M.S. THESIS

Fast Low-Rank Matrix Approximations in l_1-Norm Using an Alternating Projected Gradient Method

교차로 투영되는 Gradient 방법을 통한 l_1-Norm 기반의
빠른 낮은 차수 행렬 근사

BY

EUNWOO KIM

FEBRUARY 2013

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Fast Low-Rank Matrix Approximations in l_1-Norm Using an Alternating Projected Gradient Method

교차로 투영되는 Gradient 방법을 통한 l_1-Norm 기반의
빠른 낮은 차수 행렬 근사

BY

EUNWOO KIM

FEBRUARY 2013

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Fast Low-Rank Matrix Approximations in l_1-Norm Using an Alternating Projected Gradient Method

교차로 투영되는 Gradient 방법을 통한 l_1-Norm 기반의 빠른 낮은 차수 행렬 근사

지도교수 최 종 호

이 논문을 공학석사 학위논문으로 제출함

2012 년 11 월

서울대학교 대학원

전기 컴퓨터 공학부

김 은 우

김은우의 공학석사 학위논문을 인준함

2012 년 12 월

위 원 장 :	최 진 영	(인)
부위원장 :	최 종 호	(인)
위 원 :	오 성 회	(인)

Abstract

A low-rank matrix approximation plays an important role in the area of computer vision and image processing. Most of the conventional low-rank matrix approximation methods are based on l_2-norm (Frobenius norm), and the principal component analysis (PCA) is the most popular among them. However, this can give a poor approximation for the data contaminated by outliers (including missing data) because this exaggerates the negative effect of the outliers. Recently, in order to overcome this problem, various methods based on the l_1-norm have been proposed. Despite the robustness of the l_1-norm-based methods, these require heavy computational effort. In this paper, we propose a robust and fast low-rank factorization method based on the l_1-norm, which finds proper projection and coefficient matrices using an alternating projected gradient method. This gives a good approximation as the other l_1-based methods, but its execution time is much faster. The proposed method is applied to several problems to demonstrate its fast computational time in matrix approximation.

Keywords: Low-rank matrix approximations, Subspace analysis, l_1-Norm, Alternating projected gradient, QR decomposition, Structure and motion

Student Number: 2011-20811

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Low-rank matrix approximations such as vector and tensor based decomposition has attracted much attention in the area of subspace computation, data reconstruction, and dimensionality reduction. Since real-world data is usually high dimensional, we need to reduce the data to a smaller dimension for fast computation without degrading performance. Usually data can be well represented by a smaller number of parameters, and reducing the data dimension does not only reduce the computing time but also removes unwanted noise components. One of the most popular methods addressing these issues is the principal component analysis (PCA) [1].

PCA transforms data to a low-dimensional subspace that maximizes the variance of a given data based on Euclidean distance (l_2-norm). Although the conventional l_2-norm-based PCA (l_2-PCA) has been utilized in many problems, it is sensitive to outliers and missing data because the l_2-norm sometimes can amplify the negative effects of these data. This makes recognition or computer vision systems fail to achieve good performance.

As an alternative to solve these problems, l_1-norm-based analysis has been proposed, which is more robust to outliers and missing data [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. These techniques assume a Laplacian noise model instead of a Gaussian noise model.

Hwakins et al. [2] proposed a robust version of singular value decomposition (SVD) by applying an alternating l_1-regression algorithm and a weighted median technique to overcome outliers and missing data. Although this method is robust to such data, it takes a long time to find a local minimum by the alternating minimization technique. Ke and Kanade [3, 5] presented convex programming and weighted median approaches based on an alternating minimization for the l_1-norm-based cost function effectively, but these methods are computationally expensive.

Ding et al. [4] proposed r_1-PCA, which combines the merits of l_1-norm and l_2-norm. Unlike the conventional l_1-ALP, l_1-AQP [3, 5], it can suppress the negative effect of outliers, and its execution time is shorter. However, this method is less robust compared to other l_1-norm-based approaches. Kwak [6] suggested a PCA method based on l_1-norm maximization (PCA-l_1) to find successive projections, using a greedy approach to maximize the l_1-norm of the projected data in a feature space. These two methods, r_1-PCA and PCA-l_1, adopt cost functions, which is different from the cost function based on l_1-norm in [3], [5], and this results in the degradation of robustness to outliers.

Brooks et al. [7] presented a method for finding successive, orthogonal l_1-norm-based hyperplanes to minimize the l_1-norm cost function. This approach finds a global optimal solution when the dimension is reduced only by one, but it is no longer optimal when the reduction is more than one, and it takes a long time to find the solution by the linear programming.

Additionally, Eriksson and Hengel [8, 10] proposed a weighted low-rank matrix ap-

proximation using the l_1-norm in the presence of missing data. They generalized the Wiberg algorithm [12] using the l_1-norm and consider projection and coefficient simultaneously without using an alternating minimization procedure. However, it finds the solution by the convex programming, which is computationally intensive.

In this paper, we propose an alternating projected gradient algorithm that solves the l_1-based factorization problem which can handle outliers and missing data at a significantly reduced computational cost. Even though based on an alternating minimization method, this algorithm gives a fast convergence rate owing to the novel method of finding the update direction. This method is originated from the fact that there are numerous projection and coefficient matrices that gives the same multiplication result, but the convergence speed greatly depends on how to choose the projection and coefficient matrices. The update direction is determined so that the directional derivative with respect to the updated low-rank approximation is minimized. After finding the update direction, we use the weighted median algorithm to find a step size. However, unlike the other method [3] that applies the weighted median algorithm columnwise, we apply it to the whole matrix in order to reduce computational burden. The proposed method is about hundreds to thousands times faster compared to other l_1-norm-based methods in solving the problems in Chapter 4.

We demonstrate the competitiveness of the proposed method in terms of performance and computational speed for the examples such as face reconstruction from occluded or noisy images. In addition, the proposed method is applied to the non-rigid structure from motion problem [13] using a well-known multi-view benchmark data set.

This thesis is organized as follows. In the next Chapter, we briefly review a method of low-rank matrix approximations based on l_1-norm and discuss the drawbacks of these methods. In Chapter 3, we propose a fast low-rank matrix approximation algorithm

based on l_1-norm. In Chapter 4, we present various experimental results to evaluate the proposed methods with respect to other well-known subspace analysis methods. Finally, we present our conclusion in Chapter 5.

Chapter 2

Preliminaries on Low-Rank Matrix Approximations in l_1-Norm

In this Chapter, we briefly review the low-rank matrix approximation methods based on l_1-norm.

A low-rank matrix approximation in l_2-norm is one of the most popular methods in computer vision and pattern recognition area for preprocessing data. This is known as the principal component analysis (PCA), subspace learning, matrix factorization, etc. However, this method can be sensitive to outliers and missing data because the cost function based on l_2-norm magnify the influence of these. Therefore, l_2-based low-rank approximations may find projections that is far from optimal. Unlike the l_2-norm, the l_1-norm is more robust to outliers and missing data in statistical estimation [2], [3], [4], [5], [6], [7], [8], [9], [10]. A minimization problem based on the l_1-norm can be regarded as a maximum likelihood estimation problem under the Laplacian distribution [3].

We first consider an approximation problem for vector $\mathbf{y} = (y_1, y_2, ..., y_d)^T$ by a multi-

plication of vector $\boldsymbol{x} \in R^d$ and scalar α

$$\boldsymbol{y} = \alpha \boldsymbol{x} + \boldsymbol{\delta}, \tag{2.1}$$

where $\boldsymbol{\delta}$ is a noise vector whose elements has an independently and identically distributed (i.i.d.) Laplacian distribution [3]. The probability model for (2.1) can be written as

$$p(\boldsymbol{y}|\boldsymbol{x}) \sim exp\{-\frac{||\boldsymbol{y} - \alpha \boldsymbol{x}||_1}{s}\}, \tag{2.2}$$

where $|| \cdot ||_1$ denotes the l_1-norm, and $s(> 0)$ is a scaling constant [5]. We assume for a moment that $\boldsymbol{x}$ is given. In this case, to maximize the log likelihood of the observed data is equivalent to minimize the following error function :

$$E(\alpha) = \min_{\alpha} ||\boldsymbol{y} - \alpha \boldsymbol{x}||_1. \tag{2.3}$$

Then, the l_1-norm-based minimization problem can be written as the following [3] :

$$\begin{aligned} E(\alpha) &= \min_{\alpha} \sum_i |\boldsymbol{y_i} - \alpha \boldsymbol{x_i}| \\ &= \min_{\alpha} \sum_i |\boldsymbol{x_i}| |\frac{\boldsymbol{y_i}}{\boldsymbol{x_i}} - \alpha|. \end{aligned} \tag{2.4}$$

The global optimal point for (2.4) can be solved by the weighted median technique for set $\{\frac{\boldsymbol{y_i}}{\boldsymbol{x_i}} | i = 1, ..., K\}$ with the i-th weight $|\boldsymbol{x_i}|$ [14], [15], [16], [3], [17]. If $\boldsymbol{x_i} = 0$, then the corresponding i-th term is ignored because it has no effect in finding the solution.

The representation problem of (2.1) can be generalized to the problem of matrix approx-

imation. Now, we consider the l_1 approximation of matrix Y such that

$$\min_{P,X} ||Y - PX||_1, \tag{2.5}$$

where Y is an observation matrix and P and X are projection and coefficient matrices. Here, $||Y||_1 = \sum_{ij} |Y_{ij}|$ and Y_{ij} is the (i, j)-th element of Y. In general, (2.5) is a non-convex problem, because both P and X are unknown variables. Ke and Kanade proposed two ways of solving this problem, one by applying the alternating convex minimization and the other by applying alternating weighted median approaches [3], [5]. If one of the two variables is known, we can easily find the optimal solution by using these approaches. However, the proposed weighted median method computes the subspace bases one by one, and therefore is potentially easier to be trapped into a bad local minimum [3]. They preferred the convex programming methods, which were more efficient than the weighted median method. By minimizing the cost function over one variable while the other is fixed and then the roles of the variables are exchanged alternately, the optimization process can be performed efficiently. Such minimization technique based on alternating iteration has been widely used in subspace analysis [2], [3], [5] [18], and can be written as

$$\begin{aligned} P^{(t)} &= \arg\min_{P} ||Y - PX^{(t-1)}||_1 \\ X^{(t)} &= \arg\min_{X} ||Y - P^{(t)}X||_1, \end{aligned} \tag{2.6}$$

where the superscript t denotes the alternating iteration number.

However, these alternating approaches are computationally very expensive and requires a large memory when the dimension of matrices is large. In the next Chapter, we propose a fast method that finds the solution for (2.5) at a much reduced computa-

tional cost.

Chapter 3

Fast Low-Rank Matrix Approximations in l_1-Norm Using an Alternating Projected Gradient Method

As previously mentioned, it requires too much time and memory to solve an l_1-norm-based problem by a linear or quadratic programming to use for practical problems. To overcome this, we propose a low-rank matrix factorization using l_1-norm by an alternating projected gradient update.

3.1 Gradient update

We first describe the problem of low-rank matrix approximation in l_1-norm by gradient update. The cost function for the matrix approximation can be written as :

$$\min_{P,X} J(P,X) = \min_{P,X} ||Y - PX||_1, \tag{3.1}$$

where $Y \in R^{m\times n}$, $P \in R^{m\times r}$, and $X \in R^{r\times n}$ are the observation, projection, and coefficient matrices, respectively.

When differentiating $|x|$, we consider $|x|$ as the limit of $\sqrt{x^2+\varepsilon^2}$, i.e.,

$$|x| = \lim_{\varepsilon\to 0} \sqrt{x^2+\varepsilon^2}. \tag{3.2}$$

This gives

$$\begin{aligned}\frac{\partial |x|}{\partial x} &= \lim_{\varepsilon\to 0} \frac{\partial \sqrt{x^2+\varepsilon^2}}{\partial x} = \lim_{\varepsilon\to 0} \frac{x}{\sqrt{x^2+\varepsilon^2}} \\ &= \operatorname{sgn}(x),\end{aligned}$$

where $\operatorname{sgn}(x)$ is the signum function of x.

In this way, we can differentiate (3.1) with respect to (w.r.t.) X and its derivative is

$$\nabla_X J(P,X) = -P^T \operatorname{sgn}(Y - PX). \tag{3.3}$$

Here, $\mathrm{sgn}(Y)$ for matrix Y represents a matrix whose (i, j)-th element is $\mathrm{sgn}(Y_{ij})$. Now, we consider the update of X by the steepest gradient update method.

$$\begin{aligned} J(\alpha|P, X) &= \min_{\alpha} ||Y - P(X + \alpha(-\nabla_X J(P, X)))||_1 \\ &= \min_{\alpha} ||Y' - \alpha(-P\nabla_X J(P, X))||_1 \\ &= \min_{\alpha} ||Y' - \alpha A||_1 \\ &= \min_{\alpha} \sum_{i,j} |A_{ij}| |\frac{Y'_{ij}}{A_{ij}} - \alpha|, \end{aligned} \tag{3.4}$$

where $Y' = Y - PX$ and $A = -P\nabla_X J(P, X)$. Note that in our algorithm, we apply the weighted median algorithm update for either P or X at a time, unlike Ke and Kanade [3] which applied the weighted median algorithm columnwise, to reduce the total computation time.

Now, we apply the weighted median method [14], [15], [16], [3], [17] to the ratio Y'_{ij}/A_{ij} with weight $|A_{ij}|$ to get the step size α that minimizes the cost function (3.4).

Finally, Y' and X are updated as

$$\begin{aligned} Y' &\leftarrow Y' - \alpha(-P\nabla_X J(P, X)), \\ X &\leftarrow X + \alpha(-\nabla_X J(P, X)). \end{aligned} \tag{3.5}$$

For P, we can also differentiate (3.1) w.r.t. P in the same manner and denote the derivative by $\nabla_P J(P, X)$, then

$$\nabla_P J(P, X) = -\,\mathrm{sgn}(Y - PX)X^T, \tag{3.6}$$

and Y' and P are

$$\begin{aligned} Y' &\leftarrow Y' - \alpha(-\nabla_P J(P, X)X), \\ P &\leftarrow P + \alpha(-\nabla_P J(P, X)). \end{aligned} \tag{3.7}$$

The projection and coefficient matrices P and X are updated alternatingly until convergence is achieved.

We reexamine the minimization problem (3.1), because there are numerous pairs of P and X that give the same result of PX. If $P' = PH^{-1}$ and $X' = HX$ for some nonsingular matrix H, then

$$\begin{aligned} J(X'|P') &= \min_{X'} ||Y - P'X'||_1 \\ &= \min_{X} ||Y - PX||_1. \end{aligned} \tag{3.8}$$

The cost function in choosing the optimal step size for this problem can be written as

$$J(\beta|P', X') = \min_{\beta} ||Y' - \beta P'(-\nabla_{X'} J(P', X'))||_1, \tag{3.9}$$

where $\nabla_{X'} J(P', X') = -P'^T \operatorname{sgn}(Y - P'X')$ and β is a step size. When H is orthogonal and $\alpha = \beta$, (3.4) and (3.9) are the same because of the relation $P'P'^T = PH^{-1}H^{-1^T}P^T = PP^T$. If it is not the case, then the gradient of (3.9) changes depending on H, i.e.,

$$\min_{\alpha} ||Y' - \alpha(-P\nabla_X J(P, X))||_1 \neq \min_{\beta} ||Y' - \beta(-P'\nabla_{X'} J(P', X'))||_1. \tag{3.10}$$

This means that the direction of gradient depends on the choice of P. Therefore, it is important to find P such that we can get a gradient which provides fast rate of convergence

in minimizing (3.4).

3.2 Finding an optimal direction using the alternating gradient update

In the previous section, we have shown that the direction of gradient depends on the representation of P, which can influence the convergence rate to solution. To resolve this problem, we describe how to find an optimal direction for fast convergence. We reformulate (3.1) as :

$$\begin{aligned} \min_{\Delta X'} J(\Delta X'|P) = \min_{\Delta X'} ||Y' - P\Delta X'||_1, \\ s.t. \quad ||P\Delta X'||_F^2 = \epsilon^2, \end{aligned} \tag{3.11}$$

where $P\Delta X'$, denotes a small perturbation of Y' and ϵ is a small positive value. We want to find the direction that minimize the directional derivative of the cost function with respect to $P\Delta X'$. By introducing a Lagrange multiplier, the cost function becomes

$$||Y' - P\Delta X'||_1 + \lambda(\mathrm{tr}(\Delta X'^T P^T P \Delta X') - \epsilon^2), \tag{3.12}$$

where tr is the trace operator ($||A||_F^2 = \mathrm{tr}(A^T A)$). Differentiating (3.12) w.r.t. $\Delta X'$ and equating it to zero, we obtain

$$-\mathrm{sgn}(Y' - P\Delta X')^T P + \lambda \Delta X'^T P^T P = 0,$$

which gives

$$\begin{aligned}\Delta X' &= \frac{1}{\lambda}(P^T P)^{-1} P^T \operatorname{sgn}(Y' - P\Delta X') \\ &= \frac{1}{\lambda} P^+ \operatorname{sgn}(Y' - P\Delta X'),\end{aligned} \tag{3.13}$$

where $P^+ = (P^T P)^{-1} P^T$ is the pseudo inverse of P.

By applying (3.13) to $||P\Delta X'||_F^2 = \epsilon^2$, we get

$$\operatorname{tr}(\frac{1}{\lambda^2} \operatorname{sgn}(Y' - P\Delta X')^T P^{+^T} P^T P P^+ \operatorname{sgn}(Y' - P\Delta X')) = \epsilon^2.$$

From this, we obtain

$$\frac{1}{\lambda} = \frac{\epsilon}{\sqrt{g}}, \tag{3.14}$$

where $g = \operatorname{tr}(\operatorname{sgn}(Y' - P\Delta X')^T P^{+^T} P^T P P^+ \operatorname{sgn}(Y' - P\Delta X'))$. From (3.13) and (3.14), we get

$$\Delta X' = \frac{P^+ \operatorname{sgn}(Y' - P\Delta X')}{\sqrt{g}} \cdot \epsilon. \tag{3.15}$$

Because we are more interested in the relative values of the elements of $\Delta X'$, we set $\frac{1}{\sqrt{g}} = 1$. Since $\Delta X'$ linearly depends on ϵ, we find the optimal gradient direction $\Delta X'$ of the object function (3.11) for an infinitesimal ϵ as

$$\Delta X = \lim_{\epsilon \to 0} \frac{\Delta X'}{\epsilon} = \lim_{\epsilon \to 0} \frac{P^+ \operatorname{sgn}(Y' - P\Delta X') \cdot \epsilon}{\epsilon}, \tag{3.16}$$

ϵ is canceled in the above equation but $\Delta X'$ remains inside the signum function. Note that $\lim_{\epsilon \to 0} \Delta X' = 0$, we obtain

$$\Delta X = P^+ \operatorname{sgn}(Y'). \tag{3.17}$$

This result can be obtained by an orthogonal projection of $\frac{\partial ||Y'-P\Delta X'||_1}{\partial P\Delta X'}\Big|_{||P\Delta X'||_F \to 0} = \operatorname{sgn}(Y')$ into the subspace of $P\Delta X'$, i.e.,

$$PP^+ \operatorname{sgn}(Y') = P\Delta X \ = \lim_{\epsilon \to 0} \frac{P\Delta X'}{\epsilon}. \tag{3.18}$$

With the direction $P\Delta X$, we reformulated the problem (3.4) as the problem of finding the optimal step size α in the following ;

$$\min_{\alpha} ||Y' - \alpha P\Delta X||_1 = \min_{\alpha} ||Y' - \alpha PP^+ \operatorname{sgn}(Y')||_1. \tag{3.19}$$

In calculating PP^+, it is more efficient to calculate QQ^T $(= PP^+)$ where Q is obtained by the QR decomposition of P, i.e., $P = QR$.

There are two important insights for this updating rule. First, $P\Delta X = PP^+ \operatorname{sgn}(Y')$ is the same as the orthogonal projection of $\frac{\partial ||Y'-P\Delta X'||_1}{\partial P\Delta X'}\Big|_{||P\Delta X'||_F \to 0} = \operatorname{sgn}(Y')$ into the subspace of $\{P\Delta X' | \Delta X' \in R^{r\times n}\}$. This is why we call the proposed method as the alternating projected gradient method. Second, this new update direction is analogous to the Gauss-Newton update direction in least-squares. The Gauss-Newton direction of $||F(x)||^2$ is given as $-\nabla_x F(x)^+ F(x)$. If we regard $F(x)$ in this expression as $\frac{\partial ||F(x)||^2}{\partial F(x)}$ ignoring scale, then it is amazingly similar to the expression of $\Delta X = P^+ \operatorname{sgn}(Y')$. Hence, we may consider this new update direction as an extension of the Gauss-Newton method to l_1-norm and expect it to be better than the normal gradient direction.

As in (3.17), we can obtain ΔP in the same manner under the constraint ($||\Delta P' X||_F^2 = \epsilon^2$) as

$$\Delta P = \operatorname{sgn}(Y') X^+, \tag{3.20}$$

and the problem of finding the optimal step size is

$$\min_{\alpha'} ||Y' - \alpha' \Delta P X||_1 = \min_{\alpha'} ||Y' - \alpha' \operatorname{sgn}(Y') X^+ X||_1. \tag{3.21}$$

In this case, we use the $Q^T Q$ for $X^+ X$ in computing $\Delta P X$.

3.3 Fast low-rank matrix approximation in l_1-norm using an alternating projected gradient update

Here, we summarize the proposed algorithm. First, we update P while X is fixed in (3.1). The direction of gradient depends on X. Since X is not orthogonal, we apply the QR decomposition to X :

$$\begin{aligned} X^T &= X'^T R, \\ P R^T X' &= P' X', \end{aligned} \tag{3.22}$$

where orthogonal matrix X'^T and upper triangular matrix R are obtained from the QR decomposition and $PR^T = P'$. Then, we can compute ΔP by using X' and find the optimal step size using the weighted median algorithm.

Once the update of P has terminated, we update X with P fixed. Again, we apply the QR decomposition to P for the optimal direction of ΔX. The update rule is similar to that of P update. Then, we continue to update P and X alternatingly, and the overall procedure is described in Algorithm 1.

Numerical errors can be generated in the implementation process, and to deal with this

we modify the signum function as :

$$\mathrm{sgn}'(x) = \begin{cases} 1 & x \geq \gamma, \\ 0 & -\gamma < x < \gamma, \\ -1 & x \leq -\gamma, \end{cases}$$

where γ is a threshold which is a small positive value. Through this, we can find a better solution despite the difficulties that may arise due to numerical errors.

In the proposed algorithm, the cost function decreases monotonically in the process of alternating minimization. The alternating minimization procedure is performed until the difference of the previous and current residual Y is small enough :

$$|Y^{(t-1)}| - |Y^{(t)}| < \theta, \tag{3.23}$$

where θ is a small positive number.

In the proposed method, the step size α can be determined by using the weighted median algorithm. In the weighted median algorithm, we may use the divide and conquer algorithm like the quick-select [19], [17], which can find the solution in linear time on average. However, in practice, it is faster to use built-in sorting functions in MATLAB when the number of elements is not large. Moreover, since we are applying the weighted median algorithm to find the step size which does not need to accurate, it is better to calculate the weighted median of randomly selected samples. To see how the weighted median depends on the number of samples, we consider a problem of finding an approximate weighted median from a set consisting of an infinite number of elements. We assume that the weight of each element is the same to make the problem simple.

Algorithm 1 Fast l_1-norm-based Matrix Approximation using an Alternating Projected Gradient method (l_1-APG)

Input : $Y \in \mathbb{R}^{d \times n}$, the subspace dimension k
Output : $P \in \mathbb{R}^{d \times k}, X \in \mathbb{R}^{k \times n}$
Initialize P and X randomly
while residual Y does not converge **do**
  # P update (Fix X, compute P)
  Compute $X'^T R' \leftarrow X^T$ by QR decomposition
  Set $P' \leftarrow P R'^T$
  Compute $\Delta P' \leftarrow \mathrm{sgn}'(Y - P'X')X'^T$
  Compute α using weighted median to minimize the following
    $\min_\alpha |Y - (P' + \alpha \Delta P')X'|$
  Update P' and residual Y' as
    $P' \leftarrow P' + \alpha \Delta P'$
    $Y' \leftarrow Y' - \alpha \Delta P' X'$
  # X update (Fix P', compute X')
  Compute $P'' R'' \leftarrow P'$ by QR decomposition
  Set $X'' \leftarrow R'' X'$, $P \leftarrow P''$
  Compute $\Delta X'' \leftarrow P^T \mathrm{sgn}'(Y - PX'')$
  Compute α using weighted median to minimize the following
    $\min_\alpha |Y' - P(X'' + \alpha \Delta X'')|$
  Update X and residual Y as
    $X \leftarrow X'' + \alpha \Delta X''$
    $Y \leftarrow Y'' - \alpha P \Delta X''$
end while

Then the cumulative probability that the sample median of 2m+1 samples is less than the $100q\%$ quantile of original elements is equal to the cumulative probability that the success is no more than m for a binomial distribution $B(2m+1, 1-q)$. Since the CDF of a binomial distribution can be represented in terms of the regularized incomplete beta function, the result is given as

$$\begin{aligned} F(q; 2m+1) &= P(Z \leq m) \\ &= I_q(m+1, m+1), \end{aligned} \tag{3.24}$$

where Z is the binomial random variable and I_q is the incomplete beta function. This expression can be calculated numerically, and we have found that

$$F(1/2 + 0.005; 10^5 + 1) - F(1/2 - 0.005; 10^5 + 1) \approx 0.998.$$

This means that if we use 10^5 samples, then the sample median resides within the $\pm 0.5\%$ range of the true median with probability 0.998. Even if this result applies for the case of finding the median, the result is also meaningful for the weighted median if the weights are moderately distributed. In the examples in Chapter 4, we selected 10^5 samples if the number of elements is greater than 10^5, and then applied a built-in sorting function in MATLAB to find the weighted median.

3.4 Fast weighted low-rank matrix approximation in l_1-norm at the presence of missing data

In real applications, there are not only outliers but also missing data in dealing with the problems of image processing and computer vision. Because missing data can have a negative effect on vision and recognition systems, we consider the problem of low-rank

matrix approximations using the l_1-norm at the presence of missing data.

The problem can be formulated as

$$\min_{P,X} J(P, X|W) = \min_{P,X} ||W \odot (Y - PX)||_1, \tag{3.25}$$

where $\odot$ is the component-wise multiplication or Hadamard product. Here, $W \in R^{m \times n}$ is a weight matrix, whose element W_{ij} is 1 if y_{ij} is known, and is 0 if y_{ij} is unknown. Similar to the problem (3.11), we can formulate the weighted low-rank matrix factorization in l_1-norm under the constraint ($||P\Delta X'||_F^2 = \epsilon^2$) as

$$\begin{aligned} \min_{\Delta X'} J(\Delta X'|P, W) = \min_{\Delta X'} ||(W \odot (Y' - P\Delta X'))||_1, \\ s.t.\ \ ||P\Delta X'||_F^2 = \epsilon^2. \end{aligned} \tag{3.26}$$

By introducing a Lagrange multiplier, the cost function becomes

$$||W \odot (Y' - P\Delta X')||_1 + \lambda(||P\Delta X'||_F^2 - \epsilon^2), \tag{3.27}$$

and reformulating (3.27) into a vector form as

$$||\overline{W}\text{vec}(Y') - \overline{W}(I \otimes P)\text{vec}(\Delta X')||_1 + \lambda(||(I \otimes P)\text{vec}(\Delta X')||_F^2 - \epsilon^2), \tag{3.28}$$

where $\overline{W} = \text{diag}(\overline{\boldsymbol{w}}) \in R^{mn \times mn}$, $\overline{\boldsymbol{w}} = (\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, ..., \boldsymbol{w}_n^T)^T \in R^{mn \times 1}$, $\boldsymbol{w}_i$ is the i-th column vector of W, and I denotes an $m \times m$ identity matrix. Assume that P is fixed

for a moment. Differentiating (3.28) w.r.t. $\text{vec}(\Delta X')$ and equating it to zero, we obtain

$$\begin{aligned}
&-(I\otimes P)^T\overline{W}\,\text{sgn}(\overline{W}\text{vec}(Y')-\overline{W}(I\otimes P)\text{vec}(\Delta X'))+\lambda(I\otimes P)^T(I\otimes P)\text{vec}(\Delta X')\\
&=-(I\otimes P)^T\overline{W}\,\text{sgn}(\overline{W}\text{vec}(Y')-\overline{W}(I\otimes P)\text{vec}(\Delta X'))+\lambda(I\otimes P^TP)\text{vec}(\Delta X')\\
&=0.
\end{aligned}\tag{3.29}$$

Then, the solution for (3.29) can be found by the same procedure as in the previous section as the following :

$$\begin{aligned}
\text{vec}(\Delta X') &= \frac{1}{\lambda}(I\otimes P^TP)^{-1}(I\otimes P)^T\overline{W}\,\text{sgn}(\overline{W}\text{vec}(Y')-\overline{W}(I\otimes P)\text{vec}(\Delta X'))\\
&= \frac{1}{\lambda}(I\otimes P^+)\overline{W}\,\text{sgn}(\overline{W}\text{vec}(Y')-\overline{W}(I\otimes P)\text{vec}(\Delta X'))\\
&= \frac{1}{\lambda}(I\otimes P^+)\overline{W}\text{vec}(\text{sgn}(W\odot(Y'-P\Delta X'))),
\end{aligned}\tag{3.30}$$

where $P^+=(P^TP)^{-1}P^T$. In the derivation of this equation, we use the formula such as $(A\otimes B)^{-1}=A^{-1}\otimes B^{-1}$, $(A\otimes B)^T=A^T\otimes B^T$, and $\text{vec}(A\odot B)=\text{diag}(\overline{\boldsymbol{a}})\text{vec}(B)$, $\overline{\boldsymbol{a}}=(\boldsymbol{a}_1^T,\boldsymbol{a}_2^T,...,\boldsymbol{a}_n^T)^T$, and $\boldsymbol{a}_i$ is the i-th column vector of A. Because we are interested in the direction of $\text{vec}(\Delta X')$ rather than its scale, we set $-\frac{1}{\lambda}$ as 1.

From the relations $\lim_{\epsilon\to 0}\Delta X'=\Delta X$ and $\lim_{\epsilon\to 0}\Delta X'=0$ as in the previous section, and the relation $\text{vec}(A\odot B)=\text{diag}(\overline{\boldsymbol{a}})\text{vec}(B)$, the above equation becomes

$$\begin{aligned}
\text{vec}(\Delta X) &= (I\otimes P^+)\overline{W}\text{vec}(\text{sgn}(W\odot Y'))\\
&= (I\otimes P^+)\text{vec}(W\odot\text{sgn}(W\odot Y')).
\end{aligned}\tag{3.31}$$

Because the elements of W are either 0 or 1, (3.31) can be rewritten as

$$\begin{aligned} \text{vec}(\Delta X) &= (I \otimes P^{+})\text{vec}(\text{sgn}(W \odot Y')) \\ &= \text{vec}(P^{+}\,\text{sgn}(W \odot Y')), \end{aligned} \tag{3.32}$$

and this gives

$$\Delta X = P^{+}\,\text{sgn}(W \odot Y'). \tag{3.33}$$

Similar to (3.19), the cost function to find the step size α becomes

$$\begin{aligned} \min J(\alpha|P, \Delta X, W) &= \min_{\alpha} ||W \odot (Y' - \alpha P \Delta X)||_1 \\ &= \min_{\alpha} ||W \odot Y' - \alpha W \odot (PP^{+}\,\text{sgn}(W \odot Y'))||_1. \end{aligned} \tag{3.34}$$

Compared to (3.19), the only difference is the presence of W in the cost function.

When we vary P for a fixed X, we can obtain ΔP and the cost function to find the step size, similar to (3.33) and (3.21), respectively.

$$\Delta P = \text{sgn}(W \odot Y')X^{+}, \tag{3.35}$$

$$\begin{aligned} \min J(\alpha'|\Delta P, X, W) &= \min_{\alpha'} ||W \odot (Y' - \alpha' \Delta P X)||_1 \\ &= \min_{\alpha'} ||W \odot Y' - \alpha' W \odot (\text{sgn}(W \odot Y')X^{+}X)||_1. \end{aligned} \tag{3.36}$$

We can solve (3.34) and (3.36) using the l_1-minimization algorithm presented in the previous section.

Chapter 4

Experimental Results

We evaluated the performance of the proposed method (l_1-APG) by experimenting with various data, and compared with other methods (PCA-l_1 [6], r_1-PCA [4], l_1-PCA* [7], l_1-AQP [5], l_2-PCA [20]) in terms of reconstruction error and execution time. In the experiments, we used Huber's M-estimator [4] for r_1-PCA and the initial projection matrix of r_1-PCA and PCA-l_1 was set equal to the outcome of l_2-PCA. The initial projection and coefficient matrices of l_1-APG, l_1-AQP, and l_1-PCA* were set to zero and to random numbers selected from the Gaussian distribution with zero mean and variance of 1, respectively. In addition, l_1-APG used the fast version of weighted median method which has been described in Chapter 3. Each experiments were repeated for 10 times and the results were averaged.

We also performed experiments with missing data using the weighted version of the proposed method in Section 3.4, and the average performance of 100 experiments were compared to those of other methods that can handle the case of missing data (l_1-ALP [5], l_1-Wiberg [10]). To show the usefulness of the proposed algorithm, we also applied

l_1-APG to the structure and motion problem [13]. All the experiments were conducted using MATLAB.

4.1 Experiments with outliers

Firstly, we applied the proposed method to an example with outliers. We generated a 1200×30 matrix B whose elements are randomly selected from the Gaussian distribution with zero mean and variance of 1, and applied the QR decomposition to this matrix. We also generated a 30×100 matrix C where each element is randomly selected from the Gaussian distribution with zero mean and the variance that is decreased exponentially for each row from 5 to 0.05, i.e. the variance of the k-th row is $5 \times 0.01^{\frac{k-1}{r-1}}$. We generated a 1200×30 mean matrix $\overline{Y0}$ consisting of the same columns whose elements are selected randomly in the range of [0, 1]. From matrices B and C, and $\overline{Y0}$, $Y0$ was generated as $Y0 = BC + N + \overline{Y0}$, where N is a noise matrix of 1200×30, whose elements have Gaussian distribution with zero mean and variance of 0.01. Then, we constructed Y by replacing 10 percent of the elements for the 20 percent randomly selected samples in $Y0$ by outliers that was uniformly distributed in the range of [0, 1]. In the experiment, we extracted 30 projection vectors and we set the stopping condition (3.23) for each algorithm as 10^{-3}. We compare the proposed method to PCA-l_1, r_1-PCA, l_2-PCA, l_1-AQP, l_1-PCA* in terms of canonical correlation (CC) [21].

Figure 4.1 shows the canonical correlation between matrix B and estimated projection matrix P for various conventional PCA methods and the proposed method. As shown in this figure, l_1-APG and l_1-AQP have the largest canonical correlation compared to the other methods because these methods use the same l_1-norm-based cost function, which removes effect of outliers effectively. Table 4.1 shows the average canonical correla-

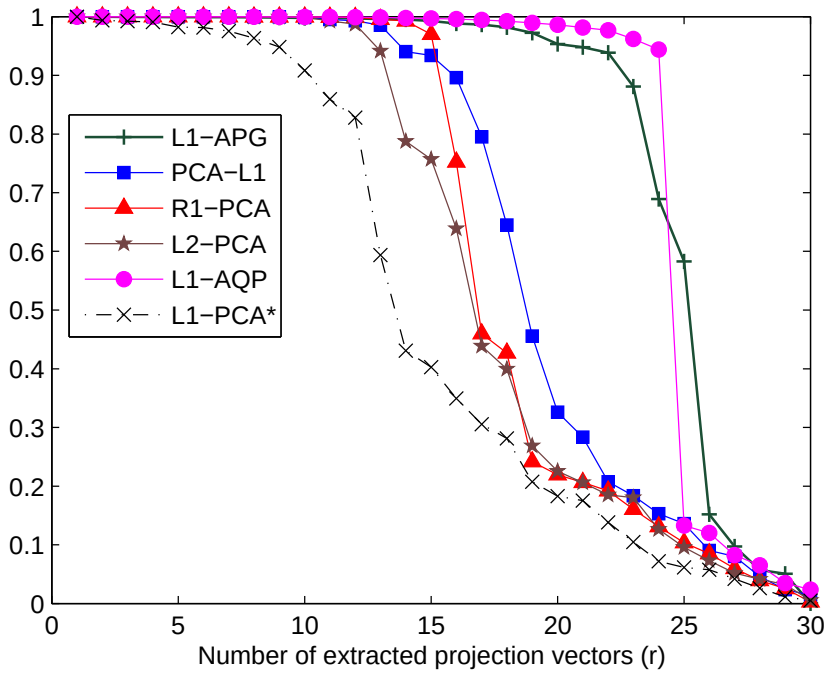


Figure 4.1 Canonical correlation in an example

tion, execution time, and number of iterations to solving this problem. Although l_1-AQP gives nearly the same mean canonical correlation as l_1-APG, it is computationally very expensive. The execution time is approximately 1000 times longer for l_1-AQP compared to l_1-APG, l_1-PCA*, l_2-PCA, and l_1-PCA*.

In addition, we experimented two examples to measure the reconstruction error and execution time. The problems were constructed in different sizes with the parameters of: (m=60, n=40, r=10) and (m=70, n=100, r=15). These examples were generated in the same procedure as the canonical correlation experiment. The results were averaged over 10 trials. Also, some gaussian noise was added to this data and 10 percent of the elements were set to outliers for 20 percent of randomly selected samples.

In the experiment, the average reconstruction error $E_2(r)$ was calculated as the follow-

Table 4.1 Mean CC, Execution time, and Number of iterations of an example with outliers

Method	Mean CC	Execution time (sec)	No. of iterations
l_1-APG	0.8093	0.362	10.92
PCA-l_1	0.6398	0.114	5.38
r_1-PCA	0.6025	0.029	6.71
l_2-PCA	0.5814	0.006	0
l_1-AQP	0.8099	1547.9	9.23
l_1-PCA*	0.4961	1019.3	0

ing [6] :

$$E_2(r) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{y_i^{org}} - P_r P_r^T \boldsymbol{y_i}||_2, \tag{4.1}$$

where n is the number of samples, P_r is r-projection vectors obtained from the training set, $\boldsymbol{y_i}$ and $\boldsymbol{y_i^{org}}$ are the i-th test sample with and without outliers and occlusion.

The average reconstruction error and execution time are shown in Table.4.2. l_1-APG is 3% - 10% larger in the reconstruction error than l_1-AQP and is 30% - 460% smaller than the rest. But, l_1-AQP took approximately 500 - 600 times longer than l_1-APG in solving these problems.

4.2 Face reconstruction

We applied various method in face reconstruction problems and compared their performances. In the experiments, 166 images from Multi-PIE face database [22] were used, which were resized to 100×120 pixels. The intensity of each pixel was normalized to have the value in the range of [0, 1]. Each 2-D image was converted to a 12000-dimensional vector. Some examples of the face database are shown in Fig. 4.2. To

Table 4.2 The average reconstruction error $E_2(r)$ and execution time of two examples with outliers.

	m=60, n=40, r=10		m=70, n=100, r=15	
Method	Error (E_2)	Time (sec)	Error (E_2)	Time (sec)
l_1-APG	1.2421	0.0229	1.1348	0.0549
PCA-l_1	1.9865	0.0081	1.8046	0.0133
r_1-PCA	1.8977	0.0046	1.6010	0.0056
l_2-PCA	2.2906	0.0006	2.0678	0.0011
l_1-AQP	1.2064	13.6101	1.0284	28.4881
l_1-PCA*	4.4244	23.5591	5.8978	1127.4

evaluate the reconstruction performance, we divided the image data into 90 training images and 76 test images. The average value of the test images using the l_1-norm is 6725.4.

To generate outliers, 30 percent of the images were randomly selected, and for those images, fifteen percent of the pixels in an image was replaced by outliers, which were randomly selected in [0, 1]. In another experiment, to generate occlusions, 30 percent of the images were randomly selected one more time, and each of those images was occluded by a randomly located rectangle, whose size varied in the range of 30×30 to 60×60 and each pixel of the rectangle has a random value randomly selected in [0, 1].

We first constructed the median image $\overline{Y}$, whose value at the location (i, j) was obtained by taking the median value of the pixels at the location (i, j) of training images. To reconstruct the face images, we subtracted the median image from each training image and then extracted the projection matrices from them using various methods including l_1-APG. We also subtracted the median image from a i-th test image Y_i^{test}, and reconstructed the test image using the projection matrix extracted from a method and then

Figure 4.2 Sample images in the Multi-PIE face database.

adding the median image as $Y_{Recon} = PP^T(Y_i^{test} - \overline{Y}) + \overline{Y}$. Figure 4.3 shows some examples of test face images with occlusions (first and third row) and outliers (second row) and its reconstructed faces with 30 extracted projection vectors. In the figure, the proposed method gives the most similar results to the original images and the occlusion block and outliers almost disappeared.

Figure 4.4(a) shows the average reconstruction errors $E_1(r)$ in the presence of occlusions for various number (r) of extracted projection vectors. In this experiment, the average reconstruction error $E_1(r)$ based on the l_1-norm was calculated as :

$$E_1(r) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{y_i^{org}} - P_r P_r^T \boldsymbol{y_i}||_1. \tag{4.2}$$

We could not apply l_1-AQP to this problem because it required too much memory space. In the figure, we can see that the reconstruction error of l_1-APG is the lowest compared

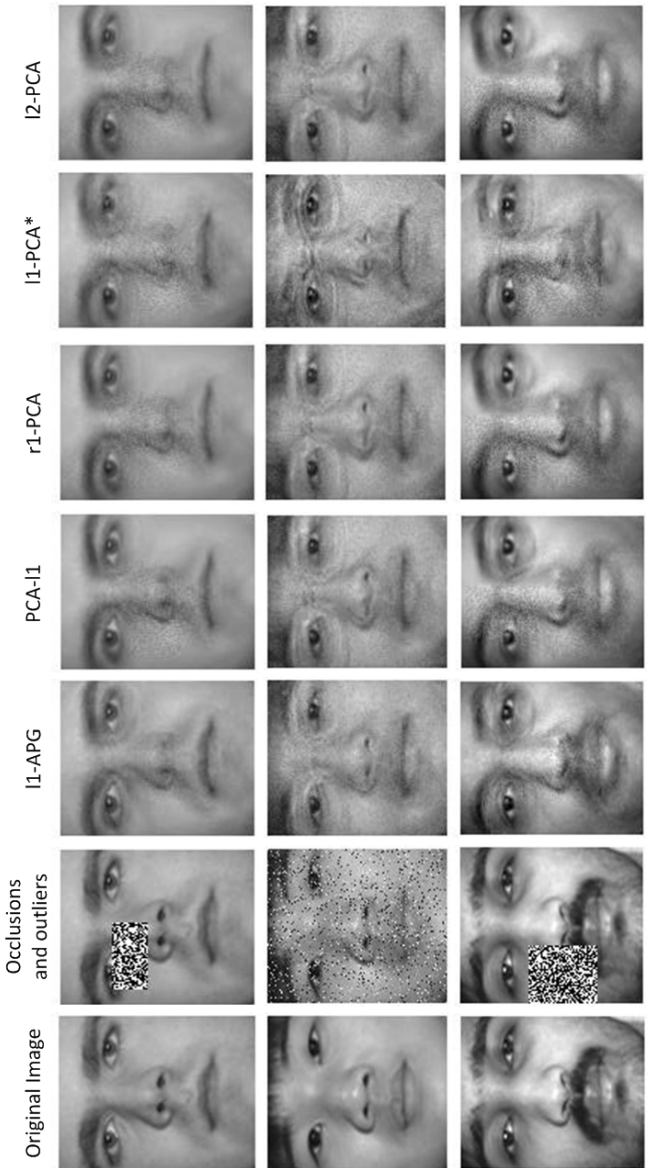
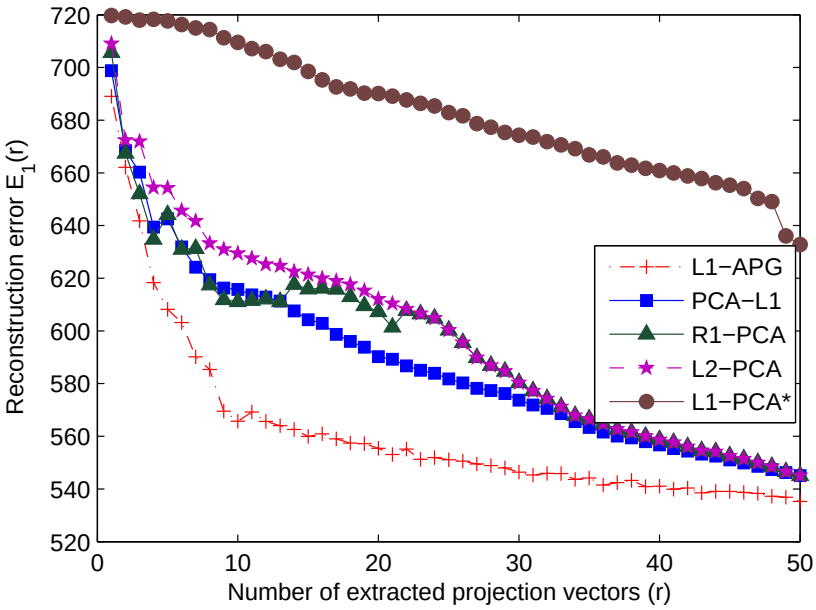
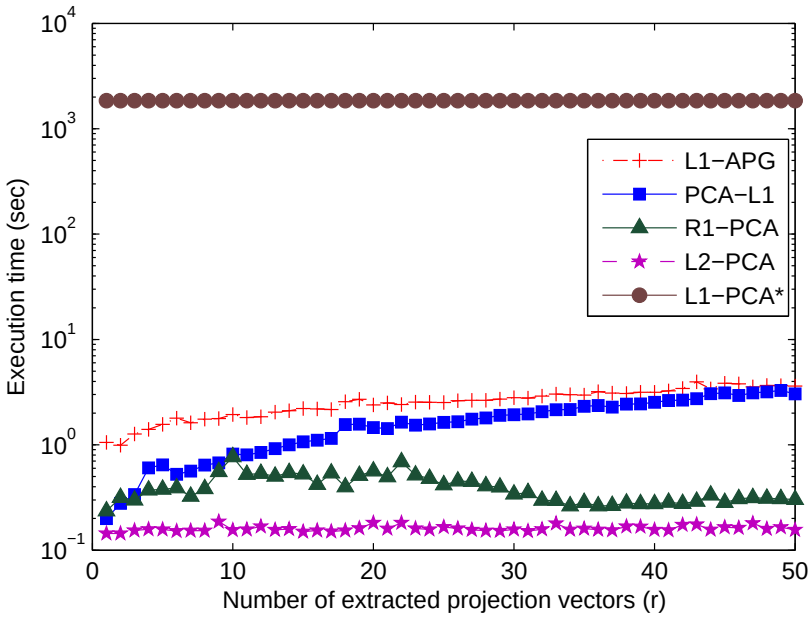


Figure 4.3 Face image with occlusions and outliers, and their reconstructed faces. First column: Original faces. Second column: Occluded and outlying faces. Third column: l_1-APG. Fourth column: PCA-l_1. Fifth column: r_1-PCA. Sixth column: l_1-PCA*. Last column: l_2-PCA.

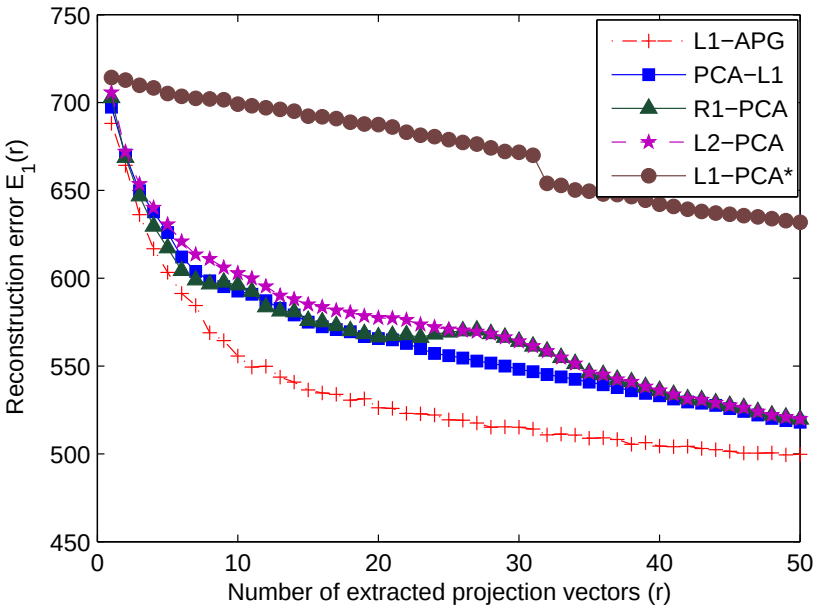


(a) Reconstruction error $E_1(r)$ with occlusions

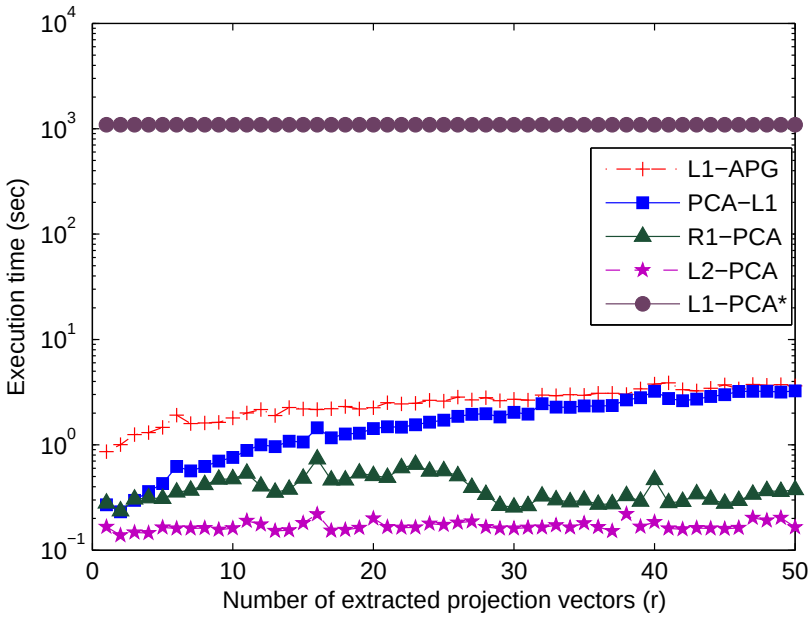


(b) Execution time with occlusions

Figure 4.4 Average reconstruction error $E_1(r)$ and execution time on the test face images with occlusions



(a) Reconstruction error $E_1(r)$ with outliers



(b) Execution time with outliers

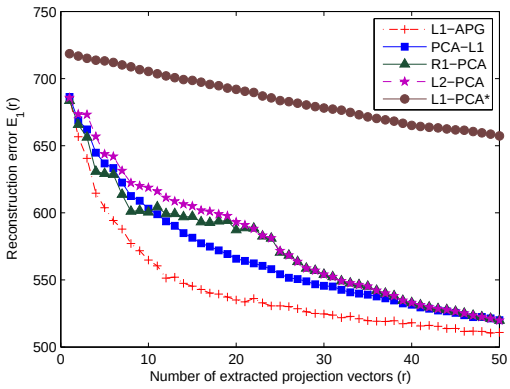
Figure 4.5 Average reconstruction error $E_1(r)$ and execution time on the test face images with outliers

to the others regardless of the number of extracted projection vectors, and the error difference between the proposed method and the other methods is large when the number of extracted projection vectors is relatively small. To get the equivalent reconstruction performance of l_1-APG with 30 projection vectors, the other methods need about 50 projection vectors, which is 66% more in the number of projection vectors. Figure 4.4(b) shows the execution time to find the projection vectors. Although l_1-APG took more execution time than the other three method except l_1-PCA*, it shows the least reconstruction error.

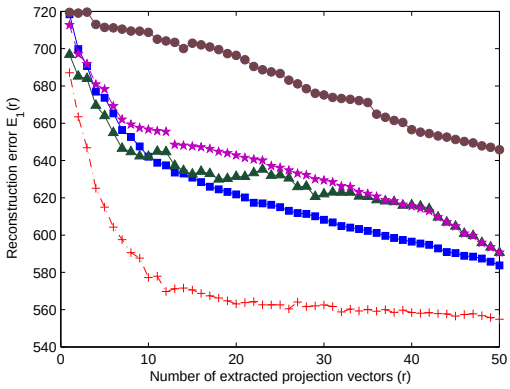
Figures 4.5(a) and 4.5(b) show the average reconstruction error and execution time for the test images with 15 percent outliers for different numbers of extracted projection vectors. The proposed method still shows the best performance compared to the other methods as in the case of images with occlusion.

For the next experiment, to measure the reconstruction error E_1 under the various conditions, we performed the experiment by varying the percentage of occluded images in the training samples and with the occlusion size varying in the range of 30×30 to 60×60, and its results are shown in Fig. 4.6. Also, after randomly selecting 50% of the training images, we performed the experiment by varying the percentage of outliers in the selected images, and its results are shown in Fig. 4.7. The proposed method is more robust than the other methods in both cases especially when the percentage of occluded images and the percentage of outliers are increased.

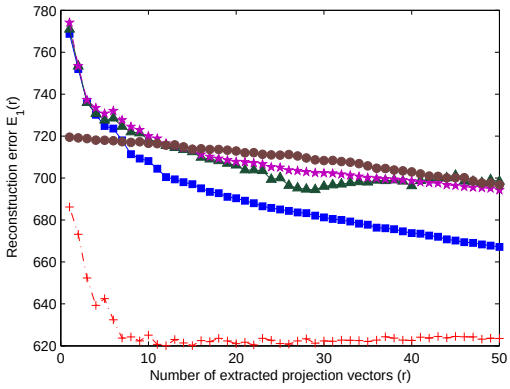
For the last experiment in this section, we resized the images of 100×120 pixels to 40×50 pixels because l_1-AQP could not be applied to an image of 100×120. Thirty percent of the training images are selected first, and 10 percent of pixels in the selected images were replaced with outliers. The same procedure is also applied to the test images. Figure 4.8(a) shows the average reconstruction error of the test images with out-



(a)

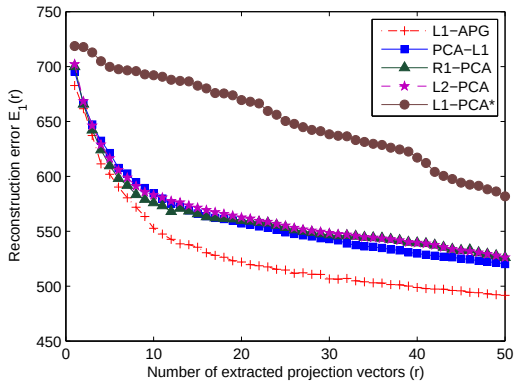


(b)

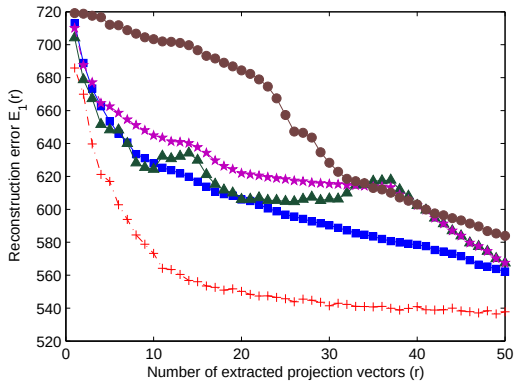


(c)

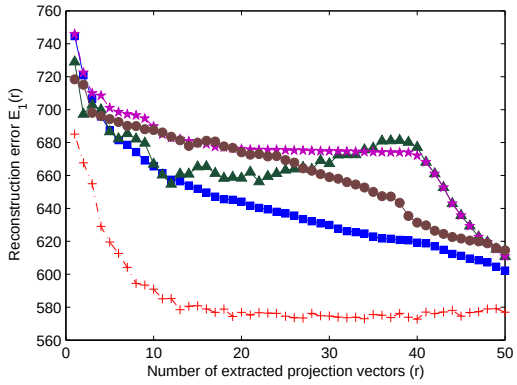
Figure 4.6 Average reconstruction error $E_1(r)$ for different percentages of occluded images in the training samples. (a) 20 percent. (b) 50 percent. (c) 100 percent.



(a)

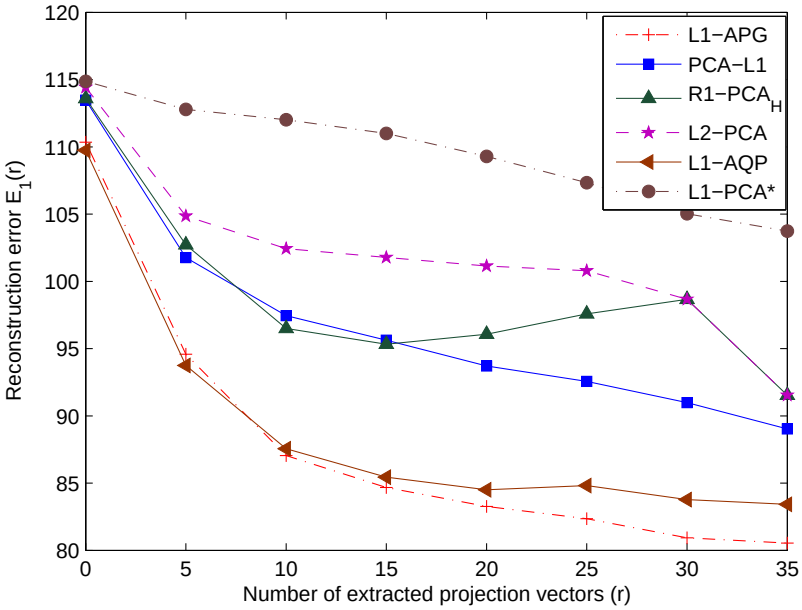


(b)

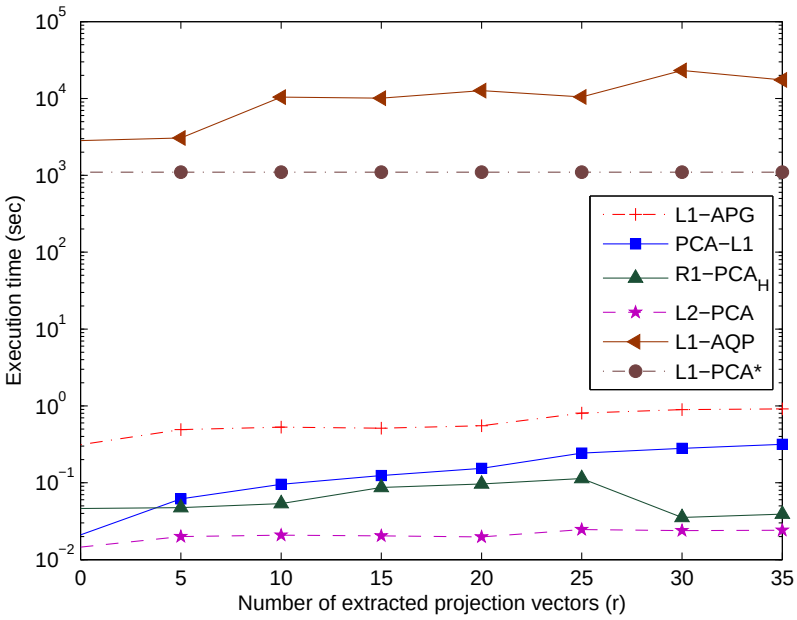


(c)

Figure 4.7 Average reconstruction error $E_1(r)$ for different percentages of outliers in an image. (a) 5 percent. (b) 15 percent. (c) 25 percent.



(a)



(b)

Figure 4.8 Average reconstruction error $E_1(r)$ and execution time for 40×50 face images with outliers

liers for various methods. The average value of test images using the l_1-norm is 1121. l_1-APG shows the best performance in all the cases. In this figure, l_1-AQP has similar reconstruction errors to the proposed method when the number of projection vectors is samll, because they have the same cost function. The performance of l_1-AQP gets a little bit worse than that of the proposed method when the number of extracted projection vectors increases. To get an equivalent reconstruction performance, l_1-APG needs much less projection vectors compared to the others. Moreover, the computation time is more than a thousand times longer for l_1-AQP than l_1-APG. Figure 4.8(b) shows the average execution time of various methods in the log-scale. Although the execution time was longer for l_1-APG compared to the other methods that were not based on l_1-cost function, it was much shorter compared to l_1-AQP and l_1-PCA*.

4.3 Experiments with missing data

We performed experiments in the presence of missing data using l_1-APG, l_1-Wiberg [10] and l_1-ALP [5], which can handle the missing data. To construct the an example, we generated a (m $\times$ r) matrix B and a (r $\times$ n) matrix C whose elements were uniformly distributed in the range of [0, 1]. We also generated a (m $\times$ n) noise matrix N whose elements had Gaussian distribution with zero mean and the variance of 0.2. The observation matrix $Y0$ was set as $Y0 = BC + N$. We constructed Y by replacing 10 percent of the elements in $Y0$ by outliers that was uniformly distributed in the range of [-6, 6]. We randomly selected 20 percent of the elements of matrix W and set them to zero while the other elements were set to one. In another experiment, we randomly selected 40 percent of the elements of matrix W and set them to zero while the other elements were set to one. We generated three sets of small example with the parameters: (m=7, n=12,

r=3), (m=10, n=15, r=5), and (m=20, n=15, r=6).

The l_1-Wiberg algorithm was implemented based on [10], which we modified the MATLAB program provided in the previous version [8] (available at http://cs.adelaide.edu.au/∼anders/code/cvpr2010.html). The initial values for the projection matrix of l_1-APG and l_1-ALP were set to random numbers which were selected from the Gaussian distribution of mean zero and variance of one. The initial values for the projection matrix of l_1-Wiberg was set to random numbers which were selected in the range of [0, 1].

Table.4.3 shows the average results for 100 examples with outliers and 20 percent of missing data. We also performed the same experiment with 40 percent missing data. The reconstruction errors are 30% - 300% greater for l_1-ALP and l_1-Wiberg compared to l_1-APG. These errors becomes greater as m, n, r, and the number of missing data become larger. The execution time is 100 - 1000 times shorter for l_1-APG compared to the other methods. Because of the execution time, l_1-Wiberg and l_1-ALP are impractical to use for a large-size data.

4.4 Face reconstruction with missing data

In this section, we show the face image reconstruction in the presence of missing data and an occlusion block for the Multi-PIE face database [22] which were resized as 100×120 pixels as in the previous section. We randomly selected a block, whose side length varied from 20 to 40 in an image that was selected from 20 percent of each training and test images, and considered the pixels in the block as missing. Similarly, we selected a block, whose side length varied from 30 to 60 in an image which was selected from 30 percent of each training and test images, and replaced the pixels by random numbers uni-

Table 4.3 The average results for 100 examples with 20 percent and 40 percent missing data (l_1-norm, second)

20% Missing	m=7, n=12, r=3		m=10, n=15, r=5		m=20, n=15, r=6	
Algorithm	Error	Execution Time	Error	Execution Time	Error	Execution Time
l_1-APG	2.277	0.002	2.714	0.003	5.819	0.005
l_1-Wiberg	2.916	3.097	5.366	4.211	7.489	42.691
l_1-ALP	2.846	0.411	5.434	1.181	8.579	9.812
40% Missing	m=7, n=12, r=3		m=10, n=15, r=5		m=20, n=15, r=6	
Algorithm	Error	Execution Time	Error	Execution Time	Error	Execution Time
l_1-APG	2.334	0.001	2.754	0.002	6.241	0.004
l_1-Wiberg	4.613	0.314	11.022	0.103	10.584	27.522
l_1-ALP	4.782	0.256	11.061	0.566	18.973	15.199

formly distributed in [0, 1]. The number of projection vectors was set to 30, hence the dimension of P is 12,000$\times$30. We did not apply l_1-ALP and l_1-Wiberg to this problem because of their heavy computational time and memory requirements.

Figure 4.9 shows the reconstructed face images using the proposed method. The occlusions and missing blocks were almost disappeared in the reconstructed images. The average reconstruction error E_2 for a test image is 7.1 and it took 2.64 sec to extract the 30 projection vectors.

4.5 Non-rigid motion estimation

Non-rigid motion estimation [13] with outliers and missing data from image sequences can be considered as a factorization problem. In this problem, l_1-norm-based factorization can be applied to restore the sequences in 2-dimensional space (2D tracks) effectively in the presence of outliers and missing data. In this experiment, we used the well-known benchmark shark sequence [23] consisting of 91 tracked points for each non-rigid shark shape in 240 frames.

To demonstrate the robustness of the proposed method, we replaced 10 percent of the pixels in a frame by outliers whose intensity was in the range of [-25, 25], whereas the intensity of a pixel is in the range of [-105, 105]. In addition, we set 20 percent of tracked points as missing in each frame. The number of shape basis was set to 2, and all the methods were initialized using the result obtained by applying l_2-PCA.

The result of reconstruction error [23] for each frame can be seen in Fig. 4.10. We compared the proposed method with l_1-ALP and iterative SVD method [24] and the stopping criterion for each algorithm was set to 10^{-6}. As shown in the figure, the error is less for l_1-APG compared to l_1-ALP and the iterative SVD. l_1-ALP has very high

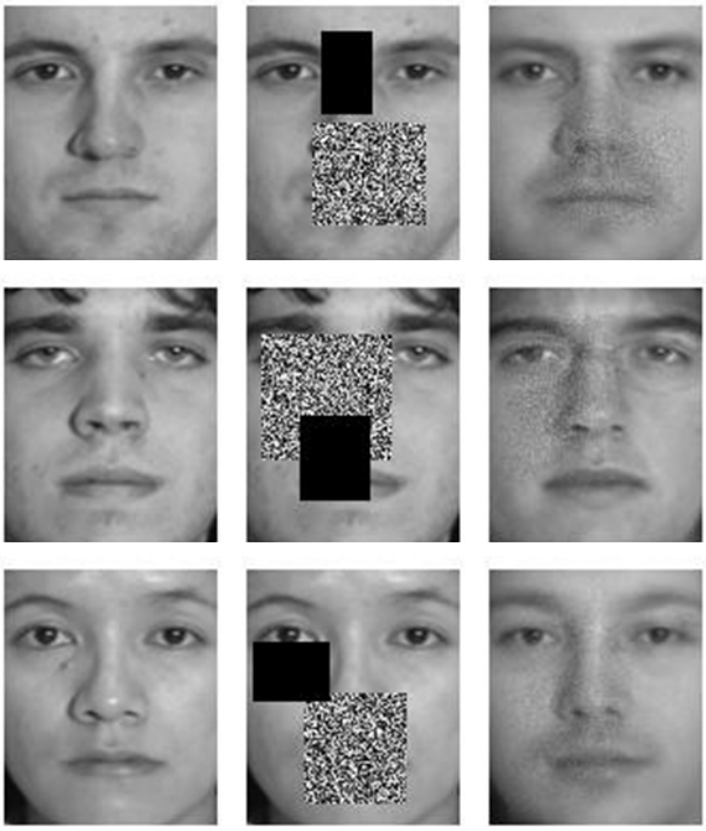


Figure 4.9 Face images with missing data and occlusion, and their reconstructed images. First column: Original images. Second column: Images with missing data and occlusion. Last column: Reconstructed images using the proposed method.

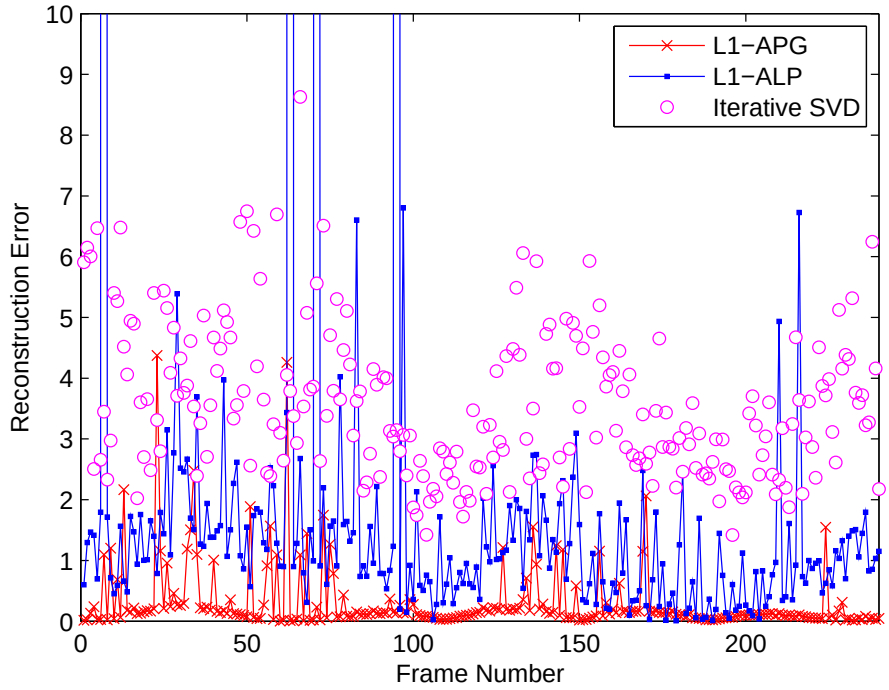
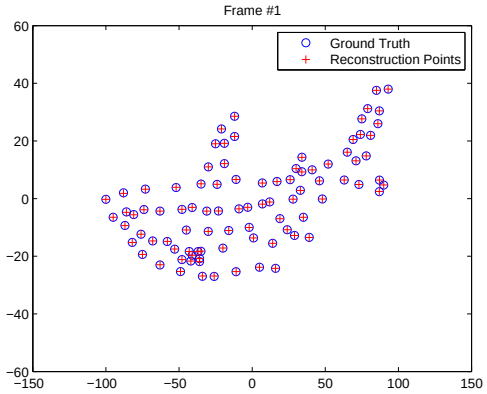


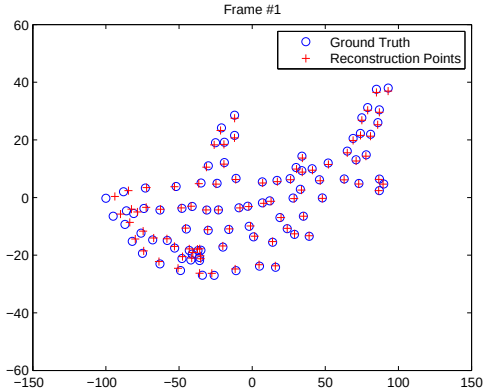
Figure 4.10 Reconstruction error for the shark sequence

error values for some of the frames. The average reconstruction errors [23] for each frame were 0.31 for l_1-APG, 6.95 for l_1-ALP, and 3.61 for the iterative SVD and the total execution time was 44 sec, 23 hour, and 0.4 sec, respectively. In this experiment, the l_1-Wiberg method was not applied because of its heavy memory requirements.

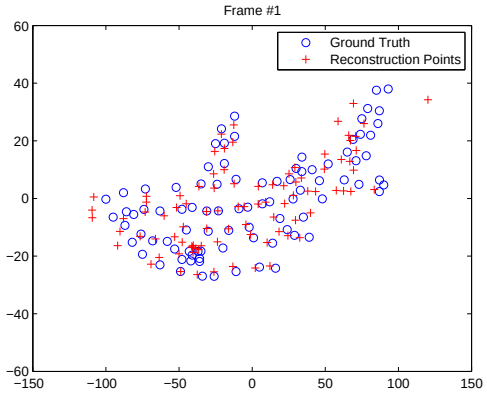
The reconstruction results of l_1-APG, l_1-ALP, and the iterative SVD for three selected frames are shown in Fig. 4.11, Fig. 4.12, and Fig. 4.13. We can see that l_1-APG shows better results than the other two methods.



(a)

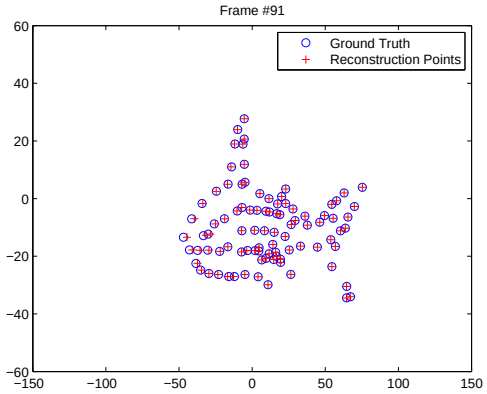


(b)

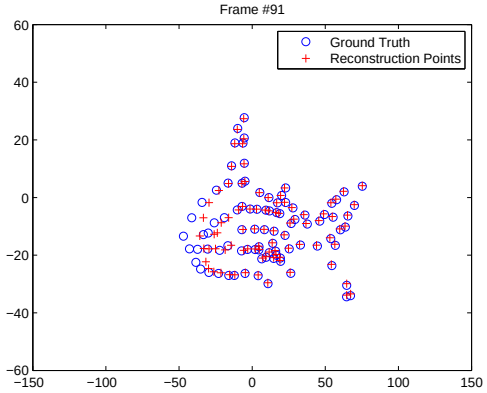


(c)

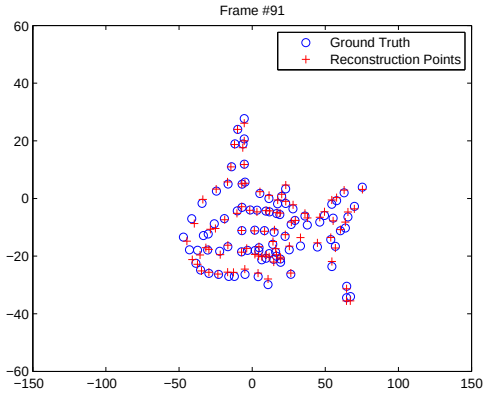
Figure 4.11 Non-rigid shape estimation from shark image sequences at frame t=1. (a): The proposed method. (b): l_1-ALP. (c): Iterative SVD.



(a)

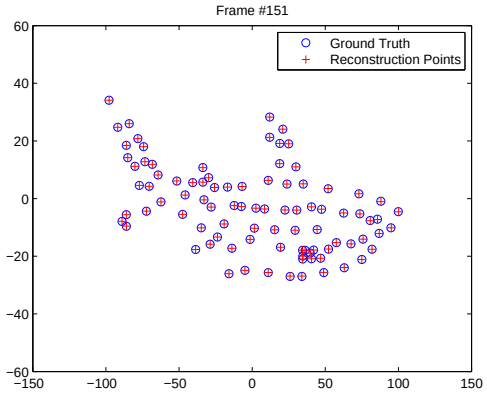


(b)

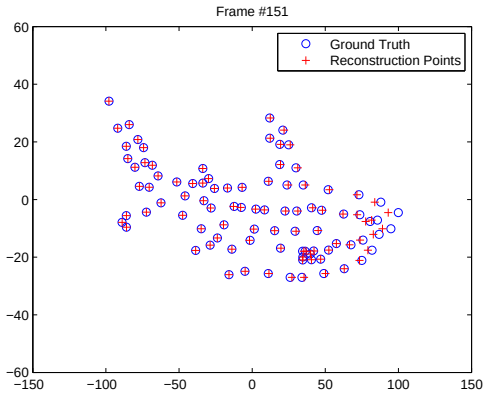


(c)

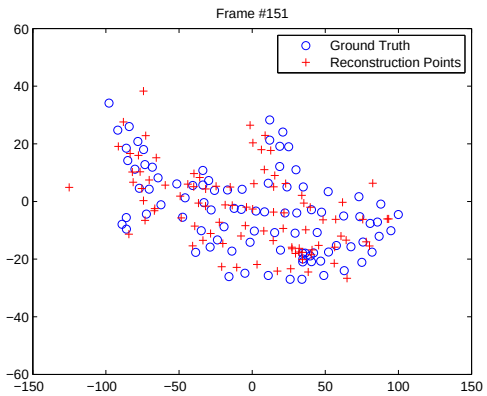
Figure 4.12 Non-rigid shape estimation from shark image sequences at frame t=91. (a): The proposed method. (b): l_1-ALP. (c): Iterative SVD.



(a)



(b)



(c)

Figure 4.13 Non-rigid shape estimation from shark image sequences at frame t=151. (a): The proposed method. (b): l_1-ALP. (c): Iterative SVD.

Chapter 5

Conclusion

In this thesis, we have proposed a novel method for low-rank matrix approximation in l_1-norm, which are based on the alternating projected gradient method. We also showed how to apply the proposed method when some of the data are missing. The conventional methods based on l_2-norm are sensitive to outliers, which can lead to poor approximation. Several l_1-norm approaches were introduced to overcome this shortcoming but make use of the convex programming and consequently require much computational time and memory. The proposed method overcomes these two shortcomings without degrading the approximation performance.

The proposed method is more robust to outliers and missing data than other conventional methods that is not based on l_1-norm. The proposed method is much faster and requires much less memory space compared to the other methods based on l_1-norm such as l_1-ALP, l_1-AQP, and l_1-Wiberg. It utilizes the alternating projected gradient method to find a proper gradient, and converges to a solution in a small number of iterations.

The proposed method was applied to face reconstruction problems under the circum-

stances of occlusions and missing data, and non-rigid motion recovery of the shark sequence. The experimental results show that the proposed method provides excellent reconstruction performances compared to the other methods except in a few cases. Even in these cases, the error difference between the best method and the proposed method is very small. The execution time is approximately 100 - 1000 times faster for the proposed method compared to the other l_1-based methods in the examples of Chapter 4. This enables the proposed method to be applied in practical problems.

Bibliography

[1] I. T. Jolliffe, "Principal component analysis," in *Principal Component Analysis*, 1986.

[2] D. M. Hawkins, L. Liu, and S. S. Young, "Robust singular value decomposition," *Technical report, National Institute of Statistical Sciences*, 2001.

[3] Q. Ke and T. Kanade, "Robust subspace computation using l_1 norm," *Technical Report CMU-CS-03-172, Carnegie Mellon Univ.*, Aug. 2003.

[4] C. Ding, D. Zhou, X. He, and H. Zha, "r_1-pca: Rotational invariant l_1-norm principal component analysis for robust subspace factorization," in *in Proc. of IEEE International Conf. Machine Learning*, 2006.

[5] Q. Ke and T. Kanade, "Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *in Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, June 2005.

[6] N. Kwak, "Principal component analysis based on l_1-norm maximization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, no. 9, pp. 1672–1680, 2008.

[7] J. Brooks, J. Dula, and E. Boone, "A pure l_1-norm principal component analysis," *Technical report, Optimization Online*, 2010.

[8] A. Eriksson and A. van den Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm," in *in Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, June 2010.

[9] P. B. A. Baccini and A. Falguerolles, "An l_1-norm pca and a heuristic approach," *Ordinal and symbolic data analysis. In E. Diday, Y. Lechevalier, and O. Opitz, editors*, pp. 359–368, 1996. Springer.

[10] A. Eriksson and A. van den Hengel, "Efficient computation of robust weighted low-rank matrix approximations using the l_1 norm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1681–1690, 2012.

[11] S. Jot and J. P. Brooks, "pcal_1: Three l_1-norm pca methods," *R package version 1.2*, 2012.

[12] T. Wiberg, "Computation of principal components when data are missing," *in Proc. Second Symp. Computational Statistics*, pp. 229–236, 1976.

[13] Y. Shen, Z. Wen, and Y. Zhang, "Shape and motion from image streams under orthography: factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 2011.

[14] F. Y. Edgeworth, "On a new method of reducing observations relating to several quantities," *Philosophical Magazine*, vol. 25, pp. 184–191, 1888.

[15] C.Gurwitz, "Weighted median algorithms for l_1 approximation," *BIT 30*, pp. 301–310, 1990.

[16] G. R. Arce, "A general weighted median filter structure admitting negative weights," *IEEE Trans. Signal Processing*, vol. 46, no. 12, pp. 3195–3205, Dec. 1998.

[17] A. Rauh and G. R. Arce, "A fast weighted median algorithm based on quickselect," in *in Proc. IEEE conf. Image Processing*, Sep. 2010, pp. 105–108.

[18] I. Csiszar and G. TTusnady, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplement Issue*, pp. 1:205–237, 1984.

[19] C. A. R. Hoare, "Algorithm 65: find," *Communications of the ACM*, vol. 4, no. 7, pp. 321–322, July 1961.

[20] P. N. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces versus. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[21] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative learning and recognition of image set classes using canonical correlations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1005–1018, June 2007.

[22] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," in *in Proc. IEEE conf. Automatic Face and Gesture Recognition*, Sep. 2008.

[23] L. Torresani, A. Hertzmann, and C. Bregler, "Learning non-rigid 3d shape from 2d motion," in *in Proc. Neural Information Processing Systems*, 2003.

[24] M. Kurucz, A. A. Benczúr, and B. Torma, "Methods for large scale svd with missing values," in *KDDCup 2007*, 2007.

한글 초록

저 차원 행렬 근사는 컴퓨터 비전이나 이미지 프로세싱 분야에서 널리 사용되고 중요한 역할을 하는 방법 중 하나이다. 주 성분 분석이라고도 불리는 저 차원 행렬 근사 방법들의 대부분은 l_2-norm을 기반으로 수행되고 있다. 하지만 이와 같은 방법들은 데이터가 outliers나 missing data를 포함하고있을 경우, 이러한 원치 않는 값들을 과장하여 표현함으로써 시스템의 성능을 저하시킬 수 있기 때문에 적절한 방법이라고 할 수 없다. 최근에 이러한 문제들을 극복하고자, l_2-norm 대신 l_1-norm을 기반으로 한 다양한 방법들이 제안되었다. l_1-norm을 기반으로 한 행렬 근사 방법들은 outlier나 비어있는 값들에 대하여 강인하지만, convex 프로그래밍을 사용하기 때문에 계산량이 매우 많다는 단점을 가지고 있다. 따라서, 본 논문에서는 같은 곱의 결과를 가지는 수 많은 기저행렬과 계수행렬들 사이에서 가장 적절한 gradient의 방향을 위한 기저 및 계수 행렬을 찾아 강인하면서 매우 적은 시간으로 수행할 수 있는 l_1-norm 기반의 저 차원 행렬 근사 방법을 제안한다. 교차로 투영되는 gradient 방법을 통해 l_1-norm 기반의 목적 함수를 그대로 사용하면서, 선형 또는 이차의 convex 프로그래밍을 사용하는 방법에 비해 수행 시간을 상당히 줄일 수 있었다. 제안하는 방법은 다른 방법들보다 성능의 우수함을 보이기 위해 몇몇 패턴 인식 및 컴퓨터 비전 문제에 대해 적용하여 실험을 하였으며, 결론적으로 우수한 성능을 보여 현실적인 응용에 적합함을 확인할 수 있었다.

주요어 : 낮은 차수 행렬 근사, l_1-norm, QR 분해, 교차로 투영되는 gradient기법, 구조와 모션

학번 : 2011-20811

감사의 글

대학원에 기대 반, 걱정 반으로 들어온 지 벌써 2년이라는 시간이 지났습니다. 2년이 지난 지금 생각해보면 많은 아쉬움들이 남아있습니다. 이제 졸업을 하게 되고, 다시 새로운 시작을 위한 발걸음을 내딛게 됩니다. 이 새로운 시작을 맞이 할 수 있도록 그동안 많은 도움과 관심을 주신 분들에게 짧게 글로나마 감사의 마음을 전하고자 합니다.

먼저 제가 대학원 생활을 무사히 마칠 수 있도록 물심양면으로 보살펴 주신 부모님께 감사드립니다. 많은 관심과 격려 덕분에 철없는 아들이 이제 졸업을 하게 되었고, 새로운 시작을 하는데에 많은 도움을 주셨습니다. 또한 어릴적부터 지금까지 많은 도움과 관심을 준 누나에게도 감사드립니다.

지도 교수님이신 최종호 교수님 감사드립니다. 석사과정 동안 교수님의 지도 아래 많은 것을 배울 수 있었고, 무사히 졸업을 할 수 있게 되었습니다. 그리고 바쁘신 와중에도 시간을 내어 학위논문을 심사해 주신 최진영 교수님, 저의 박사과정 지도교수님이 되시며 많은 조언을 해주신 오성회 교수님께도 깊이 감사드립니다.

2년 동안 연구실 생활을 함께 했던 제어 및 시스템 연구원분들께도 감사드립니다. 먼저 연구실 생활 처음부터 많은 도움과 조언을 해주셨던 지용이형과 상일이형, 많이 뵙지는 못했지만 많은 조언을 주셨던 정훈이형, 그리고 제가 석사논문을

쓸 수 있었고 진로와 관련해서도 아낌없는 도움을 주신 민식이형과 논문을 쓰는데 도움을 주셨던 곽노준 선배님, 짧았지만 재미있었던 추억이 많은 승호, 그리고 바로 윗 선배로써 처음 대학원에 들어왔을 때 많은 도움을 주었던 준희형과 진학을 위해 많은 도움을 주셨고 앞으로도 비슷한 연구를 계속 같이 할 정찬이형, 동갑으로 같이 연구실에 들어와 큰 힘이 되었던 동기 민규 모두 감사드립니다. 또한 같은 연구분야는 아니지만 졸업 할 때까지 많은 도움을 받았고 즐겁게 생활할 수 있었던 형호형과 민호형, 그리고 영수형 모두 감사드립니다. 선배님들의 격려와 충고덕에 무사히 졸업을 할 수 있게 되었습니다.

이제 2년간의 석사과정을 마치고 박사과정으로써 새로운 발걸음을 하게 됩니다. 모든 분들의 격려와 충고를 발판삼아 다시 도약할 수 있도록 최선을 다하겠습니다. 감사합니다.