object.

All results are broken up into two main portions. The first subsection is initial testing using synthetic images, and the second subsection uses images take from the OSIRIS-REx spacecraft. Images used in the "Bennu:<date>" notation were taken using OSIRIS-REx's PolyCam during the Approach phase of the spacecraft's arrival at Bennu.[12,13]

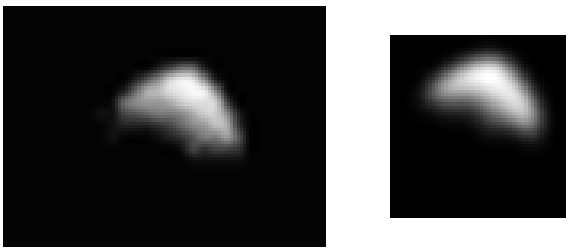

Figure 4. Cropped scene (left) compared with template (right) (image not to scale)

Synthetic Images. For an initial test, a scenario was created that used fully synthetic images. This was used to initially test if the method developed would result in reasonable values. As it can be seen in Fig. 4, the scene sub-image and template have generally the same shape. Also, the shapes are unique to an asymmetric body and high phase angle. When objects are symmetric and/or being illuminated so that the shape is symmetric about either one of the eigen axes, there can still be a rotation ambiguity. In particular, imagine approaching a spherical object that is being fully illuminated. Scale and translation can be found, but there is no uniqueness to a blob's silhouette/shape.

Figs. 4 and 5 are an image and template comparison as well as a scene object reconstruction. However, due to the image resolution (1944 x 2592 pixels), Fig. 4 is not to scale. Table 1 shows the estimated values for the transformation and the fitness value computed. The reconstruction matches the shape well. However, detail is lacking in both images. Due to such a blurry template, the method was only able to find the scale and rotation within 0.064 and 1.75° of the actual scene and template difference. With the difference in scale and rotation, it is not surprising that the translation does not match well either. Further results will consider images taken from the Approach phase of the OSIRIS-REx mission.

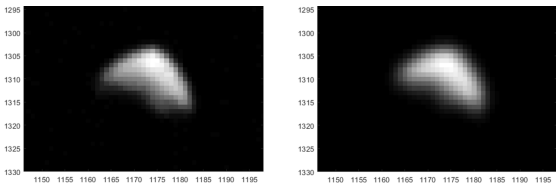

Figure 5. Image reconstructed: scene object (left) and template (right)

Table 1. Transformation and Fitness Values

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	1.6	1.5442	-	x	x
Rot. (θ)	0	1.7481	-	x	x
Horz. (T_X)	1162.7	1158.5	1163	x	x
Vert. (T_Y)	1302.5	1297.8	1301	x	x
Fitness (c)	-	2.79e-05	-	-	-

Bennu: 2018-10-21 Epoch . On October 21st, 2018, the OSRISIS-REx spacecraft was able to take images, using its PolyCam, in which Bennu reached about 10 pixels in diameter on the detector array. A cropped image of Bennu can be seen in Fig. 6. There is not much detail on the body. In fact, Bennu barely has a distinguishable shape. Additionally, the shape almost seems symmetric, and the phase angle is nearly zero.

Table 2 shows the results of the four methods being tested for scale, rotation, and translation. This initial result corresponds to a scale factor of 1 and a rotation of 0°. The considered method estimated the scale and rotation well for such a small object. However, the cross-correlation was able to estimate a more correct translation.

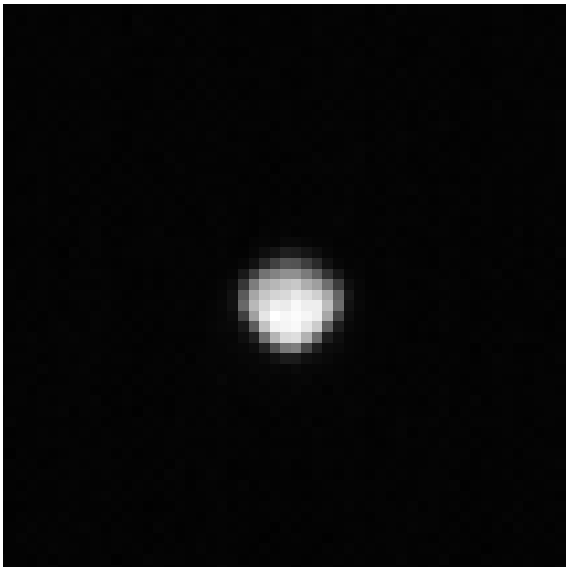

Figure 6. Cropped image of Bennu taken on 2018-10-21 using PolyCam

Table 2. Initial Transformation and Fitness Values for Template Rotation of 0° at Image Epoch of 2018-10-21

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	1.0	0.96	-	0.56	0.97
Rot. (θ)	0	1.2	-	30	0.57
Horz. (T_X)	502.8	504.2	503	505.5	391.1
Vert. (T_Y)	510.2	511.3	510	509.6	409.5
Fitness (c)	-	9.6e-06	-	-	-

Rotation -30°: To see how the method works with different scale errors and rotation, the template is shrunk, enlarged, and rotated. Here specifically, the template is rotated -30° from the scene object. A scale estimation of 2.0 indicates that the template is smaller by a factor of two than actual object (here 5 pixels wide). At this width of pixels, the object may appear as if it symmetric or even a point source. However, when the scale factor is 0.1, the template is 10 times larger than that of the scene object (≈100 pixels wide).

Table 3 shows the results of changing the scale of the template when it is rotated -30° from the scene object. Shrinking the template by a factor of 2 created too small of an object to see much of any blob detail. The template can be seen in Fig. 7. An object, such as this, lacks any sufficient detail for rotation and possibly shape. This can

Table 3. Transformation and Fitness Values for Template Rotation of -30° at Image Epoch of 2018-10-21

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	2.0	1.8	-	1.0	x
Rot. (θ)	30	0.0	-	180	x
Horz. (T_X)	504.9	502.1	505	505.2	x
Vert. (T_Y)	513.6	510.9	514	506.1	x
Fitness (c)	-	0.0062	-	-	-
Scale (K)	1.0	0.97	-	0.76	x
Rot. (θ)	30	27.3	-	66	x
Horz. (T_X)	502.3	507.3	503	502.2	x
Vert. (T_Y)	510.3	509	510	510.2	x
Fitness (c)	-	6.51e-05	-	-	-
Scale (K)	0.5	0.51	-	0.68	x
Rot. (θ)	30	27.7	-	16.4	x
Horz. (T_X)	495.1	506	-6	497.7	x
Vert. (T_Y)	503.3	507.9	-8	504.3	x
Fitness (c)	-	0.0003	-	-	-
Scale (K)	0.1	0.104	-	0.081	x
Rot. (θ)	30	28.8	-	-89.4	x
Horz. (T_X)	448.8	503.3	-19	395.6	x
Vert. (T_Y)	452.9	501.8	-29	314.4	x
Fitness (c)	-	0.133	-	-	-

be seen within the table. With such a small template, the demonstrated method was unable to correctly find the rotation. However, the approximation of the scale is within 10 percent of the actual scale, and the translation is only a few pixels off, due the scale and rotation error.

When examining the other scales, the method herein is able to estimate the scale accurately (less than 4 percent error) and the rotation within 9 percent. Note, the translation does not match the centroid compared translation. This is due to the method estimating the translation of the rotated and scaled object, not just pure translation between centers-of-brightness.

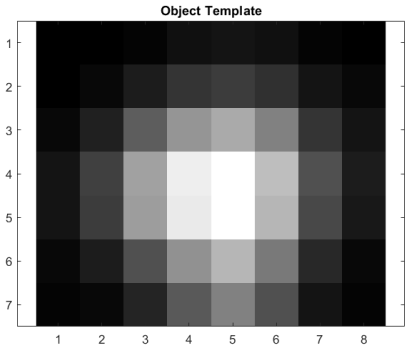

Figure 7. Template generated for Bennu on 2018-10-21, with a scale factor 0.5 and rotation of -30°

Cross-correlation performs well for finding the translation. However, once the template becomes too large, cross-correlation fails to find a reasonable translation solution. The Log-Polar formulation is able to estimate the translation well, until the template scale becomes too large. Furthermore, the SURF method fails to find any key-points and is unable to predict any parameters.

Bennu: 2018-10-25 Epoch . As the OSRIS-REx spacecraft continued to approach Bennu, the object filled more pixels on the sensor array. At this time, Bennu appeared to be approximately 20 pixels in diameter. A cropped image can be seen in Fig. 8. Shown in the figure, the object Bennu is starting to have more detail compared to what it was just four days prior. Similarly, a test case is conducted to see how all the methods fair with no rotation and scale difference. The results are given in Table 4.

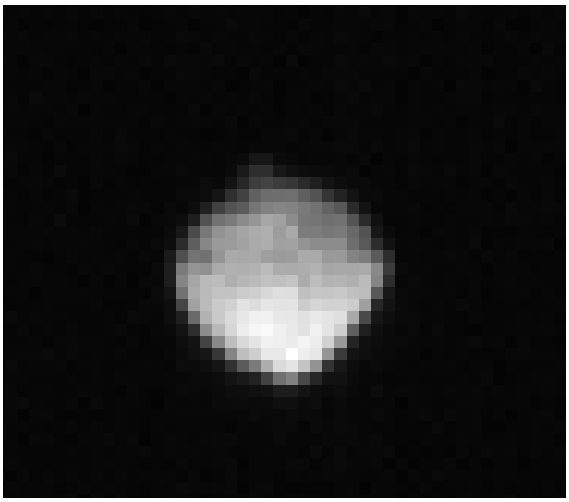

Figure 8. Cropped image of Bennu taken on 2018-10-25 using PolyCam

All methods other than SURF give a solution. Cross-correlation, as expected, finds exactly (to the nearest pixel) the correct translation from the original image. Log-Polar is able to determine a correct scale but does not provide a correct rotation and gives a slightly different translation value. The method described herein has a 1 percent error in scale, calculates a rotation of 1.8°, and has a similar translation to the true value. Reasons for the difference in translation are due to the slight difference in scale and the small calculated rotation.

Rotation 60°: To test the scale and rotation estimation given a more detailed object and template, the template is rotated by 60° and is subjected to the same scale changes as is done in the previous case. Note, when the scale is 0.1, the template object is approximately 200 pixels in diameter.

Table 5 provides the results from the rotation and scale tests. Cross-correlation is unable to determine the correct translation, except for when there is no scale difference between the template and scene object.

Table 4. Initial Transformation and Fitness Values for Template Rotation of 0° at Image Epoch of 2018-10-25

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	1.0	0.99	-	1	x
Rot. (θ)	0	1.8	-	18	x
Horz. (T_X)	492.9	494.5	493	489.5	x
Vert. (T_Y)	509.2	510	509	505.6	x
Fitness (c)	-	1.2e-05	-	-	-

Surprisingly, Log-Polar is able to determine a scale that is similar to the actual value, but fails to provide suitable rotations and, in instances other than when the scale is 1.0 case, translations. SURF is not able to register any key points. However, the method herein is able to find a solution for all cases. The largest scale difference is the first case in the table, where it also has the largest error in rotation, which is 4.3°. Otherwise, scale and rotation are estimated very well. Rotation estimates when the template is the size of Bennu or larger are less than 1.6°, and the scales have errors less than 2.1 percent, respectively.

Table 5. Transformation and Fitness Values for Template Rotation of 60° at Image Epoch of 2018-10-25

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	2.0	1.91	-	1.7	x
Rot. (θ)	300	304.3	-	0	x
Horz. (T_X)	498.8	489.8	1026	493	x
Vert. (T_Y)	515.6	527.7	-3	511	x
Fitness (c)	-	0.0015	-	-	-
Scale (K)	1.0	0.979	-	1.0	x
Rot. (θ)	300	298.4	-	-90	x
Horz. (T_X)	494.7	492.5	495	495.2	x
Vert. (T_Y)	510.2	527.7	510	510.9	x
Fitness (c)	-	8.48e-06	-	-	-
Scale (K)	0.5	0.4998	-	0.52	x
Rot. (θ)	300	299.1	-	-168.3	x
Horz. (T_X)	486.5	494.5	990	489.3	x
Vert. (T_Y)	499.3	525.6	-8	508.4	x
Fitness (c)	-	0.0001	-	-	-
Scale (K)	0.1	0.1003	-	0.14	x
Rot. (θ)	300	299.9	-	179.8	x
Horz. (T_X)	419.8	509.9	868	37.9	x
Vert. (T_Y)	409.7	517	-39	612.3	x
Fitness (c)	-	0.066	-	-	-

Bennu: 2018-10-28 Epoch. At this particular date during approach, the asteroid Bennu extended approximately 50 pixels in diameter. Fig. 9 gives a cropped image of what Bennu looked like on the image plane. Bennu started to show some surface features, and a large boulder could be seen on the lower left portion of the image. Surface shadows were even able to be seen.

For preliminary testing purposes, all methods attempted to provide information when there is neither rotation nor scale differences. These results can be seen in Table 6. As expected, cross-correlation finds the proper translation. SURF fails to find any matching key points, resulting in no parameters being estimated. Log-Polar is able to solve for all parameters. However, only the translation is close to the actual value. Considering the method herein, it is able to determine the scale within 0.1 percent, find the correct rotation (significant figure rounding), and estimate a translation that is within 1 pixel in either dimension.

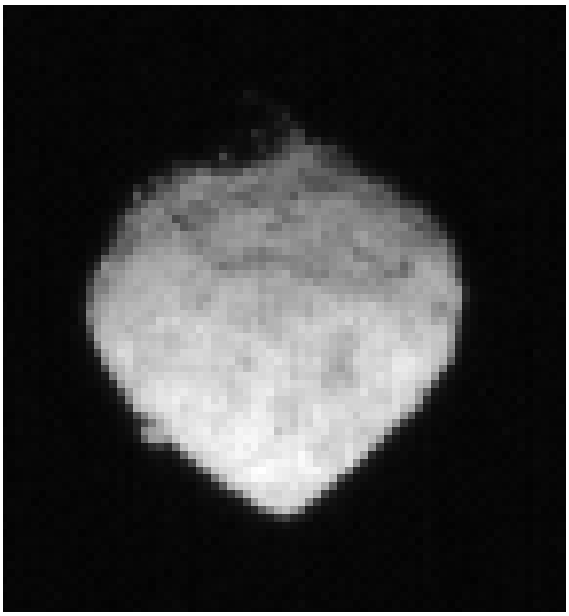

Figure 9. Cropped image of Bennu taken on 2018-10-28 using PolyCam

Table 6. Initial Transformation and Fitness Values for Template Rotation of 0° at Image Epoch of 2018-10-28

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	1.0	1.001	-	0.82	x
Rot. (θ)	0	0	-	57.9	x
Horz. (T_X)	458.4	459.4	458	454.5	x
Vert. (T_Y)	467.2	467.8	467	469.4	x
Fitness (c)	-	1.21e-05	-	-	-

Rotation 0°: In interest of testing capabilities, this scenario only scales the template; no rotation is introduced. When the template has a scale of 0.1, the object is approximately 500 pixels in diameter.
Table 7 shows the results for different template scales. The herein method estimates the scale well. However,

the rotation does not match well when the template is 10 times larger than the actual Bennu, having a difference of 3.75°. Regardless, this method was able to outperform all other methods tested (except for cross-correlation in the scale factor of 1.0 case).

Looking further into the cause of the rotation error for the last case, the template being 10 times larger, it was discovered that some of the initial Gaussian kernel standard deviations, rotated axes, and blob detection level numbers needed to be adjusted. However, no better solution was found when changing these values after the reported value in Table 7.

Table 7. Transformation and Fitness Values for Template Rotation of 0° at Image Epoch of 2018-10-28

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	2.0	1.996	-	1.53	x
Rot. (θ)	0	359.7	-	88.7	x
Horz. (T_X)	472.5	458.5	-6	464	x
Vert. (T_Y)	483.7	470.5	-8	483	x
Fitness (c)	-	0.004	-	-	-
Scale (K)	1.0	1.001	-	0.82	x
Rot. (θ)	0	0	-	57.9	x
Horz. (T_X)	458.4	459.4	458	454.5	x
Vert. (T_Y)	467.3	467.8	467	469.4	x
Fitness (c)	-	1.21e-05	-	-	-
Scale (K)	0.5	0.504	-	0.516	1.18
Rot. (θ)	0	359.6	-	-101.7	150.2
Horz. (T_X)	429.9	459.4	-17	456	454.7
Vert. (T_Y)	434	463.8	-31	463	595.6
Fitness (c)	-	0.0012	-	-	-
Scale (K)	0.1	0.10001	-	0.212	x
Rot. (θ)	0	356.25	-	119.5	x
Horz. (T_X)	201.6	455.7	-92	387.9	x
Vert. (T_Y)	167.8	425.7	-155	395.2	x
Fitness (c)	-	0.0036	-	-	-

Bennu: 2018-11-02 Epoch. The final scenario investigated occurs on November 2, 2018. Here, Bennu is approximately 200 pixels wide on PolyCam's sensor array. Fig. 10 shows what Bennu looked like. Surface shadowing, boulders, craters, and other features could be discerned at this time. However, these are not the finest resolution images taken by the OSIRIS-REx mission.

As with the other scenarios, the zero rotation and scale factor of 1.0 is applied for initial comparison. Table 8 gives the determined values from all four methods. Cross-correlation agrees very well with the actual value for translation. Log-Polar fails to find a solution. In this initial look, SURF is able to find a reasonable solution. The scale is within 3.0 percent; the rotation only has an error of 0.57°; translation values are only a few pixels different than the actual value. Considering the method described in this paper, scale error is only 0.2 percent, rotation differs by 0.02°, and either translation component is within 1.1 pixels.

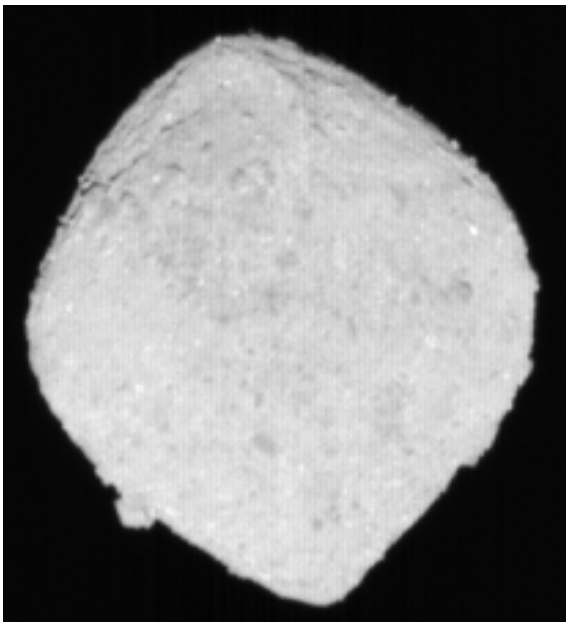

Figure 10. Cropped image of Bennu taken on 2018-11-02 using PolyCam

Table 8. Initial Transformation and Fitness Values for Template Rotation of 0° at Image Epoch of 2018-11-02

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	1.0	0.998	-	x	0.97
Rot. (θ)	0	0.018	-	x	0.57
Horz. (T_X)	386.7	387.8	387	x	391.1
Vert. (T_Y)	404.9	405.7	405	x	409.5
Fitness (c)	-	3.38e-05	-	-	-

Rotation 90°: In this final scenario, the template is rotated 90° from Bennu's actual orientation along the boresight direction. Scale variations range from two times smaller (100 pixels wide) to 10 times larger (2000 pixels wide) than Bennu's PolyCam size. A representation of the scale and rotation error for the 10 times scale case can be seen in Fig. 11.

Case results are found in Table 9. In the scenario, Log-Polar is able to find reasonable results in all cases. However, the estimated rotation for the 0.5 scale case differs greatly. Again, when the scale of the template deviates too much from the 1.0 scale scenario, cross-correlation is unable to determine the proper translation. SURF is unsuccessful in determining correct parameters. The method developed matches very well to scale and rotation. For this 90° rotation case, the estimated translation components vary greatly. This is due the translation of the scaled and rotated object. As it can be seen, the

values for the vertical component are much larger than that of the horizontal component. These differences are thought to be due to the 90° rotation.

Table 9. Transformation and Fitness Values for Template Rotation of 90° at Image Epoch of 2018-11-02

Fitted variable	Actual Value	This Work	Cross Corr.	Log Polar	Matlab SURF
Scale (K)	2.0	2.009	-	2.09	4.1
Rot. (θ)	270	269.7	-	-87.0	155.2
Horz. (T_X)	444	389.3	450	374	642
Vert. (T_Y)	461.3	630.3	517	394	984.6
Fitness (c)	-	7.79e-06	-	-	-
Scale (K)	1.0	1	-	0.98	3.24
Rot. (θ)	270	270	-	-89.2	-80.7
Horz. (T_X)	390	390.7	389	388.9	610.1
Vert. (T_Y)	405.7	628.3	404	406.8	173.9
Fitness (c)	-	5.96e-06	-	-	-
Scale (K)	0.5	0.5004	-	0.54	6.4
Rot. (θ)	270	270.01	-	-4.39	160.3
Horz. (T_X)	283	393.6	642	390.3	160.3
Vert. (T_Y)	295.5	626.4	-67	394.7	2367.8
Fitness (c)	-	0.0001	-	-	-
Scale (K)	0.1	0.1001	-	0.108	2.27
Rot. (θ)	270	270.08	-	-91.4	41.3
Horz. (T_X)	-581.4	407.6	-494.3	-385.6	-1395
Vert. (T_Y)	-593.2	616.1	-455.5	-389.8	1510
Fitness (c)	-	0.003	-	-	-

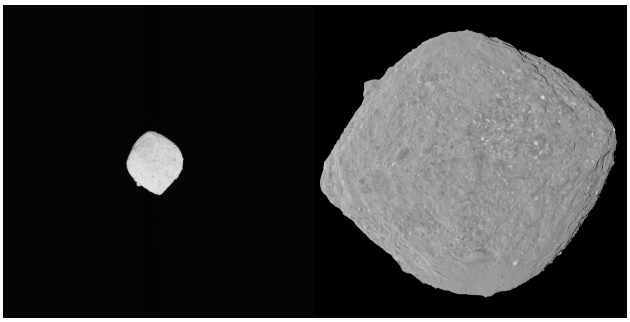

Figure 11. Image (left) and generated template (right) at 90° rotation and template 10 times larger than the image object

To have a visual sense of the described method's performance, Fig. 12 shows what the reconstructed image looks like for the 10 times scale difference case. The result of the reconstruction measures well when compared with the actual image. This is to be expected, since the amount of detail in Bennu matches well with the template.

Future Work. Future work involves further investigation into blob development and their bounders as well as blurring levels of the image. This will help create more unique shapes when the template or image objects

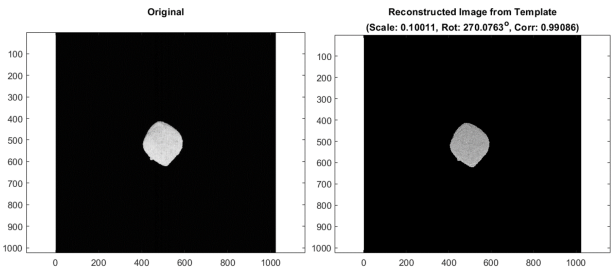

Figure 12. Image (left) and reconstructed object (right) from the scenario in Fig. 11

are small (less than 10 pixels in diameter). Implementing edge detection might help in reducing scenarios where the Gaussian filter eliminates true boundaries due to too large of a sigma value. Further research into better estimating rotation is also required. Finally, it is also desired to increase the speed at which the method executes.

Conclusion. A new formulation for image and template scale, rotation, and translation estimation is given. Scenarios involving real images from the OSIRIS-REx mission are included. This method is suitable for space mission imaging when an image and or template lack sufficient detail, have large boresight rotation, and/or involve large-scale errors. The method described outperforms more common techniques when large-scale errors and rotations are introduced. Further work needs to be done to investigate better ways to represent the bounders of blobs.

Acknowledgments. This material is based upon work supported by NASA under Contract NNM10AA11C issued through the New Frontiers Program. The authors are grateful to the entire OSIRIS-REx Team for making the encounter with Bennu possible.

References.

[1] F. Zhao, Q. Huang, and W. Gao, "Image matching by normalized cross-correlation," in 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 2, pp. II–II, IEEE, 2006.

[2] G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," in Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 1, pp. 493–496, IEEE, 2000.

[3] S. Zokai and G. Wolberg, "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations," IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1422–1434, 2005.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European conference on computer vision, pp. 404–417, Springer, 2006.

[6] M. C. Nolan, C. Magri, E. S. Howell, L. A. Benner, J. D. Giorgini, C. W. Hergenrother, R. S. Hudson, D. S. Lauretta, J.-L. Margot, S. J. Ostro, et al., "Shape model and surface properties of the OSIRIS-REx target asteroid (101955) Bennu from radar and lightcurve observations," Icarus, vol. 226, no. 1, pp. 629–640, 2013.

[7] N. Otsu, "A threshold selection method from gray-level histograms," IEEE transactions on systems, man, and cybernetics, vol. 9, no. 1, pp. 62–66, 1979.

[8] L. I. Smith, "A tutorial on principal components analysis," tech. rep., University of Otago, Department of Computer Science, University of Otago, PO Box 56, Dunedin, Otago, New Zealand, 2002.

[9] R. Bro, E. Acar, and T. G. Kolda, "Resolving the sign ambiguity in the singular value decomposition," Journal of Chemometrics: A Journal of the Chemometrics Society, vol. 22, no. 2, pp. 135–140, 2008.

[10] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," CVGIP: Graphical Models and Image Processing, vol. 54, no. 5, pp. 438–460, 1992.

[11] Z. Huang and J. Leng, "Analysis of hu's moment invariants on image scaling and rotation," in 2010 2nd International Conference on Computer Engineering and Technology, vol. 7, pp. V7–476, IEEE, 2010.

[12] P. G. Antreasian, M. C. Moreau, C. D. Adam, A. French, J. Geeraert, K. M. Getzandanner, D. E. Highsmith, J. M. Leonard, E. J. Lessac-Chenen, A. H. Levine, et al., "Early navigation performance of the OSIRIS-REx approach to Bennu," 2019.

[13] B. Rizk, C. D. d'Aubigny, D. Golish, C. Fellows, C. Merrill, P. Smith, M. Walker, J. Hendershot, J. Hancock, S. Bailey, et al., "Ocams: the OSIRIS-REx camera suite," Space Science Reviews, vol. 214, no. 1, p. 26, 2018.