

## Chapter 2

# Background and Literature Review

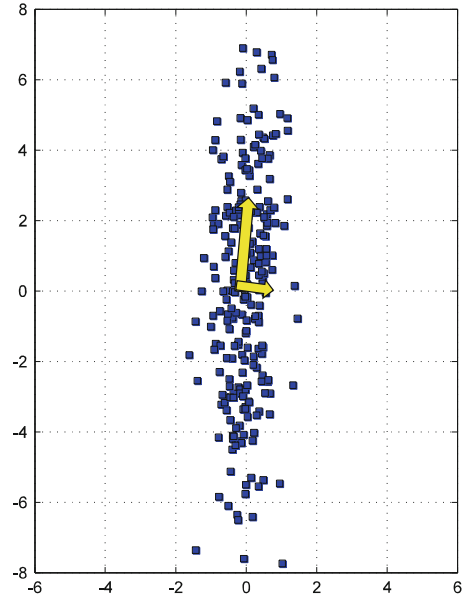
**Abstract** In this chapter, we first review the problem of linear subspace estimation and present example problems where the conventional method (PCA) is typically used. Consequently, we discuss the most prominent advances in low-rank optimization, which is the main theoretical topic of this book. Since the various low-rank formulations discussed in this book fall into several computer vision domains, we additionally review the latest techniques in each domain, including video denosing, turbulence mitigation, background subtraction, and activity recognition.

### 2.1 Linear Subspace Estimation

Consider a set of points drawn from an unknown distribution, as illustrated in blue in Fig. 2.1. For simplicity and ease of illustration, two dimensional points are considered; however, this discussion can be extended to any number of dimensions. The most popular method to estimate an orthogonal basis set is the Principal Component Analysis, where a set of orthonormal eigenvectors and their corresponding eigenvalues are computed such that the reprojection along these directions has a minimum reconstruction error. As discussed in the previous chapter, these eigenvectors (basis) can be sorted according to the variance along each basis direction using the corresponding eigenvalue. In practise, the basis vectors with insignificant variance are ignored since they typically correspond to noise. This also results in a major reduction of dimensionality, which is often desirable in many machine learning techniques which scale exponentially with data dimension. Figure 2.1 illustrates the computed basis in yellow, where the length of the vectors are weighted using the eigenvalues (the longer vector correspond to higher eigenvalue and higher variance).

This forms the basic structure of a wide variety of computer vision and machine learning applications. For example, consider a set of images of size  $100 \times 100$ , which show the face of a human in different light conditions, poses, etc.  $10K$  dimensions are required to represent each image. Alternatively, PCA can be applied,

**Fig. 2.1** The *squares* show sample data points drawn from an unknown distribution, and the *arrows* show the estimated basis vectors computed using PCA. The length of the basis vector is proportional to the amount of variance in the data along the corresponding direction



and the major  $k$  basis can be extracted. Consequently, each image can be projected along each basis vector, and represented as only a set of  $k$  coefficients. These coefficients can then be employed for instance to compare or classify the faces. This is essentially the eigenfaces approach for face recognition originally proposed by Turk and Pentland [129].

A wide variety of extensions to PCA have been proposed, from which we briefly discuss the most prominent approaches.

- **Kernel PCA (KPCA) [109]:** Kernel PCA is an extension for PCA to handle the case where the data points lie in a non-linear subspace. Embedding this data into a higher dimensional space (feature space) allows it to behave linearly (can be linearly separated). In KPCA, this non-linear mapping is never found. Instead, since the data points always appear in a dot product form, the kernel trick is used, in which each point is represented using the distances to all other points in order to form a kernel matrix. Consequently, Eigen Value Decomposition is applied on this new representation. Since the actual high dimensional embedding is not explicitly computed, the kernel-formulation of PCA is restricted in that it does not compute the principal components themselves, but the projections of the data onto those components. Kernel PCA has been demonstrated to be useful in several applications such as novelty detection [55] and image de-noising [92].
- **Probabilistic PCA [127]:** The standard definition of PCA lacks an associated probabilistic model for the observed data. Therefore, probabilistic PCA addresses that by deriving PCA within a density estimation framework. This is appealing because it enables comparison with other probabilistic approaches and permits the application of PCA in Bayesian inference-based methods. Probabilistic PCA

assumes a Gaussian noise model and formulates the solution for the Eigen vectors as a Maximum Likelihood parameter estimation problem. Consequently, the parameters of the principal subspace are computed iteratively using Expectation Maximization algorithm (EM). Since probabilistic PCA estimates a generative model, it can also handle the case where the data vectors have missing values.

- PCA for noise in exponential family [27]: PCA implicitly minimizes a squared loss function, which intrinsically assumes a Gaussian noise model. However, for data that is not real-valued, other noise models often fit better. For example, a Bernoulli distribution better describes binary data, and a Poisson distribution may better fit integers, and an exponential for positive-valued data. All these distributions have probability density functions which can be formulated as members of the exponential family (similar to a Gaussian) and thus PCA can be extended to incorporate that.
- Generalized PCA (GPCA) [87]: GPCA extends PCA to handle the case where the data points are drawn from multiple subspaces. This case is far more difficult, since the problem inherently includes data segmentation. GPCA is an algebraic geometric approach to data segmentation; therefore, it is different than probabilistic approaches, such as the Gaussian Mixture Models (GMM), or spectral clustering methods . . . etc.

One of the critical issues in PCA is robustness against outliers. PCA is sensitive to outliers; which often falsely contribute in the computed basis, and result in a potentially inaccurate basis. Popular outlier detection methods such as RANSAC cannot be employed since they often require prior knowledge of a model which fits the data, which is often unknown or non-existing. This is inarguably the most prominent reason behind the research for a method which can detect the outliers and estimate the basis simultaneously. In the coming section, we discuss the recent developments in low-rank optimization, which represents one of the most successful approaches in the area of robust subspace estimation.

## 2.2 Low-Rank Representation

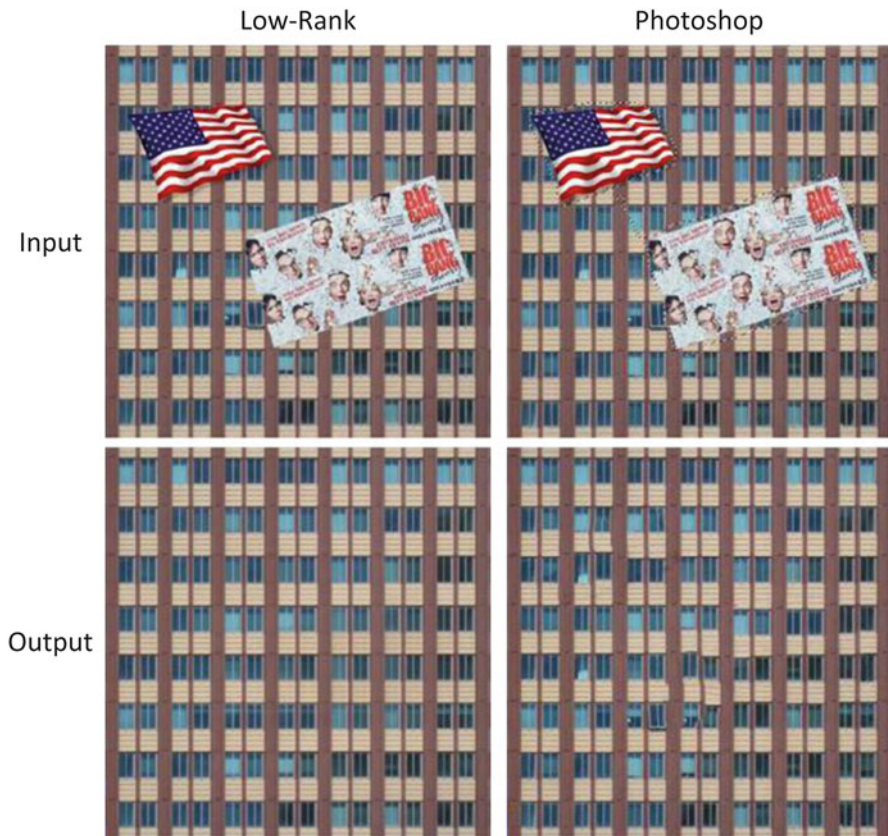
A low-rank structure can be identified as a set of observations which can be represented using a low number of basis. We observe such low-rank structures frequently in our everyday life. For instance, building facades are typically composed of groups of repetitive structures, which together form a low-rank space (each group corresponds to a certain basis vector in the space of the facade). Similarly, the frames of a video may also correspond to a low-rank subspace because, in principle, the video can be divided into groups of frames, where the frames of each group are similar or linearly correlated (the frames of a group may vary only in multiplicative factors such as illumination changes). Refer to Fig. 2.2. It is also not difficult to think of various other examples in image processing, web data ranking, and bioinformatic data analysis.



**Fig. 2.2** Example low-rank structures. *Left*: pictures of a human face under different illuminations. *Middle*: a building facade. *Right*: a sequences of frames (video)

When observing samples of such low-rank structures, they are often contaminated with noise which can be gross and difficult to model (e.g. non-Gaussian sparse errors). For instance, Fig. 2.3 shows a building facade occluded by a flag and a flyer. The rows (or columns) containing the occluded pixels do not lie in the low-rank space. Therefore, it is desirable to recover the inherent low-rank structure using these noisy samples, and separate the noise, which also can often be of a specific interest.

In the previous chapter, we showed that noisy data points drawn from a linear subspace can be stacked in the columns of a matrix, and decomposed into a low-rank component and a sparse noise component by minimizing the nuclear norm and the  $\ell_1$  norm, respectively. Surprisingly, the Robust PCA problem can be solved under broad conditions via convex optimization techniques such as the Augmented Lagrange Multiplier (ALM) [76] and the Accelerated Proximal Gradient (APG) [77]. These approaches fall in the computational methods for finding the desired decompositions based on the Alternating Directions Method of Multipliers (ADMM), which was first introduced in the mid-1970s by Gabay and Mercier [43] and Glowinski and Marroco [45], and is the current method of choice for large-scale non-smooth convex optimization, as in [46, 53, 123, 146]. It is also possible to solve RPCA with other methods such as for example [1], where sufficient conditions can be obtained in order to find an optimal solution, recovering the low-rank and sparse components of the matrix. In this book, we focus only on the ALM method to solve our Low-Rank formulations because of its stability and convergence speed.



**Fig. 2.3** Example showing a building facade occluded by a flag and a flyer. The rank of the matrix containing this image is unnecessarily high due to the occlusion. Low-rank optimization makes it possible to detect the noise (the occlusion in this example), and thus recover the original low-rank structure (the image of the facade in this example) (The picture is taken from [88])

The general method of ALM solves a constrained problem of this form:

$$\min f(X) \text{ s.t. } h(X) = 0, \quad (2.1)$$

where  $X \in \mathbb{R}^{M \times N}$ ,  $f : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}$ , and  $h : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{M \times N}$ . The augmented lagrange function for Eq. (2.1) is defined as

$$L(X, Y, \mu) = f(X) + \langle Y, h(X) \rangle + \frac{\mu}{2} \|h(X)\|_F^2, \quad (2.2)$$

where  $\|\cdot\|_F$  is the Frobenius norm which is equal to the square root of the sum of squared elements in the matrix,  $Y$  is a Lagrange multiplier matrix,  $\mu$  is a positive scalar, and  $\langle A, B \rangle$  denotes the matrix inner product ( $\text{trace}(A^T B)$ ). Under some

rather general conditions, when  $\{\mu_k\}$  is an increasing sequence and both  $f$  and  $h$  are continuously differentiable functions, the Lagrange multiplier converges to the optimal solution by following the algorithm outlined in Algorithm 1.

---

**Algorithm 1:** The general method of Augmented Lagrange Multiplier

---

```

while not converged do
    Solve  $X_{k+1} = \min_X L(X, Y_k, \mu_k)$ ;
     $Y_{k+1} = Y_k + \mu_k h(X_{k+1})$ ;
    Update  $\mu_k$  to  $\mu_{k+1}$ ;
end
Output  $X_k$ .

```

---

In the case of RPCA problem as in Eq. (1.7), the aforementioned ALM algorithm can be applied with  $X = (A, E)$ ,  $f(X) = \|A\|_* + \lambda \|E\|_1$ , and  $h(X) = D - A - E$ . Note that solving Eq. (1.7) using ALM involves solving a joint optimization for  $A$  and  $E$ , i.e.  $(A_{k+1}, E_{k+1}) = \min_{A,E} L(A, E, Y_k, \mu_k)$ . However, as shown in [76], solving for each of  $A$  and  $E$  separately is sufficient for them to converge to the optimal solution (in this case ALM is referred to as inexact ALM).

Ever since the basic formulation of Robust PCA was proposed in [20], where a low rank matrix was recovered from a small set of corrupted observations through convex programming, numerous applications in low-rank and sparse optimization-based image and video processing have followed. From these method, in this section, we briefly review the most related articles.

Low-rank optimization was employed in [60] for video de-noising, where serious mixed noise was extracted by grouping similar patches in both spatial and temporal domains, and solving a low-rank matrix completion problem. Additionally, in [101], linear rank optimization was employed to align faces with rigid transformations, and concurrently detect noise and occlusions. In [145], Yu and Schuurmans proposed an efficient solution to subspace clustering problems which involved the optimization of unitarily invariant norms. Another variant of such space-time optimization techniques is the total variation minimization, where for instance in [22], Chan et al. posed the problem of video restoration as a minimization of anisotropic total variation given in terms of  $\ell_1$ -norm, or isotropic variation given in terms of  $\ell_2$ -norm. Consequently, the Lagrange multiplier method was used to solve the optimization function. Moreover, the low-rank constraint has been vigorously employed in other computer vision problems such as tracking [44], feature fusion [141], face recognition [24], and saliency detection [115].

### 2.3 Turbulence Mitigation and Video Denoising

Conventional approaches for turbulence mitigation are mainly focused on registration-based techniques, where the deformation caused by the turbulence is overcome by aligning the frames to an undistorted template. In [117, 147, 148],



both the turbulence deformation parameters and a super-resolution image were recovered using B-Spline registration. Moreover, in [125], Tian and Narasimhan proposed recovering the large non-rigid turbulence distortions through a “pull-back” operation that utilizes several images with known deformations. Averaging-based techniques are also popular for video de-noising and turbulence mitigation, including pixel-wise mean/median, non-local means (averaging image patches which have similar appearance although not spatially close) [15, 81], fourier-based averaging [134], and speckle imaging [48, 49, 105]. Another category of methods for turbulence mitigation is the lucky region approach [21, 34, 63], where the least distorted patches of the video are selected based on several quality statistics, then those selected patches are fused together to compose the recovered video.

The earlier work in reconstructing an underwater sequence focused on finding the center of the distribution of patches in the video through clustering [33, 34], and manifold embedding [36], or employing the bispectrum (higher order correlation) to average the frames in the fourier domain [134]. The state of the art in this area, however, is the model-based tracking [124], where the characteristics of the water waves were employed to estimate the water basis using PCA. In such work, the optimal number and size of the basis remain vague; therefore, we argue that the estimated basis can be under-fitted or over-fitted depending on the selected basis. Other than requiring an orthographic camera and a given water height, the basis are additionally obtained by simulation using a single parameter differential equation for the waves, with the assumption that the surface fluctuations are small compared to the water’s height. Such a simple model with low parameter space is quite limited, and does not fully represent the actual scenario which can be much more complicated and dependant over several other factors such as the container size, external forces, and camera calibration; hence, the results from [124] are not quite satisfactory. Later in [125], it was proposed that the large non-rigid distortion caused by effects such as the water waves cannot be overcome through traditional B-spline registration, but rather by utilizing training images with predefined deformations. While the distorted video is typically the only available piece of information in such a problem, [125] assumes additional given training images, or a template from which we can generate samples; therefore, their work is considered out of the scope of comparison with the method discussed in this book.

In the context of blur kernel estimation which is used in the robust registration stage of the underwater scene reconstruction approach, the latest advances focused on deblurring a single image [64, 139], or a motion blurred video [2, 73]. However, our problem layout is different in that the underwater sequence is not blurry, but its mean is extremely blurry and noisy such that it cannot be deblurred. Thus, we are interested in rather blurring the frames in order to aid the registration. For those reasons, we discuss how to estimate a spatially varying blur kernel, which encodes the difference in blur between the mean and the frames by using the motion estimated at each iteration of the discussed algorithm. The robust registration approach we present in this book is not very sensitive to the frame blurring operation; therefore, in principal, other good blur estimation algorithms could be employed.

## 2.4 Moving Object Detection

Moving object detection is a widely investigated problem. When the scene is static, moving objects can be easily detected using frame differencing. A better approach would be to use the mean, the median, or the running average as the background [30]. The so-called eigenbackground [95] can also be obtained using PCA. However, when the scene is constantly changing because of noise, light changes, or camera shake, the intensities of image pixels can be considered as independent random variables, which can be represented using a statistical model such as a Gaussian, a mixture of Gaussians, or a kernel density estimator. The model can then be used to compute the probability for each pixel to belong to either the background or the foreground. Examples of such approaches include [37, 120, 149]. Additionally, the correlation between spatially proximal pixels could also be employed to improve the background modelling using a joint domain (location) and range (intensity) representation of image pixels such as in [113]. For comprehensive surveys on background subtraction techniques, the reader may refer to [29, 38, 52, 91, 102, 111].

## 2.5 Motion Trajectories and Activity Recognition

Motion trajectories have been employed in a variety of problems for human action representation and recognition [5, 47, 54, 61, 65, 93, 104, 142]. Many tracking-based methods are used or could be adapted for trajectory acquisition (for a comprehensive review on tracking techniques, the reader may refer to relevant surveys such as [59]). Usually, a single trajectory can be acquired by simple techniques such as temporal filtering [47]. The tracking entity typically is either a human body part (head, hand, foot, etc.) or a person as a whole. For the simultaneous tracking of multiple points, KLT [128] is a popular choice [54, 65, 99]. Statistical mixture models are also developed for multi-trajectory tracking [104]. Some specific tracking strategies (e.g. Significant Motion Point (SMP) [93]) are designed to handle complicated and subtle full-body motions. A common drawback among tracking-based methods is that it is difficult to obtain reliable trajectories for the reasons discussed in the previous section. In addition, several studies assume that the motion trajectories are already available [135], or they rely on manual annotations [142], or the so-called semi-automatic manner [5]. In contrast, the particle advection approach presented in book work is fully automatic and is very easy to implement.

Particle trajectories have been previously used to model crowded scenes in [136], where the flows normally occupy the whole frame, and the camera is static; thus, such dense trajectories could be directly employed. In contrast, the method discussed in this book adopts the particle trajectories for recognizing actions in videos acquired from a moving camera, which imposes several challenges since the actions usually only cover a small part of the frame, and more importantly,



the obtained trajectories combine both the camera motion and the object motion. Therefore, a novel approach is employed to detect the foreground trajectories and extract their object-induced component, which, in principal, requires estimating the background motion subspace. A large variety of subspace estimation methods exist in the literature such as PCA-based and RANSAC-based approaches. Such methods are, however, sensitive to noise which is considerably present in our scenario since a significant number of trajectories can be contaminated with the foreground motion. Fortunately, sparsity-based matrix decomposition methods such as [19, 79, 116] which have been primarily employed in image denoising domain, have shown that a robust estimation of an underlying subspace can be obtained by decomposing the observations into a low rank matrix and a sparse error matrix. Therefore, in this book, we show how Robust PCA [19] can be adopted to extract the object motion relevant to the action of interest.

The acquired motion trajectories can be represented by certain descriptors to identify the underlying spatio-temporal characteristics. Wu and Li [135] proposed a signature descriptor that can provide advantages in generalization, invariants, and compactness, etc. Ali et al. [5] showed that the features based on chaotic invariants for time series analysis perform very well in modelling semi-automatically obtained trajectories. Meanwhile, “trajecton” was proposed in a Bag-of-Words context [54] for trajectory-based action recognition. Messing et al. [99] investigated the temporal velocity histories of trajectories as a more representative feature for recognizing actions. The approach discussed in this book employs the particle trajectories and chooses the chaotic invariants [5] as a trajectory descriptor. It should be noted that the algorithms in [136] was used for computing the chaotic features as they have been shown more robust than [5].

Aside from trajectory features, a variety of feature representations have been developed for action recognition such as appearance features [31], shape-based representation [7], volumetric features (e.g. Poisson equation-based features [112], 3D Haar feature [121]), spatiotemporal interest points [28, 69, 100], motion history image (MHI) [13], and kinematic features [4].

The action categories typically handled by these approaches are short and simple clips of activities such as waving, jumping, running, clapping, etc. In contrast, actions which are long and complex (referred to as events) pose inherently different set of challenges and thus handled by different approaches, which we discuss in the coming section. Examples of such complex events are birthday party, making a fire, riding a bike, etc.

## 2.6 Complex Event Recognition

Compared to the traditional action recognition, complex event recognition is more challenging, mainly because the complex events have significantly longer lengths and diverse contents. Early methods for complex event recognition used low-level features such as Dollar [32], SIFT [86], MBH [132], and MFCC [68], and showed

promising results as in [12, 83, 110, 131]. Additionally, pooling of these low-level features was proposed in [74], where features such as SIFT and color were fused in order to improve the complex event recognition. Moreover, [122] used seven different low-level features in a Bag-of-Words (BoW) framework. However, these approaches reveal limitations in representing semantic concepts because they seek high-level class labels using only low-level features.

On the other hand, concept-based complex event recognition [58, 82, 85, 140] has recently flourished and shown promising results in high-level semantic representation of videos with complicated contents. This approach is particularly appealing for the purposes of retrieval and filtering of consumer media. The semantic concepts inherently represent the building blocks of the complex events; therefore, they naturally fit the complex event recognition task. For instance, in [85] and [82], a large dataset is collected to train concept detectors. However, their concepts are not suitable for complex videos as they have been recorded in well constrained conditions. Additionally, Loui et al. in [85] collected a benchmark dataset containing 25 concepts; however, the concepts are based on static images, not videos. On the other hand, concepts have been also employed in other computer vision problems such as image ranking and retrieval [119], and object classification [40]. In that, the concepts were used in the form of attributes [40], which can be considered as concepts with small granularity [58].

The most recent works on complex event recognition are [140] and [58]. The former utilized 62 action concepts as well as low-level features in a latent SVM framework, while the latter used an unsupervised approach (deep learning [70]) to find the data driven concepts in order to describe the high level semantics of the complex videos. Data driven concepts showed promising performance; however, they do not provide any conceptual description of the video.

## 2.7 Summary

The review of existing literature provided a summary of the prominent approaches for low-rank optimization in general, and the other relevant problems of underwater scene reconstruction, turbulence mitigation, background subtraction, and activity recognition. The method we present in this book for underwater scene reconstruction is generic, simple, and robust, which in contrast to the pervious methods, does not require a known template such as [125], the camera's height [124], or a special illumination [71]. The method is similar to the previous state the art [124] in that it works on a short sequence (61 frames) rather than 800 in [36] and 120 in [134], but more importantly, superior to [124] in performance and processing time.

Furthermore, previous work in moving object detection in dynamic scenes mostly focused on detecting the objects and did not consider recovering the background. Conversely, previous work in turbulence mitigation did not deal with detecting moving objects in the scene. In this book, we pose the two problems of moving object detection and turbulence mitigation as one application for the

three-term low-rank decomposition approach, which we discuss in detail in Chap. 4. We demonstrate how to decompose a turbulent video into separate background, foreground, and turbulence components. It is important to note that this method is not directly comparable to background subtraction or turbulence mitigation approaches because it focuses on optimizing the two problems jointly; though, the method still delivers competitive results on each task separately.

Moreover, a common drawback among pervious tracking-based activity recognition methods is that it is difficult to obtain reliable trajectories for the reasons discussed earlier. In addition, several studies assume that the motion trajectories are already available [135], or they rely on manual annotations [142], or the so-called semi-automatic manner [5]. In contrast, the particle advection presented in this book is fully automatic. Additionally, in order to handle the moving camera, previous methods typically require motion compensation, moving object detection and object tracking. The errors from these daunting steps propagate, which further complicates the activity recognition task. In contrast, the method we discuss here does not follow these steps, and accordingly avoids the aforementioned difficulties through the use of low-rank optimization, where we decompose the trajectories into their camera-induced and object-induced components in one convex optimization step, which can be efficiently solved.

In the complex event recognition domain, one of the closest approaches to the method we present in this book is [141], where the low-rank constraint was enforced on the detection scores for the purpose of late fusion of different training models. In contrast to the method from [141], the method we discuss here estimates the low-rank subspace of concept scores, which, to the best of our knowledge, has never been used before. More importantly, not only a low-rank concept representation is obtained, but also the user annotation is incorporated in order to encourage the low-rank estimation to follow a semantic pattern.

Robust Subspace Estimation Using Low-Rank  
Optimization

Theory and Applications

Oreifej, O.; Shah, M.

2014, VI, 114 p. 41 illus., 39 illus. in color., Hardcover

ISBN: 978-3-319-04183-4