Computing the relations among three views based on artificial neural network

Ying Kin Yu Kin Hong Wong Siu Hang Or
Department of Computer Science and Engineering
The Chinese University of Hong Kong
E-mail: ykyu.hk@gmail.com, {khwong, shor}@cse.cuhk.edu.hk

***Abstract*—Multiple view geometry plays an important role in the traditional problem on the camera pose estimation from 2-D images. Given only 3 images of the same scene, one may apply the well-known fundamental matrix or trifocal tensor to make a robust computation of the poses among the three camera views. Practically, extracting the pose parameters from them is not straight forward. Multiple solutions arise and the final computed value may deviate from the actual one even if the closest solution is chosen since the process is highly non-linear. In this paper, we are going to investigate the use of a machine learning approach to solve the problem. An artificial neural network is trained with noisy point features from three views to estimate the camera poses among them. It is found that the proposed method is able to compute accurate pose parameters under noisy environment. The effects of applying different amount of noises to the training and testing datasets have also been studied.**

***Index Terms*—Trifocal tensor, Multiple view geometry, Neural network, Machine learning, Camera pose estimation**

I. Introduction

The estimation of relative position and orientation among image views is a fundamental research problem for many of the futuristic computer vision applications such a augmented reality, video surveillance and autonomous vehicles. Traditionally, such estimation problem is tackled in an analytical way. In this paper, we are going to address this issue based on a machine learning framework. Here we focus on computing the relation of point features among three image views.

To set up the geometric relationship among three views, the trifocal tensor can be employed traditionally. With the relative pose of three camera views, the tensor, denoted by T, can be computed in a straightforward manner. Having an accurate tensor T, one can establish the corresponding point features among the three images. However, camera pose estimation from images is an inverse of the above process and is much more complicated. Features are first extracted from the images and correspondences can be setup. With the corresponding features from three images, the trifocal tensor can be calculated by forming an over-determined system. The 27 elements of the trifocal tensor can be found algebraically. The noise from outlying point features can be rejected using robust estimation algorithm such as Random sample consensus (RANSAC) [1]. Then the camera matrices consisting of the rotation and translation parameters can be extracted from the trifocal tensor [2], [3], [4]

The major disadvantage of the above approach is the high computational requirement. For N potential feature matches among three views, it draws 6 points from them for each computation of the tensor in the RANSAC procedure. Usually, several hundreds of trials are required for an accurate estimation. Moreover, if our final goal is to compute the rotation and translation among the views, the deviation of the values of the estimated trifocal tensor from the actual one will make the final camera pose extracted erroneous. It is because the extraction process is analytical and the relation between the image projection and camera pose is highly non-linear. A small amount of noise in the image points could cause a great error in the extracted rotation and translation parameters.

In this paper, a machine learning based method is investigated for the estimation of the trifocal tensor relation and subsequently the rotation and translation among the views. Corresponding points contaminated with different level of noises are input to the artificial neural network for training and testing. The errors of the camera poses computed from it are studied under various situations. Intuitively, the proposed neural network implementation is able to learn and identify noisy inputs in the training process. The noise contains in the input point features can be filtered off automatically. Experimental results show that a feed forward neural network having one hidden layer is sufficient to learn the function that computes camera poses from 2-D point features in three image views.

This paper is organized as follow. Section II explains the related work of our project. In Section III, the theory of multiview geometry will be introduced. In Section IV we discuss the method of using multilayer feed forward artificial neural networks for the pose estimation problem. The experiments and results are shown in Section V. Discussion and future work is found in Section VI. Finally, the conclusion is in Section VII.

II. Related Work

There are a number of research projects that attempt to tackle the pose estimation problem based on neural network. Wunsch et. al. [5] trained a classifier with different views of an object. To avoid presenting a large number of real images with ground truth to the system, the training was done using synthetic images generated from a CAD model.

The trained network is able to respond to an input pattern with the approximate orientation parameters of the presented view at a high speed.

Benton and Chu in [6] applied a multi-layered feed forward network solution to estimate the relative pose of the two cameras in the 3-D space. A network with 32 inputs, 20 hidden and 3 output units was used to accommodate 8 corresponding points of input in two views. The proposed network was configured as a classifier and quantized rotation angles were recovered in the process.

Kendall et. al. in [7], [8] adopted a Bayesian convolutional neural network to regress the 6-degree-of-freedom camera pose relative to the world coordinate frame from a single RGB image. The training of the proposed network was end-to-end. Neither the explicit feature extraction procedure nor graph optimization was necessary. The system could estimate the pose in real-time with a measure of model uncertainty.

Melekhov et. al. in [9] proposed another convolutional neural network based method to compute the relative pose between two cameras. Their network takes RGB images directly from two cameras and calculates the relative rotation and translation in the forward pass. With the use of transfer learning from a large scale classification dataset, the proposed network was again trained in an end-to-end manner. Their results were compared with feature-based approaches using SIFT or ORB descriptors.

The work presented here focuses on the recovery of camera poses from three views based on the training of a neural network instead of solving the problem in an analytical way using trifocal tensor. The trifocal tenor was applied to estimate camera pose recursively with extended Kalman filter in the literature [10], [11], [12]. Our work is a bit similar to [6] in a way that features in the views are assumed to be extracted and matched properly. The proposed neural network is able to compute the exact pose with errors less than 5mm and 0.5 degrees under 1 pixel of image noise while the method in [6] is only able to predict an approximate quantized values.

III. Multiple View Geometry

Point and line correspondences in different camera views can be related together in a mathematical way. Given two views of the same scene, points and lines can be related by one another through a matrix as follows:

$$x' F x = 0 \tag{1}$$

$$l' = F x \tag{2}$$

F is the fundamental matrix, x, x' and l' are a point in the first image, the corresponding point and line in the second image, respectively. This is the well-known epipolar geometry. If three image views are available, then points or lines or a mixture of them can be constrained by a 3 dimensional tensor as below:

$$[x^{'}]_x(\sum_i x^i T_i)[x^{"}]_x = 0_{(3x3)} \tag{3}$$

$$[x^{'}]_x(\sum_i x^i T_i)[l^{"}]_x = 0_{(3x3)} \tag{4}$$

$$[l^{'}]_x(\sum_i x^i T_i)[l^{"}]_x = 0_{(3x3)} \tag{5}$$

Here T is the trifocal tensor, which sets up the geometric relation among three views. $x^{"}$ and $l^{"}$ are the point and line in the third camera view, respectively. T contains the pose information of the cameras. To estimate the 3-D camera motion among the images, one has to compute the 27 elements in the trifocal tensor T. There are several ways to accomplish the task. In case that the input point features are perfect, a linear method based on direct solution of a set of linear equations can be applied after appropriate data normalization. Otherwise, iterative methods that minimizes the algebraic error, geometric error or Sampson approximation to geometric error can be used. The most common way to solve such problem is the application of the RANSAC framework. The trifocal tensor T can be computed in a robust manner even if there are considerable amount of outlying features in the input. Further details of these traditional methods can be found in reference [4].

Once an accurate tensor T is calculated, the rotation and translation components of the camera views are extracted. This is done by finding the epipoles, e' and e" in the image views. The fundamental matrix F_{21} and F_{31} relating the first and the second, and the first and the third view, respectively, can be calculated as

$$F_{21} = [e^{'}]_x[T_1, T_2, T_3]e^{''} \tag{6}$$

$$F_{31} = [e^{'}]_x[T_1^T, T_2^T, T_3^T]e \tag{7}$$

Assuming the first camera matrix $P = I$, the projection matrices of the second camera P' and third camera P" can be computed as

$$P^{`} = [[T_1, T_2, T_3]e^{''}|e^{`}] \tag{8}$$

$$P^{''} = [(e^{''}e_T^{''} - 1)[T_1^T, T_2^T, T_3^T]e^{'}|e^{''}] \tag{9}$$

Given the camera intrinsic parameters, rotation and translation of the views can be found. More details about their derivation are available in [4].

The goal of our work is to compute the two sets of pose parameters among the three camera views. The pose parameters are expressed as the translation t_x, t_y, t_z and rotation yaw, pitch, roll in the paper.

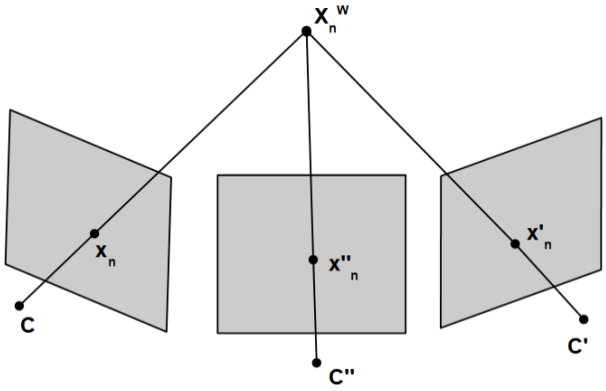

Figure 1. This picture illustrates the relation of points in three views related by a trifocal tensor. C, C' and C" represent the centers of three cameras.

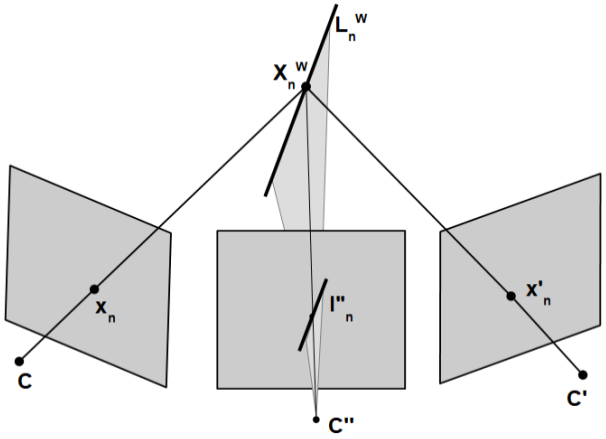

Figure 2. The figure above illustrates the point and line relation in three views constrained by a trifocal tensor. C, C' and C" represent the centers of three cameras.

IV. Our Multi-layer Feed Forward Artificial Neural Network

We propose to use an artificial neural network to learn the computation of the 6-degree-of-freedom camera pose from point features directly. There is no single mathematical function that can calculate the six translation and rotation parameters from 2-D point coordinates in three views.

By definition, the connection of neurons in a feed forward artificial neural network do form no cycles. In this network, the information moves forward from the input nodes via the hidden nodes and finally to the output nodes in only one direction [13].

A multi-layer feed forward neural network is used to estimate the pose parameters directly from the corresponding image features from three views. Our neural network consists of one hidden layer. The nodes are interconnected in a feed-forward way. Each neuron in the hidden layer has directed connections to the nodes in the output layer. A sigmoid function is applied as an activation function in the hidden layer while a linear transfer function is used in the output layer. According to the universal approximation theorem for

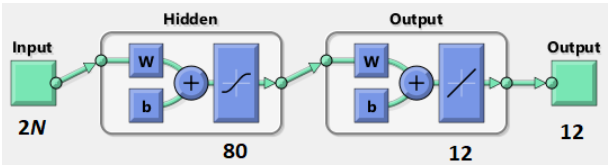

Figure 3. The architecture of our feed forward neural network. It consists of $2N$, 80 and 12 neurons in the input, hidden and output layer, respectively. Here N denotes the number of points input to the network for pose computation. A sigmoid and a linear transfer function are applied at the hidden and output layer.

neural networks, it is sufficient to use a single hidden layer to approximate a continuous function that maps intervals of real numbers to some output interval of real numbers. So, a deep neural network is not necessary in our case. An illustration of the network architecture can be found in Fig.3.

To train the neural network, the coordinates of the corresponding points from three image views are used as the input of the neural network. A minimum of 6 points is employed. With the assumption that the pose of the first camera has no translation and rotation, i.e. P =I, the target output of the network is the two sets of pose parameters of the second and the third camera relative to the first image. For N point correspondences in the views, there are $2N, 80$ and 12 neurons in input, hidden and output layer of our feed forward neural network, respectively. With the corresponding point features, the camera poses can be estimated directly by computing a forward pass of the trained network. The cost function used is typical Euclidean distance measure and we used standard Levenberg-Marquardt optimization with backpropagation to estimate the network weights.

V. Experiments and Results

In the experiment, synthetic data were used to train the proposed neural network to obtain the estimation of pose parameters from point matches. In the settings, the three cameras were viewing at a synthetic scene consisting of a synthetic structure with 100 random features. The 3-D structure had a volume of $0.13m^3$ and was placed at 0.33m away from the first camera. With the assumption that the first camera was in the world center, the second and the third camera were moved away from the world randomly with a pose of 0 to 10 degrees for the yaw pitch, row and 0 to 0.25 meters for the translation along the x, y and z axis. 10000 sets of synthetic data were generated, within which 7000, 1500 and 1500 sets were used for training, validation and testing, respectively.

The proposed neural network was first trained with 6 point features contaminated by low level of noise. 6 is the minimum number of points for pose computation using the traditional trifocal tensor-based method. We also trained the network with points of with increasing input noise level so as to investigate how the 2-D image noises affect the estimation accuracy.

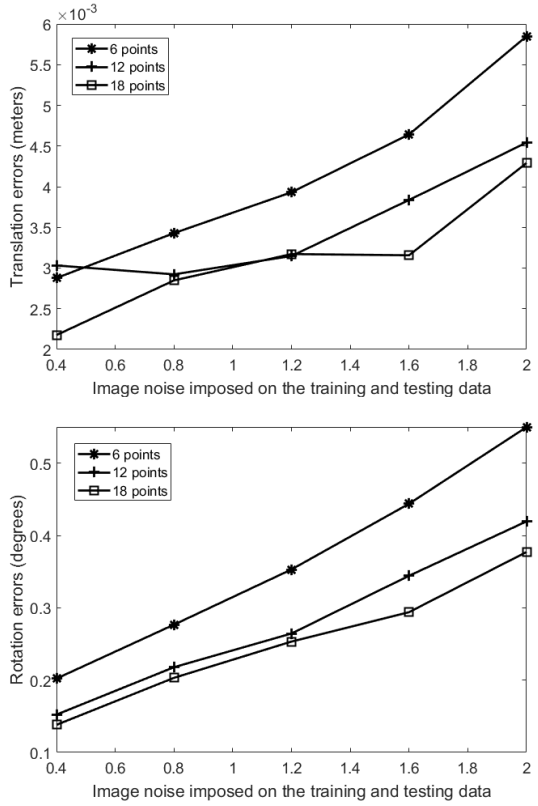

Figure 4. An investigation of the relation of number of input point features, the changes of input noise level and the estimated pose errors was carried out. We used 6, 12, 18 points having image noises from 0.4 to 2 pixels to train and test the networks. The image noises of the training and testing data were matched in this experiment.

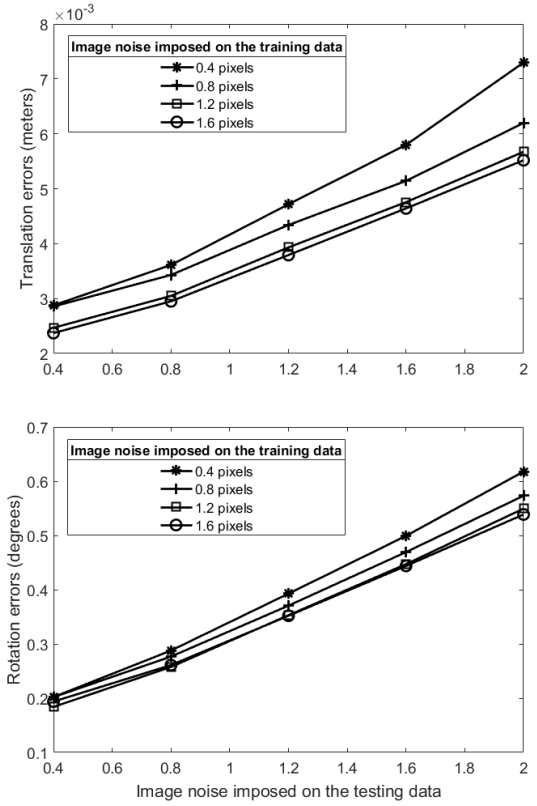

Figure 5. The effects of mismatched noise level in the training and testing data set are shown. 6 input points were used to train the neural network. The lines in each graph represent the networks trained with different noise level. The horizontal axis indicates the noise of the testing data while the vertical axis indicates the recovered pose errors.

Fig. 4 below shows the results. The errors of the recovered pose gradually increases as the noise level of the input point features increases. In addition to 6 point features, we re-trained the network with 12 and 18 point correspondences to examine the relation between the number of input point features and the resulting pose parameter errors. From Fig. 4, it is found that the larger number of input points, the smaller is the recovered pose errors in general.

We also studied the effects of mismatched noise level in the training and testing data sets. For example, the proposed network was trained with points having a Gaussian noise of 0.8 pixels but tested with data points having a noise of 2 pixels. The results are shown in Figs. 5, 6 and 7. It can be revealed that the performance of the network may not be the best even if one trains the neural network using data having the same noise statistics as the testing data.

VI. Discussion and Future Work

With a minimum of 6 point correspondences, the trained neural network is able to give a reasonably accurate solution of the pose parameters. The estimation accuracy can be increased if more input points are available. This is similar to the RANSAC procedure in a way that the quality of solution could be improved if redundant information is available under a noisy environment. One may expect that the estimation results is perfect if the noise level injected to the training data is the same as that of the testing data. It is shown that this is not the case in our experiment. The computed pose parameters are more accurate if the noise in the training data is a bit larger than the testing data. This may be due to the problem of over fitting during network training.

It is well-known that the performances of most of the machine learning based approaches depend quite a lot on the training data. If the actual situation of the application is similar to the training environment, the neural network based approach could have a better performance than the traditional method that solves for the solution analytically. However, its accuracy degrades significantly if this assumption does not hold. In our experiment, the proposed neural network may be unable to estimate the camera pose if the testing values are out of the range of the training data.

In the future, we are going to work out a method to train a universal neural network that is able to compute most of the commonly encountered camera poses in the three view geometry. Our goal is to use a limited set of synthetic training data to make a neural network that is able to estimate the pose parameters of three cameras from real image data. So performing a forward pass on the trained neural network will

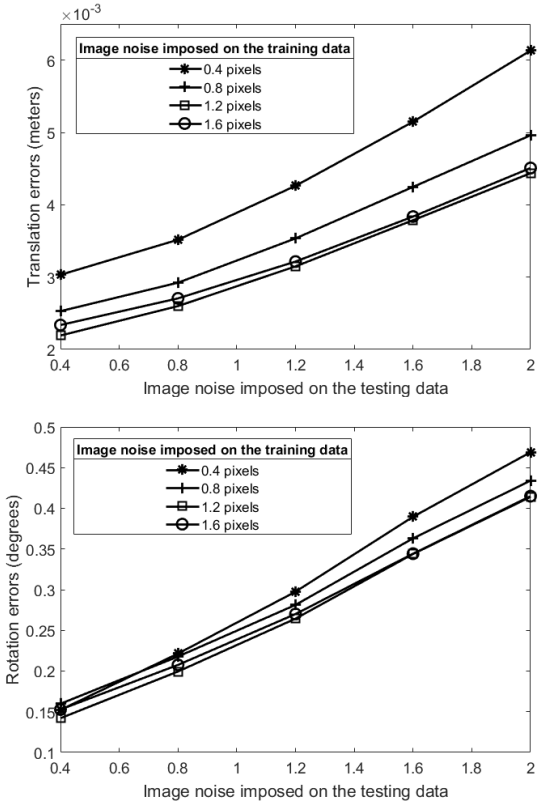

Figure 6. The effects of mismatched noise statistics in the training and testing data are shown. 12 input pionts were used in the experiment. The lines in each graph indicate the networks trained with different noise level. The horizontal axis represents the noise of the testing data while the vertical axis denotes the recovered pose errors.

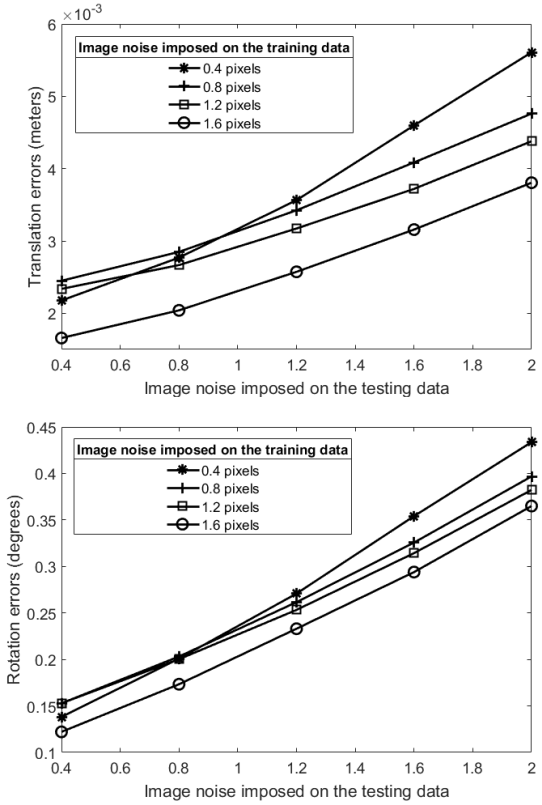

Figure 7. The effects of mismatched noise level in the training and testing data with 18 input points are shown. The lines represent the networks trained with different noise statistics. The horizontal axis indicates the noise of the testing data while the vertical axis indicates the recovered pose errors.

be equivalent to the procedure to recover the translation and rotation parameters using trifocal tensor in a practical sense.

VII. Conclusion

We have successfully trained a feed forward neural network to estimate the camera positions and orientations from three image views. This machine learning based approach is able to avoid the complicated algebraic procedure to compute the trifocal tensor and extract the pose parameters. The calculation of the epipoles in the traditional methods during pose extraction makes the recovered pose erroneous when the input point features are contaminated with noises. It is inevitable even if a RANSAC procedure is applied to select good point features. The proposed neural network solution is more stable under the situation that it is applied to compute the camera poses within the ranges of the training data set. The effects of the number of available points and the mismatches of training and testing data have been investigated in the experiment.

Acknowledgement

This work is supported by a direct grant (Project Code: 4055045) from the Faculty of Engineering of the Chinese University of Hong Kong.

References

[1] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[2] Richard I Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.

[3] Philip HS Torr and Andrew Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and vision Computing*, 15(8):591–605, 1997.

[4] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] Patrick Wunsch, Stefan Winkler, and Gerd Hirzinger. Real-time pose estimation of 3d objects from camera images using neural networks. In *IEEE International Conference on Robotics and Automation 1997*, volume 4, pages 3232–3237. IEEE, 1997.

[6] Ryan G Benton and Chee-hung Henry Chu. Camera pose estimation by an artificial neural network. In *International Conference on Neural Information Processing*, pages 604–611. Springer, 2006.

[7] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International conference on computer vision 2015*, pages 2938–2946, 2015.

[8] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE International Conference on Robotics and Automation 2016*, pages 4762–4769. IEEE, 2016.

[9] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks.

In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017.

[10] Ying Kin Yu, Kin Hong Wong, Siu Hang Or, and Chen Junzhou. Controlling virtual cameras based on a robust model-free pose acquisition technique. *IEEE Transactions on Multimedia*, 11(1):184–190, 2009.

[11] Ying Kin Yu, Kin Hong Wong, Siu Hang Or, and Michael Ming Yuen Chang. Robust 3-d motion tracking from stereo images: A model-less method. *IEEE Transactions on Instrumentation and Measurement*, 57(3):622–630, 2008.

[12] Kai Ki Lee, Ying Kin Yu, Kin Hong Wong, and Michael Ming Yuen Chang. Tracking 3-d motion from straight lines with trifocal tensors. *Multimedia Systems*, 22(2):181–195, 2016.

[13] Kevin Gurney. *An introduction to neural networks*. CRC press, 1997.