Optimized Projections for Compressed Sensing via Direct Mutual Coherence Minimization

Can yi Lu, Student Member, IEEE, Huan Li and Zhouchen Lin, Senior Member, IEEE

Compressed Sensing (CS) is a new sampling/data acquisition theory based on the existence of a sparse representation of a signal and a projected dictionary PD, where P ∈ R^{m×d} is the projection matrix and D ∈ R^{d×n} is the dictionary. To exactly recover the signal with a small number m of measurements, it is expected that the projected dictionary PD is of low mutual coherence. Several previous methods attempt to find the projection P such that the mutual coherence of PD is as low as possible. However, they do not minimize the mutual coherence directly and thus these methods may be far from optimal. Also, the solvers they use lack convergence guarantee and thus the quality of their obtained solutions is not guaranteed. This work aims to address these issues. We propose to find an optimal projection matrix by minimizing the mutual coherence of PD directly. This leads to a nonconvex nonsmooth minimization problem. We then approximate it by smoothing and solve it by alternating minimization. We further prove the convergence of our algorithm. To the best of our knowledge, this is the first work which directly minimizes the mutual coherence of the projected dictionary and has convergence guarantee. Numerical experiments demonstrate that the proposed method can recover sparse signals better than existing ones.

I. INTRODUCTION

Compressed Sensing (CS) [1, 2] is a new sampling/data acquisition theory asserting that one can exploit sparsity or compressibility when acquiring signals of interest. It shows that signals which have a sparse representation with respect to appropriate bases can be recovered from a small number of measurements. A fundamental problem in CS is how to construct a measurement matrix such that the number of measurements is near minimal.

Consider a signal x ∈ R^d which is assumed to have a sparse representation with respect to a fixed overcomplete dictionary D ∈ R^{d×n} (d < n). This can be described as

\[ x = D\alpha, \]

where \( \alpha \in R^n \) is a sparse representation coefficient, i.e., \( \|\alpha\|_0 \ll n \). Here \( \|\alpha\|_0 \) denotes the \( \ell_0 \)-norm which counts the number of nonzero elements in \( \alpha \). The solution to problem (1) is not unique since \( d < n \). To find an appropriate solution in the solution set of (1), we need to use some additional structures of D and \( \alpha \). Considering that \( \alpha \) is sparse, we are interested in finding the sparsest representation coefficient \( \alpha \). This leads to the following sparse representation problem

\[ \min_{\alpha} \|\alpha\|_0, \text{ s.t. } x = D\alpha. \]

However, the above problem is NP-hard and thus is challenging to solve. Some algorithms, such as Basis Pursuit (BP) [3] and Orthogonal Matching Pursuit (OMP) [4], can be used to find suboptimal solutions.

An interesting theoretical problem is that under what conditions the optimal solution to (2) can be computed. If the solution is computable, can it be exactly or approximately computed by BP or OMP? Some previous works answer the above questions based on the mutual coherence of the dictionary D [6].

**Definition 1:** Given \( D = [d_1, \ldots, d_n] \in R^{d×n} \), its mutual coherence is defined as the largest absolute and normalized inner product between different columns of D, i.e.,

\[ \mu(D) = \max_{i\neq j} \frac{|d_i^T d_j|}{\|d_i\| \cdot \|d_j\|}. \]

The mutual coherence measures the highest correlation between any two columns of D. It is expected to be as low as possible in order to find the sparsest solution to (2).

**Theorem 1:** [6, 11, 8] For problem (2), if \( \alpha \) satisfies

\[ \|\alpha\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(D)} \right), \]

then the following results hold:

- \( \alpha \) is the solution to (2).
- \( \alpha \) is also the solution to the following convex \( \ell_1 \)-minimization problem

\[ \min_{\alpha} \|\alpha\|_1, \text{ s.t. } x = D\alpha, \]

where \( \|\alpha\|_1 = \sum_i |\alpha_i| \) is the \( \ell_1 \)-norm of \( \alpha \).
- \( \alpha \) can be obtained by OMP.

The above theorem shows that if the mutual coherence of D is low enough, then the sparsest solution to (2) is computable. Thus, how to construct a dictionary D with low mutual coherence is crucial in sparse coding. In CS, to reduce the number of measurements, we face a similar problem on the sensing matrix construction.

The theory of CS guarantees that a signal having a sparse representation can be recovered exactly from a small set of linear and nonadaptive measurements. This result suggests that it may be possible to sense sparse signals by taking far fewer measurements than what the conventional Nyquist-Shannon sampling theorem requires. But note that CS differs from classical sampling in several aspects. First, the sampling theory typically considers infinite-length and continuous-time signals. In contrast, CS is a mathematical theory that focuses on measuring finite-dimensional vectors in \( R^n \). Second, rather than sampling the signal at specific points in time, CS systems
typically acquire measurements in the form of inner products between the signal and general test functions. At last, the ways to dealing with the signal recovery are different. Given the signal \( x \in \mathbb{R}^d \) in (4), CS suggests replacing these \( n \) direct samples with \( m \) indirect ones by measuring linear projections of \( x \) defined by a proper projection or sensing matrix \( P \in \mathbb{R}^{m \times d} \), i.e.,

\[
y = Px,
\]

such that \( m \ll d \). It means that instead of sensing all \( n \) elements of the original signal \( x \), we can sense \( x \) indirectly by its compressed form \( y \) in a much smaller size \( m \). Surprisingly, the original signal \( x \) can be recovered from the observed \( y \) by using the sparse representation in (1), i.e., \( y = PD\alpha \) with the sparest \( \alpha \). Thus the reconstruction requires solving the following problem

\[
\min_{\alpha} \|\alpha\|_1, \quad \text{s. t.} \quad y = M\alpha,
\]

where \( M = PD \in \mathbb{R}^{m \times n} \) is called the effective dictionary. Problem (5) is also NP-hard. As suggested by Theorem 11, if the mutual coherence of \( PD \) is low enough, then the solution \( \alpha \) to (5) is computable by OMP or by solving the following convex problem

\[
\min_{\alpha} \|\alpha\|_1, \quad \text{s. t.} \quad y = M\alpha.
\]

Finally, the original signal \( x \) can be reconstructed by \( x = D\alpha \). So it is expected to find a proper projection matrix \( P \) such that \( \mu(PD) \) is low. Furthermore, many previous works [9], [10] show that the required number of measurements for recovering the signal \( x \) by CS can be reduced if \( \mu(PD) \) is low.

In summary, the above discussions imply that by choosing an appropriate projection matrix \( P \) such that \( \mu(PD) \) is low enough, the true signal \( x \) can be recovered with high probability by efficient algorithms. At the beginning, random projection matrices were shown to be good choices since their columns are incoherent with any fixed basis \( D \) with high probability [11]. However, many previous works [9], [12], [10] show that well designed deterministic projection matrices can often lead to better performance of signal reconstruction than random projections do. In this work, we focus on the construction of deterministic projection matrices. We first give a brief review on some previous deterministic methods.

### A. Related Work

In this work, we only consider the case that \( D \) is fixed while \( P \) can be changed. Our target is to find \( P \) by minimizing \( \mu(M) \), where \( M = PD \). If each column of \( M \) is normalized to have unit Euclidean length, then \( \mu(M) = \|G\|_{\infty,\text{off}} \), where \( G = (g_{ij}) = M^{T}M \) is named as the Gram matrix and \( \|G\|_{\infty,\text{off}} = \max_{i \neq j} |g_{ij}| \) is the largest off-diagonal element of \( |G| \). Several previous works used the Gram matrix to find the projection matrix \( P \) [9], [12], [10]. We give a review on these methods in the following.

1) The Algorithm of Elad

The algorithm of Elad [9] considers minimizing the \( t \)-averaged mutual coherence defined as the average of the absolute and normalized inner products between different columns of \( M \) which are above \( t \), i.e.,

\[
\mu_t(M) = \frac{\sum_{1 \leq i,j \leq k, i \neq j} \chi_t(|g_{ij}|)}{\sum_{1 \leq i,j \leq k} \chi_t(|g_{ij}|)},
\]

where \( \chi_t(x) \) is the characteristic function defined as

\[
\chi_t(x) = \begin{cases} 
1, & \text{if } x \geq t, \\
0, & \text{otherwise}
\end{cases}
\]

and \( t \) is a fixed threshold which controls the top fraction of the matrix elements of \( |G| \) that are to be considered.

To find \( P \) by minimizing \( \mu_t(M) \), some properties of the Gram matrix \( G = M^{T}M \) are used. Assume that each column of \( M \) is normalized to have unit Euclidean length. Then

\[
\begin{align*}
\text{diag}(G) &= 1, \\
\text{rank}(G) &= m.
\end{align*}
\]

The work [9] proposed to minimize \( \mu_t(M) \) by iteratively updating \( P \) as follows. First, initialize \( P \) as a random matrix and normalize each column of \( PD \) to have unit Euclidean length. Second, shrink the elements of \( G = M^{T}M \) (where \( M = PD \)) by

\[
g_{ij} = \begin{cases} 
\gamma g_{ij}, & \text{if } |g_{ij}| \geq t, \\
\gamma \text{sign}(g_{ij}), & \text{if } t > |g_{ij}| \geq \gamma t, \\
g_{ij}, & \text{if } \gamma t > |g_{ij}|,
\end{cases}
\]

where \( 0 < \gamma < 1 \) is a down-scaling factor. Third, apply SVD and reduce the rank of \( G \) to be equal to \( m \). At last, build the square root of \( G \): \( S^{T}S = G \), where \( S \in \mathbb{R}^{m \times n} \), and find \( P = SD^{1} \), where \(^{1}\) denotes the Moore-Penrose pseudoinverse.

There are several limitations of the algorithm of Elad. First, it is suboptimal since the \( t \)-averaged mutual coherence \( \mu_t(M) \) is different from the mutual coherence \( \mu(M) \) which is our real target. Second, the proposed algorithm to minimize \( \mu_t(M) \) has no convergence guarantee [4]. So the quality of the obtained solution is not guaranteed. Third, the choices of two parameters, \( t \) and \( \gamma \), are crucial for the signal recovery performance in CS. However, there is no guideline for their settings and thus in practice it is usually difficult to find their best choices.

2) The Algorithm of Duarte-Carajalino and Sapiro

The algorithm of Duarte-Carajalino and Sapiro [12] is not a method that is based on mutual coherence. It instead aims to find the sensing matrix \( P \) such that the corresponding Gram matrix is as close to the identity matrix as possible, i.e.,

\[
G = M^{T}M = D^{T}P^{T}PD \approx I,
\]

where \( I \) denotes the identity matrix. Multiplying both sides of the previous expression by \( D \) on the left and \( D^{T} \) on the right, it becomes

\[
DD^{T}P^{T}PD = DD^{T}.
\]

\(^{1}\)In this paper, an algorithm converges means that any accumulation point is a stationary point.
Let $DD^T = VΛV^T$ be the eigen-decomposition of $DD^T$. Then (10) is equivalent to

$$ΛV^TP^TPVΛ = Λ.$$  

(11)

Define $Γ = PV$. Then they finally formulate the following model w.r.t. $Γ$

$$\min \| Λ - ΛΓ^TΓΛ \|_F.$$  

(12)

After solving the above problem, the projection matrix can be obtained as $P = ΓV^T$.

However, usually the signal recovery performance of the algorithm of Duarte-Carajalino and Sapiro is not very good. The reason is that $M$ is overcomplete and the Gram matrix $G$ cannot be an identity matrix. In this case, simply minimizing the difference between the Gram matrix $G$ and the identity matrix does not imply a solution $M$ with low mutual coherence.

3) The Algorithm of Xu et al.

The algorithm of Xu et al. [10] is motivated by the well-known Welch bound [13]. For any $M ∈ R^{m×n}$, the mutual coherence $μ(M)$ is lower bounded, e.g.,

$$μ(M) ≥ \sqrt{\frac{n - m}{m(n - 1)}}.$$  

(13)

The algorithm of Xu et al. aims to find $M$ such that the off-diagonal elements of $G = M^TM$ approximate the Welch bound well. They proposed to solve the following problem

$$\min_G \| G - G_Λ \|_F$$  

s.t. $G_Λ = G^T_Λ$, $\text{diag}(G_Λ) = 1$, $\| G_Λ \|_{∞,off} ≤ μ_W$,  

(14)

where $μ_W = \sqrt{\frac{n - m}{m(n - 1)}}$. The proposed iterative solver for the above problem is similar to the algorithm of Elad. The main difference is the shrinkage function used to control the elements of $G$. See [10] for more details.

However, their proposed solver in [10] for (14) also lacks convergence guarantee. Another issue is that, for $M ∈ R^{m×n}$, the Welch bound [13] is not tight when $n$ is large. Actually, the equality of [13] can hold only when $n ≤ \frac{m(m+1)}{2}$. This implies that the algorithm of Xu et al. is not optimal when $n > \frac{m(m+1)}{2}$.

B. Contributions

There are at least two main issues in the previous methods reviewed above. First, none of them aims to find $P$ by directly minimizing $μ(PD)$ which is our real target. Thus the objectives of these methods are not optimal. For their obtained solutions $P$, $μ(PD)$ is usually much larger than the Welch bound in [13]. Second, the algorithms of Elad and Xu et al. have no convergence guarantee and thus they may produce very different solutions given slightly different initializations. The convergence issue may limit their applications in CS.

To address the above issues, we develop Direct Mutual Coherence Minimization (DMCM) models. First, we show how to construct a low mutual coherence matrix $M$ by minimizing $μ(M)$ directly. This leads to a nonconvex and nonsmooth problem. To solve our new problem efficiently, we first smooth the objective function such that its gradient is Lipschitz continuous. Then we solve the approximate problem by proximal gradient which has convergence guarantee. Second, inspired by DMCM, we propose a DMCM based Projection (DMCM-P) model which aims to find a projection $P$ by minimizing $μ(PD)$ directly. To solve the nonconvex DMCM-P problem, we then propose an alternating minimization method and prove its convergence. Experimental results show that our DMCM-P achieves the lowest mutual coherence of $PD$ and also leads to the best signal recovery performance.

II. LOW MUTUAL COHERENCE MATRIX CONSTRUCTION

In this section, we show how to construct a matrix $M ∈ R^{m×n}$ with low mutual coherence $μ(M)$ by DMCM. Assume that each column of $M$ is normalized to unit Euclidean length. Then we aim to find $M$ by the following DMCM model

$$\min_{M ∈ R^{m×n}} μ(M) = \| M^TM - I \|_{∞,off}$$  

s.t. $\| M_i \|_2 = 1$, $i = 1, \cdots, n$,  

(15)

where $M_i$ (or $(M_i)_i$) denotes the $i$-th column of $M$. The above problem is equivalent to

$$\min_{M ∈ R^{m×n}} f(M) = \| M^TM - I \|_{∞}$$  

s.t. $\| M_i \|_2 = 1$, $i = 1, \cdots, n$,  

(16)

where $\| A \|_{∞} = \max_{i,j} |a_{ij}|$ denotes the $ℓ_∞$-norm of $A$. Solving the above problem is not easy since it is nonconvex and its objective is nonsmooth. In general, due to the nonconvexity, the globally optimal solution to (16) is not computable. We instead consider finding a locally optimal solution with convergence guarantee.

First, to ease the problem, we adopt the smoothing technique in [14] to smooth the nonsmooth $ℓ_∞$-norm in the objective of (16). By the fact that the $ℓ_1$-norm is the dual norm of the $ℓ_∞$-norm, the objective function in (16) can be rewritten as

$$f(M) = \| M^TM - I \|_{ℓ_∞} = \max_{\|V\|_1 ≤ 1} \langle M^TM - I, V \rangle,$$

where $\|V\| = \sum_{ij} |v_{ij}|$ denotes the $ℓ_1$-norm of $V$. Since $\{V \| V\|_1 ≤ 1\}$ is a bounded convex set, we can define a proximal function $d(V)$ for this set, where $d(V)$ is continuous and strongly convex on this set. A natural choice of $d(V)$ is $d(V) = \frac{1}{2} \| V \|_F^2$, where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. Hence, we have the following smooth approximation of $f$ defined in (16):

$$f_{ρ}(M) = \max_{\|V\|_1 ≤ 1} \langle M^TM - I, V \rangle - \frac{ρ}{2} \| V \|_F^2,$$  

(17)

where $ρ > 0$ is a smoothing parameter. Note that the smooth function $f_{ρ}$ can approximate the nonsmooth $f$ with an arbitrary precision and it is easier to be minimized. Indeed, $f$ and $f_{ρ}$ have the following relationship

$$f_{ρ}(M) ≤ f(M) ≤ f_{ρ}(M) + ργ,$$

where $γ = \max_{V} \{ \frac{1}{2} \| V \|_F^2 | \|V\|_∞ ≤ 1 \}$. For any $ε > 0$, if we choose $ρ = \frac{ε}{γ}$, then $|f(M) - f_{ρ}(M)| ≤ ε$. This implies that if $ρ$ is sufficiently small, then the difference between $f$
With the above property, problem (18) can be solved by the L
While \( k < K \)
Output: \( M^* = \text{PG}(M_k, \rho) \).
while \( k < K \) do
1) Compute \( V_k \) by solving (21);
2) Compute \( M_{k+1} \) by solving (19);
3) \( k = k + 1 \).
end while

and \( f_\rho \) can be very small. This motives us to use \( f_\rho \) to replace \( f \) in (16) and thus we have the following relaxed problem

\[
\min_{M \in \mathbb{R}^{m \times n}} f_\rho(M) \\
\text{s.t. } ||M_i||_2 = 1, \ i = 1, \ldots, n.
\] (18)

As \( f_\rho \) can approximate \( f \) at an arbitrary precision, solving (18) can still be regarded as directly minimizing the mutual coherence. Problem \( \text{PG} \) is easier to solve since \( \nabla f_\rho(M) = M(V^* + V^T) \), where \( V^* \) is the optimal solution to (17), is Lipschitz continuous. That is, for any \( M_1, M_2 \in \mathbb{R}^{m \times n} \), there exists a constant \( L = 1/\rho \) such that

\[
||\nabla f_\rho(M_1) - \nabla f_\rho(M_2)||_F \leq L ||M_1 - M_2||_F.
\]

With the above property, problem (18) can be solved by the proximal gradient method which updates \( M \) in the \((k+1)\)-th iteration by

\[
M_{k+1} = \arg\min_{M} \frac{1}{2} ||M - (M_k - \alpha \nabla f_\rho(M_k))||_F^2 \\
\text{s.t. } ||M_i||_2 = 1, \ i = 1, \ldots, n,
\] (19)

where \( \alpha > 0 \) is the step size. To guarantee convergence, it is required that \( \alpha < \rho \). In this work, we simply set \( \alpha = 0.99\rho \).

The above problem has a closed form solution by normalizing each column of \( M_k - \alpha \nabla f_\rho(M_k) \), i.e.,

\[
(M_{k+1})_i = \frac{(M_k - \alpha \nabla f_\rho(M_k))_i}{||(M_k - \alpha \nabla f_\rho(M_k))_i||_2}.
\] (20)

To compute \( \nabla f_\rho(M_k) = M_k(V_k + V_k^T) \), where \( V_k \) is optimal to (17) when \( M = M_k \), one has to solve (17) which is equivalent to the following problem

\[
V_k = \arg\min_{V} \frac{1}{2} ||V - (M_k^T M_k - I)||_F \\\n\text{s.t. } ||V||_1 \leq 1.
\] (21)

Solving the above problem requires computing a proximal projection onto the \( \ell_1 \) ball. This can be done efficiently by the method in (15).

Iteratively updating \( V \) by (21) and \( M \) by (19) leads to the Proximal Gradient (PG) algorithm for solving problem (18). We summarize the whole procedure of PG for (18) in Algorithm [1]. It can be easily seen that the per-iteration cost of Algorithm [1] is \( O(m^2n + mn^2) \). For nonconvex problems, e.g., (18), it was proved that PG is guaranteed to converge.

Theorem 2: Let \( \{M_k\} \) be the sequence generated by PG in Algorithm [1]. Then \( \{M_k\} \) is bounded and any accumulation point \( M^* \) of \( \{M_k\} \) is a stationary point.

Algorithm 2 Solve (18) by PG with continuation trick.
Initialze: \( \rho > 0, \alpha = 0.99\rho, \eta > 1, M, t = 0, T > 0 \).
while \( t < T \) do
1) \( M = \text{PG}(M, \rho) \) by calling Algorithm [1];
2) \( \rho = \rho/\eta, \alpha = 0.99\rho; \)
3) \( t = t + 1 \).
end while

Though PG is guaranteed to converge, the obtained suboptimal solution to (18) may be far from optimal to problem (16) which is our original target. There are two important factors which may affect the quality of the obtained solution by PG. First, due to the nonconvexity of (18), the solution may be sensitive to the initialization of \( M \). Second, the smoothing parameter \( \rho > 0 \) should be small so that the objective \( f_\rho \) in (18) can well approximate the objective \( f \) in (16). However, if \( \rho \) is directly set to a very small value, PG may decrease the objective function value of (18) very slowly. This can be easily seen from the updating of \( M \) in (19), where \( \alpha < \rho \).
To address the above two issues, we use a continuation trick to find a better solution to (16) by solving (18) with different initializations. Namely, we begin with a relatively large value of \( \rho \) and reduce it gradually. For each fixed \( \rho \), we solve (18) by PG in Algorithm [1] and use its solution as a new initialization of \( M \) in PG. To achieve a better solution, we repeat the above procedure \( T \) times or until \( \rho \) reaches a predefined small value \( \rho_{min} \).

We summarize the procedure of PG with the continuation trick in Algorithm [2].

Finally, we would like to emphasize some advantages of our DCMC model (16) and the proposed solver. A main merit of our model (16) is that it minimizes the mutual coherence \( \mu(M) \) directly and thus the mutual coherence of its optimal solution can be low. Though the optimal solution is in general not computable due to the nonconvexity of (16), our proposed solver, which first smooths the objective and then minimizes it by PG, has convergence guarantee. To the best of our knowledge, this is the first work which directly minimizes the mutual coherence of a matrix with convergence guarantee.

III. LOW MUTUAL COHERENCE BASED PROJECTION

In this section, we show how to find a projection matrix \( P \) such that \( \mu(PD) \) can be as low as possible. This is crucial for signal recovery by CS associated to problem (3). Similar to the DCMC model shown in (16), an ideal way is to minimize \( \mu(PD) \) directly, i.e.,

\[
\min_{P \in \mathbb{R}^{m \times d}} ||(PD)^T (PD) - I||_\infty \\
\text{s.t. } ||PD_i||_1 = 1, \ i = 1, \ldots, n.
\] (22)

However, the constraint of (22) is more complex than the one in (16), and thus (22) is much more challenging to solve. We instead consider an approximate model of (22) based on the following observation.

Theorem 3: For any \( M_1, M_2 \in \mathbb{R}^{m \times n} \), if \( M_1 \rightarrow M_2 \), then \( \mu(M_1) \rightarrow \mu(M_2) \).

It is easy to prove the above result by the definition of the mutual coherence of a matrix. The above theorem indicates
Algorithm 3 Solve (24) by Alternating Minimization.

Initialize: $k = 0$, $P_k \in \mathbb{R}^{m \times d}$, $M_k \in \mathbb{R}^{m \times n}$, $\rho > 0$, $\alpha = 0.99\rho$, $\beta > 0$.

Output: $\{P^*, M^*\} = \text{AM}(M_k, P_k, \rho, \beta)$.

while $k < K$ do
1) Compute $V_k$ by solving (21);
2) Compute $M_{k+1}$ by solving (25);
3) Compute $P_{k+1}$ by solving (26);
4) $k = k + 1$.
end while

that the difference of the mutual coherences of two matrices is small when the difference of two matrices is small. This motivates us to find $M$ such that $\mu(M)$ is low and the difference between $M$ and PD is small. So we have the following approximate model of (22):

$$
\min_{P, M} \|M^T M - I\|_\infty + \frac{1}{2\beta} \|M - PD\|_F^2.
$$

where $\beta > 0$ trades off $\mu(M)$ and the difference between $M$ and PD. To distinguish from the DMCM model in (16), in this paper we name the above model as DMCM based Projection (DMCM-P).

Now we show how to solve (23). First, we smooth $\|M^T M - I\|_\infty$ as $f_\rho(M)$ defined in (17). Then problem (23) can be approximated by the following problem with a smooth objective:

$$
\min_{M, P} F(M, P) = f_\rho(M) + \frac{1}{2\beta} \|M - PD\|_F^2.
$$

where both $\rho$ and $\beta$ are small, $f_\rho$ is very close to $f$. So is $\mu(PD)$ to $\mu(M)$ because $\|M - PD\|_\infty$ has to be small. Thus solving problem (24) can still be regarded as minimizing the mutual coherence directly. We propose to alternately update $P$ and $M$ to solve problem (24).

1. Fix $P = P_k$ and update $M$ by

$$
M_{k+1} = \arg\min_M \langle \nabla f_\rho(M_k), M - M_k \rangle + \frac{1}{2\alpha} \|M - M_k\|_F^2 + \frac{1}{2\beta} \|M - PD\|_F^2.
$$

2. Fix $M = M_{k+1}$ and update $P$ by solving

$$
P_{k+1} = \arg\min_P \|M_{k+1} - PD\|_F^2.
$$

Algorithm 4 Solve (24) by AM with continuation trick.

Initialize: $\rho > 0$, $\alpha = 0.99\rho$, $\beta > 0$, $\eta > 1$, $M$, $P$, $t = 0$, $T > 0$.

while $t < T$ do
1) $(P, M) = \text{AM}(P, M, \rho, \beta)$ by calling Algorithm 3
2) $\rho = \rho/\eta$, $\alpha = 0.99\rho$;
3) $\beta = \beta/\eta$;
4) $t = t + 1$.
end while

which has a closed form solution $P = M_{k+1}D^\dagger$.

Iteratively updating $P$ by (26) and $M$ by (25) leads to the Alternating Minimization (AM) method for (24). We summarize the whole procedure of AM in Algorithm 3. It can be easily seen that the per-iteration cost of Algorithm 5 is $O((d + m)n^2 + n^3)$. We can prove that any accumulation point of AM is a stationary point of (24).

Theorem 4: Assume that $D$ in problem (24) is of full row rank. Let $\{(M_k, P_k)\}$ be the sequence generated by Algorithm 3. Then the following results hold:

(i) $F(M_k, P_k)$ is monotonically decreasing.
(ii) $\lim_{k \to \infty} M_k = M$.
(iii) The sequence $\{(M_k, P_k)\}$ is bounded.
(iv) Any accumulation point of $\{(M_k, P_k)\}$ is a stationary point.

The proof of Theorem 4 can be found in Appendix. Note that to guarantee the convergence of Algorithm 3 Theorem 4 requires $D$ in problem (24) to be of full row rank. Such an assumption usually holds in CS since $D \in \mathbb{R}^{d \times n}$ is an overcomplete dictionary with $d < n$.

Though AM is guaranteed to converge, the obtained solution to (24) may be far from optimal to problem (23) which is our original target. In order for (24) to approximate (23) well, $\rho > 0$ should be small. On the other hand, $\beta > 0$ should also be small such that the difference between $M$ and PD is small and thus $\mu(PD)$ can well approximate $\mu(M)$. Similar to Algorithm 3, we use a continuation trick to achieve a good solution to (24). Namely, we begin with a relatively large value of $\rho > 0$ and $\beta > 0$ and reduce them gradually. For each fixed pair $(\rho, \beta)$, we solve (24) by AM in Algorithm 3 and use its solution as a new initialization of $P$ and $M$ in AM. We repeat the procedure $T$ times or until $\rho$ and $\beta$ reach predefined small values $\rho_{\text{min}}$ and $\beta_{\text{min}}$. We summarize the procedure of AM with the continuation trick in Algorithm 4.

Finally, we would like to emphasize some advantages of our DMCM-P over previous methods. The main merit of our DMCM-P is that it is the first model which minimizes $\mu(PD)$ directly and the proposed solver also has convergence guarantee. The algorithms of Elad [9] and Xu et al. [10] are also mutual coherence based methods. But their objectives are suboptimal and their solvers lack convergence guarantee.

IV. Numerical Results

In this section, we conduct several experiments to verify the effectiveness of our proposed methods by comparing them with previous methods. The experiments consist of two parts.
The first part shows the values of mutual coherence. The second part shows the signal recovery errors in CS.

A. Comparing the Mutual Coherence

This subsection presents two experiments to show the effectiveness of DMCM and DMCM-P, respectively. In the first experiment, we show that our DMCM is able to construct a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ with lower mutual coherence than previous methods do. We compare DMCM with

- Random: random matrix whose elements are drawn independently from the standard normal distribution.
- Elad: the algorithm of Elad [9] with $\mathbf{D} = \mathbf{I}$.
- Xu: the algorithm of Xu et al. [10] with $\mathbf{D} = \mathbf{I}$.
- Duarte: the algorithm of Duarte-Carajalino and Sapiro [12] with $\mathbf{D} = \mathbf{I}$.
- Welch bound: the Welch bound [13] shown in (13).

Note that the compared algorithms of Elad [9], Xu et al. [10] and Duarte-Carajalino and Sapiro [12] were designed to find a projection $\mathbf{P}$ such that $\mathbf{M} = \mathbf{PD}$ has low mutual coherence. They can still be compared with our DMCM by setting $\mathbf{D}$ as the identity matrix $\mathbf{I}$.

To solve our DMCM model in (18), we run Algorithm 2 for 15 iterations and Algorithm 1 for 1000 iterations. In Algorithm 2 we set $\rho_0 = 0.5$ and $\eta = 1.2$. $\mathbf{M}$ is initialized as a Gaussian random matrix. In the method of Elad, we follow [9] to set $t = 0.2$ and $\gamma = 0.95$. In the method of Xu, we try multiple choices of the convex combination parameter $\alpha$ and set it as 0.5 which results in the lowest mutual coherence in most cases. The method of Duarte do not need special parameters. All the compared methods have the same random initializations of $\mathbf{P}$ (except Duarte, which has a closed form solution).

The compared methods are tested on three settings with different sizes of $\mathbf{M} \in \mathbb{R}^{m \times n}$: (1) $m = [6 : 2 : 16], n = 60$; (2) $m = [10 : 5 : 35], n = 120$; and (3) $m = [10 : 10 : 50], n = 180$. Note that the constructed matrices may not be the same for the compared methods with different initializations. So for each choice of size $(m, n)$, we repeat the experiment for 100 times and record the means and standard deviations of the mutual coherences of the constructed matrices $\mathbf{M}$. The means and standard deviations of mutual coherences v.s. the number $m$ of measurements are shown in Figure 1. It can be seen that the matrix constructed by our DMCM achieves much lower mutual coherences than previous methods do. The main reason is that our DMCM minimizes the mutual coherence of $\mathbf{M}$ directly, while the objectives of all the previous methods are indirect. It can also be seen that the standard deviations of our method is close to zero, while some other compared methods may not be stable in some cases. A possible reason is that the solver of our method has convergence guarantee, while other methods do not.
Mutual Coherence of $Xu$ et al. [10] and Duarte-Carajalino and Sapiro [12] on the $P$ iterations. In Algorithm 4, we set run Algorithm 4 for 15 iterations and Algorithm 3 for 1000 this experiment. To solve our DMCM-P model in (23), we report the results in Figure 3. It can be seen that our method

**TABLE I** Comparison of running time (in seconds) of DMCM-P, Elad, Xu and Duarte on problem (23) under different settings.

| $m$ | $d$ | $n$ | DMCM-P | Elad | Xu | Duarte |
|-----|-----|-----|--------|------|----|-------|
| 10  | 40  | 60  | 181    | 5    | 5  | 0.0023 |
| 20  | 60  | 120 | 582    | 8    | 8  | 0.004  |
| 10  | 30  | 60  | 838    | 14   | 12 | 0.004  |

For the second experiment in this subsection, we show that for given $D \in \mathbb{R}^{d \times n}$ our DMCM-P is able to compute a projection $P \in \mathbb{R}^{m \times d}$ such that $PD \in \mathbb{R}^{m \times n}$ has low mutual coherence. We choose $D$ to be a Gaussian random matrix in this experiment. To solve our DMCM-P model in (23), we run Algorithm 4 for 15 iterations and Algorithm 3 for 1000 iterations. In Algorithm 4, we set $\rho_0 = 0.5$, $\beta = 2$ and $\eta = 1.2$. $P$ is initialized as a Gaussian random matrix.

We compare our DMCM-P with the algorithms of Elad [9], Xu et al. [10] and Duarte-Carajalino and Sapiro [12] on the mutual coherence of $PD$. We test on three settings: (1) $m = [6 : 2 : 16]$, $n = 60$, $d = 30$; (2) $m = [10 : 5 : 35]$, $n = 120$, $d = 60$; and (3) $m = [10 : 10 : 50]$, $n = 180$, $d = 90$. Figure 2 shows the mutual coherence of $PD$ as a function of the number $m$ of measurements. It can be seen that our DMCM-P achieves the best projection such that $PD$ has the lowest mutual coherences in all the three settings. So are the standard deviations. Note that our algorithm does not use any special property of $D$. So it is expected to work for $D$ in other distributions as well. We test our method in the case that the elements of $D$ are uniformly distributed in $[0, 1]$ and report the results in Figure 3. It can be seen that our method still outperforms other methods in both mean and standard deviation.

Furthermore, Figure 4 shows the distribution of the absolute values of inner products between distinct columns of $PD$ with $m = 20$, $n = 120$, and $d = 60$. It can be seen that our DMCM-P has the shortest tail, showing that the number of elements in the Gram matrix that are closer to the ideal Welch bound is larger than the compared methods. Such a result is consistent with the lowest mutual coherences shown in Figure 2.

Finally, we report the running time of the algorithms of Elad, Xu, Duarte and our DMCM-P in Table I. The settings of the algorithms are the same as those in Figure 2 and the running time is reported based on different choices of $m$, $d$ and $n$. It can be seen that Duarte is the fastest method since it has a closed form solution. Our DMCM-P is not very efficient since we use the continuation trick in Algorithm 4 which repeats Algorithm 3 many times. Note that speeding up the algorithm, although valuable, is not the main focus of this paper. Actually, for many applications the projection matrix $P$ can be computed offline. So we leave the speeding-up issue as future work.

**B. Comparing the CS Performance**

In this subsection, we apply the optimized projection by our DMCM-P to CS. We first generate a $T$-sparse vector $x = D\alpha$, where $\alpha \in \mathbb{R}^d$. The locations of nonzeros are chosen randomly and their values obey a uniform distribution in $[-1, 1]$. We choose the dictionary $D \in \mathbb{R}^{d \times n}$ as a Gaussian random matrix. Then we apply different projection matrices $P$ learned by our DMCM-P, random projection matrix, and the algorithms of Elad [9], Xu et al. [10] and Duarte-Carajalino.
At last, we solve problem (5) by OMP to obtain \( \hat{\alpha} \). We compare the performance of projection matrices computed by different methods using the relative reconstruction error \( \|x - \hat{x}\|_2/\|x^*\|_2 \), where \( x^* \) is the ground truth. A smaller reconstruction error means better CS performance.

We conduct two experiments in this subsection. The first one changes the number \( m \) of measurements and the second one changes the sparsity level \( T \). For every value of the aforementioned parameters we perform 3000 experiments and calculate the average relative reconstruction error.

In the first experiment, we set \( m = [6 : 2 : 16] \), \( n = 60 \), \( d = 30 \) and \( T = 2 \). Figure 5 shows the average relative reconstruction error v.s. the number \( m \) of measurements (\( T \) is fixed). The CS performance improves as \( m \) increases. Also, as expected, all the optimized projection matrices produce better CS performance than the random projection does, and our proposed DMCM-P consistently outperforms the algorithms of Elad, Xu et al. and Duarte-Carajalino and Sapiro.

In the second experiment, we set \( m = 18 \), \( n = 180 \) and \( d = 90 \) and vary the sparsity level \( T \) from 1 to 6. Figure 6 shows the average relative reconstruction error as a function of the sparsity level \( T \) (\( m \) is fixed). The CS performance also improves as \( T \) decreases. Also, our DMCM-P consistently outperforms random projection and other deterministic projection optimization methods. This is due to the low mutual coherence of \( PD \) thanks to our optimized projection method as verified in the previous experiments.

We also test the noisy case. We add Gaussian random noise with 0 mean and 0.01 variance to each element of the observation \( y \) and then recover the true signal from this noisy \( y \). This time we test with \( D \) in another different distribution and another choice of the ratio \( n/d \). We generate elements of \( D \) by a uniform distribution on \([0,1]\). We choose \( m = [6 : 2 : 16] \), \( d = 40 \) and \( n = 60 \). Figure 7 shows the relative reconstruction error v.s. the number of measurements. It can be seen that our method also achieves the best performance in almost all cases.

V. CONCLUSIONS

This paper focuses on optimizing the projection matrix in CS for reconstructing signals which are sparse in some overcomplete dictionary. We develop the first model which aims to find a projection \( P \) by minimizing the mutual coherence of \( PD \) directly. We solve the nonconvex problem by alternating minimization and prove the convergence. Simulation results show that our method does achieve much lower mutual coherence of \( PD \), and also leads to better CS performance. Considering that mutual coherence is important in many applications besides CS, we expect that the proposed construction will be useful in many other applications as well, besides CS.

There is some interesting future work. First, though we give the first solver with convergence guarantee in Algorithm
for (16), the obtained solution is not guaranteed to be globally optimal due to the nonconvexity of the problem. It is interesting to investigate when the obtained solution is globally optimal. Second, it is valuable to find faster solvers. For example, we may consider solving (16) and (22) by Alternating Direction Method of Multiplier (ADMM) after introducing some auxiliary variables, which may be more efficient than our current solvers. But proving its convergence for nonconvex problems, (16) and (22), will be challenging.

APPENDIX

In this section, we give the proof of Theorem 4.

Definition 2: [17], [16] Let \( g \) be a proper and lower semicontinuous function.

1) For a given \( x \in \text{dom } g \), the Fréchet subdifferential of \( g \) at \( x \), written as \( \partial g(x) \), is the set of all vectors \( u \in \mathbb{R}^n \) which satisfies

\[
\liminf_{y \to x, y \neq x} \frac{g(y) - g(x) - \langle u, y - x \rangle}{\|y - x\|} \geq 0.
\]

2) The limiting-subdifferential, or simply the subdifferential, of \( g \) at \( x \in \mathbb{R}^n \), written as \( \partial g(x) \), is defined through the following closure process

\[
\partial g(x) := \{ u \in \mathbb{R}^n : \exists x_k \to x, g(x_k) \to g(x), \ u_k \in \partial g(x_k) \to u, k \to \infty \}.
\]

Proposition 1: [17], [16] The following results hold:

1) In the nonsmooth context, the Fermat’s rule remains unchanged: If \( x \in \mathbb{R}^n \) is a local minimizer of \( g \), then

\[
0 \in \partial g(x).
\]

2) Let \((x_k, u_k)\) be a sequence such that \(x_k \to x, u_k \to u, g(x_k) \to g(x)\) and \(u_k \in \partial g(x_k)\). Then \(u \in \partial g(x)\).

3) If \( f \) is a continuously differentiable function, then \( \partial (f + g)(x) = \nabla f(x) + \partial g(x) \).

Proof of Theorem 4: First, we define

\[
h(M) = \begin{cases} 0, & \text{if } \|M\|_2 = 1, \ i = 1, \cdots, n, \\ \infty, & \text{otherwise.} \end{cases}
\]

Then (25) can be rewritten as

\[
M_{k+1} = \arg\min_M \langle \nabla f_\rho(M_k), M - M_k \rangle + \frac{1}{2\alpha} \|M - M_k\|_F^2 + \frac{1}{2\beta} \|M - P_k D\|_F^2 + h(M).
\]

By the optimality of \( M_{k+1} \), we have

\[
h(M_{k+1}) + \langle \nabla f_\rho(M_k), M_{k+1} - M_k \rangle + \frac{1}{2\alpha} \|M_{k+1} - M_k\|_F^2 + \frac{1}{2\beta} \|M_{k+1} - P_k D\|_F^2.
\]

h(M_k) + \frac{1}{2\beta} \|M_k - P_k D\|_F^2,
\]

and

\[
0 \in \partial h(M_{k+1}) + \nabla f_\rho(M_k) + \frac{1}{\alpha}(M_{k+1} - M_k) + \frac{1}{\beta}(M_{k+1} - P_k D).
\]

From the Lipschitz continuity of \( \nabla f_\rho(M) \), we have

\[
F(M_{k+1}, P_k) = f_\rho(M_{k+1}) + \frac{1}{2\beta} \|M_{k+1} - P_k D\|_F^2
\]

\[
\leq f_\rho(M_k) + \langle \nabla f_\rho(M_k), M_{k+1} - M_k \rangle + \frac{1}{2\beta} \|M_{k+1} - P_k D\|_F^2.
\]

Add (28) and (29), we have

\[
h(M_{k+1}) + F(M_{k+1}, P_k)
\]

\[
\leq h(M_k) + f_\rho(M_k) + \left(\frac{1}{2\alpha} - \frac{1}{2\beta}\right) \|M_{k+1} - M_k\|_F^2
\]

\[
+ \frac{1}{2\beta} \|M_k - P_k D\|_F^2.
\]

Second, from the optimality of \( P_{k+1} \) to problem (26), we have

\[
F(M_{k+1}, P_{k+1}) \leq F(M_{k+1}, P_k),
\]

and

\[
0 = \nabla P F(M_{k+1}, P_{k+1}) = (M_{k+1} - P_{k+1} D)D^T.
\]

By the assumption that \( D \) is of full row rank, (32) implies that

\[
P_{k+1} = M_{k+1}D^T(DD^T)^{-1}.
\]

Combining (30) and (31) leads to

\[
h(M_{k+1}) + F(M_{k+1}, P_{k+1})
\]

\[
\leq h(M_k) + F(M_k, P_k) + \left(\frac{1}{2\alpha} - \frac{1}{2\beta}\right) \|M_{k+1} - M_k\|_F^2.
\]

So \( h(M_k) + F(M_k, P_k) \) and \( F(M_k, P_k) \) are monotonically decreasing. Summing all the above inequalities for \( k \geq 0 \), it follows that

\[
h(M_0) + F(M_0, P_0)
\]

\[
\geq \sum_{k=0}^{+\infty} \left(\frac{1}{2\alpha} - \frac{1}{2\beta}\right) \|M_{k+1} - M_k\|_F^2
\]

\[
\geq 0.
\]

This implies that \( M_{k+1} - M_k \to 0 \). Hence \( P_{k+1} - P_k \to 0 \) by using (33).

Third, note that \( F(M, P) \) is coercive, i.e., \( F(M, P) \) is bounded from below and \( F(M, P) \to +\infty \) when \( \|M, P\|_F \to +\infty \). It can be seen from (34) that \( F(M_k, P_k) \) is bounded. Thus \( \{M_k, P_k\} \) is bounded. Then there exists an accumulation point \( (M^*, P^*) \) and a subsequence \( \{M_{k_j}, P_{k_j}\} \) such that \( (M_{k_j}, P_{k_j}) \to (M^*, P^*) \) as \( j \to +\infty \). Now we prove that \( (M^*, P^*) \) is a stationary point.
From (29) we have

\[
\begin{align*}
\nabla f_\rho(M_{k+1}) - \nabla f_\rho(M_k) - \frac{1}{\alpha}(M_{k+1} - M_k) \\
- \frac{1}{\beta}(P_{k+1} - P_k)D \\
= - \frac{1}{\alpha}(M_{k+1} - M_k) - \nabla f_\rho(M_k) - \frac{1}{\beta}(M_{k+1} - P_k)D \\
+ \frac{1}{\beta}(M_{k+1} - P_{k+1})D \\
\leq \partial h(M_{k+1}) + \nabla f_\rho(M_{k+1}) + \frac{1}{\beta}(M_{k+1} - P_{k+1})D \\
= \partial h(M_{k+1}) + \nabla M F(M_{k+1}, P_{k+1}).
\end{align*}
\]

Thus

\[
\|\nabla f_\rho(M_{k+1}) - \nabla f_\rho(M_k) - \frac{1}{\alpha}(M_{k+1} - M_k) \\
- \frac{1}{\beta}(P_{k+1} - P_k)D\|_F \\
\leq \|\nabla f_\rho(M_{k+1}) - \nabla f_\rho(M_k)\|_F + \frac{1}{\alpha}\|M_{k+1} - M_k\|_F \\
+ \frac{1}{\beta}\|(P_{k+1} - P_k)D\|_F \\
\leq \frac{1}{\alpha}\|M_{k+1} - M_k\|_F + \frac{1}{\alpha}\|M_{k+1} - M_k\|_F \\
+ \frac{1}{\beta}\|P_{k+1} - P_k\|_F\|D\|_F. \\
\rightarrow 0.
\]

(37)

Since \(F(M, P)\) is continuously differentiable, we have \(F(M_{k+1}, P_{k+1}) \rightarrow F(M^*, P^*)\). As \(h(M_k) = 0\) for all \(k\) and the set \(\{M: \|M\|_2 = 1, i = 1, \cdots, n\}\) is closed, we have \(h(M^*) = 0\) and \(F(M_k, P_k) + h(M_{k+1}) \rightarrow F(M^*, P^*) + h(M^*)\). Combing (32), (36), (37) and using Proposition 1 we have

\[
0 \in \partial h(M^*) + \nabla F(M^*, P^*). \\
\]

Thus \((M^*, P^*)\) is a stationary point.

REFERENCES

[1] Emmanuel J Candès, Justin Romberg, and Terence Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, 2006.

[2] David L Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] Balas Kausik Natarajan, “Sparse approximate solutions to linear systems,” SIAM Journal on Computing, vol. 24, no. 2, pp. 227–234, 1995.

[4] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, “Atomic decomposition by basis pursuit,” SIAM Journal on Scientific Computing, vol. 20, no. 1, pp. 33–61, 1998.

[5] Yagvensh Chandra Pati, Ramin Rezaifar, and PS Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seven Aislomar Conference on. IEEE, 1993, pp. 40–44.

[6] Rémi Gribonval and Morten Nielsen, “Sparse representations in unions of bases,” IEEE Transactions on Information Theory, vol. 49, no. 12, pp. 3320–3325, 2003.

[7] David L Donoho and Michael Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via \(\ell_1\) minimization,” Proceedings of the National Academy of Sciences, vol. 100, no. 5, pp. 2197–2202, 2003.

[8] Joel A Tropp, “Greed is good: Algorithmic results for sparse approximation,” IEEE Transactions on Information Theory, vol. 50, no. 10, pp. 2231–2242, 2004.

[9] Michael Elad, “Optimized projections for compressed sensing,” IEEE Transactions on Signal Processing, vol. 55, no. 12, pp. 5695–5702, 2007.

[10] Jianping Xu, Yiming Pi, and Zongjie Cao, “Optimized projection matrix for compressive sensing,” EURASIP Journal on Advances in Signal Processing, vol. 2010, pp. 43, 2010.

[11] Emmanuel Candes and Terence Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies,” IEEE Transactions on Information Theory, vol. 52, no. 12, pp. 5406–5425, 2006.

[12] Julio Martin Duarte-Carvajalino and Guillermo Sapiro, “Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization,” IEEE Transactions on Image Processing, vol. 18, no. 7, pp. 1395–1408, 2009.

[13] Lloyd Welch, “Lower bounds on the maximum cross correlation of signals (corresp.),” IEEE Transactions on Information Theory, pp. 397–399, 1974.

[14] Yuri Nesterov, “Smooth minimization of non-smooth functions,” Mathematical Programming, vol. 103, no. 1, pp. 127–152, 2005.

[15] Yoram Singer John Duchi, Shai Shalev-Shwartz and Tushar Chandra, “Efficient projections onto the \(\ell_1\)-ball for learning in high dimensions,” in International Conference of Machine Learning, 2008.

[16] Shoham Sabach Jérôme Bolte and Marc Teboulle, “Proximal alternating linearized minimization for nonconvex and nonsmooth problems,” Mathematical Programming, vol. 146, no. 1-2, pp. 459–494, 2014.

[17] Rockafellar R. Tyrrell and Wets. Roger, Eds., Variational Analysis, Springer, 1998.

Canyi Lu is currently a Ph.D. student with the Department of Electrical and Computer Engineering at the National University of Singapore. His current research interests include computer vision, machine learning, pattern recognition and optimization. He was the winner of the Microsoft Research Asia Fellowship 2014.

Zhouchen Lin received the Ph.D. degree in Applied Mathematics from Peking University in 2000. He is currently a Professor at Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a Guest Professor at Beijing Jiaotong University. Before March 2012, he was a Lead Researcher at Visual Computing Group, Microsoft Research Asia. He was a Chair Professor at Northeast Normal University, a Guest Professor at Shanghai Jiao Tong University and Southeast University, and a Guest Researcher at Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer vision, machine learning, pattern recognition, and numerical optimization. He is an Associate Editor of IEEE Trans. Pattern Analysis and Machine Intelligence and International J. Computer Vision, an area chair of CVPR 2014, ICCV 2015, NIPS 2015, AAAI 2016, CVPR 2016, and IJCAI 2016, and a Senior Member of the IEEE.
Huan Li received the bachelor degree in medical information from Central South University in 2011 and the master degree in software engineering from Peking University in 2014. He is currently a Ph.D. student at Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. His research interests include optimization and machine learning.