AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
AIAA Guidance, Navigation, and Control Conference

Spacecraft Pose Estimation using Principal Component Analysis and a Monocular Camera

Jian-Feng Shi,[a] Steve Ulrich*
Carleton University, Ottawa, Ontario K1S 5B6, Canada
Stephane Ruel,†
Neptec Design Group Ltd., Kanata, Ontario K2K 1Y5, Canada

The method of Principal Components Analysis (PCA) is widely used in statistical data analysis for engineering and the sciences. It is an effective tool for reducing the dimensionality of datasets while retaining majority of the data information. This paper explores the method of using PCA for spacecraft pose estimation for the purpose of proximity operations, and adapts a novel kernel based PCA method called Euler-PCA to denoise spacecraft imagery using a single optical or thermal camera.

I. Introduction

AUTONOMOUS proximity operation (ProxOps) is taking on a more vital role in modern space missions such as rendezvous and docking, re-supply, on-orbit servicing, inspection, orbital debris removal, and space structure assembly.[1] A critical enabling technology for autonomous ProxOps is real-time client satellite (CS) pose estimation. The relative position and orientation, or pose, is computed between the client and the servicing vehicle using photo or thermal camera images. All missions flown in the past performing close range ProxOps have used cooperative targets on the CS, *i.e.* spacecraft equipped with calibrated fiducial markers. Amongst the various fiducial markers, circular pattern markers are the most popular visual cues used in space applications. Pose estimation resulting from the projection of elliptical to circular shapes is a well known geometry problem. Equation for circle pose estimation is provided by Shiu and Ahmad,[2] Guru,[3] Zheng *et al*.,[4] and Lu *et al*.[5] For example, Ogilvie *et al*.[6] show the use of circular pattern in the Orbital Express Demonstration Manipulator System (OEDMS) Probe Fixture Assembly (PFA) target. Unfortunately, while fiducial markers provide highly accurate aid to pose estimation, servicing missions for commercial satellites will unlikely to have these markers to be available on the CS due to non-standard industry requirements from satellite manufactures and desires to avoid higher production and launch costs. Furthermore, at distances of 100-200 meters away from the CS, the fiducial markers are too small to be effective as pose estimation aids. The use of vision-based cameras offers lower power, mass and volume requirements for the servicing spacecraft (SS).[7]

There are many methods for determining an object pose using image analysis. Non-model based approach include the use of stereo egomotion estimation,[8] optic flow,[9,10] and structure from motion (SfM).[11,12] Non-model based approach do not require knowledge of the target object but are challenged by the loss of particular tracking feature due to image condition or perspective projection. Another approach is model-based, this is known as *model-to-image registration*, where some knowledge of the target shape is available. These include matching shape outlines,[13,14] or by region-based pose estimation using model segmentation.[15,16] Alternatively, image features such as SIFT (Scale-Invariant Feature Transformation),[17] SURF (Speeded-Up Robust Features),[18] ORB (Oriented FAST and Rotated BRIEF)[19] or non-linear diffusion methods like AKAZE (Accelerated KAZE)[20] maybe extracted from training and testing images to solve for the *correspondence problem* using *perspective-n-points* (PnP) methods.[21–23] The aforementioned methods all require some features to be extracted from the captured image, and therefore can be vulnerable to the loss of these features due to image or motion conditions. Furthermore, feature extraction for technique like SIFT, can be computationally costly and PnP methods like $Soft$POSIT suffers from local minimum

[a]PhD Candidate, Department of Mechanical and Aerospace Engineering,1125 Colonel By Drive. Senior Member AIAA.
*Assistant Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.
†Director, Exploration Technology Programs, 302 Legget Drive.

trapping when trying to predict large angle misalignments. This study focus on the matching of the image itself rather than the image features while using the method of Principal Component Analysis (PCA) to reduce the data dimension.

The method of PCA, also known as Karhunen-Loéve (KL) transformation,[24] is widely use in statistical analysis over various fields such as meteorology, biosciences, medicine, and computer vision.[25] It is an effective tool to reduce data dimensions which was earliest described by Pearson[26] and Hotelling.[27] A large body of work in computer vision using PCA has been concentrated in facial recognition where Sirovich and Kirby first represented faces in the *eigenpictures*,[28] and Turk and Pentland developed techniques to track and recognize head images while coining the term *eigenfaces*.[29] The *eigenface* methodology has been widely duplicated for facial detection.[30] Following work by Turk and Pentland, a large body of research was developed to improve face detection by enhancement of the testing procedure through weighting,[31] 3D face reconstruction,[32,33] robustness to illumination and view,[34] generalized 2-D PCA (G2DPCA) for increase robustness,[35] and the use of kernels for non-linear feature extraction,[24,36–39] PCA has been adopted to iterative closest point (ICP)[40] methods for spacecraft pose estimation using LIDAR,[41] and using PCA to reduce image feature descriptor dimensions.[42] This study shall focus on an appearance-based algorithm that utilize the *parametric eigenspace* in reduction of spacecraft image dimensions.[43] Additionally, this study experiments with the use of a recently developed Kernel PCA (KPCA) method proposed by Liwicki *et al*.[39] called Euler-PCA (ePCA) for image denoising, where it uses a robust dissimilarity measure based on the Euler representation of complex numbers.

This paper is organized as follows, Sec. II provides basic theories on PCA, dimensional reduction, reduced calculation of the eigenspace using Singular Value Decomposition (SVD), pose selection criteria, and the algorithm for direct linear PCA. Section III explains the use of Euler representation of complex numbers for PCA and provide the algorithm for ePCA. Section IV provides definition for coordinate system and relative pose. Software architecture shall also be discussed in Sec. IV. Section V provide the experimental results and discussions for using PCA on the computer simulated Envisat model. Section VI discuss the hardware and 3D Computer-Aided Design (CAD) models used in generating the training and testing dataset. Section VII provide the results and discussions for a Radarsat scaled model pose estimation using ePCA and PCA. Finally, Sec. VIII concludes this study.

II. Pose Matching

The goal for PCA is to match the object from the camera image to a stored matrix of images that has been transformed to its eigenspaces. Assumptions for the baseline PCA are: each image contains one object only, the object is non-occulted, and the target object is viewed by a weak perspective. For spacecraft imagery, these assumptions are not unreasonable at the distance of 100-200 meters from the target CS, where typically only the CS is visible in the camera frame. Another requirement of PCA is the object must be completely visible, due to lighting conditions, this is a harder requirement to comply to. The work around to this challenge is to include reasonable number of known lighting variations in the training set. A more detailed discussion of the visibility requirement is provided in the results and discussion section (Sec. V and VII) of this paper. Finally, to have effective scale invariance, the images are normalized in size while the pixel values are normalized to one. This means the image frame is the minimum rectangle enclosing the largest appearance of the object.

Given an image of $\mathbf{I}$ with resolution $R \times C$, where the i^{th} and j^{th} pixel of $\mathbf{I}$ is normalized to $\hat{I}_{ij} = I_{ij}/\bar{I}$, where

$$\bar{I} = \left[\sum_{i=1}^{R} \sum_{j=1}^{C} I_{ij}^2 \right]^{1/2} \tag{1}$$

$\hat{\mathbf{I}}$ is resized into an array $\hat{\mathbf{x}}$, where

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{I}_{11} & \hat{I}_{12} & \dots & \hat{I}_{1C} & \hat{I}_{21} & \hat{I}_{22} & \dots & \hat{I}_{RC} \end{bmatrix}^T \tag{2}$$

The dimension of $\hat{\mathbf{x}}$ is the resoultion of the image $N = R \times C$. Given a training image sequence of M images, the mean value of $\hat{\mathbf{x}}$ is

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{j=1}^{N} \hat{\mathbf{x}}_j \tag{3}$$

The training data matrix $\mathbf{X}$ is formed as a $N \times M$ matrix where $\mathbf{X} \in \mathbb{R}^{N \times M}$

$$\mathbf{X} = \hat{\mathbf{X}} - \bar{\mathbf{x}}\mathbf{1}^T = \begin{bmatrix} \hat{\mathbf{x}}_1 & \dots & \hat{\mathbf{x}}_M \end{bmatrix} - \bar{\mathbf{x}}\mathbf{1}^T \tag{4}$$

where $\mathbf{1}$ is an array of ones of the size $M \times 1$. The covariance matrix of $\mathbf{X}$ is

$$\bar{\mathbf{Q}} = \frac{\mathbf{Q}}{M-1} = \frac{\mathbf{X}\mathbf{X}^T}{M-1} \tag{5}$$

Note $\bar{\mathbf{Q}}$ and $\mathbf{Q}$ has the same eigenvectors and therefore it is more common to use $\mathbf{Q}$ for direct PCA, and $\mathbf{Q} \in \mathbb{R}^{N \times N}$. The covariance matrix $\mathbf{Q}$ can be transformed into the eigenspace, by

$$\mathbf{\Lambda} = \mathbf{P}^T \mathbf{Q} \mathbf{P} \tag{6}$$

where $\mathbf{P}$ is the matrix of eigen-unit-vectors, $\mathbf{P} \in \mathbb{R}^{N \times N}$

$$\mathbf{P} = \begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 & \dots & \hat{\mathbf{e}}_N \end{bmatrix} \tag{7}$$

By *Eignspace Representation Theorem*,[43] the j^{th} normalized image array (columns from the full image matrix) can be reconstructed by a linear combination of the eigenspace unit vectors

$$\hat{\mathbf{x}}_j = \bar{\mathbf{x}} + \sum_{i=1}^{N} g_{ji} \hat{\mathbf{e}}_i \tag{8}$$

Proof
Transform $\mathbf{X}$ from Eq. (4) by applying the eigenspace transformation matrix $\mathbf{P}^T$,

$$\mathbf{G} = \mathbf{P}^T \mathbf{X} \tag{9}$$

hence,

$$\hat{\mathbf{X}} = \bar{\mathbf{x}} \mathbf{1}^T + \mathbf{P}\mathbf{G} \tag{10}$$

where $\mathbf{g}_j \in \mathbb{R}^N$ and $\mathbf{G} \in \mathbb{R}^{N \times M}$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \dots & \mathbf{g}_M \end{bmatrix} \tag{11}$$

The individual $\mathbf{g}_j$ is therefore the columns of $\mathbf{X}$ transformed into the eigenspace. Visually, the sets of training images containing various combination of pose and lighting conditions traces out a multi degree-of-freedom (DOF) manifold in the eigenspace represented by the $\mathbf{g}_j$ vectors.

Now consider two images for comparison n and m. The image correlation function of two images $\hat{\mathbf{I}}_n$ and $\hat{\mathbf{I}}_m$, is defined by the symbol $\circ$,

$$c = \hat{\mathbf{I}}_n \circ \hat{\mathbf{I}}_m \triangleq \sum_{i=1}^{R} \sum_{j=1}^{C} \hat{\mathbf{I}}_n(i,j) \hat{\mathbf{I}}_m(i,j) \tag{12}$$

the larger the value c, the more similar $\hat{\mathbf{I}}_n$ and $\hat{\mathbf{I}}_m$ is to each other. Equation 12 can be expressed by $\hat{\mathbf{x}}$ from Eq. (2) as the inner dot product,

$$c = \hat{\mathbf{I}}_n \circ \hat{\mathbf{I}}_m = \langle \hat{\mathbf{x}}_n, \hat{\mathbf{x}}_m \rangle = \hat{\mathbf{x}}_n^T \hat{\mathbf{x}}_m \tag{13}$$

where $\langle ., . \rangle$ is the inner dot product. From Eq. (13), the Euclidean distance between the two normalized images can be related to the image correlation by

$$c = 1 - \frac{1}{2} \|\hat{\mathbf{x}}_n - \hat{\mathbf{x}}_m\|_2^2 \tag{14}$$

where c is between 0 and 1. Therefore, the maximum correlation is the minimum Euclidean distance. Since the eigenspace $\mathbf{g}_j$ is a linear transformation of $\hat{\mathbf{x}}_j$ (ref. Eq. (9)), it is easily shown the Euclidean distance of eigenspace vectors is the Euclidean distance between the two images.

$$\|\mathbf{g}_n - \mathbf{g}_m\|_2^2 = \|\hat{\mathbf{x}}_n - \hat{\mathbf{x}}_m\|_2^2 \tag{15}$$

A. Dimensional Reduction

The advantage of converting the image data into eigenspace is the ability to reduce the so-called *curse of dimensionality*. Examining Eq. (8), only the *principal components* of $\mathbf{g}_j$ contains majority of the image information, the smaller insignificant terms can therefore be discarded as they will not affect the summation. When computing Eq. (6), the eigenvalue is ordered from large to small, this also order the eigenspace transformation matrix $\mathbf{P}$ and the components in the vector $\mathbf{g}_j$. Equation 8 can be represented to dimension K instead of N where $N \gg K$. The new smaller $\mathbf{P}$ and $\mathbf{G}$ matrices are denoted by $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{G}}$ as follows,

$$\tilde{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{e}}_1 & \hat{\mathbf{e}}_2 & \dots & \hat{\mathbf{e}}_K \end{bmatrix} \tag{16}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} \tilde{\mathbf{g}}_1 & \tilde{\mathbf{g}}_2 & \dots & \tilde{\mathbf{g}}_M \end{bmatrix} \tag{17}$$

where $\tilde{\mathbf{P}} \in \mathbb{R}^{N \times K}$, $\tilde{\mathbf{g}}_j \in \mathbb{R}^K$, and $\tilde{\mathbf{G}} \in \mathbb{R}^{K \times M}$. Experiments show the K dimension can be as low as 10 where as N is 76,800 for a low resolution 320x240 size image. This represent a significant reduction of processing needed compare to a direct comparison of images.

B. Low-rank Matrix Approximation

Although Eq. (6) appears tobe elegant, $\mathbf{P}$ cannot be quickly solved in practice as a $N \times N$ eigenspace basis vectrix given the enormous size of N. A low-rank matrix approximation is typically used to remedy this problem. Consider the singular-value-decomposition (SVD) of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{18}$$

Substitute Eq. (18) into Eq. (5) produces

$$\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{19}$$

Compare Eq. (19) with Eq. (6), $\mathbf{U}$ is the eigenspace vectrix $\mathbf{P}$, and $\mathbf{\Sigma}$ is the square root of the eigenvalue matrix $\mathbf{\Lambda}$ assuming $M \geq N$. Substituting Eq. (18) into Eq. (9) produces

$$\mathbf{G} = \mathbf{P}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{\Sigma}\mathbf{V}^T \tag{20}$$

The advantage of PCA is the ability to use only a very few number of principal components from the eigenspace that contains majority of the data information. Selecting the first K terms from $\mathbf{P}$ instead of the full N^{th} dimension, Eq. (20) becomes

$$\tilde{\mathbf{G}} = \tilde{\mathbf{P}}^T\mathbf{X} = \tilde{\mathbf{\Sigma}}\mathbf{V}^T \tag{21}$$

The low-rank matrix approximation[44] of Eq. (18) is therefore,

$$\mathbf{X} = \tilde{\mathbf{P}}\tilde{\mathbf{\Sigma}}\mathbf{V}^T \tag{22}$$

Equation (22) is solved using the *svds()* function from MATLAB[45,46] with K value input. Other methods for solving low-rank approximation SVD exists as provided by Rokhlin *et al.*[47] and Halko *et al.*[48]

C. Testing Image

Given a test image captured during operations, it is first converted into an image array $\hat{\mathbf{x}}_t$ by the same steps from Eq. (2). It is then transformed into the eigenspace by using the following,

$$\tilde{\mathbf{g}}_t = \tilde{\mathbf{P}}^T(\hat{\mathbf{x}}_t - \bar{\mathbf{x}}) \tag{23}$$

where $\tilde{\mathbf{g}}_t \in \mathbb{R}^K$. This is compared with all vector columns in the $\tilde{\mathbf{G}}$ matrix to determine the minimum Euclidean distance match $\tilde{\mathbf{g}}_\wedge \in \mathbb{R}^K$.

$$\tilde{\mathbf{g}}_\wedge = \arg\min_{\tilde{\mathbf{g}}_j \in \tilde{\mathbf{G}}} \|\tilde{\mathbf{g}}_t - \tilde{\mathbf{g}}_j\|_2^2 \tag{24}$$

The training image will have known pose tag associated with it. If the match falls within two nearest training images, the pose is linearly interpolated between the two poses.

D. Pose Matching Algorithm

The pose matching algorithm is separated in two parts: a training portion (ref. Algorithm 1) that is computed off-line prior to operational use, and a testing portion (ref. Algorithm 2) that computes the pose of the spacecraft in the captured image during operations. The second portion of the PCA software is computed in real-time (RT) with the input of every captured image frame.

Algorithm 1. PCA_LEARN

procedure COMPUTE_DATABASE($\mathbf{I}_j\,(j=1,\ldots,M)\,,K$)
  Obtain a set of M images by rotating an CAD representation of the CS from the SS camera
  Tag every image $\mathbf{I}_j$ with the associated pose information
  for all M image in the dataset **do**
    Segment CS from background if required
    Normalize the image by scale by cropping the CS image
    Normalize pixel intensity using Eq. (1)
    Convert the normalized image into $\hat{\mathbf{x}}$ representation by Eq. (2)
  Compute average image vector $\bar{\mathbf{x}}$ of the entire database by Eq. (3)
  Form image database $\mathbf{X}$ in accordance to Eq. (5)
  Perform low-rank matrix approximation SVD by Eq. (18), $\tilde{\mathbf{P}} = svds(\mathbf{X}, K)$
  Obtain $\tilde{\mathbf{G}}$ by transforming image database into eigenspace by Eq. (21)
  return $\bar{\mathbf{x}}$,$\tilde{\mathbf{G}}$,$\tilde{\mathbf{P}}$

Algorithm 2. PCA_TEST

procedure POSE_RECOGNITION($\mathbf{I}_t\,,\bar{\mathbf{x}}\,,\tilde{\mathbf{G}},\tilde{\mathbf{P}}$)
  Perform steps 5 to 7 from Algorithm 1 on captured image $\mathbf{I}_t$ and form $\hat{\mathbf{x}}_t$
  Compute eigenspace representation of captured image $\tilde{\mathbf{g}}_t$ using Eq. (23)
  Find minimum matching $\tilde{\mathbf{g}}_j$ in accordance to Eq. (24)
  Compare dataset image $j+1$ and $j-1$ for second minimum match
  Interpolate pose ($\mathbf{t}$ and $\mathbf{R}$) between minimum match and second minimum match
  return $\mathbf{t}$,$\mathbf{R}$

III. Image Denoising

This study leverages the breakthroughs from kernel-based machine learning and support vector machine (SVM).[49–51] In order to gain robustness, the image is denoised by using non-linear feature extracting KPCA and by operating in high-dimensional feature spaces via the use of pre-images.[36,37] The kernel machine maps the input space $\hat{\mathbf{x}}$ into a higher dimensional feature space or *reproducing kernel Hilbert space* (rkHs) $\mathcal{H}$ using a non-linear function $\phi(\hat{\mathbf{x}}_j)$. The so called *kernel trick*, exploits the similarity measure between images by using the inner dot product.[24] In general, the kernel κ can be formulated in rkHs as

$$\kappa(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_m) = \langle \phi(\hat{\mathbf{x}}_n), \phi(\hat{\mathbf{x}}_m) \rangle_{\mathcal{H}} \tag{25}$$

The objective then, is to improve the baseline pose estimation by denoising the image data. This is done by solving an optimization problem[37] such that,

$$\hat{\mathbf{x}}_\wedge = \arg\min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \|\phi(\hat{\mathbf{x}}_l) - \boldsymbol{\beta}\boldsymbol{\beta}^T \phi(\hat{\mathbf{x}}_j)\|_2^2 \tag{26}$$

where $\boldsymbol{\beta}$ is a linear subspace orthonormal basis in rkHs. Some commonly used Kernels that are projective (polynomial, exponential) or radial (Gaussian, Laplacian) can be found in a recent survey of KPCA by Honeine and Richard.[24] The Euler-PCA (ePCA) is a KPCA based on the Euler representation of complex numbers introduced by Liwicki *et al.* (2013).[39] In ePCA rkHs, the l_2-norm can be related to a dissimilarity measure originally introduced by Fitch *et al.*[52] (see Sec. A for more details), this effectively provide a fast and statistically more robust estimation. In ePCA, the normalized image $\hat{\mathbf{x}}$ is mapped onto a complex number representation $\mathbf{z}_j \in \mathbb{C}^N$

$$\mathbf{z}_j = \frac{1}{\sqrt{2}} e^{i\alpha\pi\hat{\mathbf{x}}_j} = \frac{1}{\sqrt{2}} \left(\cos(\alpha\pi\hat{\mathbf{x}}_j) + i\sin(\alpha\pi\hat{\mathbf{x}}_j)\right) \tag{27}$$

Grouping $\mathbf{z}_j$ gives the data matrix $\mathbf{Z} \in \mathbb{C}^{N\times M}$ in rkHs

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_M \end{bmatrix} \tag{28}$$

The Hermitian kernel κ for ePCA is represented by $\mathbf{K} \in \mathbb{C}^{M\times M}$ and can then be formulated as

$$\mathbf{K} = \mathbf{Z}^H\mathbf{Z} \tag{29}$$

where $(.)^H$ is the complex conjugate transposition of a matrix. The eigen decomposition of Eq. (29) is

$$\mathbf{\Lambda}_c = \mathbf{P}_c^H\mathbf{K}\mathbf{P}_c \tag{30}$$

where $\mathbf{P}_c \in \mathbb{C}^{M\times M}$. Using similar method as Eq. (22), the lower dimension $\tilde{\mathbf{P}}_c \in \mathbb{C}^{M\times R}$ is found,

$$\mathbf{K} = \tilde{\mathbf{P}}_c\tilde{\mathbf{\Sigma}}_c\mathbf{V}_c^H \tag{31}$$

Using $\tilde{\mathbf{P}}_c$ in Eq. (30), the lower dimension $\tilde{\mathbf{\Lambda}}_c \in \mathbb{C}^{R\times R}$ can be found

$$\tilde{\mathbf{\Lambda}}_c = \tilde{\mathbf{P}}_c^H\mathbf{K}\tilde{\mathbf{P}}_c \tag{32}$$

Note that for $M \geq R$, $\tilde{\mathbf{\Sigma}}_c$ and $\tilde{\mathbf{\Lambda}}_c^{1/2}$ contains the same eigenvalues. The basis mapping $\boldsymbol{\beta}$ for ePCA is represented by $\tilde{\mathbf{B}} \in \mathbb{C}^{N\times R}$ and is formulated as

$$\tilde{\mathbf{B}} = \mathbf{Z}\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c^{-1/2} \tag{33}$$

Note with this formulation of $\tilde{\mathbf{B}}$, $\tilde{\mathbf{B}}$ is the eigenspace mapping of $\tilde{\mathbf{K}} \in \mathbb{C}^{N\times N}$, where

$$\tilde{\mathbf{K}} = \mathbf{Z}\mathbf{Z}^H \tag{34}$$

hence,

$$\begin{aligned} \tilde{\mathbf{B}}^H\tilde{\mathbf{K}}\tilde{\mathbf{B}} &= \tilde{\mathbf{\Lambda}}_c^{-1/2}\tilde{\mathbf{P}}_c^H\mathbf{Z}^H\mathbf{Z}\mathbf{Z}^H\mathbf{Z}\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c^{-1/2} \\ &= \tilde{\mathbf{\Lambda}}_c^{-1/2}\tilde{\mathbf{P}}_c^H\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c\tilde{\mathbf{P}}_c^H\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c\tilde{\mathbf{P}}_c^H\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c^{-1/2} \\ &= \tilde{\mathbf{\Lambda}}_c^{-1/2}\tilde{\mathbf{\Lambda}}_c\tilde{\mathbf{\Lambda}}_c\tilde{\mathbf{\Lambda}}_c^{-1/2} \\ &= \tilde{\mathbf{\Lambda}}_c^{1/2}\tilde{\mathbf{\Lambda}}_c^{1/2} \\ &= \tilde{\mathbf{\Lambda}}_c \end{aligned} \tag{35}$$

Hence, $\tilde{\mathbf{B}}$ is also an eigenspace basis vectrix, and Eq. (35) is the equivalence to Eq. (32). Finally, the reconstruction of $\breve{\mathbf{z}}_j \in \mathbb{C}^N$ is performed,

$$\breve{\mathbf{z}}_j = \tilde{\mathbf{B}}\tilde{\mathbf{B}}^H\mathbf{z}_j \tag{36}$$

in this form, one may perform the denoising based on dissimilarity to be discussed in the next section (Sec. A). Furthermore, the advantage of the ePCA construction becomes apparent when the inverse from the rkHs (Euler complex number feature space) back to the image space is performed. Sec. B, provides a close form transformation function as result of the ePCA formulation.

A. Dissimilarity Measure

This section discuss the motivation for the ePCA transformation. The l_2-norm in Eq. (24) is optimal for *independent and identically distributed* (iid) Gaussian noise but not robust to outliers.[53–55] Previous approach found robustness in using the l_1-norm instead of the l_2-norm.[54,56] Define $\boldsymbol{\theta}_n \triangleq \alpha\pi\hat{\mathbf{x}}_n$. A fast robust correlation scheme by Fitch *et al.* ,[52] suggest the use of the following so called *dissimilarity measure*,

$$\delta \triangleq \sum_{r=1}^{N} \left(1 - \cos\left(\theta_l\left(r\right) - \theta_j\left(r\right)\right)\right) \tag{37}$$

A similarity can be found by applying the l_2-norm in the Euler complex number representation space by the following,

$$\begin{aligned} \|\mathbf{z}_l - \mathbf{z}_j\|_2^2 &= \frac{1}{2}\|\cos\left(\boldsymbol{\theta}_l\right) + i\sin(\boldsymbol{\theta}_l) - \cos\left(\boldsymbol{\theta}_j\right) - i\sin(\boldsymbol{\theta}_j)\|_2^2 \\ &= \sum_{r=1}^{N} e^{i\left(\theta_l(r)+\theta_j(r)\right)}\left(1 - \cos\left(\theta_l\left(r\right) - \theta_j\left(r\right)\right)\right) \end{aligned} \tag{38}$$

The optimizing of Eq. (38), is essentially the ePCA form of Eq. (26). This is the basis for the denoising pre-image computation step that is performed in the following section (Sec. B).

B. Pre-image Computation

Typically, the method to convert from the rkHs to the optimized input image space is to perform the numerical method of gradient ascent. Liwicki *et al.*[39] showed the ePCA procedures for this optimization. This is done by reforming Eq. (26) in the argument of the maxima form

$$\hat{\mathbf{x}}_{\wedge} = \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re\left(\phi\left(\hat{\mathbf{x}}_l\right)^H \boldsymbol{\beta}\boldsymbol{\beta}^H \phi\left(\hat{\mathbf{x}}_j\right)\right) \tag{39}$$

where $\Re(.)$ outputs the real number of the argument. In ePCA form, $\boldsymbol{\beta} \triangleq \mathbf{B} = \mathbf{Z}\hat{\mathbf{B}}$, then Eq. (39) becomes

$$\hat{\mathbf{x}}_{\wedge} = \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re\left(\mathbf{z}\left(\hat{\mathbf{x}}_l\right)^H \mathbf{Z}\left(\mathbf{X}\right)\hat{\mathbf{B}}\hat{\mathbf{B}}^H \mathbf{Z}\left(\mathbf{X}\right)^H \mathbf{z}\left(\hat{\mathbf{x}}_j\right)\right) \tag{40}$$

Applying the kernel trick,[24] Eq. (40) becomes

$$\begin{aligned}\hat{\mathbf{x}}_{\wedge} &= \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} \Re\left(\left[\boldsymbol{\kappa}\left(\hat{\mathbf{x}}_l, \hat{\mathbf{x}}_1\right) \ldots \boldsymbol{\kappa}\left(\hat{\mathbf{x}}_l, \hat{\mathbf{x}}_M\right)\right] \underbrace{\hat{\mathbf{B}}\hat{\mathbf{B}}^H \left[\boldsymbol{\kappa}\left(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_1\right) \ldots \boldsymbol{\kappa}\left(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_M\right)\right]^H}_{\triangleq \mathbf{t}}\right) \\ &= \arg\max_{\hat{\mathbf{x}}_l \in \mathbf{X}} f\left(\hat{\mathbf{x}}_l\right)\end{aligned} \tag{41}$$

to maximize Eq. (41), the standard Newton's method is given as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \bigtriangledown f\left(\hat{\mathbf{x}}_{k-1}\right) \tag{42}$$

The number of steps to complete Eq. (42) is denoted as h. For completeness, the gradient of f is given as

$$\bigtriangledown f\left(\hat{\mathbf{x}}_l\right) = -\Im\left(\mathbf{H}\right)\Re\left(\mathbf{t}\right) + \Re\left(\mathbf{H}\right)\Im\left(\mathbf{t}\right) \tag{43}$$

where $\bigtriangledown(.)$ is the gradient operator, $\Im(.)$ is the imaginary portion of the argument, and $\mathbf{H} \in \mathbb{C}^{N \times M}$,

$$\mathbf{H} = \bar{\mathbf{z}}_l \mathbf{1}^T \odot \mathbf{Z} \tag{44}$$

where $\bar{\mathbf{z}}_l$ is the complex conjugate of $\mathbf{z}_l$, and $\odot$ is the Hadamard point-wise operator. Equations 41 to 44 will denoise the input as it maps the feature space image back to input space. However, the cost of this computation for the entire training set is costly, on the order of $\mathcal{O}\left(hNM^2\right)$. One of the advantage of the ePCA feature space is the ability to optimally transform back to the pixel domain by a close formulation base simply on the angle of the complex number $\breve{\mathbf{Z}}$,

$$\breve{\mathbf{x}}_j = \frac{\angle \breve{\mathbf{z}}_j}{\alpha\pi} \tag{45}$$

where $\angle(.)$ computes the angle of the complex number. For the same value of $\hat{\mathbf{x}}_l = \hat{\mathbf{x}}_j = \hat{\mathbf{x}}$, the rkHs representation is $\phi\left(\hat{\mathbf{x}}\right) = \mathbf{z}$. The optimality can then be shown by combining Eq. (36), Eq. (45), and Eq. (26).

$$\begin{aligned}\hat{\mathbf{x}}_{\wedge} &= \arg\min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \| \frac{1}{\sqrt{2}} e^{i\angle\left(\mathbf{B}\mathbf{B}^H\mathbf{z}\right)} - \mathbf{B}\mathbf{B}^H\mathbf{z} \|_2^2 \\ &= \arg\min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \sum_{r=1}^{N} \left(\frac{1}{\sqrt{2}} e^{i\angle\left(\mathbf{B}_r\mathbf{B}_r^H z_r\right)} - R\left(\mathbf{B}_r\mathbf{B}_r^H z_r\right) e^{i\angle\left(\mathbf{B}_r\mathbf{B}_r^H z_r\right)}\right)^2 \\ &= \arg\min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \sum_{r=1}^{N} \left(\frac{1}{\sqrt{2}} - R\left(\mathbf{B}_r\mathbf{B}_r^H z_r\right)\right)^2 \\ &= \arg\min_{\hat{\mathbf{x}}_l \in \mathbf{X}} \| \frac{1}{\sqrt{2}}\mathbf{1} - R\left(\mathbf{B}\mathbf{B}^H\mathbf{z}\right) \|_2^2\end{aligned} \tag{46}$$

where $R(.)$ is the length of the complex number *phasor*. Note that $\mathbf{1}$ here is the array of ones with dimension $N \times 1$. For all image in data bin, replace $\mathbf{z}_j$ with $\mathbf{Z}$ from Eq. (28), and substitute Eq. (33),

$$\begin{aligned} R\left(\mathbf{B}\mathbf{B}^H Z\right) &= R\left(\mathbf{Z}\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c^{-1/2}\tilde{\mathbf{\Lambda}}_c^{-1/2}\tilde{\mathbf{P}}_c^H\mathbf{Z}^H\mathbf{Z}\right) \\ &= R\left(\mathbf{Z}\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c^{-1}\tilde{\mathbf{P}}_c^H\mathbf{Z}^H\mathbf{Z}\right) \\ &= R\left(\mathbf{Z}\left(\tilde{\mathbf{P}}_c\tilde{\mathbf{\Lambda}}_c\tilde{\mathbf{P}}_c^H\right)^{-1}\mathbf{K}\right) \\ &= R\left(\mathbf{Z}\mathbf{K}^{-1}\mathbf{K}\right) \\ &= R\left(\mathbf{Z}\right) \\ &= \sqrt{\Re(\mathbf{Z})^2 + \Im(\mathbf{Z})^2} \\ &= \sqrt{\left(\frac{\cos\mathbf{\Theta}}{\sqrt{2}}\right)^2 + \left(\frac{\sin\mathbf{\Theta}}{\sqrt{2}}\right)^2} \\ &= \frac{1}{\sqrt{2}}\mathbf{1} \end{aligned} \tag{47}$$

where $\mathbf{1}$ is a matrix of ones of the dimension $N \times M$ and $\mathbf{\Theta} \in \mathbb{R}^{N\times M}$ is

$$\mathbf{\Theta} = [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_M] = \alpha\pi\left[\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_M\right] = \alpha\pi\hat{\mathbf{X}} \tag{48}$$

Compare Eq. (47) and Eq. (46), the minimum error distance is always zero for the same input image vector. Hence, Eq. (45) is the exact solution for the approximate optimal inverse mapping.

C. Denoising Algorithm

In this section, Algorithm 3 and Algorithm 4 shall be provided for denoising of the training dataset and the testing images respectively.

Algorithm 3. ePCA_TRAIN

procedure TRAIN_IMG_DENOISE($\mathbf{I}_j\,(j = 1, \ldots, M)$,α,R)
  Perform steps 2 to 8 from Algorithm 1
  Compute $\mathbf{z}_j$ in accordance to Eq. (27) and form $\mathbf{Z}$ per Eq. (28)
  Compute kernel matrix $\mathbf{K}$ per Eq. (29)
  Compute $\tilde{\mathbf{P}}_c$ using $\left[\tilde{\mathbf{P}}_c, \tilde{\mathbf{\Sigma}}_c\right] = svds(\mathbf{K}, R)$
  Set $\tilde{\mathbf{\Lambda}}_c$ from $\tilde{\mathbf{\Sigma}}_c$
  Compute $\tilde{\mathbf{B}}$ from Eq. (33)
  Compute $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^H$ and construct $\breve{\mathbf{Z}}$ from Eq. (36) and Eq. (28)
  Compute pre-image $\breve{\mathbf{X}}$ using Eq. (45)
  return $\breve{\mathbf{X}}$,$\tilde{\mathbf{B}}$,$\tilde{\mathbf{\Sigma}}$

Algorithm 4. ePCA_TEST

procedure TEST_IMG_DENOISE($\mathbf{I}_t$,α,$\tilde{\mathbf{B}}$)
  Perform steps 2 from Algorithm 2 and form $\hat{\mathbf{x}}_t$
  Compute $\mathbf{z}_t$ using Eq. (27)
  Compute $\breve{\mathbf{z}}_t$ using Eq. (36)
  Compute $\breve{\mathbf{x}}_t$ using Eq. (45)
  return $\breve{\mathbf{x}}_t$

IV. Coordinate System and Software Architecture

This section describes the pose estimation coordinate system definition and software architecture.

A. Coordinate System Definition

This section describes the coordinate system used by the pose estimation software. Two Local Vertical Local Horizontal (LVLH) orbital frames are defined as $\mathcal{F}_{CO}$ and $\mathcal{F}_{SO}$ for the client satellite (CS) and the servicing spacecraft (SS) respectively. The LVLH frame has their center fixed to the spacecraft orbit, x-axis pointed in the orbital path, z-axis pointed towards Earth J2000 inertial $\mathcal{F}_I$, and y-axis completing the right handed coordinate system. The spacecraft body coordinate system, $\mathcal{F}_{CB}$ and $\mathcal{F}_{SB}$ for CS and SS respectively, is fixed to the spacecraft body and centered on the spacecraft Center of Mass (COM) coinciding with the LVLH coordinate frame. A standard camera coordinate system $\mathcal{F}_{VW}$ is attached to the SS and has its z-axis pointing out from the camera boresight, y-axis pointing vertically down from the image center and x-axis making the right hand. The pixel coordinate (0,0) center is located at the upper left corner of the image plane. Pointing out from the camera boresight, the image x-axis pointing horizontally to the right and the y-axis pointing vertically down.

The relative pose is defined as the position and orientation difference between the camera coordinate $\mathcal{F}_{VW}$ and the CS body coordinate. The position, $\mathbf{t} = \mathbf{r}_{VW}^{VW,CB}$, is a vector beginning from $\mathcal{F}_{VW}$ and ending in $\mathcal{F}_{CB}$ expressed in the $\mathcal{F}_{VW}$ frame. The orientation, $\mathbf{R} = \mathbf{R}^{VW,CB}$, is the rotation matrix rotating a vector from the $\mathcal{F}_{CB}$ frame to the $\mathcal{F}_{VW}$ frame. The rotation matrix can be represented in as Euler angles of the pitch-yaw-roll rotation sequence, where roll is about the x-axis, pitch about the y, yaw about the z about an specified frame. Any point of interest on the CS is denoted by PW_k for the k^{th} point. It is a position vector, $\mathbf{r}_{CB}^{CB,PW_k}$, defined from $\mathcal{F}_{CB}$ to PW_k expressed in the CB frame. $\mathbf{r}_{CB}^{CB,PW_k}$ is rigidly fixed to the CS body frame. Figures 1 and 2 shows the various coordinate systems and the pose definition respectively, where f denotes the focus length of the camera.

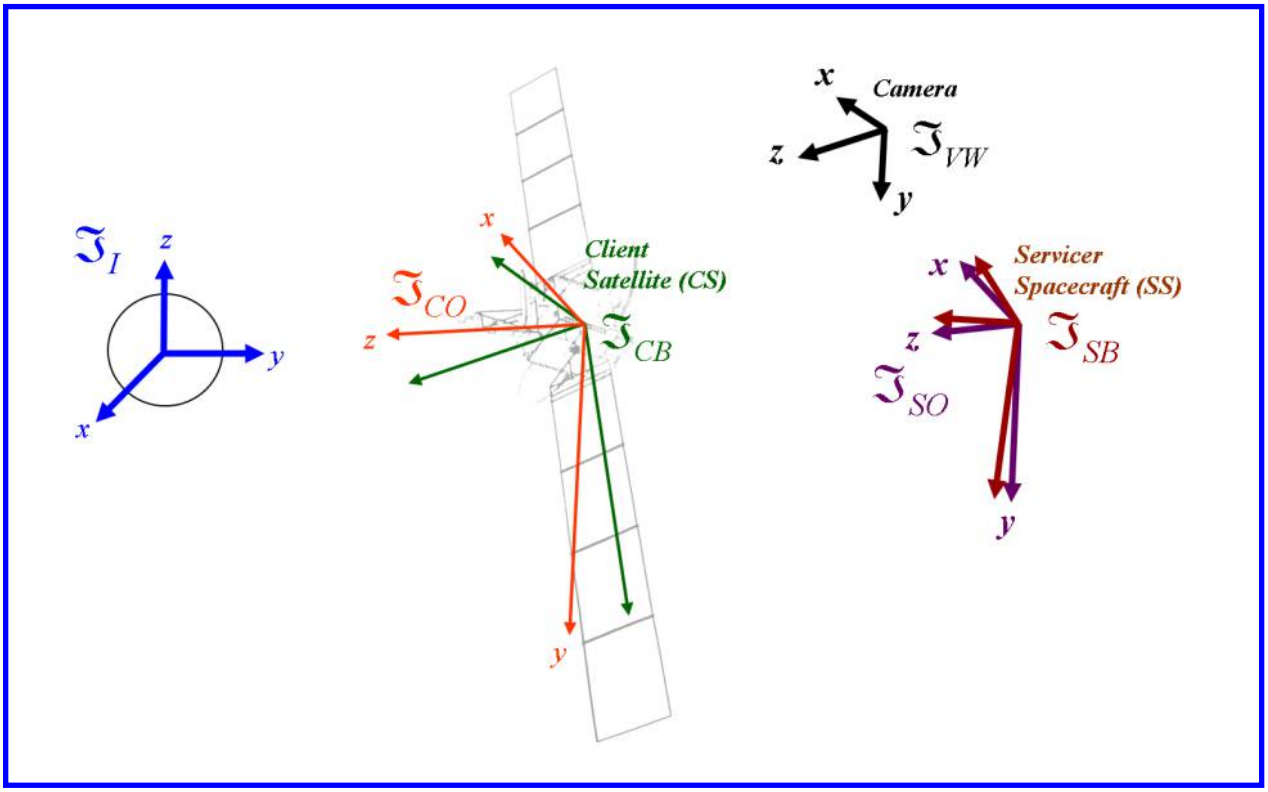


Figure 1. Coordinate system definition.

B. Software Architecture

Pose estimation software is composed of two parts, a training software and a testing software. The training software is ran prior to RT operations to process and denoise the training images and determine the denoising mapping function. After processing, the training software stores the eigenspace vectrix and reduced principal component image to disk so it can be retrieved by the testing software. The testing software is ran in RT to process and denoise the captured image and determine the pose of the CS vehicle. Figures 3 and 4 provides the training and testing software sequences respectively. In the processing step prior to PCA denoising, a Contrast Limiting Adaptive Histogram Equalization

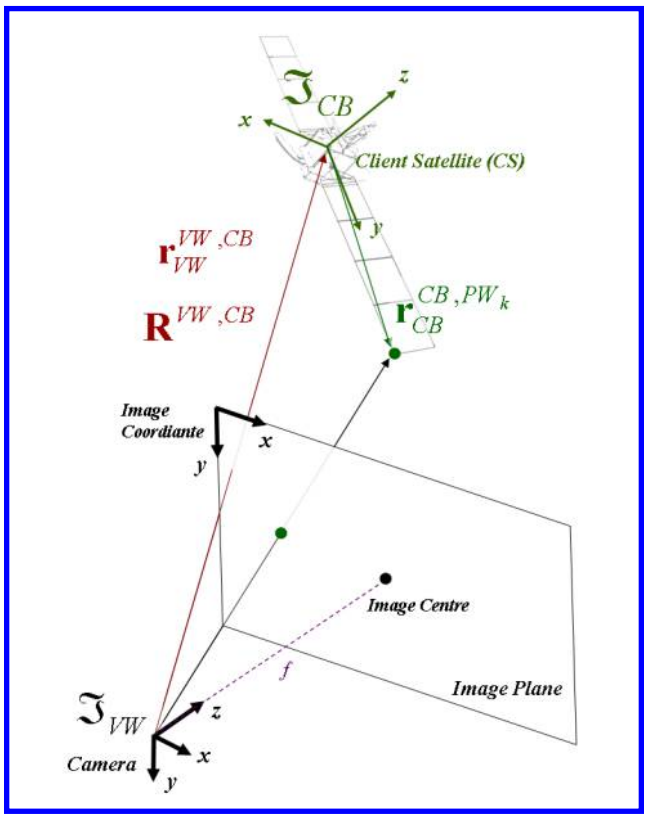


Figure 2. Relative pose definition.

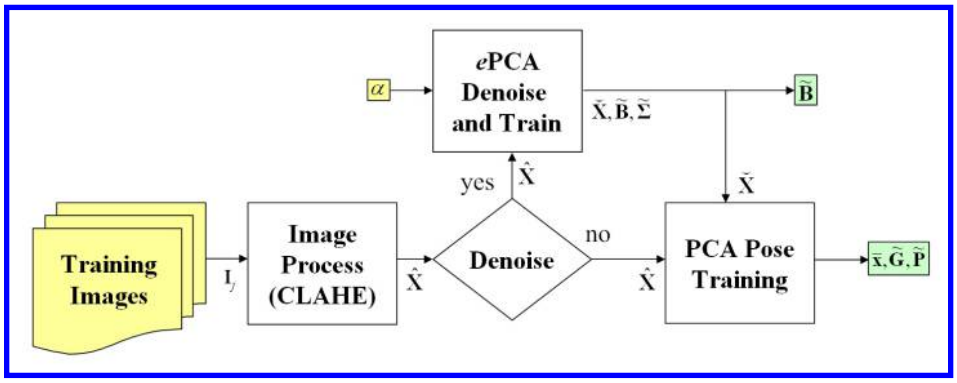


Figure 3. Training software.

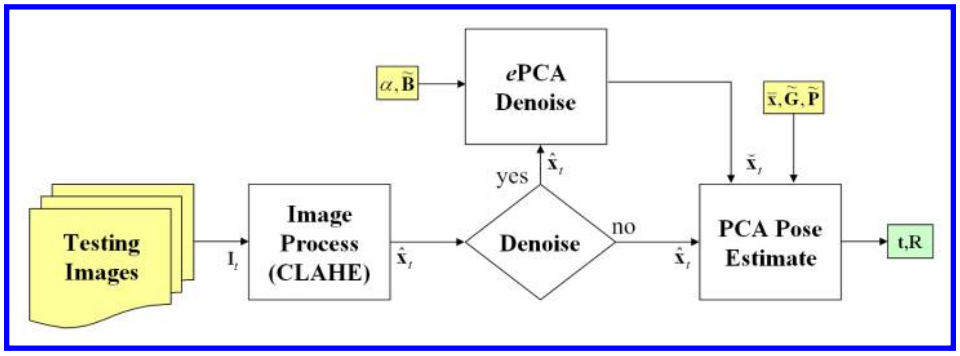


Figure 4. Testing software.

(CLAHE)[57–60] image processing method is used to enhance the input image. The CLAHE is shown to be a favorable method under low illumination conditions. Unlike a linear histogram equalization (HE), CLAHE can adjust the image histogram for various regions in the image even with the high and low histogram boundaries of the entire image already maximized. Figure 5 shows the benefits of the CLAHE image processing for various illumination conditions in a infrared camera imagery.

(a) Original (b) Normal HE (c) CLAHE 8x8

Figure 5. Image Processing using CLAHE

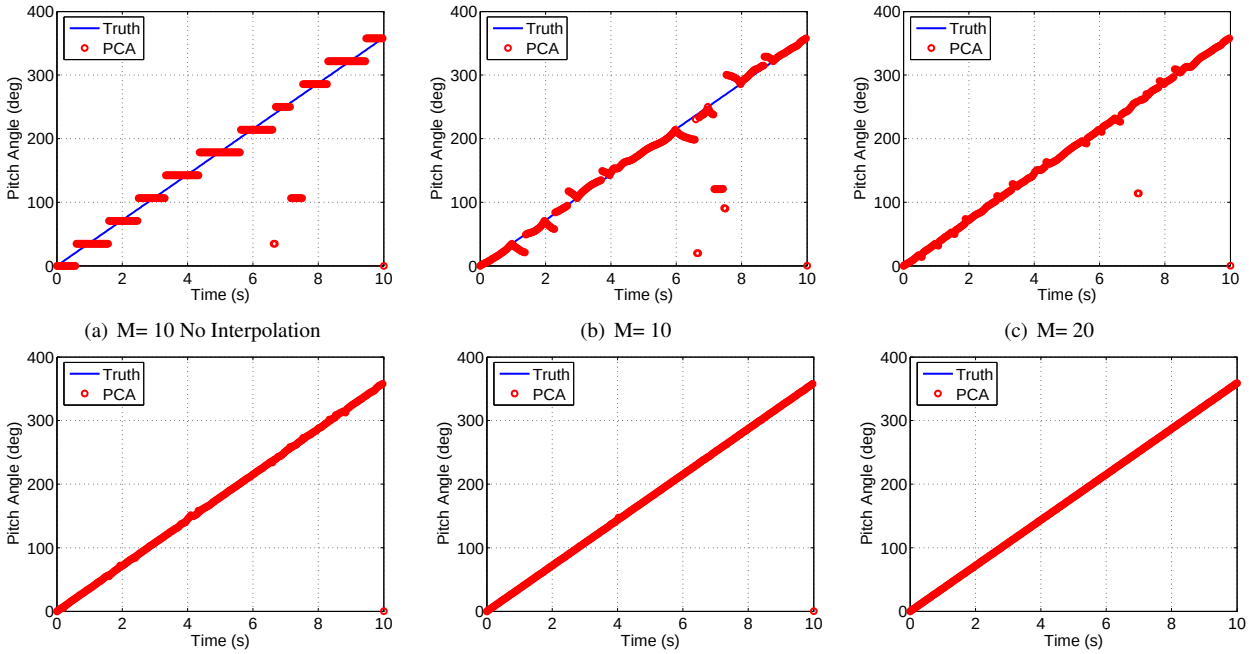
V. Numerical Simulation Results

This section provide discussion of the pose estimation results for synthetic image and real thermal camera images. Additionally, denoising results using ePCA is performed on synthetic images and real space images.[a]

A. Single-Axis Pose Results

Starting with single axis pitch rotation. Two parameters that will affect the pose estimation are the total number of training frames used (M) and the maximum principal components retained (K). In the former, increasing the total number of training images will increase estimation accuracy as each test frame have a closer point to compare to, this will also reduce ambiguity between similar pose. The trade-off is with increasing training set, the search time will increase. As pose DOF increase, the order of magnitude will increase by the total training frame per axis to the power of the pose DOF. Pose estimation (non-denoised) results for increasing number of training frames are shown in Fig. (6) for M equal to 10 to 300 frames respectively. Once matched, an interpolation step is needed to more precisely determine the pose position, this is especially important if the number of training images are low, resulting in a coarse resolution in the pose estimator. To interpolate the pose result, two nearest matches are selected after thresholding an angular tolerance, since angular pattern will wrap around in 360 degrees. The angular tolerance should be set on the order of the pose resolution based on number of training images used. For example, an angular tolerance of 45 degrees is used for the minimum M of 10 training frames spanning 360 degrees. Figures 6(a) and 6(b) shows estimation without and with interpolation. If interpolation is not performed, the pose value jumps from closest frames which results in a step pattern when the training set is coarse (*i.e.* $M = 10$). The interpolation caused an overhead on the computation that amounts to 0.7% increase to the total processing time for all 300 test frames. Furthermore, when the training set is coarse, it is more likely to result in wrongful matches in a mirror angle. This is evident in Fig. (6(a)), at every 30 degrees the choices of two neighbours -30 and +30 degrees is presented and the wrongful choice resulting in the interpolation to run backwards. This issue will correct itself as M increase in value as shown in Fig. (6). Figure 7 and Tbl. (1) shows the timing and error metric for the single axis Envisat example. The average single image testing time is the time it takes to search the database of training image principal components. This time will increase linearly with the increase of testing images, for $M = 300$, the maximum search time is $14ms$. Some outliers (OL) will be caused by wrongful matches, these outliers can be removed using the same angle tolerance. One common outlier is the 360 deg wrapping; if not removed, it will result in the maximum error ceiling for all M image trials. Figure 7 shows a large reduction in maximum angular error from 37 degrees to 5 degrees when number of training images is increased from 10 to 30 frames (a decrease of 85%), this benefit have a worthwhile cost of roughly 7% increase in the computation time.

[a]All simulations are performed on Intel® Core™ 2 Quad Q6600-2.4GHz Processor running on 32bit Windows-Vista-SP2.



(a) M= 10 No Interpolation (b) M= 10 (c) M= 20

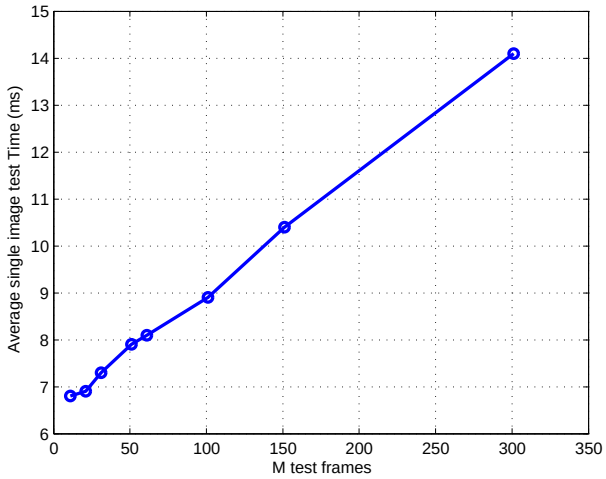
(d) M= 50 (e) M=100 (f) M=300

Figure 6. Pitch axis pose estimation vs. ground truth

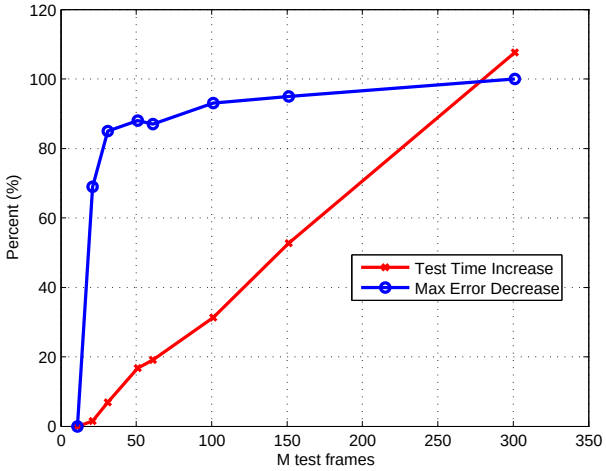
Number of Training Frames	Avg Est Time per Frame (ms)	Est. Time Percent Increase (%)	Mean Ang Err (deg)	Max Ang Err (deg)	Max Err Percent Decrease (%)
10	6.8	0.0	6.378	37.320	0
20	6.9	1.5	1.553	11.421	69
30	7.3	6.9	0.712	5.626	85
50	7.9	16.8	0.355	4.398	88
60	8.1	19.1	0.290	4.709	87
100	8.9	31.3	0.086	2.642	93
150	10.4	52.7	0.047	1.945	95
300	14.1	107.6	0.000	0.000	100

Table 1. Variations in number of training frames

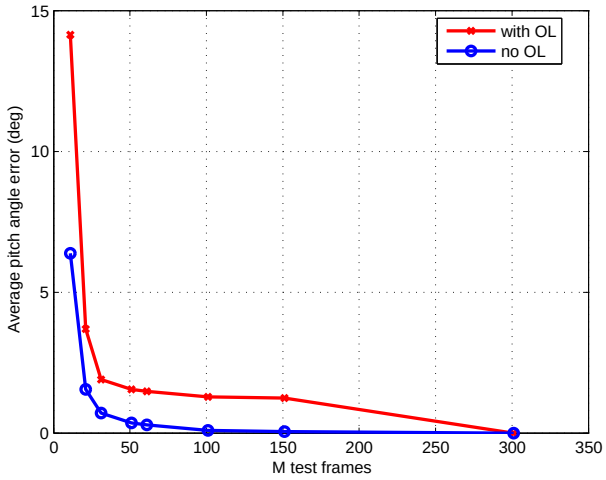
The second parameter that is important to the accuracy of the pose estimation is the number of principal components retained in the training database K. As K is increased, the training time increases due to the subset SVD computation. Figure 8 shows a linear increase of the training and testing time as a function of increasing principal components, where testing involved five cases with increasing number of training images from 10 to 100. A noticeable increase in total testing time occurred as total training images increased from 10 to 100. Figure 9 shows the timing and pose error metrics for the testing with K equaling to 1 to 20 principal components. Five M testing dataset of 10, 20, 30, 50, and 100 were compared. A large jump in average single image testing time when the training set is $M = 100$ images where as from $M = 50$ and below, testing timing remain relatively the same; while all testing times are linearly increasing with increasing principal components. The average pitch angle and maximum pitch angle both show a dip in accuracy between 5 to 10 principal components. The optimal number can be selected at 8 principal components. As the number of components increase, the angular error settles to a constant value, this is understandable since PCA suggests higher eigenvalue space contains less useful information. As noted previously, the pitch angle error is significantly higher for when training dataset is only 10 images.



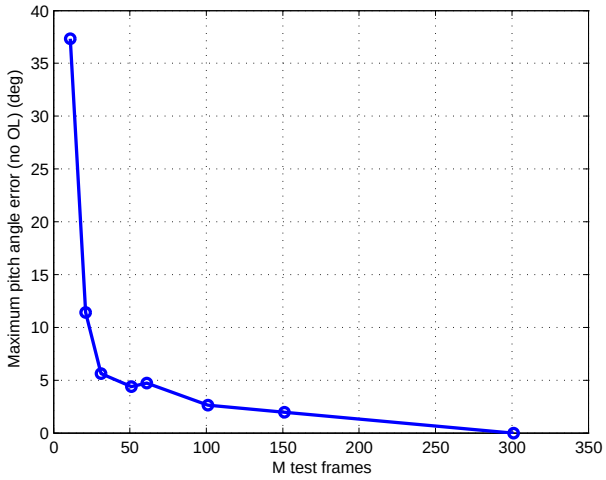
(a) Single frame test time



(b) Percent time and angle error

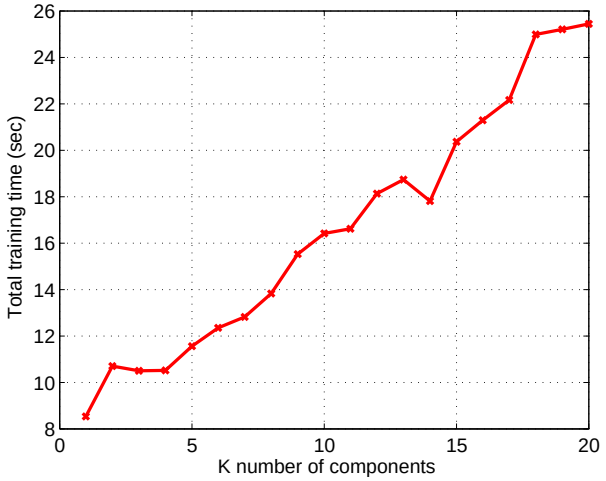


(c) Average pitch angle error

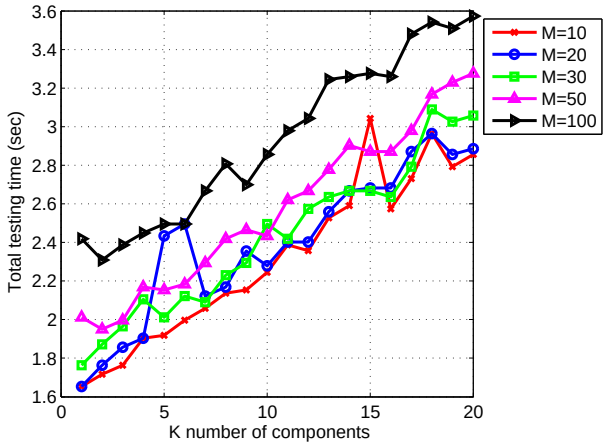


(d) Maximum pitch angle error

Figure 7. Testing metric resulting from M number of training images

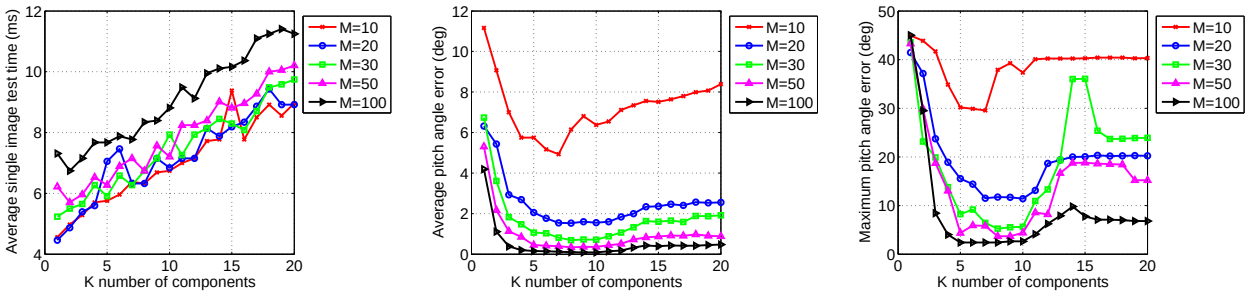


(a) Training time



(b) Testing time

Figure 8. Computation time resulting from K components



(a) Single frame test time (b) Average pitch angle error (c) Maximum pitch angle error

Figure 9. Testing metric resulting from K components

B. Multi-Axis Pose Results

Increasing the number rotational DOF, the target CS is allowed to rotate about X, Y and the Z axis. In order to produce sequential training images, the SS camera is made to revolve around a stationary CS. The camera boresight is locked onto the CS COM where the relative distance remained a constant radius as the SS camera traced out the path of a sphere. Spherical coordinates were used with full variations in inclination and right ascension as shown in Fig. (10). At each spherical location, the SS camera makes a 360 degree rotation about the camera boresight. The spherical coordinate and the boresight roll angle is then kinematically transformed into quaternion representing relative orientation between the SS camera and the CS. Based on the findings from Sec. A 30 intervals were used for each of the

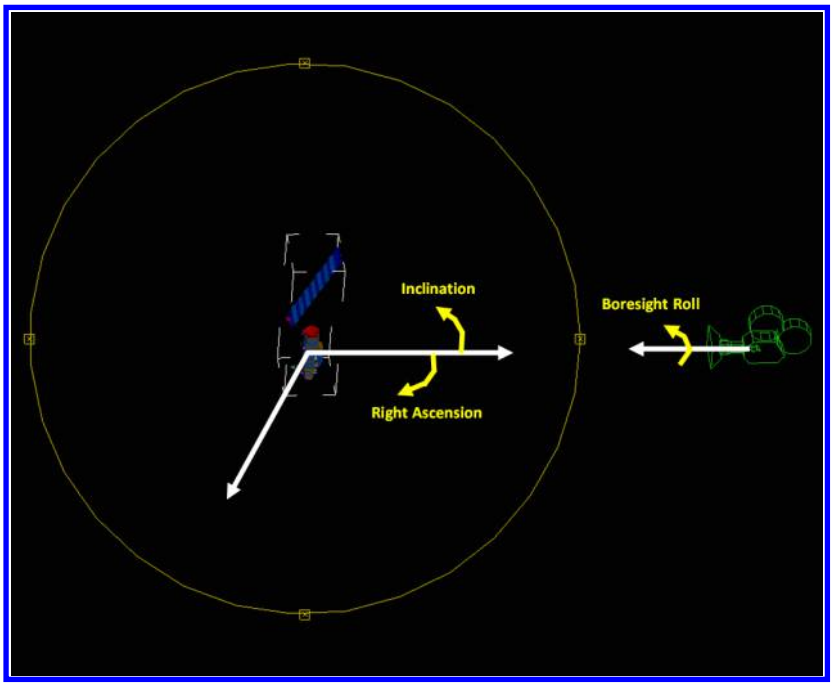
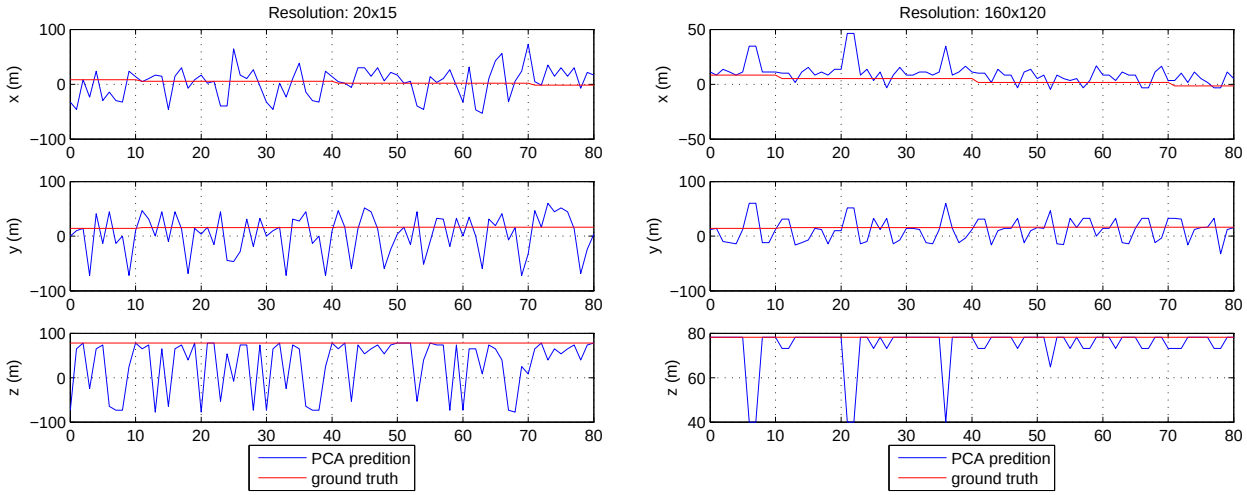


Figure 10. Method for extracting training pose

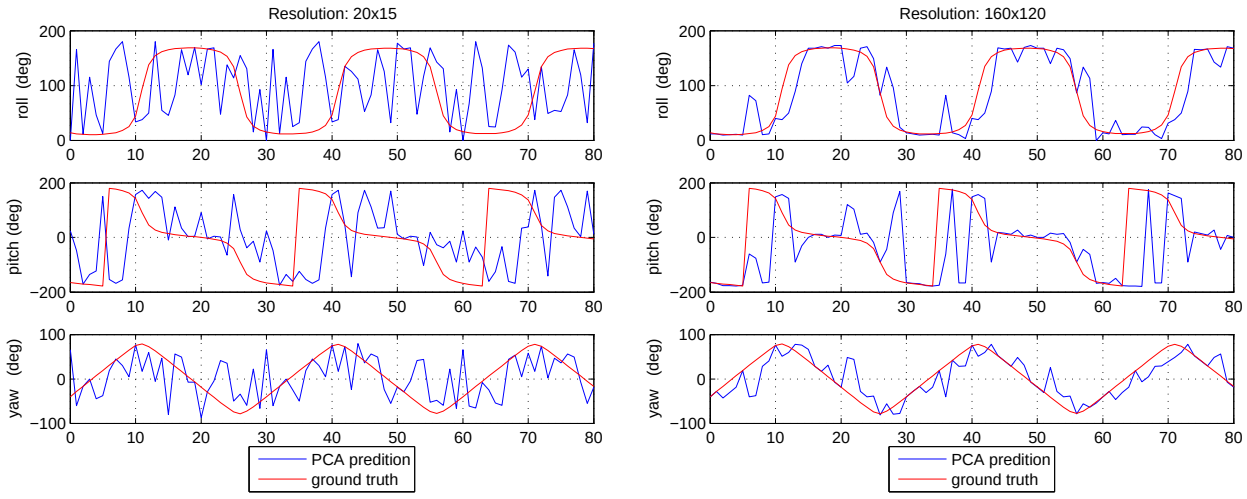
inclination, right ascension, and boresight roll axis. At the north and south poles, the redundant right ascension frames were removed. This resulted in $M = 12660$ images in the training database that covered the entire 3D orientation space. In order to save memory in calculating the SVD of the image bin, the images were pyramid down 4 levels using method used by Lowe.[17] A simple every other line pixel removal scheme was used. The four resolutions were tested for its accuracy when using the PCA method. Figure 11 shows the 4 resolution for the same image frame. Figure 12 and 13 shows the position and orientation of the lowest resolution of 20×15 and the highest resolution of 160×120 for ground truth and matched pose respectively. It is shown the low resolution pose is significantly noised where the highest resolution pose has less noise. This implies the resolution of the training image will affect the accuracy of the pose determination and filtering is required when using the PCA method for pose estimation. Figure 14(a) shows the eigenspace error norm between the test image and training images. Comparing the eigenspace error norms for various

Figure 11. Reduction in image resolution using pyramids



(a) Resolution: 20x15 (b) Resolution: 160x120

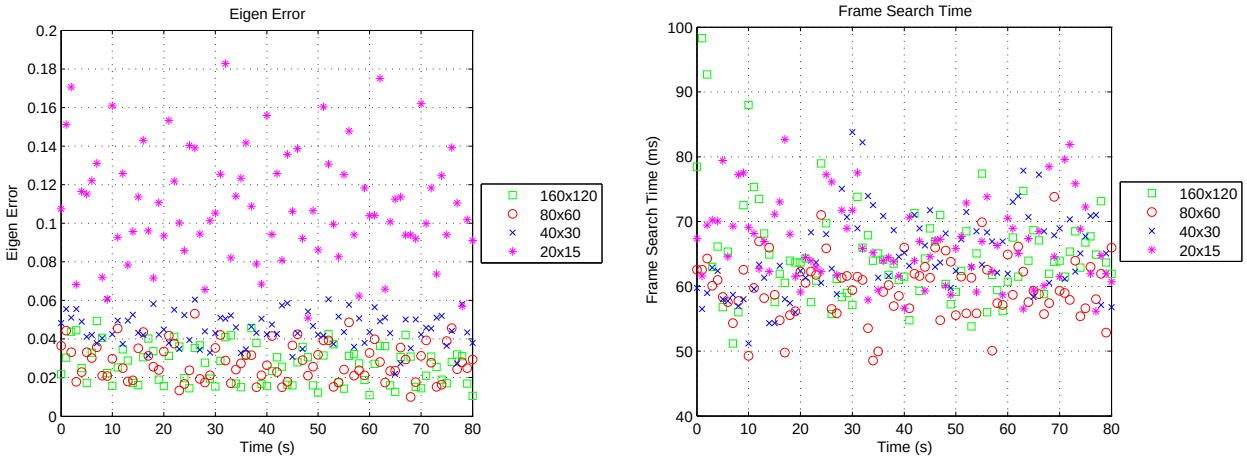
Figure 12. Position results from different training resolutions



(a) Resolution: 20x15 (b) Resolution: 160x120

Figure 13. Orientation results from different training resolutions

image bank resolutions, the lower resolution image match resulted in higher scattering compared to the finer resolution error. This is further indication the accuracy of the PCA method is sensitive to the resolution of the image bank and the test image especially for low resolution cases. Figure 14(b) shows the search time for all resolution cases. The timing result indicate resolution does not affect the overall timing where the mean search time is approximately 62.8 ms. Compared to other feature based method,[61] this timing is relatively low for target spacecraft pose estimation.



(a) Eigenspace error (b) Frame search time

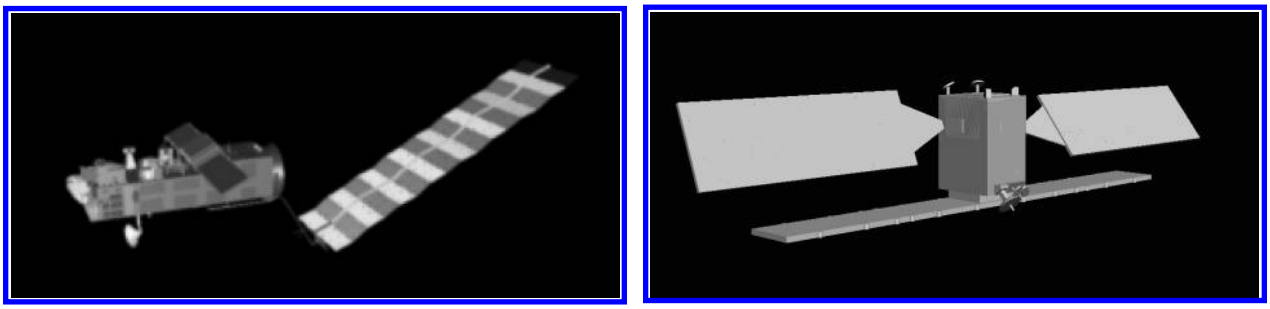
Figure 14. Eigenspace error and test search time

VI. Experimental Setup

Experiment for this study is composed of pose estimation from synthetic imagery and real thermal camera imagery. Training dataset is first developed using CAD models developed in Solid Works and imaged using a virtual camera in 3D Studio Max (3DS). The pose of the CS is feed into the 3DS animation tool plug-in and captured using a virtual camera at 30 frames per second (FPS) and 320x240 resolution. The virtual camera properties is provided in Tbl. (2). Three CAD models were developed, these are Envisat and the Radarsat Model (RSM). The models are shown in Fig. (15) for the two CS respectively.

Description	Settings
Resolution	320x240
Frames per Second (FPS)	30
Focal Length (mm)	43.5
Aperture Width (mm)	36
Field of View (FOV)	45

Table 2. Virtual Camera Properties



(a) Envisat (b) Radarsat

Figure 15. Various CS CAD models

A thermal camera image of the RSM is taken using ICI 9320P camera. The 9320P camera property is provided in Tbl. (3). The thermal camera[b], physical RSM model, and the test setup is shown in Fig. (16).

Description	Settings
Resolution (pixels)	320x240
Frames per Second (FPS)	10
Focal Length (mm)	7.5
Field of View (FOV)	40
Spectral Response (μm)	7-14
Thermal Sensitivity (mK)	$<$30
Accuracy (°C)	$\pm$1

Table 3. ICI-9320P Thermal Camera Properties

(a) Test Setup (b) Radarsat Model (c) Thermal Camera

Figure 16. Test Setup

A. Appearance Based Training Data

The PCA approach outlined in this study determines the relative spacecraft pose based on training images prior to actual flight operations. The training images are stored as reduced principal component matrix $\tilde{\mathbf{G}}$. Pose data is tagged to each image in the training set of M images. In the initial training image set, the SS points the camera directly at the CS COM and the CS is rotated about the pitch axis of the LVLH frame at the rate of 36 deg/s *i.e.* taking ten seconds to rotate one revolution. This is illustrated in Figs. (17) and (18) for Envisat and Radarsat respectively.

VII. Experimental Results

A. Image Denoising

The ePCA image denoising process was applied on the synthetic Radarsat CAD image (shown in Fig. (18)) and space imagery of the SpaceX Dragon cargo vehicle[c]. Star patterns were added to the synthetically generated RSM image, the patterns represent possible camera lens glare. Figure 19 show three images using ePCA to process the flawed input image. In all three cases, a non-occulted image is returned for pose estimation. The ePCA process essentially eliminated the occultation from the image. Ten training images for the Dragon vehicle is shown in Fig. (20). These images were taken by the ISS camera during various Dragon vehicle docking approach. All images were reduced to 320x240 resolution and some include the SSRMS robotic manipulator during the final stage free-flyer capture event. Figure 21 show the optimized and normalized training data after ePCA process. Figure 22 shows the three Dragon images with camera glare added as occultation. These were processed by the ePCA software and returned the

[b]courtesy of Infrared Camera Incorporated

[c]Images courtesy of National Aeronautics and Space Administration (NASA) and Space Exploration Technologies Corporation (SpaceX)

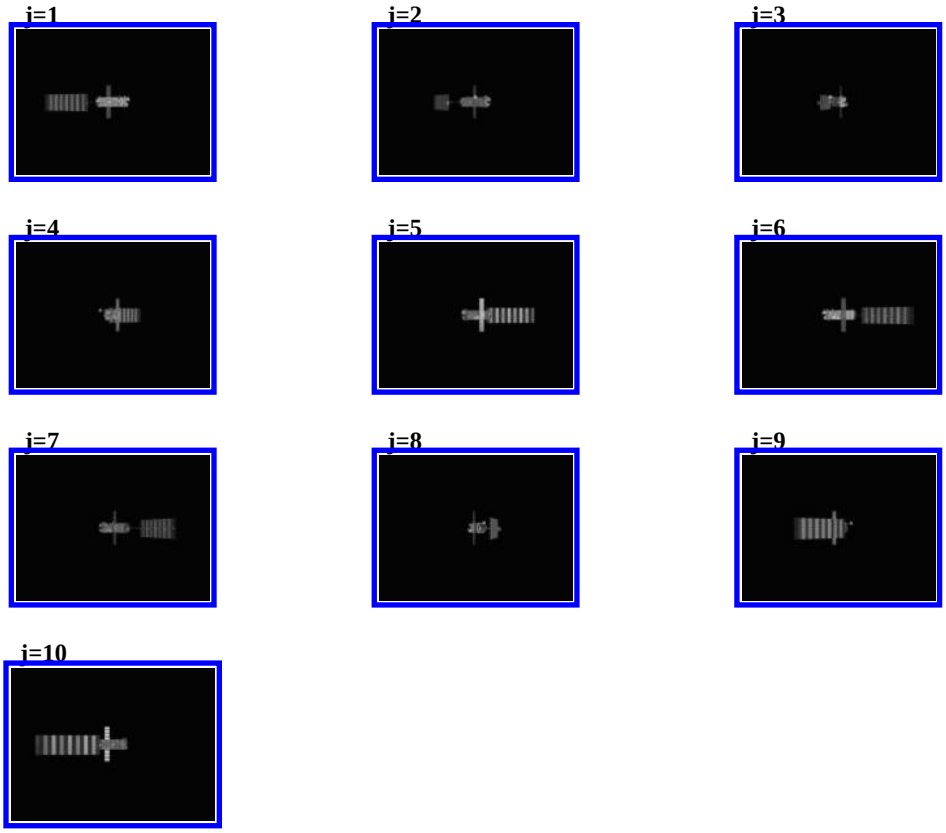


Figure 17. Training dataset: 360 deg rotation of the Envisat

optimized images below. The processed images do not have the occultation in the original testing image. Therefore, the pose estimation process will have a much easier time in matching to the training data in storage.

B. Thermal Image Pose Estimation

Thermal images of the Radarsat Model (RSM) taken by the ICI-9320 camera (ref. Tbl. (3) and Fig. (16)) were used for pose estimation. Example of the RSM thermal images are shown in Fig. (5). To work with the thermal camera image for PCA pose estimation, background clutter were removed by using adaptive background subtraction. Figure 23 shows as long as the central CS image can be reasonably obtained, the synthetic image can match the background subtracted thermal image.

VIII. Conclusions

In conclusion, this study makes use of principal components analysis for image based spacecraft pose estimation and a novel kernel based principal components analysis called Euler-PCA for image denoising. Algorithm and formulation for direct PCA and ePCA were adopted for spacecraft monocular camera and infrared camera imagery. Simulation were ran with single and multi-axis spacecraft orientation. PCA has the benefit of data compression for storing training pose imagery, while ePCA offers advantage in data robustness to distortion and occultation. PCA methods works well with single axis however, for multiple axis, sufficient hardware memory is required to make the singular value decomposition calculation in order for full image resolution to be used. The accuracy of PCA degrades as the image resolution is degraded. Filters are recommended when using PCA to remove outlier jumps resulting from low resolution matches. Finally, the time to perform end-to-end pose estimation for the multi-axis scenario is 62.8 ms. This is a relatively good speed for real-time pose estimation purposes. ePCA denoising was applied to synthetic RSM images as well as space images of SpaceX Dragon cargo vehicle. Results show in both instances, returned optimal result can be without corruption, distortions and occultation. In the future, ePCA tracking can be integrated to the software while a more efficient pose estimator from both baseline PCA and ePCA can be developed.

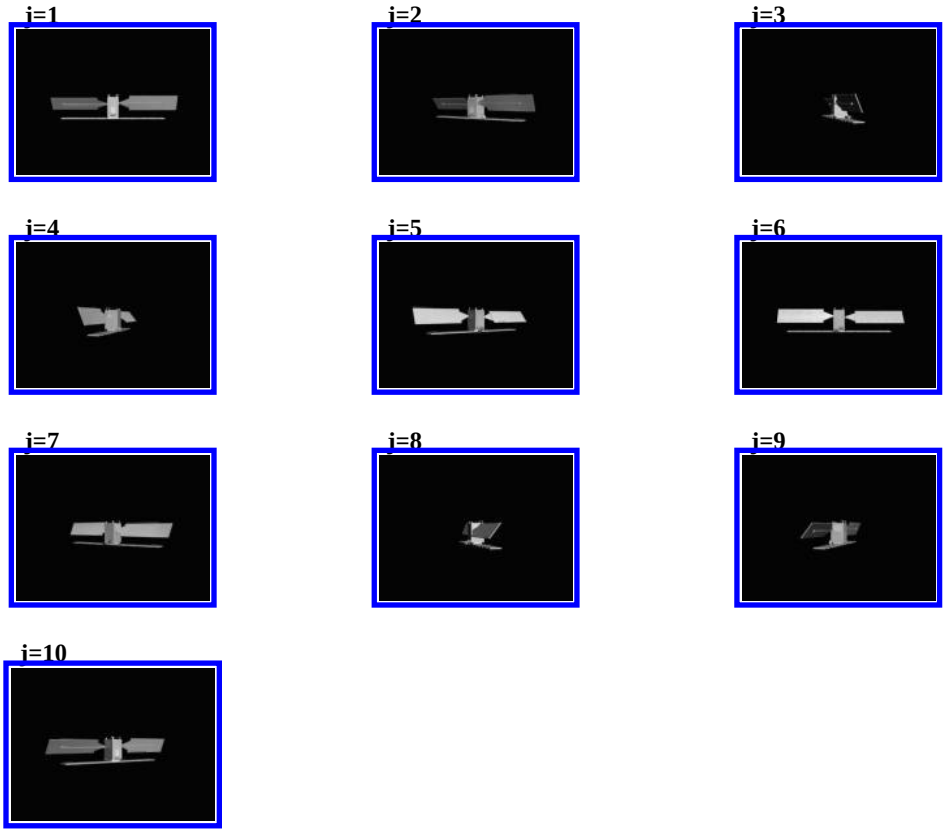


Figure 18. Training dataset: 360 deg rotation of the Radarsat

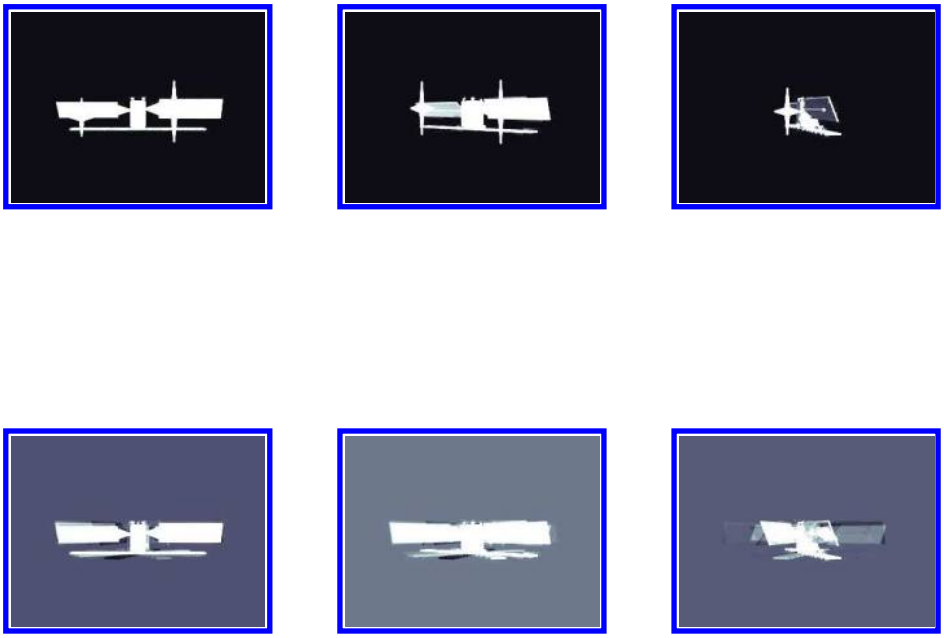


Figure 19. Synthetic Radarsat Testing Images. (Top:corrupted testing images;Bottom: ePCA optimized output

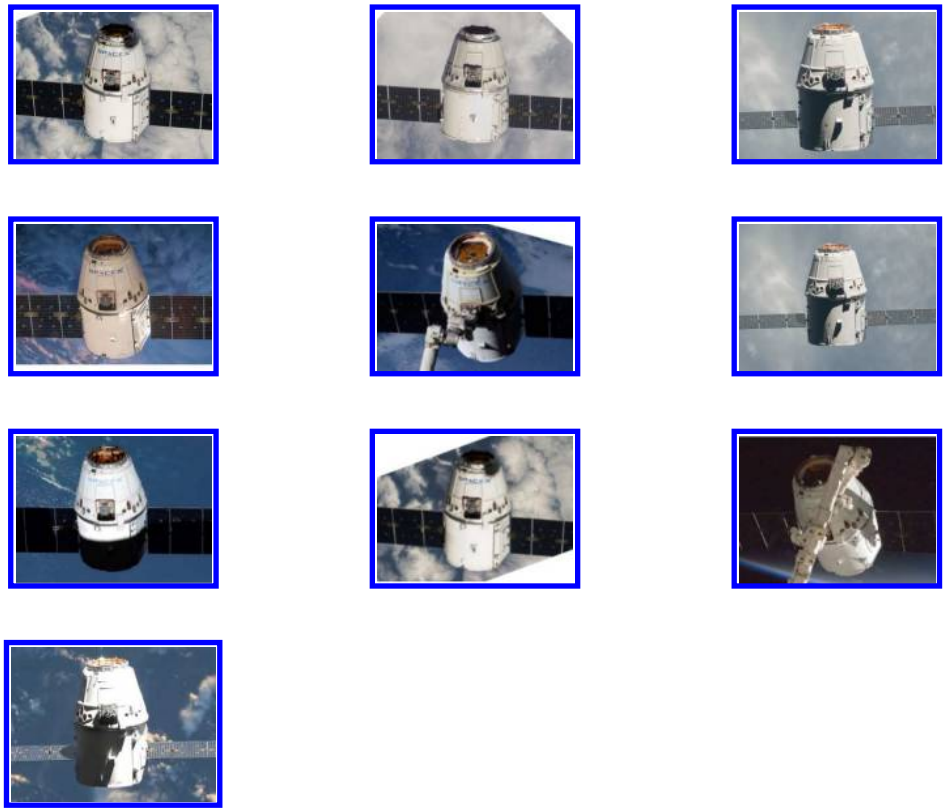


Figure 20. SpaceX Dragon Vehicle Training Images

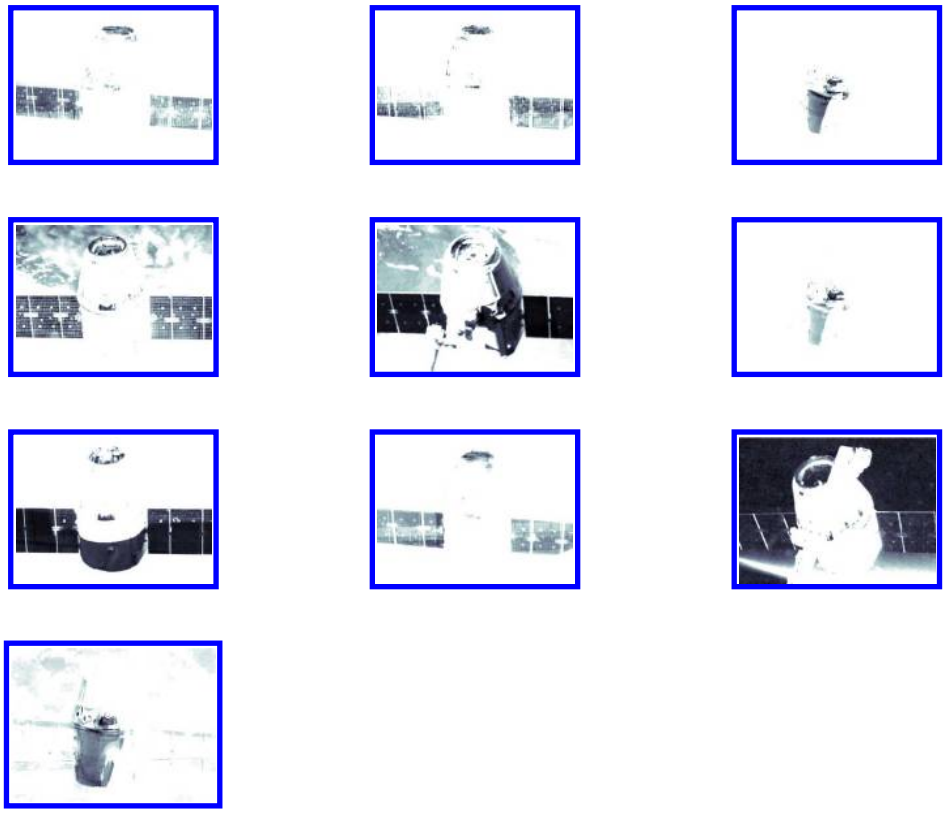


Figure 21. SpaceX Dragon Vehicle *e*PCA Optimized Training Images

Figure 22. SpaceX Dragon Vehicle ePCA Optimized Testing Images. (Top:corrupted testing images;Bottom: ePCA optimized output

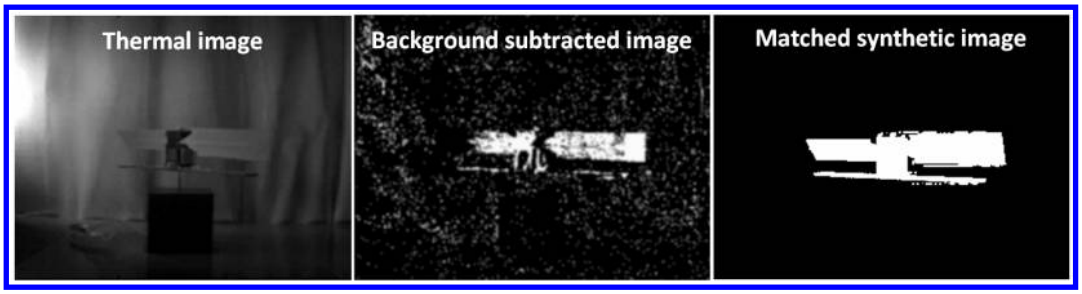


Figure 23. Thermal test image to synthetic training image match

Acknowledgements

The first author wishes to thank Dr. Gerhard Roth for his inspirational ideas and guidance. This research was jointly funded by the Natural Sciences and Research Council of Canada's Alexander Graham Bell Canada Graduate Scholarship CGSD3-453738-2014, Canadian Space Agency Space Technology Development Program and Ontario Centres for Excellence Voucher for Innovation and Productivity II Award 24053.

References

[1]Geller, D. K., "Orbital rendezvous: when is autonomy required?" *Guidance, Control, and Dynamics, AIAA Journal of*, Vol. 30, No. 4, 2007, pp. 974–981.

[2]Shiu, Y. and Ahmad, S., "3D location of circular and spherical features by monocular model-based vision," *Systems, Man and Cybernetics, Conf. Proc.*, Nov. 1989, pp. 576–581.

[3]Guru, D., Shekar, B., and Nagabhushan, P., "A simple and robust line detection algorithm based on small eigenvalue analysis," *Patter Recognition Letters*, Vol. 25, No. 1, 2004, pp. 1–13.

[4]Zheng, Y., Ma, W., and Liu, Y., "Another way of looking at monocular circle pose estimation," *Image Processing (ICIP), 15th IEEE Int Conference on*, San Diego, CA, Oct. 2008.

[5]Lu, J., Shi, Y., and Wu, S., "Monocular vision-based sensor for autonomous mobile robot localization by circular markers," *Przeglad Elektrotechniczny*, Vol. 89, No. 1b, 2013, pp. 131–133.

[6]Ogilvie, A., Allport, J., Hannah, M., and Lymer, J., "Autonomous robotic operations for on-orbit satellite servicing," *Sensors and Systems for Space Applications II, Proc. of SPIE,*, Vol. 6958, Orlando, FL, April 2008.

[7]Tweddle, B., *Computer vision based navigation for spacecraft proximity operations*, Master's thesis, Dept. of Aeronautics and Astronautics Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2010.

[8]Morency, L. and Gupta, R., "Robust Real-time Egomotion from stereo images," *Image Processing, Proc. Intl. Conf. on*, 2003.

[9]Sinclair, D., Blake, A., and Murray, D., "Robust estimation of egomotion from normal flow," *Computer Vision. Intl. Journal of*, Vol. 13, No. 1, 1994, pp. 57–69.

[10]Pauwels, K., Rubio, L., Díaz, J., and Ros, E., "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues," *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, Portland, OR, June 2013.

[11]Tomasi, C. and Kanade, T., "Shape and motion from image streams under orthography: a factorization method," *Computer Vision (IJCV), Intl. Journal of*, Vol. 9, No. 2, 1992, pp. 137–154.

[12]Pizzoli, M., Forster, C., and Scaramuzza, D., "REMODE: probabilistic, monocular dense reconstruction in real time," *Robotics and Automation (ICRA), IEEE Intl Conf. on*, Hong Kong, China, May 2014.

[13]Barandiarán, J. and Borrow, D., "Edge-based markerless 3D tracking of rigid bodies," *Artificial Reality and Telexistence, 17th Intl. Conf. on*, IEEE, 2007, pp. 282–283.

[14]Choi, C., Trevor, A., and Christensen, H., "RGB-D edge detection and edge-based registration," *Intelligent Robots and Systems (IROS), IEEE/RSJ Intl. Conf. on*, IEEE, Nov 2013, pp. 1568–1575.

[15]Hexner, J. and Hagege, R., "2D-3D Pose estimation of heterogeneous objects using a region based approach," *Computer Vision (IJCV), Intl. Journal of*, Vol. 118, No. 1, 2016, pp. 95–112.

[16]Tjaden, H., Schwanecke, U., and Schömer, E., "Real-time Monocular Segmentation and Pose Tracking of Multiple Objects," *Computer Vision (ECCV), 15th European Conf. on*, IEEE, 2016, pp. 423–438.

[17]Lowe, D., "Distinctive image features from scale-invariant keypoints," *Computer Vision, Intl. Journal of*, Vol. 60, No. 2, 2004, pp. 91–110.

[18]Bay, H., Ess, A., Tuytelaars, T., and Gool, L., "Speeded-Up Robust Features (SURF)," Vol. 110, No. 3, June 2008, pp. 346–359.

[19]Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: an efficient alternative to SIFT or SURF," *Computer Vision (ICCV), IEEE 15th Intl Conf. on*, Barcelona, Spain, Nov. 2011.

[20]Alcantarilla, P., Nuevo, J., and Bartoli, A., "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *British Machine Vision Conference (BMVC)*, Bristol, UK, Sept 2013.

[21]David, P., Dementhon, D., Duraiswami, R., and Samet, H., "SoftPOSIT: simultaneous pose and correspondence determination," *Computer Vision, Intl. Journal of*, Vol. 59, No. 3, 2004, pp. 259–289.

[22]Lepetit, V., "EPnP: an accurate O(n) solution to the PnP problem," *Computer Vision, Intl. Journal of*, Vol. 81, No. 2, 2009, pp. 155–166.

[23]Ferraz, L., Binefa, X., and Moreno-Noguer, F., "Very fast solution to the PnP problem with algebraic outlier rejection," *Computer Vision and Pattern Recognition (CVPR), Proc. of the IEEE Conf. on*, 2014, pp. 501–508.

[24]Honeine, P. and Richard, C., "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, Vol. 28, No. 2, 2011, pp. 77–88.

[25]Jolliffe, I., *Principal Component Analysis*, Springer, New York, 2nd ed., 2002.

[26]Pearson, K., "On lines and planes of closest fit to systems of points in space," *Phil. Mag*, Vol. 6, No. 2, 1901, pp. 559–572.

[27]Hotelling, H., "Analysis of a complex of statistical variables in to principal componenets," *Edu. Psychol., Journal of*, Vol. 24, 1933, pp. 417–441,498–520.

[28]Sirovich, L. and Kirby, M., "A low dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, Vol. 4, No. 3, 1987, pp. 519–524.

[29]Turk, M. and Pentland, A., "Eigenfaces for recognition," *Cognitive Neuroscience, Journal of*, Vol. 3, No. 1, 1991, pp. 71–86.

[30]Javed, A., "Face recognition based on principal component analysis," *Image, Graphics and Signal Processing, Intl. Journal of*, Vol. 5, No. 2, 2013, pp. 38–44.

[31]Cavalcanti, G., Ren, T., and Pereira, J., "Weighted modular image principal component analysis for face recognition," *Expert Systems with Applications*, Vol. 40, No. 12, 2013, pp. 4971–4977.

[32]Mena-Chalco, J., Veiho, L., Macêdo, I., and Cesar-Jr., R., "PCA-based 3D Face Photography," *Computer Graphics and image Processing (SIBGRAPI), IEEE XXI Brazilan Symp. on*, Oct. 2008, pp. 313–320.

[33]Mena-Chalco, J., Macêdo, I., and Veiho, L., "3D Face Computational Photography using PCA Spaces," *The Visual Computer*, Vol. 25, No. 10, 2009, pp. 899–909.

[34]Du, H., Hu, Q., Jiang, M., and Zhang, F., "Two-dimensional principal component analysis based on Schatten p-norm for image feature extraction," *Visual Communication and Image Representation, Journal of*, Vol. 32, 2015, pp. 55–62.

[35]Wang, J., "Generalized 2-D principal component analysis by Lp-Norm for image analysis," *Cybernetics, IEEE Trans. on*, Vol. 46, No. 3, 2016, pp. 792–803.

[36]Schölkopf, B., Smola, A., and K.R., M., "Nonlinear Component Analysis as a Kernel," *Neural Computation*, Vol. 10, No. 5, 1998, pp. 1299–1319.

[37]Mika, S., Schölkoph, B., Smola, A., Müller, K., Scholz, M., and Rätsch, G., "Kernel PCA and de-noising in feature spaces," *Advances in Neural Information Processing Systems*, Vol. 11, No. 1, 1999, pp. 536–542.

[38]la Torre, F. and Nguyen, M., "Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment," *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, IEEE, 2008.

[39]Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M., "Euler principal component analysis," *Computer Vision, Intl. Journal of*, Vol. 101, No. 3, 2013, pp. 498–518.

[40]Simon, D., *Fast and accurate shape-based registration*, Ph.D. thesis, Carnegie Millon University, Pittsburgh, Pennsylvania, 1996.

[41]Mark, L., *Principal Component Analysis for ICP Pose Estimation of Space Structures*, Master's thesis, The Dept. of Aerospace Engineering, Ryerson Univ., Toronto, Canada, 2010.
[42]Ke, Y. and Sukthankar, R., "PCA-SIFT: A more distinctive representation for local image descriptors," *Computer Vision and Pattern Recognition (CVPR), Proc. of Conf. on*, 2004, pp. 511–517.
[43]Trucco, E. and Verri, A., *Introductory Techniques for 3D Computer Vision*, Prentice-Hall, Inc, 1998.
[44]Eckart, C. and Young, G., "The Approximation of one matrix by another of lower rank," *Psychometrika*, Vol. 1, No. 3, 1936, pp. 211–218.
[45]Baglama, J. and Reichel, L., "Augmented Implicitly Restarted Lanczos Bidiagonalization Methods," *Scientific Computing, Society for Industrial and Applied Mathematics (SIAM) Journal on*, Vol. 27, No. 1, 2005, pp. 19–42.
[46]Larsen, R., "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Report Series*, Vol. 27, No. 537, 1998.
[47]Rokhlin, V., Szlam, A., and Tygert, M., "A randomized algorithm for principal component analysis," *Matrix Analysis and Applications, Society for Industrial and Applied Mathematics (SIAM) Journal on*, Vol. 31, No. 3, 2009, pp. 1100–1124.
[48]Halko, N., Martinsson, P., and J.A., T., "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions," *Society for Industrial and Applied Mathematics (SIAM) Review*, Vol. 53, No. 2, 2009, pp. 217–288.
[49]Vapnik, V., *The Nature of Statistical Learning Theory*, Springer, 1995.
[50]Boser, B., Guyon, I., and Vapnik, V., "A training algorithm for optimal margin classifiers," *Learning Theory (COLT), Proc. of conf. on*, ACM Press, Pittsburgh,PA, 1992.
[51]Schölkopf, *Support vector learning*, Ph.D. thesis, Dept. Computer Science, Univ. of Berlin, Oldenbourg Verlag, Munich, 1997.
[52]Fitch, A., Kadyrov, A., Christmas, W., and Kittler, J., "Fast Robust Correlation," *Image Processing, IEEE Trans. on*, Vol. 14, No. 8, 2005, pp. 1063–1073.
[53]De La Torre, F. and Black, M., "A Framework for Robust Subspace Learning," *Computer Vision, Intl. Journal of*, Vol. 54, No. 1–3, 2004, pp. 117–142.
[54]Kwak, N., "Principal component analysis based on L1-norm maximaization," *Pattern analysis and machine intelligence, IEEE Trans. on*, Vol. 30, No. 9, 2008, pp. 1672–1680.
[55]He, R., Hu, B., Zheng, W., and Kong, X., "Robust principal component analysis based on maximum correntropy criterion," *Image Processing, IEEE Trans. on*, Vol. 20, No. 6, 2011, pp. 1485–1494.
[56]Ke, Q. and Kanade, T., "Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming," *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, IEEE, 2005.
[57]Pizer, S., Auston, J., Perry, J., and Safrit, H., "Adaptive histogram equalization for automatic contrast enhancement of medical images," *Medicine, Proc. of SPIE XIV/PACS IV Conf.*, Newport Beach, CA, 1986.
[58]Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B., Zimmerman, J., and Zuiderveld, K., "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, Vol. 39, 1987, pp. 355–368.
[59]Pizer, S., Johnston, R., Ericksen, J., Yankaskas, B., and Muller, K., "Contrast-limited adaptive histogram equalization speed and effectiveness," *Visualization in Biomedical Computing, Proc. of the 1st Conf. on*, 1990, pp. 337–345.
[60]Zuiderveld, K., "Contrast limited adaptive histogram equalization," *Graphics gems IV*, 1994, pp. 474–485.
[61]Shi, J., Ulrich, S., and Ruel, S., "Spacecraft pose estimation using a monocular camera," *67th International Astronautical Congress (IAC) of Guidance, Control, and Dynamics, Proceedings of the*, Guadalajara, Mexico, Sept 2016, IAC-16-C1.3.4.