Tracking Image Features with PCA-SURF Descriptors

Ardhisha Pancham
CSIR, UKZN
South Africa
apancham@csir.co.za

Daniel Withey
CSIR
South Africa
dwithey@csir.co.za

Glen Bright
UKZN
South Africa
brightg@ukzn.ac.za

Abstract

The tracking of moving points in image sequences requires unique features that can be easily distinguished. However, traditional feature descriptors are of high dimension, leading to larger storage requirement and slower computation. In this paper, Principal Component Analysis (PCA) is applied to the 64-Dimension (D) Speeded Up Robust Features (SURF) descriptor to reduce the descriptor dimensionality and computational time, and suggest the minimum number of dimensions needed for reliable tracking with the Kalman Filter (KF). Tests using image sequences, from an RGB-D camera, are used to validate the performance of the reduced PCA-SURF descriptors as compared to the standard SURF descriptor.

1 Introduction

For mobile robots to operate in dynamic environments they must be able to detect and track stationary and moving objects. Tracking generally involves estimating the position and velocity of objects over a sequence of measurements. For robust tracking, objects need to be uniquely identifiable. In image data, feature descriptors allow for image features to be identified as distinct. These descriptors are usually of high dimension and processing is expensive in terms of both computational time and memory.

Principal Components Analysis (PCA) is a dimensionality reduction technique, that has been applied to reduce the dimensionality of feature descriptor vectors, in several image-related applications.

Ke et al. [4] applied PCA to the normalized gradient patch of Scale-Invariant Feature Transform (SIFT). The 3042-Dimension (D) PCA-SIFT feature descriptor was reduced to 20 dimensions and retained good performance in an image retrieval application.

Lu et al. [6] applied PCA to the Histograms of Oriented Gradient (HOG) descriptor to obtain the 20-D PCA-HOG descriptor, for simultaneous tracking, with the Particle Filter (PF), and action recognition.

PCA was applied to the 128-D Speeded Up Robust Features (SURF) feature descriptor to produce PCA-SURF feature vectors for face recognition [5].

In [10], PCA was applied to the 128-D SIFT and 64-D SURF descriptor to yield Reduced-SIFT and Reduced-SURF feature vectors, respectively. The reduced vectors were evaluated in matching and image retrieval. The use of 32 dimensions produced results similar to SIFT and SURF. To further decrease the computational time and memory used, 20 dimensions was used with a trade off in accuracy.

Euclidean distance was used for matching with either Nearest Neighbor (NN) [4, 10] or NN Distance Ratio (NNDR) [5, 10].

In the applications above, the low dimensional PCA feature descriptors consumed less computational time and memory than the standard descriptors, and increased processing speed.

Gauglitz et al. [3] deduced that the Fast Hessian detector and SURF/SIFT descriptors perform well for tracking during motion and a starting motion. Tracking was simulated via homography rather than an explicit tracking algorithm. PCA was not used.

Although PCA feature descriptors have been applied in the above image-related applications, the novelty of this paper is in applying PCA-SURF descriptors to track dynamic changes in 3D image sequences. PCA is applied to the 64-D SURF descriptor to reduce the descriptor dimensionality and computational time, and suggest the minimum number of dimensions needed for reliable tracking with the Kalman Filter (KF). Tests using image sequences are used to validate the performance of the reduced PCA-SURF descriptors as compared to the standard SURF descriptor.

The detector-descriptor combination mentioned by [3] is suitable for the tracking application herein which involves motion of the camera in a stationary environment. SURF is used as it is computationally more robust than SIFT [1]. Mahalanobis distance is used for descriptor matching with Global NN (GNN) algorithm.

The PCA-SURF method applied is this paper is independent of the tracking algorithm used. KF tracking is used for experiments as it is well suited to linear dynamic applications.

Tracking is performed on a RGB-D image dataset [8]. The advent of affordable RGB-D cameras, allows for the availability of both color and depth data from a single sensor.

The rest of the paper is organized as follows: Section 2 explains SURF, PCA and KF. Section 3 describes the method used to obtain the PCA-SURF descriptors and the tracking application. Section 4 evaluates the performance of the descriptors for tracking. Section 5 concludes the work.

2 Technical background

2.1 SURF

SURF is scale and rotation invariant. The detector is based on the Hessian matrix. The use of box filters and integral images allows for efficient computation. The descriptor consists of a distribution of Haar-wavelet responses that represent the underlying intensity pattern around the detected point [1].

2.2 PCA

PCA is a technique for dimensionality reduction [4, 10, 7]. It involves the following steps:

(1) A matrix M consisting of a set of training vectors in the high dimensional space is obtained.

(2) The mean of each dimension in M is subtracted from its respective dimension to give a mean-adjusted matrix $\overline{M}$.

(3) The covariance matrix of the mean-adjusted, training vectors in $\overline{M}$ is calculated.

(4) The eigenvectors and eigenvalues of the covariance matrix are calculated. Eigenvectors with high eigenvalues represent dimensions of greater variability.

(5) The eigenvectors are arranged in descending order of their eigenvalues. The eigenvectors with high eigenvalues are selected to form the eigenspace or projection matrix P.

(6) A new data matrix N is obtained by projecting the mean-adjusted matrix over the projection matrix, $\left(N = P \times \overline{M}\right)$. N contains vectors of n dimensions, where n is the number of eigenvectors selected.

2.3 KF

The standard KF algorithm is presented in Algorithm 1, adapted from [9]. The input of the KF is the prior probability distribution at time $k-1$, in the form of its mean $\hat{x}_{k-1}$ and covariance P_{k-1} and the measurement z_k. The output is the posterior probability distribution in the form of its mean $\hat{x}_k$ and covariance P_k.

In the prediction step, (1) and (2), the prior mean and covariance at $k-1$ are used to predict the mean $\hat{x}_{k|k-1}$ and covariance $P_{k|k-1}$, at k. A_k is the motion model and Q_k is the process noise covariance matrix.

In the measurement update step, (3) to (7), the measurement z_k is used to compute the mean $\hat{x}_k$ and covariance P_k at k. H_k is the measurement model and R_k is the measurement noise covariance matrix.

The innovation y_k is the difference between the actual measurement z_k and the predicted measurement $H_k(\hat{x}_{k|k-1})$, (3). S_k is the innovation covariance matrix, (4). The Kalman gain K_k indicates the amount by which the measurement should be included in the posterior, (5). The posterior is returned in the form of its mean and covariance in (6) and (7). The estimate $\hat{x}_k$ is also referred to as the state vector [9, 2].

3 PCA-SURF descriptors

3.1 Training

The eigenspace was computed offline with a training set for the 64-D SURF descriptor. The training set consisted of features from the first image of each of four RGB-D datasets [8]. Training images were not used for tracking tests.

A total of 2408 SURF features and descriptors were extracted from the images. PCA was applied to the descriptor vectors to estimate projection matrices.

Figure 1 shows the percentage of variance retained from PCA versus the number of descriptors. A greater percentage of variance is retained with a higher number of descriptors.

Algorithm 1:
Kalman Filter $(\hat{x}_{k-1},\, P_{k-1},\, z_k)$
Prediction

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1} \tag{1}$$

$$P_{k|k-1} = A_k P_{k-1} A_k^T + Q_k \tag{2}$$

Measurement update

$$y_k = z_k - H_k\left(\hat{x}_{k|k-1}\right) \tag{3}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{4}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{5}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k y_k \tag{6}$$

$$P_k = \left(I - K_k H_k\right) P_{k|k-1} \tag{7}$$

return $\hat{x}_k, P_k$

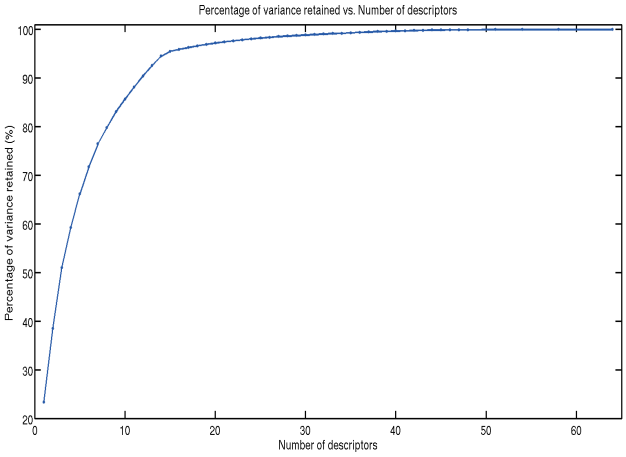

Figure 1. Percentage of variance retained versus the number of descriptors.

3.2 Tracking

The 64-D descriptors were projected to the lower feature space with the respective projection matrix. For experiments, tracking was executed with the KF and GNN. The state vector $\hat{x}_k$ for tracking a feature is shown in (8). It consisted of the feature's X, Y, Z (where Z is derived from depth) position, velocity in X, Y, Z (V_x, V_y, V_z, respectively) and the n-dimensional descriptor (d) vector, where n is the selected number of significant descriptors. A_k, H_k, Q_k (where the constant $q = 20$), R_k, and P_k, are defined in (9) to (13) respectively.

$$\hat{x}_k = [X\ Y\ Z\ V_x\ V_y\ V_z\ d_1\ d_2\ \ldots\ d_n] \tag{8}$$

$$A_k = \begin{bmatrix} I_{3\times3} & \triangle t\, I_{3\times3} & 0_{3\times n} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times n} \\ 0_{n\times3} & 0_{n\times3} & I_{n\times n} \end{bmatrix} \tag{9}$$

$$H_k = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times n} \\ 0_{n\times3} & 0_{n\times3} & I_{n\times n} \end{bmatrix} \tag{10}$$

$$Q_k = q \begin{bmatrix} \frac{\triangle t^3}{3} I_{3\times 3} & \frac{\triangle t^2}{2} I_{3\times 3} & 0_{3\times n} \\ \frac{\triangle t^2}{2} I_{3\times 3} & \triangle t\, I_{3\times 3} & 0_{3\times n} \\ 0_{n\times 3} & 0_{n\times 3} & I_{n\times n} \end{bmatrix} \quad (11)$$

$$R_k = 2 \begin{bmatrix} R_{xyz_{3\times 3}} & 0_{3\times n} \\ 0_{n\times 3} & R_{d_{n\times n}} \end{bmatrix} \quad (12)$$

$$P_k = \begin{bmatrix} R_{xyz_{3\times 3}} & R_{xyz_{3\times 3}} & 0_{3\times n} \\ R_{xyz_{3\times 3}} & 2R_{xyz_{3\times 3}} & 0_{3\times n} \\ 0_{3\times n} & 0_{3\times n} & R_{d_{n\times n}} \end{bmatrix} \quad (13)$$

R_k was composed of an upper left diagonal submatrix R_{xyz} representing the variation of the spatial dimensions and a diagonal submatrix R_d containing the maximum variance for each descriptor, as determined from the training data.

To associate features from one frame to the next, a window was sized around the respective state prediction value. An observation that fell in the window was a potential match for the track and its Mahalanobis distance was calculated. The GNN algorithm was used to find the best assignment for the tracks.

4 Results

The tests were conducted in Matlab on a Linux computer with an Intel Core i7 - M640 CPU with 2.8 GHz clock speed and 4 Gbytes of RAM.

The 64-D SURF descriptor vector was reduced to 46, 36, 32, 28, 27, 26, 20, 16, 10, 6 and 3 by PCA to estimate the minimum number of dimensions needed for reliable tracking.

Thirty frames from the freiburg1_xyz RGB-D dataset [8] were used for tracking. The camera ground truth was provided with the dataset. A Kinect camera was moved along its principal axes in X, Y and Z while viewing a stationary office environment. The first color and depth image of the sequence are shown in Figure 2.

Tracking performance with the 64-D descriptor was compared to the reduced descriptors. Five features were tracked over 30 frames. The accuracy of tracking with each descriptor dimension, for each track, was calculated as in (14). The number of correct matches for each track was verified manually.

$$Accuracy = \frac{Number\ of\ correct\ matches}{Total\ number\ of\ matches} \times 100 \quad (14)$$

Figure 2. Color image (left) and corresponding depth image (right) of RGB-D dataset [8].

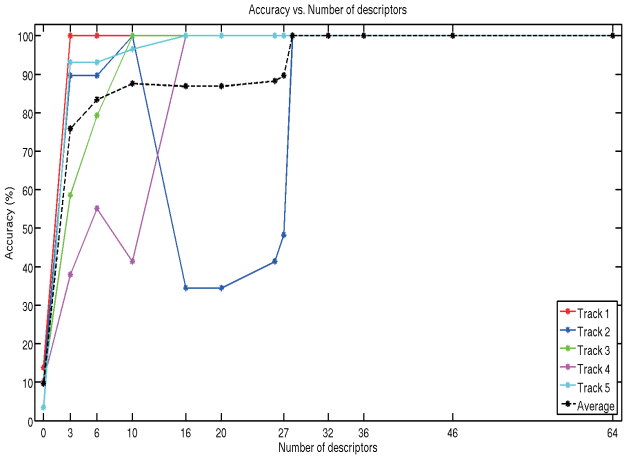

Figure 3. Accuracy versus number of descriptor dimensions.

The accuracy for each track and the average accuracy are shown in Figure 3. A track with 100% accuracy for a particular descriptor dimension was regarded as a correct track. The number of tracks correctly tracked by the KF for the descriptor dimensions tested are shown in Table 1.

The 64-D descriptor had 100% accuracy and 5 correct tracks. The accuracy and performance of 46, 36, 32, and 28-D descriptors were similar to the 64-D descriptor, shown in Figure 4.

The relatively large cluster sizes of the feature positions in world coordinates may be due to the non-synchronized color and depth images from the Kinect. There is a small time delay between the color and depth images [8]. Color and depth images were matched by time proximity and the average of their arrival times was used to identify the corresponding ground truth camera pose, used in computing the world coordinates.

Tracking performance started to decrease below the 28-D descriptor. The 27, 26, 20 and 16-D descriptors had 4 correct tracks. The 10-D descriptor had 3 correct tracks. The 6 and 3-D descriptor each had only 1 correct track. As seen in Figure 4, the average of the accuracy scores drops gradually from 28 to 10-D and then quite rapidly from 10 to 0-D. The 10-D descriptor retains a variance of about 90%, as shown in Figure 1.

The need for additional information beyond the spatial position of the feature is evidenced by the performance when the number of descriptors is zero as shown in Figure 5. None of the five tracks completed correctly, each drifted away from its original feature of interest, unable to distinguish it from other nearby features.

As the number of descriptor dimensions decreased, the accuracy and performance of the tracker decreased. The decrease in accuracy and drift in the unsuccessful tracks of the descriptor dimensions from 27 to 0 indicates that incorrect measurements were chosen to extend the track. As the number of dimensions decrease the descriptor loses its distinctiveness when compared to other descriptors and an incorrect measurement can easily be chosen.

With regard to the number of dimensions that might be required, it is visible in Figure 1 that even with 28 descriptors, about 99% of the variance is retained.

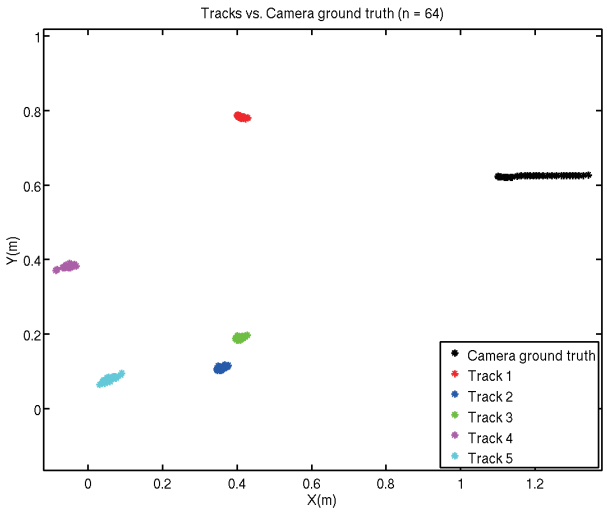

Figure 4. Tracks in world frame versus the camera ground truth for the 64-D descriptor.

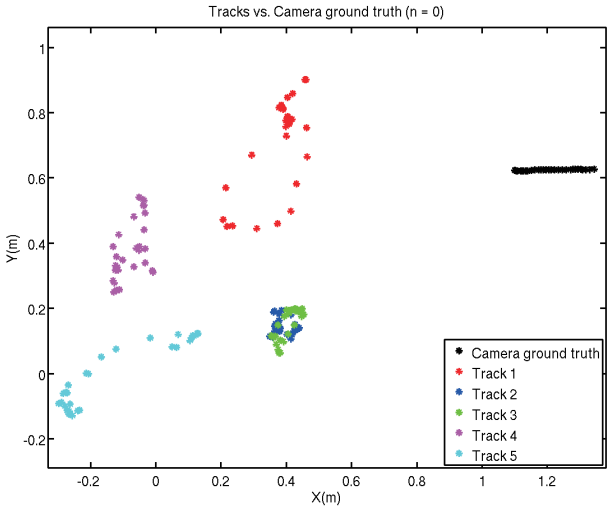

Figure 5. Tracks in world frame versus the camera ground truth for no descriptor dimensions.

This suggests that tracking should be successful with 28 descriptors.

The normalized time to track the five tracks for the different descriptors is shown in Table 1. The time decreased by about 40% from the 64-D descriptor to the 28-D descriptor.

5 Conclusions

Tracking of features in image data requires more than just the three-dimensional position and velocity estimates of the features. The natural choice for additional feature data is feature descriptor information such as SURF descriptors. These descriptors exist in a high dimensional space which can be effectively reduced using PCA.

In this paper PCA was applied to the 64-D SURF descriptor to reduce the descriptor dimensionality and computational time. Simulations validated the performance of the reduced PCA-SURF descriptors as compared to the standard SURF descriptors, providing an estimate of 28 for the minimum number of dimensions needed for reliable tracking. The low dimensionality of the PCA-SURF descriptors decreased computational time and hence optimized tracking speed.

In this work, tracking involved the KF with GNN data association. Tracking with algorithms that are more robust, e.g. [2], will be the subject of a future study and would be an important step in verifying the results of the present work.

Table 1. Descriptor dimension, number of correct tracks and normalized time to track 5 tracks.

Dimension	Correct tracks	Time(normalized)
64	5	1
46	5	0.78
36	5	0.68
32	5	0.65
28	5	0.59
27	4	0.57
26	4	0.58
20	4	0.54
16	4	0.51
10	3	0.46
6	1	0.42
3	1	0.41
0	0	0.39

References

[1] H. Bay, et al.: "SURF: Speeded Up Robust Features," *ECCV*, pp.404-417, 2006.

[2] S. S. Blackman, et al.: "Design and analysis of modern tracking systems," vol.685, Artech House Norwood, MA, 1999.

[3] S. Gauglitz, et al.: "Evaluation of interest point detectors and feature descriptors for visual tracking," *IJCV*, vol.94, no.3, pp.335–360, 2011.

[4] Y. Ke, et al.: "PCA-SIFT: A more distinctive representation for local image descriptors," *CVPR*, vol.2, pp.II-506-II-513, 2004.

[5] S. D. Lin, et al.: "Combining Speeded Up Robust Features with Principal Component Analysis in face recognition system," *IJICIC*, vol.8, no.12, pp.8545-8556, 2012.

[6] W-L. Lu, et al.: "Simultaneous tracking and action recognition using the PCA-HOG descriptor," *CRV*, pp.6-6, 2006.

[7] L. I. Smith: "A tutorial on Principal Components Analysis," vol.51, pp.52, Cornell University, USA, 2002.

[8] J. Sturm, et al.: "A benchmark for the evaluation of RGB-D SLAM systems," *IROS*, pp.573-580, 2012.

[9] S. Thrun, et al.: "Probabilistic Robotics," MIT Press, 2005.

[10] R. E. G. Valenzuela, et al.: "Dimensionality reduction through PCA over SIFT and SURF descriptors," *CIS*, pp.58-63, 2012.