# Classification of the Indoor Environment of a Mobile Robot Using Principal Component Analysis

Received 19th August 2007; accepted 7th November 2007.

*Tahir Yaqub, Jayantha Katupitya*

**Abstract:**

*Large indoor environments of a mobile robot usually consist of different types of areas connected together. The structure of a corridor differs from a room, a main hall or laboratory. A method for online classification of these areas using a laser scanner is presented in this paper. This classification can reduce the search space of localization module to a great extent making the navigation system efficient. The intention was to make the classification of a sensor observation in a fast and real-time fashion and immediately on its arrival in the sensor frame. Our approach combines both the feature based and statistical approaches. We extract some vital features of lines and corners with attributes such as average length of lines and distance between corners from the raw laser data and classify the observation based on these features. Bootstrap method is used to get a robust correlation of features from training data and finally Principal Component Analysis (PCA) is used to model the environment. In PCA, the underlying assumption is that data is coming from a multivariate normal distribution. The use of bootstrap method makes it possible to use the observations data set, which is not necessarily normally distributed. This technique lifts up the normality assumption and reduces the computational cost further as compared to the PCA techniques based on raw sensor data and can be easily implemented in moderately complex indoor environment. The knowledge of the environment can also be up-dated in an adaptive fashion. Results of experimentation in a simulated hospital building under varying environmental conditions using a real-time robotic software Player/Stage are shown.*

***Keywords:*** *mobile robot environment, PCA, classification, feature extraction, training data, bootstrap method*

## 1. Introductions and Related Work

Mobile robots have many applications like office delivery, building security, and transportation in mining, as straddles for container stacking in ships and others. Localization is the ability of a mobile robot to know its position during navigation. In indoor environments, GPS cannot be used and the ability to localize becomes crucial. Localization means finding a three dimensional state vector $[x,\ y,\ \theta]^T$ at the current time. Although a robot used for mail delivery may remain in a more structured office like environment but usually large buildings such as hospitals, universities, hotels etc have different areas linked together and a mobile robot has to move across different types of environments such as rooms, corridors and laboratories. For example a wheelchair moving in a hospital may come across these different areas of the same hospital building. A patient may be taken from a room through a corridor to laboratory and so on. In such hospital like environment, the localization task can be simplified by dividing the environment on top level. This will result in a reduced search space on a prior map. Some researchers have attempted the pose estimation problem for a mobile robot in an indoor environment using PCA. Pourraz [1] used an approach based on learning the scene by taking large number of images in the environment and then by reducing the dimensionality. Similarly Crowley [2] used PCA for the localization. An incremental strategy has been proposed by Artac [3] and [4] based on partial image updating. Jogan [5] also presented an appearance-based model of the environment using the panoramic snapshots. The model is then constructed by using approximation of this set of images. This is represented by principal components (also called *eigenvectors*) spanning the low dimensional space (also called *eigenspace*). Similarly Crowley [2], Krose[6], Artac [7] and Vlassis [8] used Principal component analysis for environment modelling and mobile robot localization. Weckesser [9] and Wu [10] presented sensor fusion techniques for constructing environment models. Rolfes[11] used random sets to statistically model the environment. Gasos[12] used fuzzy sets to represent the uncertainty in the environment. More recently [13] presented an action based environment modelling. The ultimate goal of all these techniques is to build a map to be used for location estimation and navigation of a mobile robot. However there are two main problems of perceptual aliasing and image variability as discussed in [14] which cause difficulties in making a model. We make the classification based on the variation in some correlated features. The variation is captured by principal components of feature data. We use a two-step process to accomplish this. PCA [15] is one of the oldest and best-known techniques for analysis of multivariate data. This has been used in many fields such as image analysis, climate analysis, chemometrics etc. The central idea of PCA is to reduce the dimensionality of a data set in which there are a large number of correlated variables while retaining the maximum of their variation. PCA analysis is a two-step process. The environment of the mobile robot is *learned* in the first step by taking a number of images or sensor measurements. The data obtained in this fashion is usually of many dimensions. Then these sensor observations are represented on an orthogonal basis of reduced dimensionality. The main objective of making the future inferences is achieved based on this data set. If a data set has $n$ components then we need $n \times n$ matrix to take into

account the variation of data, if we use variance covariance based approaches. The aim is to reduce the dimensionality from $n$ to $p$ where $p$ is much less than $n(p \ll n)$. In the method proposed here we extract first some valuable features from the environment using a feature extraction algorithm presented in [16] and then make use of two techniques namely bootstrap and PCA to get the classification of observations in some sensible fashion so that future observations may be able to classify within these groups in an online real-time manner.

## 2. Feature Based PCA

Principal Component Analysis (PCA) is a technique for multivariate analysis. The recommended reference is the book by Jolliffe [15]. The central idea of PCA is to reduce the dimensionality of any data set (in which there are large number of correlated variables), while still retaining the maximum variation present in the data. This reduction is achieved by transforming to a new set of variables, which are termed as principal components. These principal components are uncorrelated. In this way we can identify the patterns in the data and we can detect the similarities/dissimilarities. Usually a training data is used to model the system (The system in our case is the environment of a mobile robot), and then future inferences, about the incoming data, can be made using this model. These inferences could be just for the classification purpose or for any type of further investigations, which is of interest to us.

It is always desirable to use the raw laser data if possible because it contains the maximum information about the environment. The methods based on feature extraction are highly sensitive to outliers and hence their robustness is arguable. But use of raw laser data demands high processing power, which is usually not available in many applications. Dimensionality is referred here as the dimensionality of sensor data. Here we are using a feature-based approach and hence the number of features used for analysis will define the dimensions of data space. We are using LRF from SICK electro-optics with the resolution of 0.5° and it covers a field of view of 180°. Thus we get 361 range measurements $z_i$ in the form of $(r_i,\theta)$. Laser provides highly accurate and reliable data. The detailed characteristics can be seen in [17]. The data from laser scanner is obtained in a particular sequence. The first range measurement is at and angle of -90° and then all the way to 90° at an interval of 0.5°.

### 2.1. Principal Components

Let us say we have $v$ random variables stored in the observation vector $\mathbf{z}$. Now if $v$ is large then the variance-covariance way of getting the structure of the data would be very difficult. It would be much useful to get $u$ such that $u$ is much less than $v$ or $(u \ll v)$. The intention is to find a set of (up to $u$) linear functions namely $\alpha_1^T$ ..... $\alpha_u^T$ of the elements of $\mathbf{z}$ (where $T$ denotes the transpose), such that they are uncorrelated and capture maximum variability of the data:

$$\alpha_{\mathbf{1}}^T\mathbf{z} = \alpha_{11}z_1 + \alpha_{12}z_2 + \ldots\ldots\ldots + \alpha_{1v}z_{1v} = \sum_{i=1}^{v}\alpha_{1i}z_i$$

$$\alpha_{\mathbf{2}}^T\mathbf{z} = \alpha_{21}z_1 + \alpha_{22}z_2 + \ldots\ldots\ldots + \alpha_{2v}z_{2v} = \sum_{i=1}^{v}\alpha_{2i}z_i$$

$$\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots$$

$$\alpha_{\mathbf{u}}^T\mathbf{z} = \alpha_{u1}z_1 + \alpha_{u2}z_2 + \ldots\ldots\ldots + \alpha_{uv}z_{uv} = \sum_{i=1}^{v}\alpha_{ui}z_i$$

The first function $\alpha_1^T$ has the maximum variance (it captures the maximum variability of the data). Then the others in the descending order. All these are called the principal components (PCs). Total of upto $v$ PCs can be found but the expectation is that only $u$ will capture the maximum amount of variability in the data and $(u \ll v)$.

### 2.2. Feature Definition

Let us say that initially we defined 6 features to be detected in each laser scan. These were: *number of straight lines, number of corners, Average distance between adjacent corners in distance units, Average length of line in distance units, Slope of the largest line, Length of the largest line in distance units*. The usefulness of these features was tested and after an investigation presented in [18], the fifth feature was discarded and the remaining five were taken for further analysis. As an example, Fig-1 shows a simulated robot and the feature extracted from laser scan. Five line segments (at locations 1 to 5) and three corners were found at locations 1, 2 and 4. Other features described above (note that those are mostly the attributes of lines and corners) can also be found for all the locations used to get the training data.

## 3. Training

Data were collected at three different types of environment namely in a relatively simple structured room, in a corridor and in a complex laboratory like environment. The map used in simulation was of a hospital building provided with the Player/Stage [19] robotic simulator. Then the next step is where in each of these environmental sections, the robot would acquire the training data. The important considerations are the capability of the sensors, the quantity of training data, which we decided to obtain, the navigational strategy which is being implemented. In the hospital map used, the corridor is parallel to the x-axis, so we can get data at few locations along the corridor at an angle of 0° or 180°. Usually the rooms for patients in a hospital only differ slightly in overall structure and size. The poses were selected in an attempt to capture most of the variability. The complex environment is one, which is large and has more number of basic features at varying relative positions (not in repeating patterns).

### 3.1. Assumptions

It is assumed that the environment of the mobile robot is static. And also it is obvious that the small sample size was used based on the assumption that many patches of the environment are similar in structure or at least do

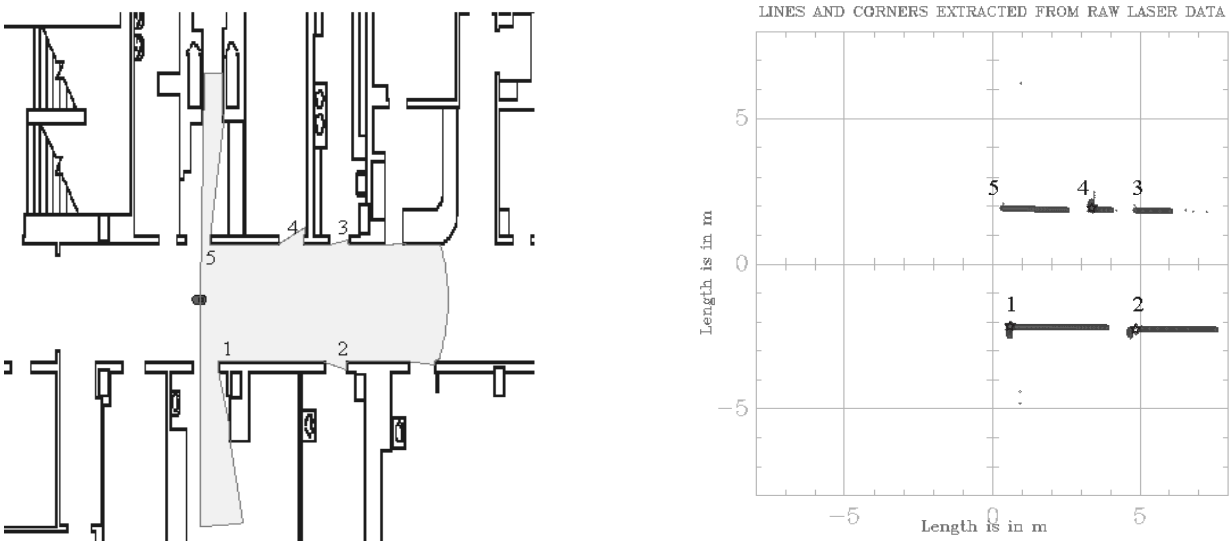


*Fig. 1. (a) Robot equipped with a laser scanner, navigating in a corridor of a hospital building, (b) Feature extracted from laser data. Different things were checked by putting ther obotin various locations and extracting features, to see the robustness of feature extraction algorithm. A star shows that the algorithm detects a corner, and the detected lines are represented by light gray color, also the points (dots) beneath these lines are the range points.*

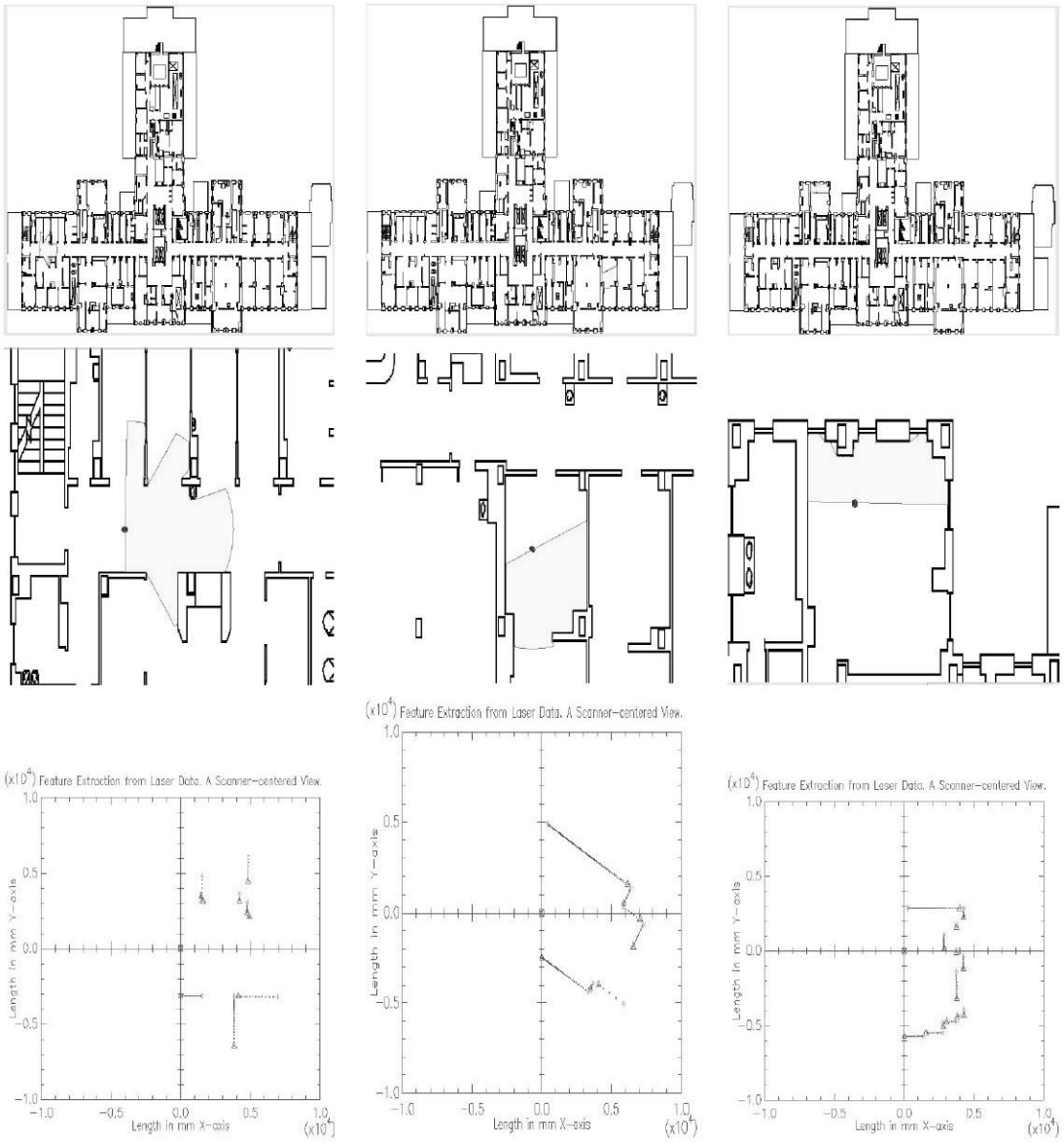


*Fig. 2. This figure shows one observation point at three different areas during training session. (a) Corridor, (b) Room, (c) Very complex Laboratory environment. The top row shows the robot in hospital. Second row shows the zoomed view and the last row shows the extracted features in the sensor frame.*

not change significantly. If the environment is very dynamic, this method may still work with some special arrangements. For example people walking on the floor adds to the complexity but the walls near the roof are clutter free. If we could mount the laser scanner in that direction, our static environment assumption may not cause any problem. But this was not tested in this work. At present most techniques treat the complexity such as moving objects and walking people as noise. The size of the hospital map used in the simulation was 100 m in both directions. If a robot occupies 1 $m^2$ space then in an area of 100 m × 100 m and with a resolution of 6°, there could be a total of 600000 possible poses. Getting the training data for all these poses is nearly impossible in real environment and time consuming even in simulation. To mitigate the effect of small sample size, a technique called bootstrapping was used as explained below.

### 3.2. Bootstrap Method

It is not viable to train using complete set of possible poses in a potential indoor environment. A technique called bootstrapping may be used to mitigate the effect. This was initially introduced by Efron and later used by many researchers and is discussed in [20]. In bootstrap methods, we can evaluate any statistics from a limited samples treated as pseudo-population. The main reason for using bootstrap is that for PCA, it is assumed that the underlying distribution is a multivariate normal which is usually not true. In the bootstrap analysis, we do not need to assume that the data is coming from and underlying normal or any other parametric distribution. Having observed a set of random samples of size $m$, we can define an empirical distribution and can calculate its standard error and confidence intervals. However we first use to calculate the mean of the feature vectors. If the environment is to be classified in $\kappa$ different environment and we decided to use $n$ features then from the training data set, $\kappa \times n$ mean values would be calculated from as many feature vectors using 300 bootstrap samples (samples with replacement), from each of these vectors. This number was used based on the discussion, which can be found in [20].

## 4. PCA Based Classification Method

Even with a static assumption, the environment is very complex and simple variances and co-variances based analysis is not much useful for discrimination between different parts of the environment even in the broader sense. The motivation to use the PCA is to estimate a linear function $\alpha_1^T \mathbf{z}$ of $\mathbf{z}$ that captures the maxi-mum of variance present in its components subject to being uncorrelated with some more similar linear functions. Here $T$ denotes the transpose and $\alpha_1$ is a vector of $v$ constants because $v$ features are being carried forward.

$$\alpha_1^T \mathbf{z} = \sum_{i=1}^{v} \alpha_{1i} z_i$$

If we take $v$ features (in our experimentation $v = 5$), then possibly there could be $v$ such uncorrelated functions called PC's which form the columns of a matrix say $\mathrm{A}$, but usually first few (say $u$) capture most of the variability of the data. This matrix $\mathrm{A}$ can be found from the covariance matrix of $\mathbf{z}$ (which would be replaced by feature matrix $F$ or $F_s$ in standardized form in the following discussion). Lets call this covariance matrix $C$. The $(i,j)th$ element of $C$ is the covariance between the $i_th$ and $j_th$ dimension and is given by:

$$\mathcal{C}_{i,j} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{z}_{ij} - \overline{\mathbf{z}_j})(\mathbf{z}_{ik} - \overline{\mathbf{z}_k}) \tag{1}$$

and

$$C\mathrm{A} = \mathrm{A}\Lambda \tag{2}$$

$\mathrm{A}$ is the matrix of eigenvectors of this covariance matrix and $\Lambda$ is the diagonal matrix whose $kth$ diagonal element $\alpha_k$ is $kth$ eigenvalue of $C$ and denotes $var(\alpha_k^T \mathbf{z})$. In our case covariance matrix has five eigenvalues. We call these values as $\alpha_1 \ldots \alpha_5$ labelled in decreasing order of magnitude. $\alpha_1$ is an eigenvector of $C$ i.e a column of $\mathrm{A}$ which corresponds to the largest eigenvalue $\lambda_1$. The eigenvalues and eigenvectors are actually computed by a complicated series of algebraic manipulations requiring on the order of $n^3$ calculations when $C$ is a $n \times n$ matrix. Therefore after the proposed feature extraction, the dimensionality of laser data reduces drastically and hence the computational time. This method can therefore be used in an adaptive manner to update the representation of the environment. Supposing that the environment is to be classified in $\kappa$ different areas. In our experimentation, $\kappa = 3$ and the areas are corridor, room and laboratory. Training data set was obtained in these three types of areas. It has also decided that $n$ features would be extracted from every sensor observation. For simplicity we assume that we take equal number of $m$ observations in all these areas. The data collected at these locations would be stored in three different matrices each having $n$ columns and $m$ rows. Let us use the superscript $C$, $R$ or $L$ denoting corridor, room and lab respectively (not as a power but simply a character). Denoting these three matrices by $F_1$, $F_2$ and $F_3$:

$$F_1 = \begin{pmatrix} F_{11}^C & \cdots & F_{1n}^C \\ \vdots & \ddots & \vdots \\ F_{m1}^C & \cdots & F_{mn}^C \end{pmatrix}$$

$$F_2 = \begin{pmatrix} F_{11}^R & \cdots & F_{1n}^R \\ \vdots & \ddots & \vdots \\ F_{m1}^R & \cdots & F_{mn}^R \end{pmatrix} \tag{3}$$

$$F_3 = \begin{pmatrix} F_{11}^L & \cdots & F_{1n}^L \\ \vdots & \ddots & \vdots \\ F_{m1}^L & \cdots & F_{mn}^L \end{pmatrix}$$

And in vector form, this can be written as:

$$F_1 = [\ F_1^C \ \ldots \ F_n^C\ ]$$
$$F_2 = [\ F_1^R \ \ldots \ F_n^R\ ] \quad (4)$$
$$F_3 = [\ F_1^L \ \ldots \ F_n^L\ ]$$

These are $\kappa \times n$ vectors, each of length $m$. The complete feature matrix $F$ of the environment is then obtained by augmenting these three matrices:

$$F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (5)$$

If $m$ measurements of $n$ features were taken in $\kappa$ different areas of the environment then this matrix would have $\kappa \times m$ rows and $n$ columns.

### 4.1. Stepwise Procedure

1. First we determine the bootstrap mean of these $\kappa \times n$ vectors and put the values in a matrix. Let us call the matrix of these values as our *basic matrix* $B$, obviously of the same dimensions $\kappa \times n$.

$$B = \begin{bmatrix} \mu_{F_1^C} \ldots\ldots\ldots \mu_{F_n^C} \\ \mu_{F_1^R} \ldots\ldots\ldots \mu_{F_n^R} \\ \mu_{F_1^L} \ldots\ldots\ldots \mu_{F_n^L} \end{bmatrix} \quad (6)$$

Here $\mu_{F_i^j}$, (where $i$=1,2,..$n$ and $j$=$C$,$R$,$L$) are the bootstrap means obtained by taking 300 samples with replacement from vectors of (4).

2. The purpose of $B$ is to find the model parameters. Mean and standard deviation of each column of $B$ would be used for standardization. The next step is the standardization of the matrix $F$. Let the standardized basic matrix be denoted by $F_s$. To find this from $B$, the mean of each column of $B$ would be calculated and subtracted from every element of the same column of matrix $F$. The difference would then be divided by the standard deviation of that column of $B$. This will result the standardized value for every element.

2. Find the covariance matrix of $F_s$. This is the matrix $C$ of (2).

4. Find the matrix of eigenvalues $\Lambda$ and matrix of eigenvectors A of the covariance matrix $C$. Matrix $\Lambda$ of eigenvalues would be a diagonal matrix. Each of the diagonal entry is an eigenvalue and has a corresponding vector (a column) in matrix A .

5. The eigenvector, which corresponds to the highest eigenvalue, is the first principal component denoted by $\alpha_1$ and so on. At this point, the question is that how many principal components should we use. The detailed discussion can be found in[15]. For this we first decide how much variation of the environment we want to capture (usually 70 to 90 percent is a reasonable value). Then the required principal components are the smallest value for which this chosen percentage is exceeded. We can check the amount of variation which is captured by each of the principal components $\alpha_1 \ldots \alpha_n$. If $\lambda_1$ is the largest eigenvalue and $\hat{\theta}$ is the ratio of the largest to total of eigenvalues, then

$$\hat{\theta} = \frac{\lambda_1}{\sum_{i=1}^{n} \lambda_i}$$

gives us the percentage of variation captured by first principal component. We got reasonable value of 0.81 captured by $\alpha_1$ and $\alpha_2$. The confidence in this value can also be calculated using bootstrap. The detailed discussion of whether the two PCs are sufficient can also be found in [15]. Now we decided to take only first two principal components. we define $n_{PC}$= Number of principal components considered for modelling. The matrix containing these two PCs as columns is denoted by $N$, which is given by.

$$N = [\alpha_1 \ \ \alpha_2]$$

Matrix $N$ would be of size $n \times n_{PC}$.

6. Now the training data, which is in the standardized matrix $F_s$, would be transformed using this matrix $N$. and we get the transformed data matrix $\tau$.

$$\tau = N^T F_s^T \quad (7)$$

This matrix $\tau$ will have the rows equals to $n_{PC}$, and columns equals to $\kappa \times m$. The transformed training data will be stored in matrix $\tau$ and there would be three separate sub-matrices inside $\tau$ and there sizes and positions can be imagined like this:

$$\tau = \begin{bmatrix} Room & Lab & Corridor \\ \underbrace{n_{PC} \times m} & \underbrace{n_{PC} \times m} & \underbrace{n_{PC} \times m} \end{bmatrix} \quad (8)$$

It would be convenient to denote each of these by $\tau_R$, $\tau_L$ and $\tau_C$, each of size $n_{PC} \times m$.

7. After the transformation, we have the feature data transformed to the frame of principal components. We can look for any clustering of the transformed feature vectors. And if found then we use a metric to classify the future oncoming observation into one of $\kappa$ areas.

### 4.2. Classification

The important model parameters were found using the matrix $B$ above. Each column provides the mean and standard deviation of each feature. Note that this is not the bootstrap mean. Bootstrap means are calculated for individual elements of matrix $B$. The means and standard deviations of columns would be used to standardize the

online sensor observation. When a sensor observation arrives, the feature extraction would be carried out and then following steps would be performed:

1. Let the laser sensor observes the environment and after feature extraction, the observation is denoted by $Q = (F1, F2, ..., Fn)$ having size $1 \times n$.
2. Standardize the observation by taking the difference of each feature value and the mean of the corresponding column of $B$ and dividing this difference by the standard deviation of that column. This is denoted by $Q_s$ having same size $1 \times n$.
3. Transform the standardized observation using all PCs or just the first two PCs. This is denoted by $\hat{Q}_s$ and is obtained by:

   $$\hat{Q}_s = N^T Q_s^T$$

   This transformed observation $\hat{Q}_s$ is a single column vector having rows equal to $n_{PC}$ i.e. 2 in our case.
4. Find three vectors of Euclidean distances between the vector $\hat{Q}_s^T$ i and the columns of three sub-matrices $\tau_R$, $\tau_L$ and $\tau_C$, shown in 8 and take the average.

   $D_R$ = Average distance between $\hat{Q}_s^T$ and the columns of $\tau_R$.
   $D_L$ = Average distance between $\hat{Q}_s^T$ and the columns of $\tau_L$.
   $D_C$ = Average distance between $\hat{Q}_s^T$ and the columns of $\tau_C$.

The minimum of these distances implies a close association of the current observation with that area of the environment on principal component axis. If two PC's are used, this can easily be seen in a graphical representation. But for more PC's, techniques such as biplots can be used for graphical representation but it's hard to visualize the closeness in high dimensional data. Classification, which we get, can be used in many ways. Each observation will give us some features and similar features would represent the same (or closer) point in the eigenspace. The other and more general way, which we used here, is to make a grid of poses such that minimum variation of observation (in eigenspace) is present in nearest poses. This makes the position tracker more robust.

### 4.3. Algorithm

**Algorithm 1**: PCA Based Classification of Environment

**Data**: Model data obtained during training (matrix $F$ of Eq-5) and current real-time sensor observation
**Result**: Location of the robot in one of the three broader Environment Classes (Areas).
**Initialization** :

1. Get matrix $B$ (Eq-6)from training data as explained above.
2. Standardize $B$ and get $F_s$.
3. Set $n_{PC} = 2$ (Assuming 2 PC's capture the desired variation)
4. Find the covariance matrix of $F_s$. This is the matrix $\mathcal{C}$ of Eq-(2).
5. Find the matrix of eigen values $\mathbf{\Lambda}$ and matrix of eigen vectors $\mathbf{A}$ of the covariance matrix $\mathcal{C}$.
6. Find the matrix $N$ of first 2 PC's $\alpha_1$ and $\alpha_2$. Transform $F_s$ using this matrix $N$, and get $\tau$ using Eq-7.

**while** *Navigating* **do**

1. Get the current sensor observation $Q$ in the form of a row vector of features having size $1 \times n$.
2. Standardize this observation and get $Q_s$, using either of the two methods (one describe in Step-2 of procedure and the other in section 5.3.)
3. Transform $Q_s$ using Eq-9 and get $\hat{Q}_s$.
4. Find the values of $D_R$, $D_L$ and $D_C$ and the minimum of these three.

**switch** *(The minimum of $D_R$, $D_L$ and $D_C$)* **do**

**case** $D_R$
The robot is in the room;
break;
**case** $D_C$
The robot is in the corridor;
break;
**otherwise**
(Obviously $D_L$ would be minimum). The robot is in the Laboratory
**end**

**end**

**end**

## 5. Experimental validation

We used a simulated robot equipped with a laser scanner under a real-time software Player/Stage [19]. The advantage of using simulation is that we can easily change the environment of the robot and verify the robustness. Before going for the validation of modelling, we need to check the robustness of feature extraction algorithm. We put the robot in different locations using a map. The locations were chosen intelligently in an effort to detect any flaws in the algorithm meaning any situations in which it might fail. As compare to real environment, the situation here was different and favourable, because while using a map, we know how many lines are there and what is the location and other attributes of these lines. We can therefore easily compare the actual and detected features. So we are able to check the accuracy and be sure that the algorithm detects the correct features. The results were excellent, and we were able to detect exactly the same lines as there were in the map. In some cases however, we got two/three small lines instead of a single long one. Merging can rectify this problem. We are positive that this type of merging is required seldom and will not affect the performance. There are many ways of representing the multidimensional data but we shall use the biplots.

*Table 1. Feature matrix F of the environment.*

| Room | | | | |
|---|---|---|---|---|
| F1 | F2 | F3 | F4 | F6 |
| 7 | 5 | 2007 | 1845 | 6677 |
| 7 | 4 | 2128 | 1520 | 5694 |
| 7 | 4 | 2128 | 1611 | 6333 |
| 6 | 3 | 3280 | 3269 | 7797 |
| 5 | 2 | 1682 | 3073 | 7797 |
| 5 | 2 | 2231 | 2973 | 7797 |
| 5 | 3 | 906 | 3269 | 7797 |
| 5 | 3 | 873 | 3277 | 7797 |
| 5 | 3 | 906 | 3269 | 7797 |
| 5 | 3 | 859 | 2850 | 7797 |
| Lab | | | | |
| F1 | F2 | F3 | F4 | F6 |
| 13 | 6 | 2085 | 882 | 3737 |
| 14 | 8 | 1350 | 928 | 2730 |
| 7 | 3 | 2130 | 945 | 2011 |
| 7 | 4 | 2350 | 970 | 2407 |
| 6 | 2 | 5970 | 2394 | 5791 |
| 14 | 8 | 1723 | 1222 | 2575 |
| 7 | 1 | 0 | 1378 | 3121 |
| 6 | 2 | 8242 | 2448 | 7602 |
| 10 | 4 | 3386 | 1268 | 4615 |
| 14 | 6 | 2493 | 951 | 2450 |
| Corridor | | | | |
| F1 | F2 | F3 | F4 | F6 |
| 9 | 4 | 3027 | 1423 | 3386 |
| 9 | 5 | 2346 | 1431 | 3667 |
| 8 | 4 | 2273 | 890 | 3587 |
| 10 | 5 | 1624 | 1408 | 5466 |
| 11 | 5 | 2651 | 748 | 2265 |
| 12 | 7 | 2927 | 931 | 2180 |
| 13 | 7 | 2535 | 946 | 2437 |
| 4 | 2 | 1189 | 1690 | 2882 |
| 4 | 2 | 1156 | 1495 | 1961 |
| 7 | 3 | 4436 | 2019 | 3983 |

### 5.1. The Experimental Data

The matrix of feature data collected for training is shown in Table-1. After standardization, we obtain the matrix shown in Table-2.

*Table 2. Standardized feature matrix $F_s$ of the environment.*

| Room | | | | |
|---|---|---|---|---|
| F1 | F2 | F3 | F4 | F6 |
| -0.3328 | 0.5332 | - 0.2220 | 0.0773 | 0.8613 |
| -0.3328 | 0 | -0.1465 | -0.2944 | 0.4247 |
| -0.3328 | 0 | -0.1465 | -0.1903 | 0.7085 |
| -0.6448 | -0.5332 | 0.5716 | 1.7059 | 1.3588 |
| -0.9569 | -1.0664 | -0.4246 | 1.4817 | 1.3588 |
| -0.9569 | -1.0664 | -0.0823 | 1.3673 | 1.3588 |
| -0.9569 | -0.5332 | -0.9083 | 1.7059 | 1.3588 |
| -0.9569 | -0.5332 | -0.9289 | 1.7150 | 1.3588 |
| -0.9569 | -0.5332 | -0.9083 | 1.7059 | 1.3588 |
| -0.9569 | -0.5332 | -0.9376 | 1.2267 | 1.3588 |
| Lab | | | | |
| F1 | F2 | F3 | F4 | F6 |
| 1.5393 | 1.0664 | -0.1734 | -1.0241 | -0.4446 |
| 1.8513 | 2.1328 | -0.6315 | -0.9715 | -0.8918 |
| -0.3328 | -0.5332 | -0.1453 | -0.9520 | -1.2112 |
| -0.3328 | 0 | -0.0082 | -0.9234 | -1.0353 |
| -0.6448 | -1.0664 | 2.2484 | 0.7052 | 0.4678 |
| 1.8513 | 2.1328 | -0.3990 | -0.6352 | -0.9607 |
| -0.3328 | -1.5996 | -1.4730 | -0.4568 | -0.7182 |
| -0.6448 | -1.0664 | 3.6646 | 0.7669 | 1.2722 |
| 0.6032 | 0 | 0.6376 | -0.5826 | -0.0546 |
| 1.8513 | 1.0664 | 0.0810 | -0.9452 | -1.0162 |
| Corridor | | | | |
| F1 | F2 | F3 | F4 | F6 |
| 0.2912 | 0 | 0.4138 | -0.4054 | -0.6005 |
| 0.2912 | 0.5332 | -0.0107 | -0.3962 | -0.4757 |
| -0.0208 | 0 | -0.0562 | -1.0149 | -0.5112 |
| 0.6032 | 0.5332 | -0.4607 | -0.4225 | 0.3234 |
| 0.9153 | 0.5332 | 0.1795 | -1.1773 | -1.0984 |
| 1.2273 | 1.5996 | 0.3515 | -0.9680 | -1.1361 |
| 1.5393 | 1.5996 | 0.1072 | -0.9509 | -1.0220 |
| -1.2689 | -1.0664 | -0.7319 | -0.1000 | -0.8243 |
| -1.2689 | -1.0664 | -0.7524 | -0.3230 | -1.2334 |
| -0.3328 | -0.5332 | 1.2921 | 0.2763 | -0.3353 |

### 5.2. Results for Simple and Complex Environment

Although a relatively small data set was used for training purpose but in simple environments classification was accurate. Even without any assistance of motion model, for the case of room around 70% of the time, the algorithm was able to correctly identify the location from where observation was coming. Similarly for simple room, the success rate was nearly 60%. However the method was failed to classify the observation coming from a complex laboratory environment and this was expected with this simple and efficient strategy. Two different programs were made, one using two PC's and the other using all the 5 PC's. There was no significance difference and the outcome was nearly the same. This fact was also clear by plotting all three types of observations in the frame of principal components. It can be seen that the laboratory data were scattered all over the graph while the observations from a room and corridor were somewhat clustered. This can be seen in Fig-3 shows the graphical representation of the training data set when two PC's were used for transformation. The resulting 2-D vectors were plotted and hence the axis are shown as PC1 and PC2 due to the fact that the 2-D transformed vector has been obtained using the first two PC's. In Fig-3-(a), the data from room is plotted red, from the corridor in blue and from lab as magenta. In Fig-3-(b), the data points from three areas were joined in a fashion that the rest of the entries are within the respective boundaries. It can be seen that with the exception of outliers, the room and corridor data falls in different and separate regions of the plot. But the data from lab shares a common region. This means that the lab data is hard to distinguish. This is because we are taking the average minimum Euclidian distance as a metric, which is efficient but has the limitations to handle complex environments.

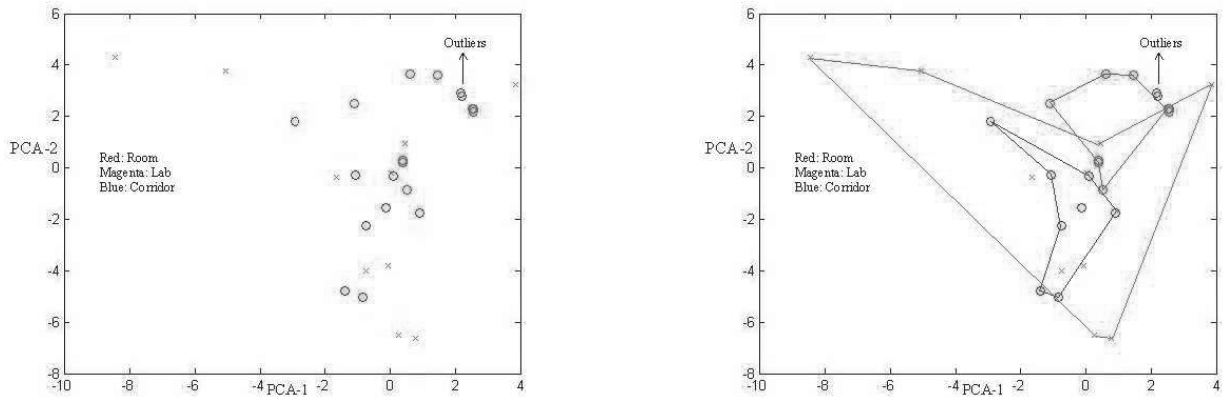


*Fig. 3. This figure shows the training data drawn in the frame of first two principal components. Instead of xy-plane, this is the plane of first 2PC's. The data was standardized using B as described in step-2 of Procedure. (b) Outer observations were joined to get a cluster-like region.*

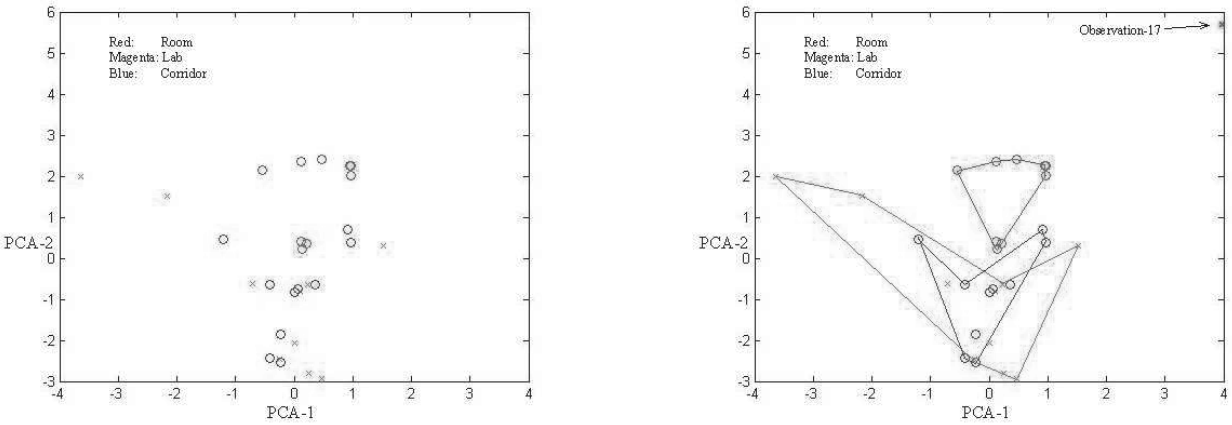


*Fig. 4. This figure shows the training data drawn in the frame of first two principal components. Instead of xy-plane, this is the plane of first 2PC's. The data was standardized using F as Method-2 above. (b) Outer observations were joined to get a cluster-like region.*

### 5.3. Method-2 for Standardization

In step (2) of procedure described above, mean and standard deviation of each column of $B$ was used for standardization. Instead the mean and standard deviation of the columns of the matrix $F$ itself can also be used to get $F_s$. To find this the mean of each column of $F$ would be calculated and subtracted from every element of the same column. The difference would then be divided by the standard deviation of that column of $F$. This will result the standardized value for every element. This approach resulted in reduced differences between the values of $D_R$, $D_L$ and $D_C$ and some outliers were also removed as can be seen in Fig-4. The outliers were removed but the observations from complex environment were still misclassified. And exactly the same number of observations was correctly classified for simple environments. The method proved to be robust for simple environment and can be used efficiently for a well-structured environments but is not capable of handling the complex environments. The solution to this problem is discussed in last section.

## 6. Conclusion and future work

The use of principal component analysis for the classification of the environment of a mobile robot is investigated. The training data was obtained in a simulated hospital building. The environment was divided in three different areas. PCA is a proven method of dimensional reduction and the intention was to get the environment perception for mobile robot navigation. The data handling capability of laser scanners is enormous. A huge amount of data is produced within a second. With this much data, interpretation is quite difficult and the dimension reduction leads to loss of discriminatory information. However After a series of experiments with data obtained from different environments, it was found that the method has good potential for structured environment. It can be used to distinguish between simple and static environments but it was found that this might not be used for complex environments without a special arrangement described earlier. Other improvements, which come to mind for complex environments, are the use of some more feature definitions and the use of very large training data. But defining more features will also impose some problems in regards to outliers and this should be kept in mind. The method does converges to a pose with acceptable error in simple static environment and a reasonable size of state decomposition if a motion model assisted with a prior map is used. There are two criticisms about PCA based methods and we tried to fix these problems. One issue is the underlying assumption of multivariate normality and the other is the outlier issue. Both of these issues are discussed in detail in [15] and the proposed solution was to use a robust way to calculate the covariance matrix. We used a novel approach of bootstrapping to overcome the issue of multivariate normality and we found the covariance, which is a robust way, but it was found that the effect of outliers is greater than the expected. The major problem in detecting an outlier in a multivariate date is that the observed value may not be inconsistent on individual variable basis but actually it is inconsistent with the correlation structure of the data and hence is an outlier. This type of outlier is really very difficult to detect. Since the volume of laser data is huge and as the amount of data increases, more outliers are also included in the data, and handling of outliers become an issue. These outliers can be treated and many ways have been proposed in the literature but the attempt to handle the outliers increases the computational burden and raises the efficiency issues. The improvements that come to mind are the use of some more feature definitions and the use of very large training data. But defining more features will also impose some problems in regards to outliers.

**AUTHORS**

**Tahir Yaqub and Jayantha Katupitya** - School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, Australia, e-mail: t.yaqub@unsw.edu.au.

**References**

[1] F. Pourraz, J.L. Crowley, "Use of eigenspace techniques for position estimation". In: *International Workshop on Advanced Motion Control, AMC*, 1998, pages 267 271.

[2] J. L. Crowley, F. Wallner, B. Schiele, "Position estimation using principal components of range data", *Robotics and Autonomous Systems*, vol. 23, 1998, pp. 267-276.

[3] M. Artac, "Mobile robot localisation with incremental pca". In: *Proceedings of the Mediter-ranean Electrotechnical Conference - MELECON*, vol. 1, 2002, pp. 192-197.

[4] M. Artac, M. Jogan, A.Leonardis, "Mobile robot localization using an incre-mental eigenspace model". In: *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 1025-1030.

[5] M. Jogan, L. Ales, "Robust localization using an omnidirectional appearance-based subspace model of environment", *Robotics and Autonomous Systems*, vol. 45, issue 1, 2003, pp. 51-72.

[6] B. J. A. Krose, N. Vlassis, R. Bunschoten, Y. Motomura, "A probabilistic model for appearance-based robot localization", *Image and Vision Computing*, vol. 19, April 2001, pp. 381-391.

[7] M. Artac, M. Jogan, and A. Leonardis, "Mobile robot localization using an incremental eigenspace model". In: *"Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington D.C., May 2002, pp. 1025-1030.

[8] N. Vlassis, B. Krose, "Robot environment modeling via principal component regression". In: *Proceedings of the 1999 IEEERSJ International Conference on Intelligent Robots and Systems*, volume 2, 1999, pp. 677-682.

[9] P. Weckesser, R. Dillmann, "Modeling unknown environments with a mobile robot", *Robotics and Autonomous Systems*, vol. 23, July 1998, pp. 293-300.

[10] Yong-Ge Wu, Jing-Yu Yang, and Ke Liu, "Obstacle detection and environment modeling based on multisensor fusion for robot navigation", *Artificial Intelligence in Engineering*, vol.10, issue 4, November 1996, pp. 323-333.

[11] S. Rolfes, M-J. Rendas. "Statistical environment repre-

sentation for navigation in natural environments". *Robotics and Autonomous Systems*, vol. 41, no. 2, 2002, pp. 129-136.

[12] J. Gasos, A. Rosetti, "Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments", *Fuzzy Sets and Systems*, vol. 107, issue 1, October 1999, pp. 1-24.

[13] S. Yamada, "Recognizing environments from action sequences using self-organizing maps", *Applied Soft Computing*, vol. 4, February 2004, pp. 35-47.

[14] B. Kuipers, P. Beeson, "Bootstrap learning for place recognition". In: *18th national conference on Artificial intelligence*, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence, pp. 174-180.

[15] I.T. Jollife, *Principal Component Analysis*, Springer-Verlag, 1986.

[16] T. Yaqub, J. Katupitiya, "Modelling the environ-ment of a mobile robot using feature-based principal component analysis". In: *IEEE interantional conference on Cybernetics, Intelligent Systems and Robotics, Automation and Mechatronics (IEEE CIS-RAM-06)*, Bangkok, Thailand, June 2006, pp. 1-6.

[17] Y. Cang, J. Borenstein, "Characterization of a 2d laser scanner for mobile robot obstacle negotiation". In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 3, Washington D.C., May 2002, pp. 2512-2518.

[18] Tahir. Yaqub. and J. Katupitiya. *Robot Motion and Control*, chapter A Parametric Repre-sentation of the Environment of a Mobile Robot for Measurement Update in a Particle Filter, Springer-Verlag Ltd, London, June 2007.

[19] B. P. Gerkey, R. T. Vaughan, A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems". In: *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, 2003, pp. 317-323.

[20] B. Efron, R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall/CRC, May 1994.