Hybrid Projection-Reflection Method for Phase Retrieval

Heinz H. Bauschke
Department of Mathematics and Statistics, University of Guelph
Guelph, Ontario N1G 2W1, Canada.

Patrick L. Combettes
Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie – Paris 6
75005 Paris, France.

D. Russell Luke
The Pacific Institute for the Mathematical Sciences, Simon Fraser University
Burnaby, British Columbia V5A 1S6, Canada.

Abstract

The phase retrieval problem, fundamental in applied physics and engineering, asks to determine the phase of a complex-valued function from modulus data and additional *a priori* information. Recently, we identified two important methods for phase retrieval, namely Fienup's Basic Input-Output (BIO) and Hybrid Input-Output (HIO) algorithms, with classical convex projection methods and suggested that further connections between convex optimization and phase retrieval should be explored. Following up on this work, we introduce a new projection-based method, termed the Hybrid Projection Reflection (HPR) algorithm, for solving phase retrieval problems featuring nonnegativity constraints in the object domain. Motivated by properties of the HPR algorithm for convex constraints, we recommend an error measure studied by Fienup more than twenty years ago. This error measure, which has received little attention in the literature, lends itself to an easily implementable stopping criterion. In numerical experiments, we found the HPR algorithm to be a competitive alternative to the HIO algorithm and the stopping criterion to be reliable and robust.

1 Introduction

The best known and most widely used projection-based methods for phase retrieval are the *Error Reduction (ER)* algorithm, that derived from the pioneering work of Gerchberg and Saxton [1], and Fienup's *Basic Input-Output (BIO)* and *Hybrid Input-Output (HIO)* algorithms [2]. In the mid-1980s Levi and Stark [3, 4] identified the ER algorithm as a *nonconvex* application of a sequential projection algorithm known in convex settings as *Projections Onto Convex Sets* or *POCS* [5, 6]. In the same spirit, we recently presented new interpretations of Fienup's algorithms: the BIO and HIO algorithms turn out to be nonconvex instances of *Dykstra's* [7] and the *Douglas-Rachford* [8] projection algorithm, respectively [9]. The POCS, Dykstra, and Douglas-Rachford projection algorithms are governed by iterations of the form

$$x_{n+1} = T(x_n), \tag{1}$$

where x_n is the current estimate of the function to be reconstructed and T is an operator that utilizes in some fashion the projectors onto the constraint sets. The convergence behavior of these algorithms is well-understood provided that all constraint sets are convex. While convergence results do not carry over from the convex setting to the nonconvex setting of phase retrieval, it is nonetheless useful to make this connection. Indeed, it not only provides different views and insights on known phase retrieval methods but, more importantly, it also suggests a systematic way of experimenting with novel algorithmic structures by the formal adaption of convex optimization algorithms to phase retrieval.

We model a *signal* as a function $x : \mathbb{X} \to \mathbb{Y}$, where $\mathbb{X}$ is either $\mathbb{R}^N$ or $\mathbb{Z}^N$, corresponding to analog or discrete signals, respectively, and where $\mathbb{Y}$ is either $\mathbb{R}$ or $\mathbb{C}$, depending on whether x is real- or complex-valued. We denote the original object, or signal, to be recovered by x_*. In the analysis of [9] the setting was $\mathbb{Y} = \mathbb{C}$. In many important applications (e.g. optical interferometry astronomy and crystallography) the original signal x_* is real-valued and nonnegative [10, 11]. The algorithms discussed here are limited to these applications. Hence, throughout this work, signals are viewed as points in $\mathcal{L}$, the Hilbert space of square integrable functions from $\mathbb{X}$ to $\mathbb{R}$. Our notation is the same as in Ref. [9]. We denote by P_X an arbitrary but fixed selection of the possibly multi-valued *projector* onto a subset X of $\mathcal{L}$. By definition, the operator $R_X = 2P_X - I$, where I is the identity operator on $\mathcal{L}$, is the *reflector* with respect to X; in other words, for every $x \in \mathcal{L}$, $P_X(x)$ is the midpoint between x and $R_X(x)$.

Many signal recovery problems can be formulated mathematically as a feasibility problem of the form

$$\text{find } \; x \in S \cap M, \tag{2}$$

where S is the set of signals in $\mathcal{L}$ that satisfy object domain constraints and M is the set of signals in $\mathcal{L}$ that satisfy Fourier domain constraints [5, 13].

In phase retrieval, M is the set of signals whose Fourier modulus agrees with some measured data m, namely

$$M = \big\{ y \in \mathcal{L} \mid \; |\widehat{y}| = m \big\}, \tag{3}$$

where $\widehat{y}$ denotes the Fourier transform of y. For the applications we consider here, the data $m : \mathbb{X} \to \mathbb{R}$ is a measurement of the magnitude of the Fourier transform of the original signal x_*,

thus m is a nonnegative even function. The projection of a signal $x \in \mathcal{L}$ onto the Fourier magnitude constraint set M, $P_M x$, is computed by replacing the magnitude of the Fourier transform of x by the known magnitude data m and inverse transforming the result:

$$P_M(x) = \mathcal{F}^{-1}(\widehat{y_0}), \tag{4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform and $\widehat{y_0}$, a particular selection of the multi-valued Fourier domain projection, is defined by

$$\widehat{y_0}(\omega) = \begin{cases} m(\omega)\dfrac{\widehat{x}(\omega)}{|\widehat{x}(\omega)|}, & \text{if } \widehat{x}(\omega) \neq 0; \\ m(\omega), & \text{otherwise.} \end{cases} \tag{5}$$

For further discussion of this projector see Example 3.6 of Ref. [9]. Note that, if x is real, then $\widehat{x}$ is Hermitian (conjugate symmetric). Since the data m is an even function, our choice of selection $\widehat{y_0}$ also has this property; thus $P_M(x)$ is real-valued, that is, the projector P_M necessarily maps the space of real-valued functions $\mathcal{L}$ to itself.

Recalling the original signal x_*, let the set $D \subset \mathbb{X}$ satisfy

$$\{t \in \mathbb{X} \mid x_*(t) \neq 0\} \subset D. \tag{6}$$

Let us denote the complement of D by $\complement D$, and by $1_{\complement D}$ the function that takes on the value 0 on D and the value 1 on $\complement D$. Critical to the analysis of Ref. [9] is the fact that the object domain constraint consists only of support information, namely

$$S = \big\{y \in \mathcal{L} \mid \ y \cdot 1_{\complement D} = 0\big\}. \tag{7}$$

We emphasize that S is a closed convex set (in fact, a linear subspace – see [6] or Example 3.14(i) of Ref. [9]), whereas M defined by Eq.(3) is closed but not convex (see Example 3.15 and Remark [111] of Ref. [9]). As shown in Example 3.14(i) of Ref. [9], the projection of a signal $x \in \mathcal{L}$ onto S is given by

$$P_S(x) = x \cdot 1_D. \tag{8}$$

In Section 5 of Ref. [9], the BIO and HIO algorithms with $\beta = 1$ were shown to fit the structure of Eq.(1) with $T = P_S P_M + I - P_M$ and $T = P_S(2P_M - I) + I - P_M$, respectively. The ER algorithm can also be put in this form by taking $T = P_S P_M$ [3]. We show below that such closed-form representations are not always easily obtained.

For the phase retrieval problems considered here, support information alone is insufficient for satisfactory signal recovery, hence additional object domain constraints are utilized. Since the original signal x_* is real-valued and nonnegative, these properties are included in the problem formulation by defining the object domain constraint set S_+ as

$$S_+ = \big\{y \in \mathcal{L} \mid \ y \cdot 1_{\complement D} = 0 \ \text{ and } \ y \geq 0\big\}. \tag{9}$$

Arguing as in Example 3.14(i)&(iii) of Ref. [9], the projection of a signal $x \in \mathcal{L}$ onto S_+ is given by

$$(\forall t \in \mathbb{X}) \ \ \big(P_{S_+}(x)\big)(t) = \begin{cases} \max\{0, x(t)\}, & \text{if } t \in D; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Altogether, the feasibility problem Eq.(2) becomes

$$\text{find } \; x \in S_+ \cap M. \tag{11}$$

While the ER algorithm for Eq.(11) can immediately be put in the form of Eq.(1) by setting $T = P_{S_+} P_M$, we have not been able to find corresponding reformulations of the BIO and HIO algorithms. Indeed, while in this case the set S_+ is still closed and convex (see Example 3.14 of Ref. [9]), the correspondences between Eqs. (19) and (20) of Ref. [9] on the one hand, and Eqs. (21) and (23) of Ref. [9] on the other hand do not hold here because key properties of the object domain projector are lost: most notably, P_S is a linear operator, but P_{S_+} is not. However, in an effort to formalize the HIO algorithm for problem Eq.(11) as a fixed point scheme of type Eq.(1), we arrive at a new algorithm for phase retrieval. This method, which we call the *Hybrid Projection Reflection (HPR)* algorithm, is a relaxation of the Douglas-Rachford projection algorithm and is therefore well-motivated by its counterpart in convex optimization. Moreover, based on our numerical results, the HPR algorithm appears to be a competitive alternative to the HIO algorithm.

The remainder of the paper is organized as follows. In Section 2, we review the HIO algorithm. The new algorithm (HPR) is introduced in Section 3 and Section 4 is devoted to numerical simulations.

2 Fienup's HIO algorithm

Let $x_0 \in \mathcal{L}$ be an initial guess and β be a fixed real relaxation parameter [14]. Fienup's HIO algorithm [2] generates a sequence of signals $(x_n)_{n\in\mathbb{N}}$ recursively as follows. Given a current iterate $x_n \in \mathcal{L}$, define

$$\gamma_n = \big\{ t \in \mathbb{X} \mid P_M(x_n) \text{ violates the object domain constraint at } t \big\}. \tag{12}$$

The next iterate x_{n+1} is given by

$$(\forall t \in \mathbb{X}) \;\; x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \notin \gamma_n; \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{if } t \in \gamma_n. \end{cases} \tag{13}$$

We show below that the form of the object domain constraint is critical to the analysis and implementation of HIO. To illustrate this we specialize the HIO algorithm to the cases in which the object domain contains: (A) only a support constraint; and (B) both support and nonnegativity constraints.

A. Support constraint only

If the object domain constraint set incorporates only a support constraint (as in Eq.(7)) and the zero crossings of $P_M(x_n)$ outside of D are assumed to be negligible [15], then Algorithm (13) can be rewritten as

$$(\forall t \in \mathbb{X}) \;\; x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \in D; \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases} \tag{14}$$

This formulation of the HIO algorithm leads to the following convenient expression in terms of reflectors and projectors (recall that $R_S = 2P_S - I$ and that $R_M = 2P_M - I$):

Proposition 1 Algorithm (14) is equivalent to

$$x_{n+1} = \tfrac{1}{2}\big(R_S(R_M + (\beta-1)P_M) + I + (1-\beta)P_M\big)(x_n). \tag{15}$$

Proof. For every $n \geq 0$, Eq.(14) yields

$$\begin{aligned}
x_{n+1} &= 1_D \cdot P_M(x_n) + 1_{\complement D} \cdot (x_n - \beta P_M x_n) \\
&= 1_D \cdot P_M(x_n) + (1 - 1_D) \cdot (x_n - \beta P_M x_n) \\
&= 1_D \cdot \big((1+\beta)P_M - I\big)(x_n) + x_n - \beta P_M(x_n) \\
&= P_S\big((1+\beta)P_M - I\big)(x_n) + x_n - \beta P_M(x_n) \\
&= \tfrac{1}{2}\big((2P_S - I)((1+\beta)P_M - I) + I + (1-\beta)P_M\big)(x_n) \\
&= \tfrac{1}{2}\big(R_S(R_M + (\beta-1)P_M) + I + (1-\beta)P_M\big)(x_n),
\end{aligned}$$

where the fourth equality follows from Eq.(8). □

Proposition 1 is an extension of Observation 5.10 of Ref. [9]. Indeed, if $\beta = 1$, then Algorithm (14) is equivalent to the following nonconvex version of the Douglas-Rachford projection algorithm:

$$x_{n+1} = \tfrac{1}{2}\big(R_S R_M + I\big)(x_n). \tag{16}$$

B. Support and nonnegativity constraints

We now turn to the situation in which the object domain constraint is given by the set S_+ of Eq.(9). As in the previous case, again assuming that the zero crossings of $P_M(x_n)$ outside of D are negligible, the HIO algorithm Eq.(13) becomes

$$(\forall t \in \mathbb{X})\ \ x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \in D \text{ and } \big(P_M(x_n)\big)(t) \geq 0; \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases} \tag{17}$$

While the HIO algorithm without nonnegativity constraint Eq.(14) can be reformulated in terms of projectors and reflectors as in Eq.(15), the nonlinearity of P_{S_+} — which is due to the nonnegativity constraint in Eq.(9) — appears to make a corresponding reformulation of Eq.(17) impossible.

3 The Hybrid Projection Reflection (HPR) algorithm

In this section, we derive a new phase retrieval algorithm based on the following two principles. Firstly, the spirit of the method should be close to that of the HIO algorithm, whose effectiveness is

well-established. Secondly, it should bear a strong connection with a convex optimization algorithm. Eq.(16) suggests the use of the Douglas-Rachford projection scheme as an algorithmic model. For the sets S_+ and M, the Douglas-Rachford algorithm is simply

$$x_{n+1} = \tfrac{1}{2}\big(R_{S_+}R_M + I\big)(x_n). \tag{18}$$

As with the HIO algorithm, it is desirable to incorporate a relaxation parameter for added flexibility. This leads to the following analog to Eq.(15),

$$x_{n+1} = \tfrac{1}{2}\big(R_{S_+}(R_M + (\beta-1)P_M) + I + (1-\beta)P_M\big)(x_n). \tag{19}$$

Next, we unravel Eq.(19) into an algorithmic prescription analogous to Eq.(17). Given an arbitrary signal $z \in \mathcal{L}$, let $z^+ = \max\{z, 0\}$ and $z^- = \min\{z, 0\}$ be its positive and negative parts, respectively. Then the following relations hold:

$$\begin{aligned}
x_{n+1} &= \tfrac{1}{2}\big(R_{S_+}(R_M + (\beta-1)P_M) + I + (1-\beta)P_M\big)(x_n) \\
&= \tfrac{1}{2}\big((2P_{S_+} - I)((1+\beta)P_M - I) + I + (1-\beta)P_M\big)(x_n) \\
&= P_{S_+}\big((1+\beta)P_M - I\big)(x_n) + x_n - \beta P_M(x_n) \\
&= \big[1_D \cdot \big((1+\beta)P_M - I\big)(x_n)\big]^+ + x_n - \beta P_M(x_n) \\
&= 1_D \cdot P_M(x_n) + 1_{\complement D} \cdot (x_n - \beta P_M x_n) - \big[1_D \cdot \big((1+\beta)P_M - I\big)(x_n)\big]^-.
\end{aligned} \tag{20}$$

The fourth equality follows from (10) and the last equality uses the identity $1 - 1_D = 1_{\complement D}$ together with the fact that $z = z^+ + z^-$ where $z = 1_D \cdot \big((1+\beta)P_M - I\big)(x_n)$. We now consider two cases:

(i) if $t \in D$, then Eq.(20) yields

$$\begin{aligned}
x_{n+1}(t) &= \big(P_M(x_n)\big)(t) - \big[\big((1+\beta)P_M - I\big)(x_n)\big]^-(t) \\
&= \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } (1+\beta)\big(P_M(x_n)\big)(t) \geq x_n(t); \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases}
\end{aligned} \tag{21}$$

(ii) if $t \notin D$, then Eq.(20) yields directly

$$x_{n+1}(t) = x_n(t) - \beta\big(P_M(x_n)\big)(t). \tag{22}$$

The following formal statement of the Hybrid Projection Reflection algorithm summarizes these results.

HPR Algorithm. Starting with an arbitrary signal $x_0 \in \mathcal{L}$, the Hybrid Projection Reflection (HPR) algorithm is governed by the recursion

$$(\forall t \in \mathbb{X})\ \ x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \in D \text{ and} \\ & \big(R_M(x_n)\big)(t) \geq (1-\beta)\big(P_M(x_n)\big)(t); \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases} \tag{23}$$

We have thus proven the following analog to Proposition 1.

Proposition 2 Algorithm (23) is equivalent to the recursion Eq.(19).

For the case $\beta = 1$ the recursion Eq.(23) specializes to

$$(\forall t \in \mathbb{X})\ \ x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \in D \text{ and } \big(R_M(x_n)\big)(t) \geq 0; \\ x_n(t) - \big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases} \tag{24}$$

Note that $x_{n+1}(t)$ is $\big(P_M(x_n)\big)(t)$ or $x_n(t) - \big(P_M(x_n)\big)(t)$, exactly as in the HIO algorithm Eq.(17) for $\beta = 1$. The crucial difference is the condition used to determine when, for $t \in D$, the update rule $x_{n+1}(t) = \big(P_M(x_n)\big)(t)$ applies: in Eq.(17), the requirement of the HIO algorithm (with $\beta = 1$) is

$$\big(P_M(x_n)\big)(t) \geq 0, \tag{25}$$

whereas in Eq.(24) the update rule depends on

$$\big(R_M(x_n)\big)(t) \geq 0. \tag{26}$$

4 HIO and HPR: a comparison.

We compare the HPR algorithm to the HIO algorithm, which is the leading iterative transform algorithm for the type of problem considered in this paper. The error reduction (ER) algorithm is not as effective as HIO for these problems and the Basic Input-Output (BIO) algorithm is seldom used as it does not work nearly as well as HIO. For our evaluation we use the two standard digital images shown in Figure 1(a)&(d).

A. Error measure

Before describing the details of our comparison, a few remarks about performance measures are necessary. Any meaningful and quantifiable comparison requires a performance measure. This is a difficult topic in the signal recovery literature. Often times what is measured in phase retrieval is the normalized root mean squared error between the magnitude of the Fourier transform of the current iterate and the Fourier magnitude data m, namely

$$E_{\mathrm{RMS}}(x_n) = \frac{\big\| \, |\widehat{x_n}| - m \big\|}{\|m\|}, \tag{27}$$

where m is assumed to be nonzero throughout this section [16]. The convex theory that guides our investigations suggests a different performance measure which we found in practice to be quite robust and reliable. This is described next.

Given an initial guess $x_0 \in \mathcal{L}$ and a parameter $\beta \in \mathbb{R}$, let $(x_n)_{n\in\mathbb{N}}$ be the sequence generated by the HPR algorithm (Algorithm (23)). The main convergence result in the convex case (see Fact 5.9 of Ref. [9]) suggests that the pertinent sequence to monitor is [17]

$$\big(P_M(x_n)\big)_{n\in\mathbb{N}} \tag{28}$$

rather than $(x_n)_{n\in\mathbb{N}}$. Alternatively, since one is often more interested in signals satisfying the object domain constraint, we recommend to monitor the sequence

$$\big(P_{S_+}\big(P_M(x_n)\big)\big)_{n\in\mathbb{N}}. \tag{29}$$

Since each signal $P_M(x_n)$ satisfies the Fourier modulus constraint and since our aim is to solve Eq.(11), i.e., to find a point in $S_+ \cap M$, it is natural to monitor the squared distance from the signal $P_M(x_n)$ to the set S_+, i.e., $\|P_{S_+}\big(P_M(x_n)\big) - P_M(x_n)\|^2$. In fact, Fienup originally suggested this quantity as an error measure for the HIO algorithm; see end of Section V on page 2764 of Ref. [2] and remark [18]. Following the standard practice in numerical analysis of relativizing the error and recalling that $\|P_M(x_n)\| = \|m\|$, we define the error measure at the nth iteration as follows

$$E_{S_+}(x_n) = \frac{\big\|P_{S_+}\big(P_M(x_n)\big) - P_M(x_n)\big\|^2}{\big\|P_M(x_n)\big\|^2} = \frac{\big\|P_{S_+}\big(P_M(x_n)\big) - P_M(x_n)\big\|^2}{\|m\|^2}. \tag{30}$$

In words, $E_{S_+}(x_n)$ measures the square of the normalized distance from $P_M(x_n)$ to S_+. Using this error measure, we compare the performance of the HPR algorithm to that of HIO for the two original images shown in Figure 1(a)&(d).

B. Initialization

We study the performance of the algorithms under noiseless and noisy settings. For noiseless data, we initialize the algorithms as in [20]. More specifically, the initial points x_0 are chosen to be random pixel magnitude maps that satisfy the object domain constraint, i.e., $x_0 \in S_+$. Such an image is shown in Figure 1(f). For noisy data, we take the initial guess to be simply the characteristic function of the support constraint shown in Figure 1(c). For both noiseless and noisy experiments, the initial guess is normalized so as to have the same energy as the known Fourier magnitude data, i.e., $\|\widehat{x_0}\| = \|m\|$.

C. Noiseless data

The data consists of the support/nonnegativity constraint shown in Figure 1(c) and the Fourier magnitude data shown in Figure 1(b)&(e). To better illustrate the differences in the algorithms, we have made the problem more difficult than it normally would be by making the support constraint larger than necessary. For more reasonable strategies for choosing the support constraint in practical applications, see Refs. [20, 21].

We select 100 random initial points x_0 generated by the method described in Section 4.B and use them to compare the performance of HIO and HPR (the same 100 random initial points are used for each algorithm). Note that these initial points have random phases in the Fourier domain, as well as random Fourier magnitudes. Moreover, since x_0 is real-valued and the data m is a real nonnegative even function, $P_M x_0$ and all subsequent iterates are also real-valued up to machine error. To compensate for limited machine precision, we take only the real part of the computed projection, that is, we compute $\mathrm{Re}(P_M x_n)$. First, we compare the mean asymptotic behavior

(1000 iterations) of two sets of realizations of the HPR and HIO algorithms, each corresponding to different values of the relaxation parameter, $\beta = 0.75$ and $\beta = 1.0$. The average value of the error metric at iteration n, $E_{S_+}(x_n)$, is shown in Figure 2 and Figure 3. These are all given in decibels (recall that the decibel value of $\alpha > 0$ is $10\log_{10}(\alpha)$).

Figure 2 and Figure 3 show the relative progress of the algorithms with respect to the error measure Eq.(30). It is clear from these graphs that the HIO algorithm reaches its smallest error value more rapidly than the HPR algorithm. Indeed, the distance from $P_M(x_n)$ to S_+ increases in the first few iterations. This is due to the path that the HPR algorithm takes toward the solution. Every trial of the HPR algorithm behaved in this manner, though ultimately HPR reduced the error metric well below that of the HIO algorithm. As we show in part D of this section, in the presence of noise, the HPR algorithm did not always reduce the value of the error metric below that achieved by the HIO algorithm. Nevertheless, in both noiseless and noisy experiments, the HPR algorithm consistently delivered superior picture quality. While we have found the error metric Eq.(30) to be a reliable indicator of picture quality, this is only relative to the algorithm it is being used to monitor. In other words, the quality of an image with $E_{S_+} = -50$ dB will not be as good as that of an image with $E_{S_+} = -55$ dB *when both images are generated from the same algorithm.* In contrast to this, an iterate of the HIO algorithm with $E_{S_+} = -55$ dB will not necessarily look better than an iterate of the HPR algorithm with $E_{S_+} = -53$ dB. In fact we have observed that it was often the case that images produced by the HPR algorithm still *looked* better than those produced by the HIO algorithm, whose error value were a few percent lower. This is an artifact of the fit of the error metric to the algorithm.

We now turn to qualitative image reconstruction results. The more than twenty years of experience that has accumulated with HIO indicates that user intervention is often necessary to achieve satisfactory results. Fienup and Wackerman [22] study several strategies to ensure that HIO reaches an acceptable stagnation point. These strategies can also be used with HPR. However, the HPR algorithm appears to be very robust and, in our trials, reaches good stagnation points *without any user intervention or case-specific heuristics.* In that sense, HPR can be described as an autonomous algorithm. This is illustrated in Figure 4 and Figure 6, where we show a typical reconstruction for each of the algorithms after 600 iterations (while good reconstructions are achievable with significantly fewer iterations, 600 iterations guarantee that HIO did reach stagnation, in agreement with Figure 2 and Figure 3). For the satellite images shown in Figure 4 the difference in picture quality is not discernible to the naked eye. However, if we now consider the worst cases produced by these algorithms, as shown in Figure 5, then HPR outperforms HIO (the worst of the 100 reconstructions is taken to be that image for which E_{S+} is the largest at iteration 600). The difference in performance between the two algorithms is much starker with the second example shown in Figure 6, where we display typical reconstructions.

To illustrate the robustness of the HPR algorithm, we perform the following experiment. First, we observe that the image of Figure 6(b) is actually a stagnation point of HIO (it essentially did not evolve through 5000 additional iterations of HIO). This image is then used as an initial point for HPR, which iterated to the stagnation point shown in Figure 7. This experiment shows that HPR can recover further significant information from images for which HIO did its best.

Our experiments with noiseless data reveal fundamental differences between the HIO and HPR

algorithms, namely the effect of changing the update rule (see Eq.(17) and Eq.(23)). These differences are born out in the presence of noise, as we illustrate next.

D. Noisy data

We investigate another aspect of robustness, namely sensitivity to noise. This is an important issue since in practical applications the Fourier data are often corrupted by additive noise. To model this phenomenon, a symmetric noise vector u is added to the magnitude data m shown in Figure 1(b)&(e). The components of the vector u are drawn from a zero mean Gaussian white noise. The signal-to-noise ratio (SNR) is $20 \log_{10} \|m\|/\|u\| = 34$ dB. We compute the mean value of the error measure E_{S+} over 100 trials with different realizations of the noise and the same initial zero-phase guess, which is generated as described in Section 4.B. Let us note that there is no direct connection between the error metric E_{S_+} and the noise level. The error E_{S_+} measures the gap between the constraint sets at a particular iterate. There is no reason this gap should scale in any predictable manner with the noise since the noisy data makes up one of the constraint sets. In principle it is possible, though unlikely, that, even with noise, the constraint sets could have a nonempty intersection. In this case the error metric E_{S_+} could be arbitrarily small even with a significant amount of noise. This is illustrated in Figures 8 and 9 where E_{S_+} is less than -60 dB while the signal to noise ratio is 34 dB.

The behavior of the algorithms in this noisy environment is described by the plots shown in Figure 8 and Figure 9. As with the experiments with noiseless data, the HIO algorithm achieves its minimum value of E_{S_+} more rapidly than the HPR algorithm. As in the noiseless case, the distance from $P_M(x_n)$ to S_+ increases in the first few iterations of the HPR algorithm. Note also in Figure 9 that the HIO algorithm achieves a lower error value than the HPR algorithm. Consistent with our observations in the noiseless experiments, the value of the error metric is only a reliable indicator of picture quality relative to a single algorithm. This is illustrated in Figure 10 (for $\beta = 1.0$) and Figure 11 (for $\beta = 0.75$), where we show typical iterates generated by the algorithms for the same random trial. The algorithms are compared at the 100, 200, and 400 iteration marks. In both cases, stagnation occurs roughly around 200 iterations and little improvement is noticeable beyond that point. The value of E_{S_+} for each of the iterates generated by HIO shown in Figure 10 and Figure 11 is lower by about 2% dB than those generated by HPR, but clearly the quality of the iterates generated by the HPR algorithm is superior.

E. Coupling algorithms

It is common practice in phase retrieval to couple algorithms in order to obtain improved performance. For instance, iterating ER after HIO has been reported to improve the estimate produced by HIO alone [20]. We perform various experiments of this sort in Figure 12 (for $\beta = 1$) and Figure 13 (for $\beta = 0.75$) where, as in Section 4.D, the SNR is 34 dB. In each case, 200 iterations of the HIO and HPR algorithms are first performed (as seen in Section 4.D, this corresponds to stagnation) and then a second algorithm is used for another 200 iterations. These experiments show that while HIO does slightly benefit from a coupling with ER (compare Figure 10(b) with Figure 12(a), and

Figure 11(b) with Figure 13(a)), it is still less competitive than HPR alone (compare Figure 12(d) with Figure 12(a), and Figure 13(d) with Figure 13(a)). On the other hand, iterating HPR after HIO improves significantly the estimate obtained by HIO (compare Figure 10(b) with Figure 12(b), and Figure 11(b) with Figure 13(b)). This shows that the behavior observed in Figure 7 persists in the presence of noise. Finally, iterating ER after HPR does not improve the estimate obtained by HPR alone (compare Figure 12(d) with Figure 12(c), and Figure 13(d) with Figure 13(c)). This lends further support to the idea of using HPR as an autonomous algorithm, obviating user intervention.

Acknowledgments

The authors gratefully acknowledge the Fields Institute in Toronto, Canada, and the Centre International de Rencontres Mathématiques in Luminy, France, where much of this research was done. The work of H. H. Bauschke was supported by the Natural Sciences and Engineering Research Council of Canada and the work of D. R. Luke was completed while he was at the Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Germany. We would also like to thank James Fienup for graciously providing us with Figure 1(a) and the referees for their careful reading of the manuscript and their comments.

E-mails: hbauschk@uoguelph.ca, plc@math.jussieu.fr, drluke@pims.math.ca

References

[1] R. W. Gerchberg and W. O. Saxton, A practical algorithm for the determination of phase from image and diffraction plane pictures, *Optik*, **35**:237–246 (1972).

[2] J. R. Fienup, Phase retrieval algorithms: A comparison, *Appl. Opt.*, **21**:2758–2769 (1982).

[3] A. Levi and H. Stark, Image restoration by the method of generalized projections with application to restoration from magnitude, *J. Opt. Soc. Amer. A*, **1**:932–943 (1984).

[4] A. Levi and H. Stark, Restoration from phase and magnitude by generalized projections, In *Image Recovery: Theory and Applications*, H. Stark, ed. (Academic Press, Orlando, FL, 1987), pp 277–320.

[5] P. L. Combettes, The foundations of set theoretic estimation, *Proc. IEEE*, **81**:182–208 (1993).

[6] D. C. Youla and H. Webb, Image restoration by the method of convex projections: Part I - theory, *IEEE Trans. Medical Imaging*, **MI-1**:81–94 (1982).

[7] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert spaces," *Lecture Notes in Statistics,* **37**:28–47 (1986).

[8] P.-L. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16**:964–979 (1979).

[9] H. H. Bauschke, P. L. Combettes, and D. R. Luke, Phase retrieval, error reduction algorithm, and Fienup variants: A view from convex optimization, *J. Opt. Soc. Amer. A*, **19**:1334–1345 (2002).

[10] R. P. Millane, Phase retrieval in crystallography and optics, *J. Opt. Soc. Amer. A*, **7**:394–411 (1990).

[11] In general, the object domain space need not be restricted to real-valued signals. For a review of a phase retrieval application in which the iterates are complex-valued see [12].

[12] D. R. Luke, J. V. Burke, and R. Lyon, Optical wavefront reconstruction: Theory and numerical methods, *SIAM Rev.*, **44**:169-224 (2002).

[13] H. Stark, ed., *Image Recovery: Theory and Application* (Academic Press, Orlando, FL, 1987).

[14] Typically $0.5 \leq \beta \leq 1$.

[15] It was pointed out in Remark 4.1 of Ref. [9] that a more literal reformulation of Eq.(13) leads to

$$(\forall t \in \mathbb{X}) \;\; x_{n+1}(t) = \begin{cases} \big(P_M(x_n)\big)(t), & \text{if } t \in D \text{ or } \big(P_M(x_n)\big)(t) = 0; \\ x_n(t) - \beta\big(P_M(x_n)\big)(t), & \text{otherwise.} \end{cases} \tag{31}$$

Under the assumption that the zero crossings of $P_M(x_n)$ outside of D are negligible, Eq.(31) reduces to Eq.(14). In digital computing, this assumption is justified by the fact that the probability of obtaining zero numbers is virtually zero.

[16] If $m = 0$, then $x = 0$ is the unique solution of the corresponding phase retrieval problem.

[17] For the BIO algorithm, this was already pointed out in Remark 5.6 of Ref. [9].

[18] We stress that monitoring the sequences $\big(P_M(x_n)\big)_{n\in\mathbb{N}}$ and $\big(P_{S_+}P_M(x_n)\big)_{n\in\mathbb{N}}$ is well-motivated from the convex consistent setting. Replace the nonconvex set M and its corresponding projector P_M with the convex set B and the corresponding projector P_B. If $S_+ \cap B \neq \emptyset$, then, using the results in Ref. [9], one can prove that $E_{S_+}(x_n) \to 0$ with equality precisely when $P_B(x_n) \in S_+ \cap B$. However, if $S_+ \cap B = \emptyset$, which is likely a better approximation of the geometry of the phase retrieval problem, then minimizing $E_{S_+}(x_n)$ corresponds to finding a "displacement vector" for S_+ and B in the sense of Ref. [19]. If the problem is feasible but $E_{S_+}(x_n)$ is positive, then the algorithm has stagnated and the value of $E_{S_+}(x_n)$ is an indicator of the quality of the stagnation point.

[19] H. H. Bauschke and J. M. Borwein, On the convergence of von Neumann's alternating projection algorithm for two sets, *Set-Valued Anal.*, **1**:185–212 (1993).

[20] J. C. Dainty and J. R. Fienup, Phase retrieval and image reconstruction for astronomy, In *Image Recovery: Theory and Application*, H. Stark, ed. (Academic Press, Orlando, FL, 1987), pp. 231–275.

[21] T. R. Crimmins, J. R. Fienup, B. J. Thelen, Improved bound on object support from autocorrelation support and application to phase retrival, *J. Opt. Soc. Amer. A*, **7**:3–13 (1990).

[22] J. R. Fienup and C. C. Wackerman, Phase retrieval stagnation problems and solutions, *J. Opt. Soc. Amer. A*, **3**:1897–1907 (1986).

List of Figures

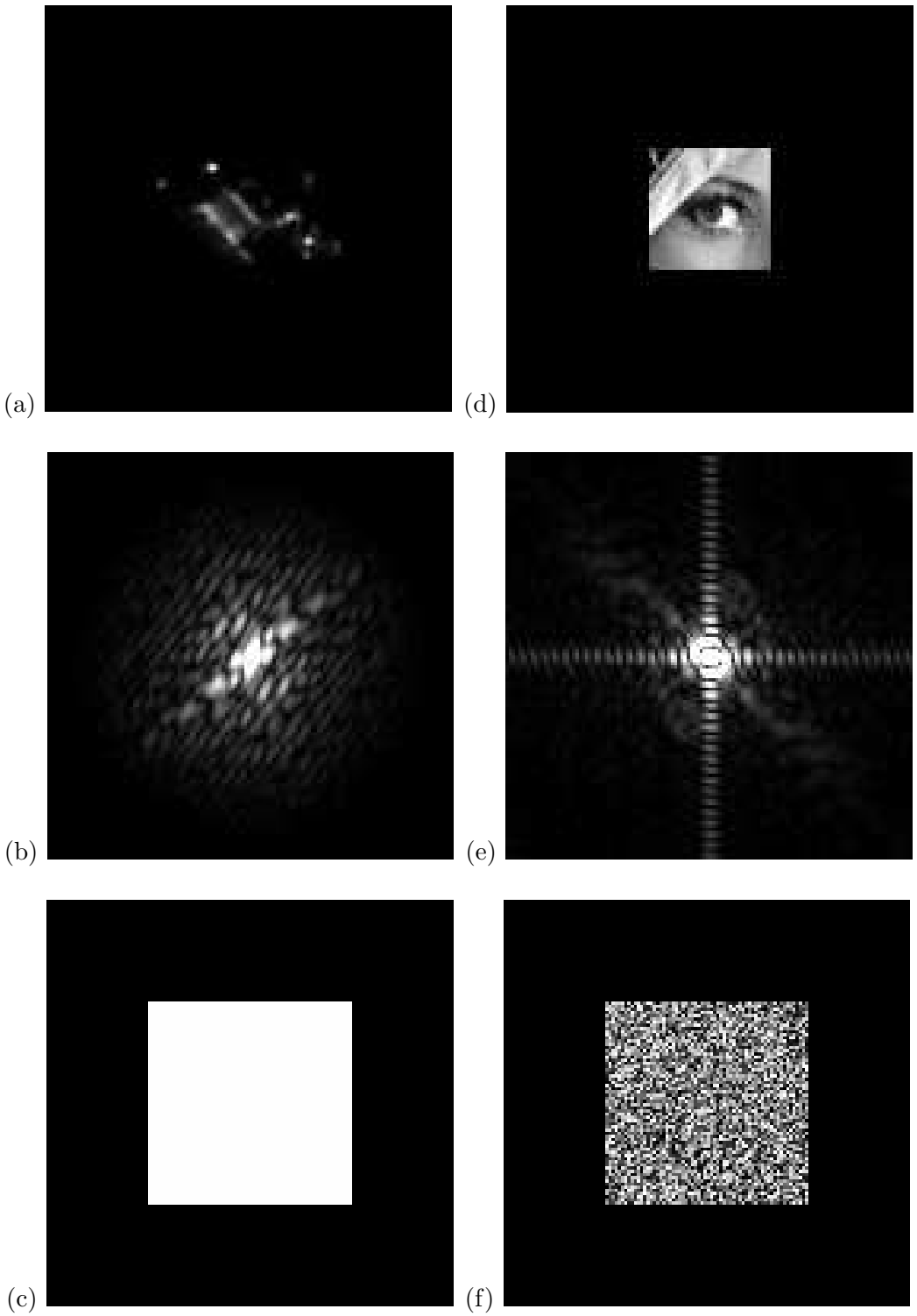

Figure 1: Original images and corresponding data used for the comparison of the HIO and HPR algorithms. (a) is a 128×128 pixel image of a satellite used in [20]. The center of (d) is a 38×38 pixel section of the standard Lena image, zero-padded to a 128×128 matrix. (b) and (e) are the Fourier magnitude data m corresponding to images (a) and (d), respectively. The same object domain support constraint of size 64×64 pixels, shown in (c), is used for each example. An initial image x_0 is shown in (f).

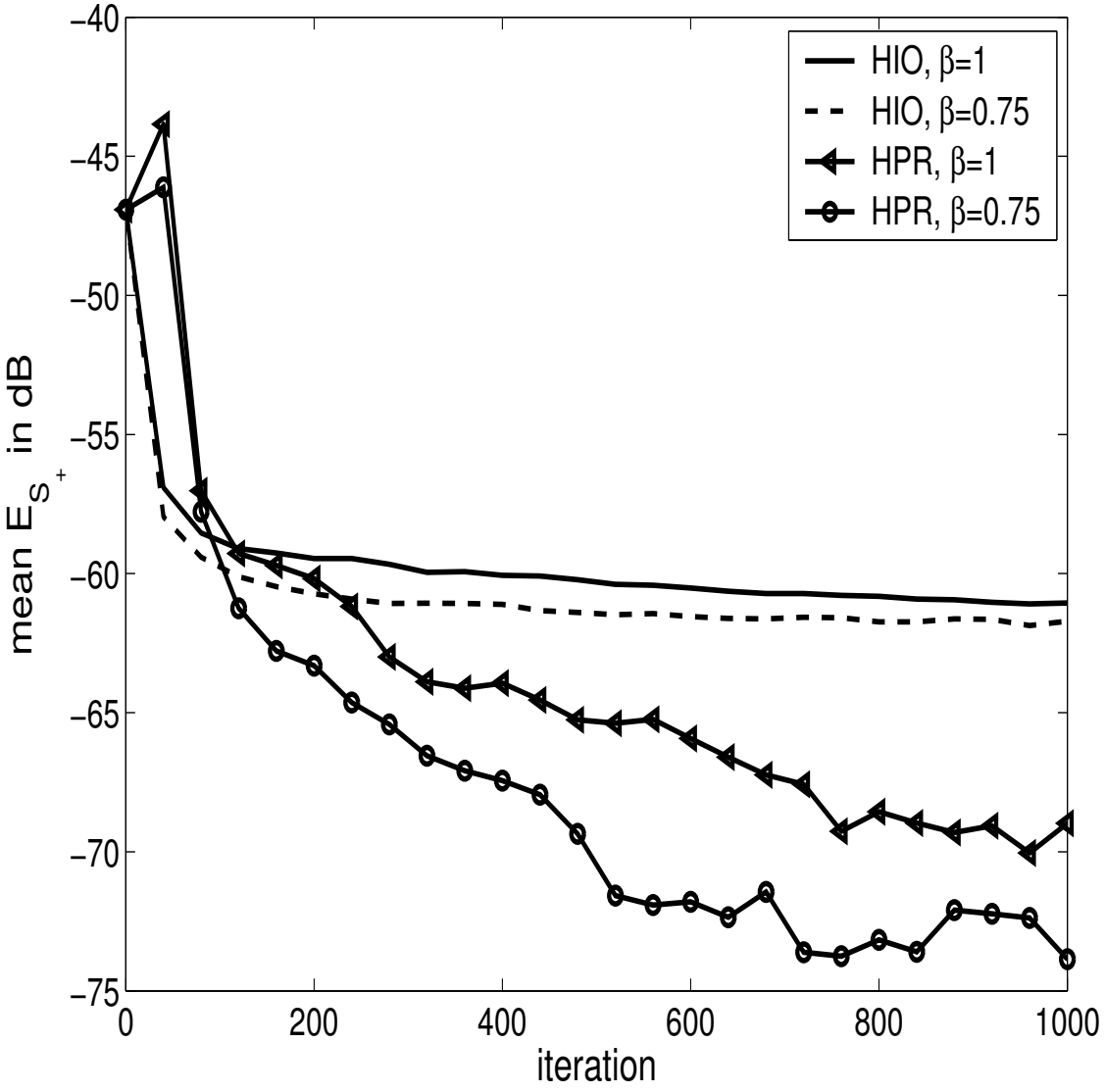

Figure 2: Asymptotic behavior of the HIO and HPR algorithms ($\beta = 0.75$ and 1.0) over 100 random trials for the original image shown in Figure 1(a). Shown here is the mean value of the performance measure E_{S_+} defined by Eq.(30) at each iteration.

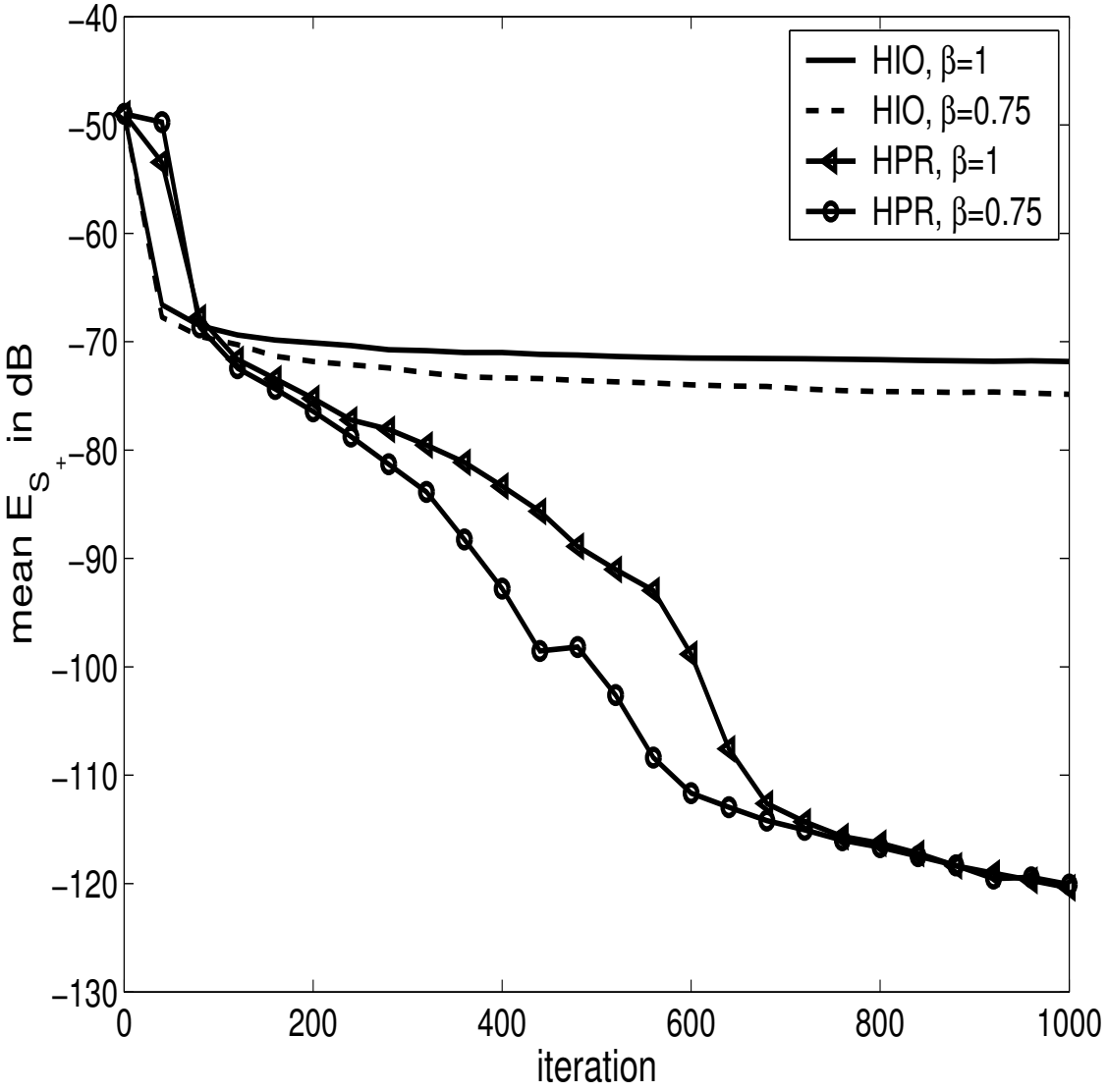

Figure 3: Asymptotic behavior of the HIO and HPR algorithms ($\beta = 0.75$ and 1.0) over 100 random trials for the original image shown in Figure 1(d). Shown here is the mean value of E_{S_+} defined by Eq.(30) at each iteration.

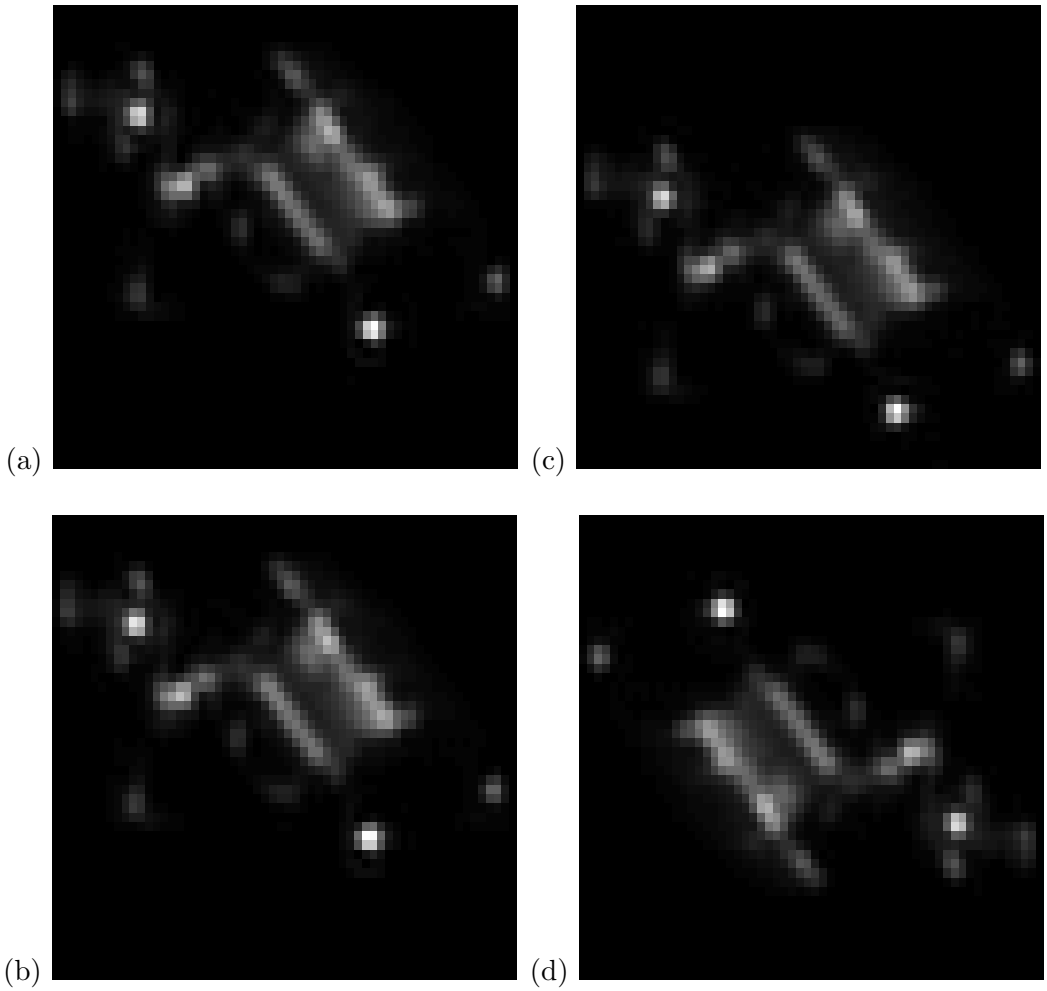

Figure 4: Typical images recovered with 600 iterations of the HIO and HPR algorithms for the data corresponding to Figure 1(a) and a random initial guess similar to that shown in Figure 1(f). (a) and (b) are reconstructed with the HIO algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively. (c) and (d) are reconstructed with the HPR algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively. The rotation of all recovered images from the original image, with the exception of (d), reflects the nonuniqueness in recovery from phase.

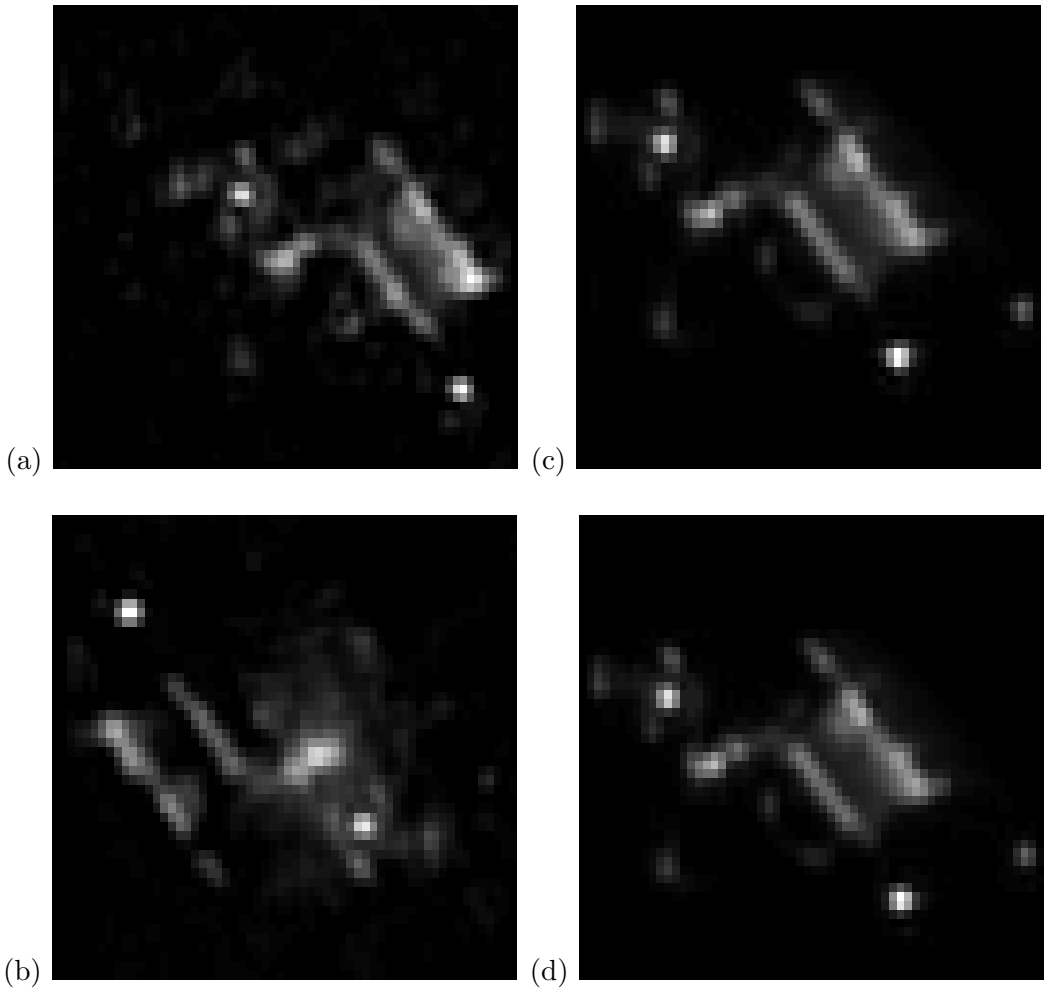

Figure 5: Worst images recovered by the HIO and HPR algorithms for the data corresponding to Figure 1(a) and a random initial guess similar to that shown in Figure 1(f). (a) and (b) are reconstructed with the HIO algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively. (c) and (d) are reconstructed with the HPR algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively. The rotation of all recovered images from the original image, with the exception of (b), reflects the nonuniqueness in recovery from phase.

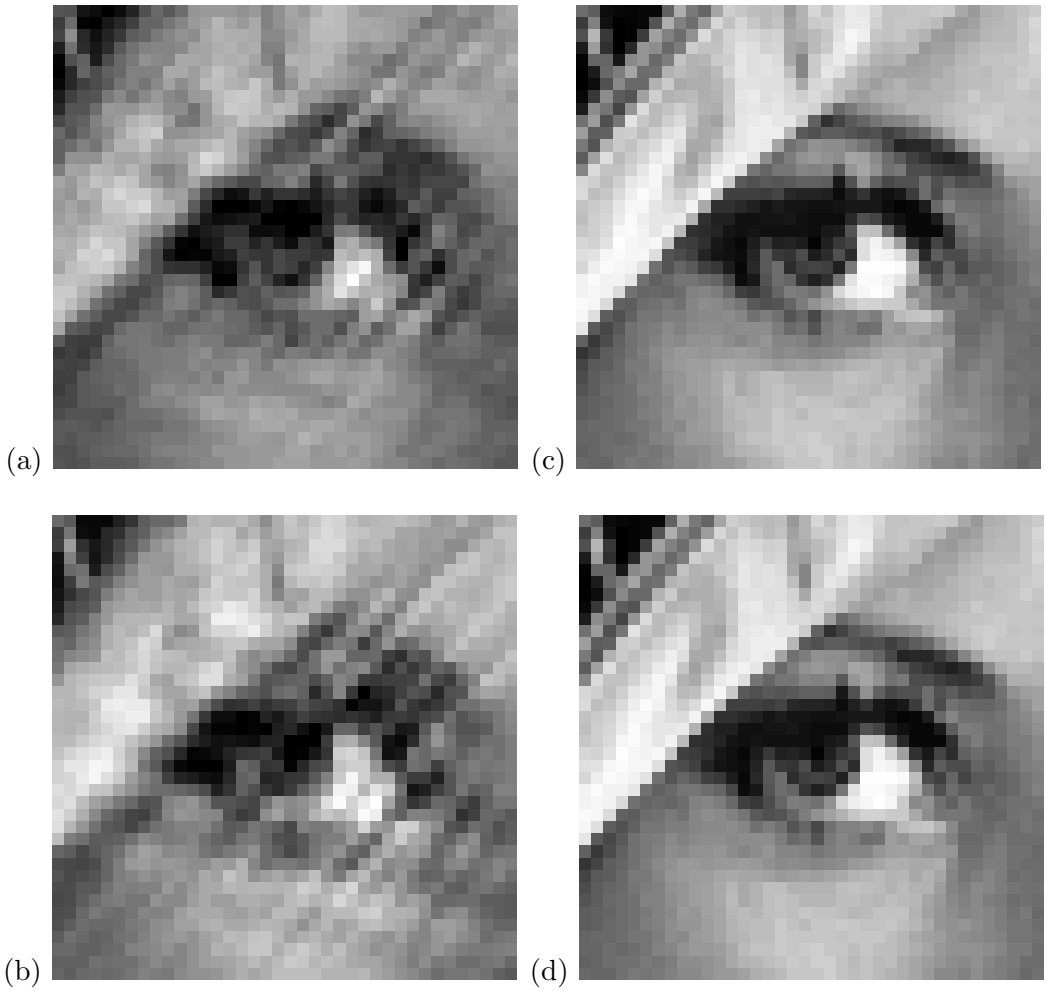

Figure 6: Typical images recovered with 600 iterations of the HIO and HPR algorithms for the data corresponding to Figure 1(d) and a random initial guess similar to that shown in Figure 1(f). (a) and (b) are reconstructed with the HIO algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively. (c) and (d) are reconstructed with the HPR algorithm for $\beta = 0.75$ and $\beta = 1.0$, respectively.

Figure 7: Using the stagnated HIO ($\beta = 1.0$) iterate shown in Figure 6(b) as the initial point, the HPR ($\beta = 1.0$) algorithm finds a stagnation point with a much better picture quality.

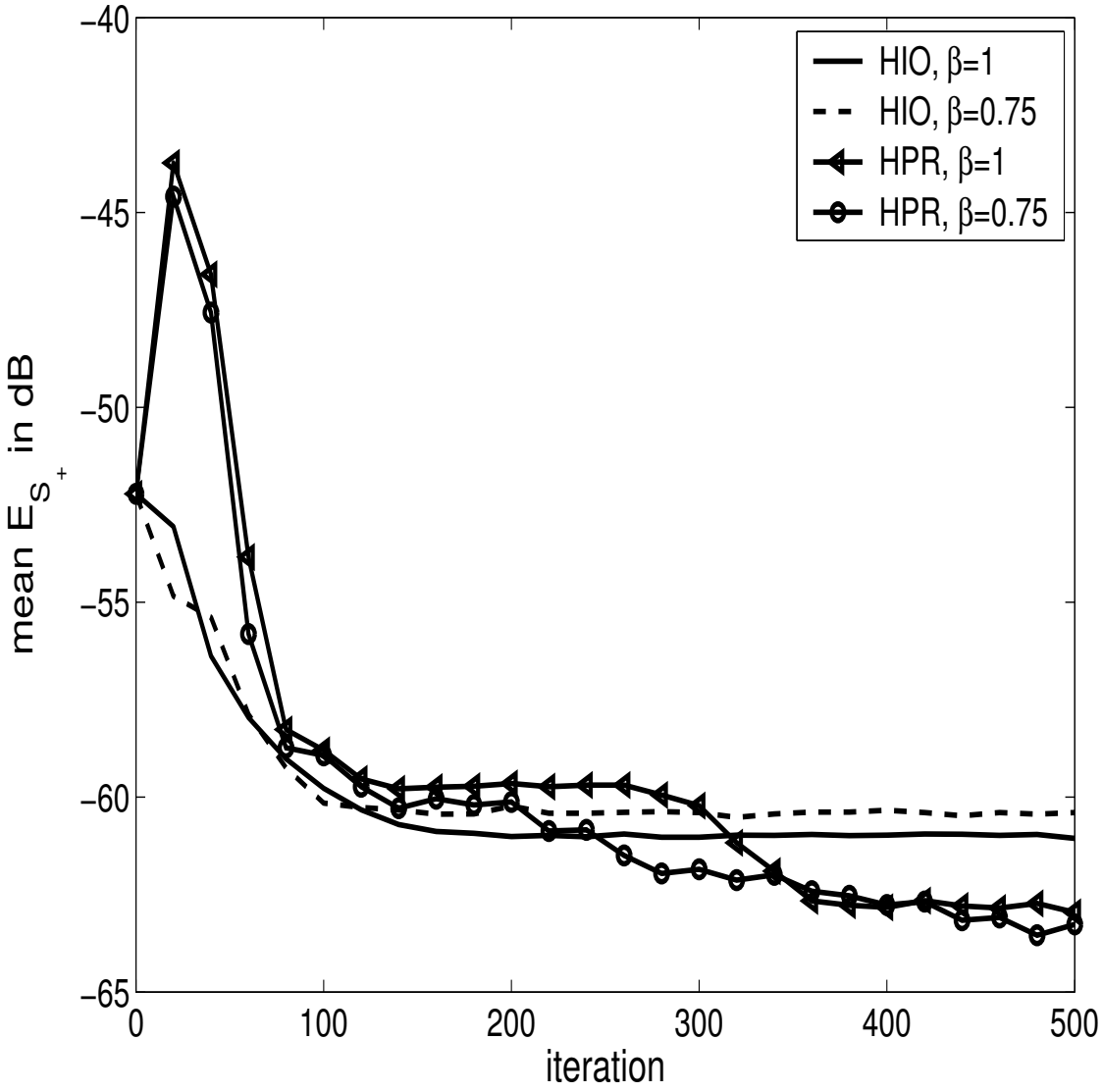

Figure 8: Behavior of the HIO and HPR algorithms ($\beta = 0.75$ and 1.0) with noise (SNR=34 dB) over 100 random trials for the original image shown in Figure 1(a). Shown here is the mean value of E_{S_+} at each iteration.

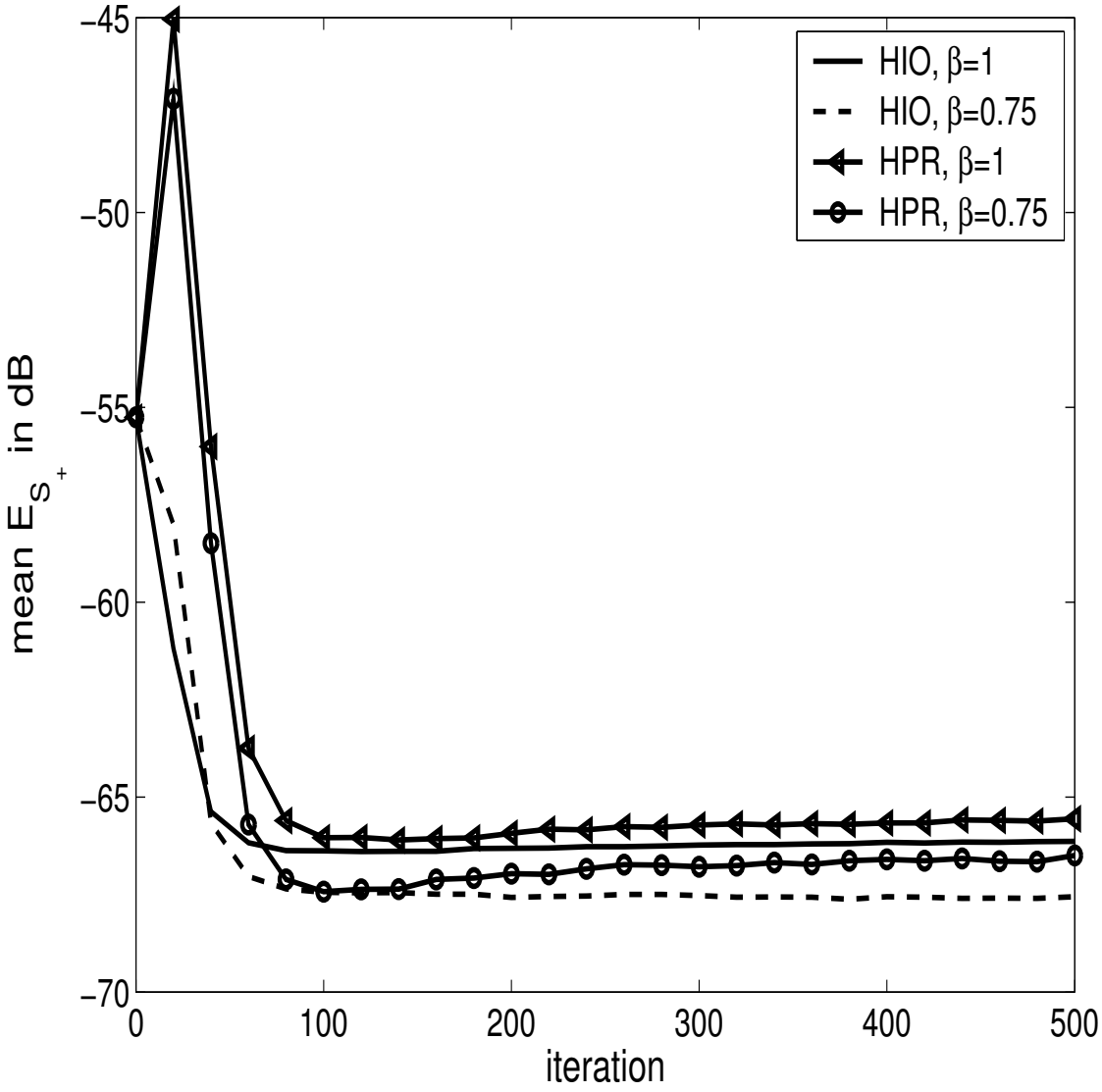

Figure 9: Behavior of the HIO and HPR algorithms ($\beta = 0.75$ and 1.0) with noise (SNR=34 dB) over 100 random trials for the original image shown in Figure 1(d). Shown here is the mean value of E_{S_+} at each iteration.

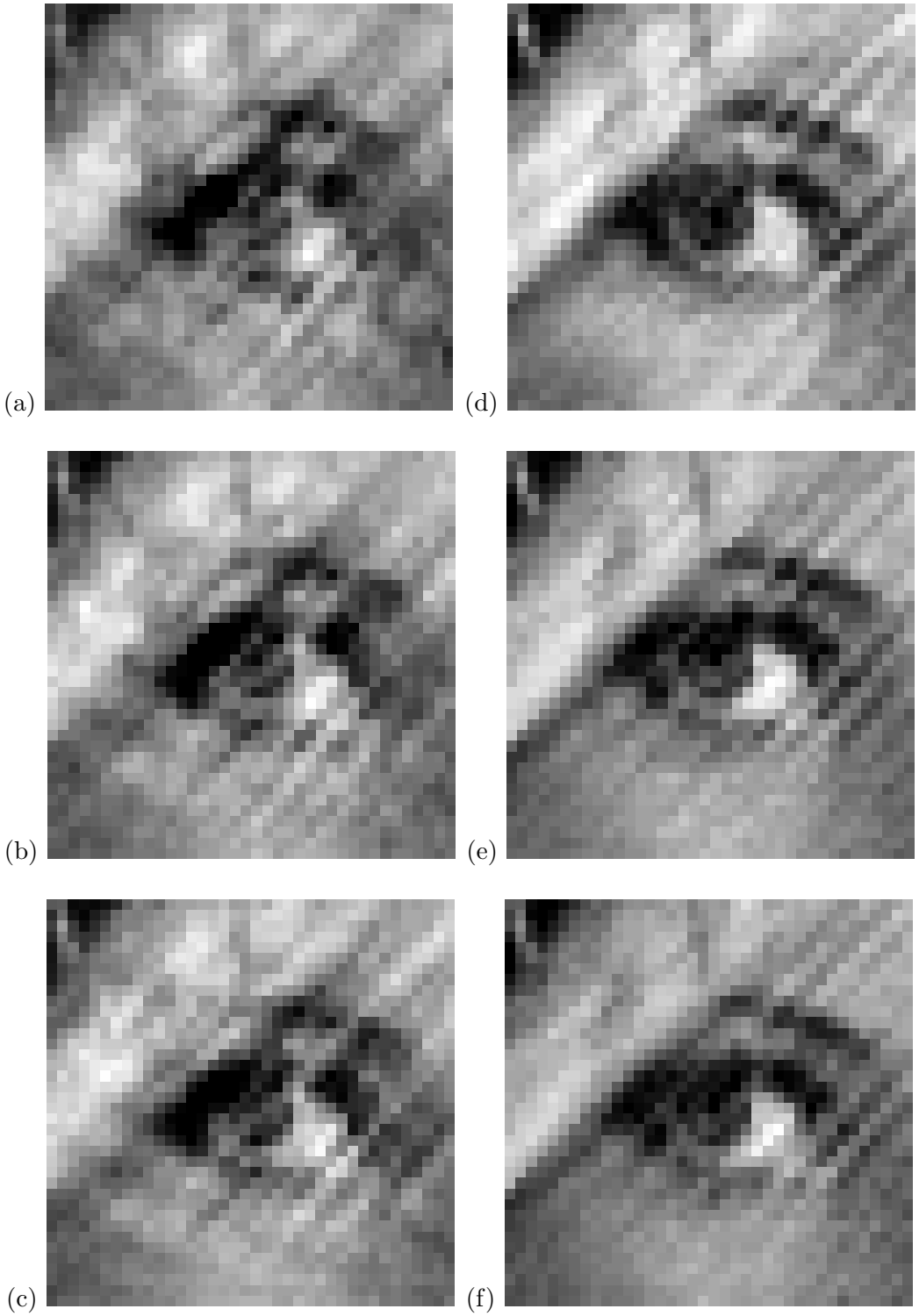

Figure 10: Typical images recovered by the HIO ($\beta = 1$) and HPR ($\beta = 1$) algorithms with noise (SNR=34 dB) for the data corresponding to Figure 1(d) and the normalized zero-phase initial guess shown in Figure 1(c). (a), (b), and (c) are reconstructed with 100, 200, and 400 iterations of the HIO algorithm, respectively; (d), (e), and (f) are reconstructed with 100, 200, and 400 iterations of the HPR algorithm, respectively.

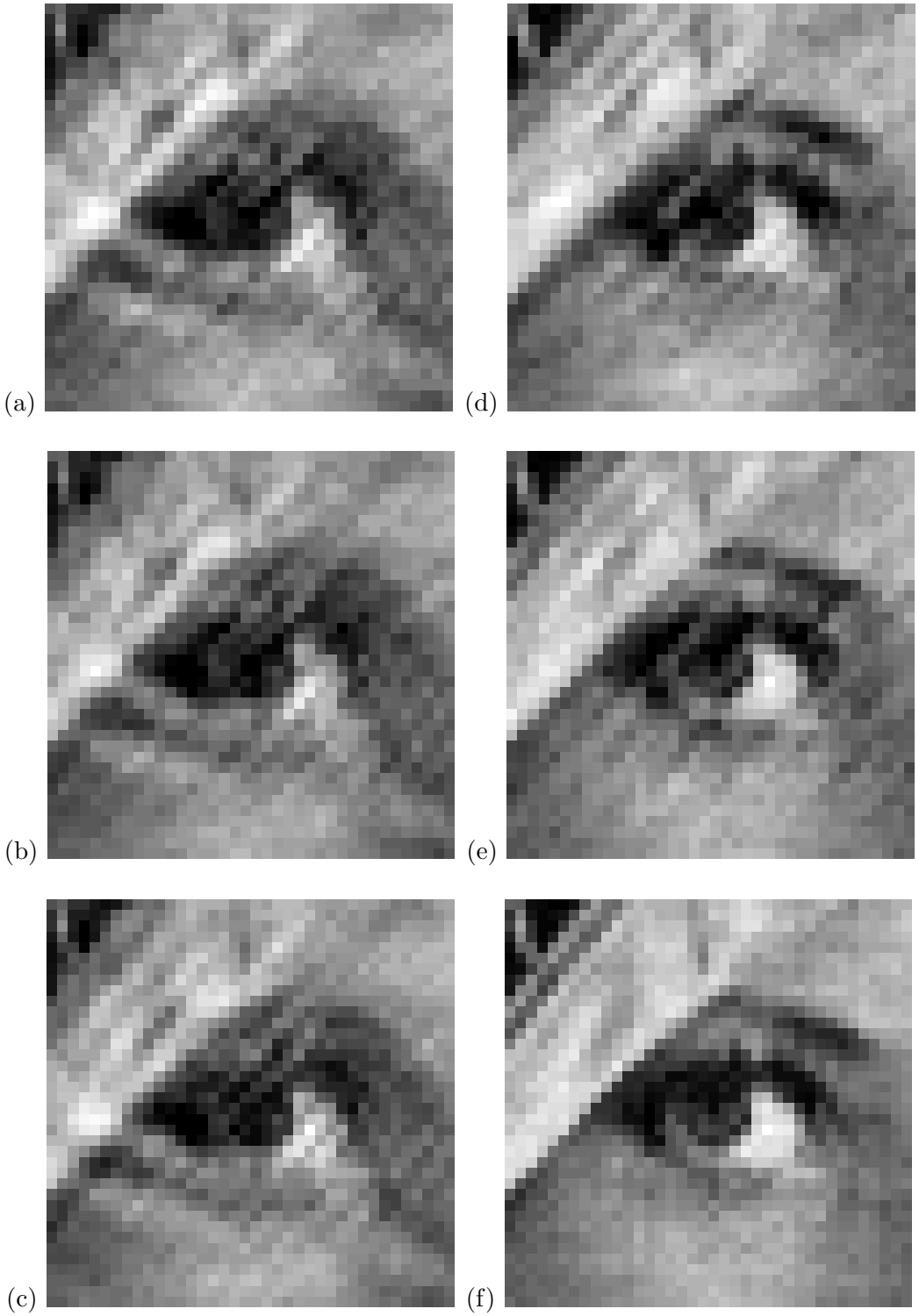

Figure 11: Typical images recovered by the HIO ($\beta = 0.75$) and HPR ($\beta = 0.75$) algorithms with noise (SNR=34 dB) for the data corresponding to Figure 1(d) and the normalized zero-phase initial guess shown in Figure 1(c). (a), (b), and (c) are reconstructed with 100, 200, and 400 iterations of the HIO algorithm, respectively; (d), (e), and (f) are reconstructed with 100, 200, and 400 iterations of the HPR algorithm, respectively.

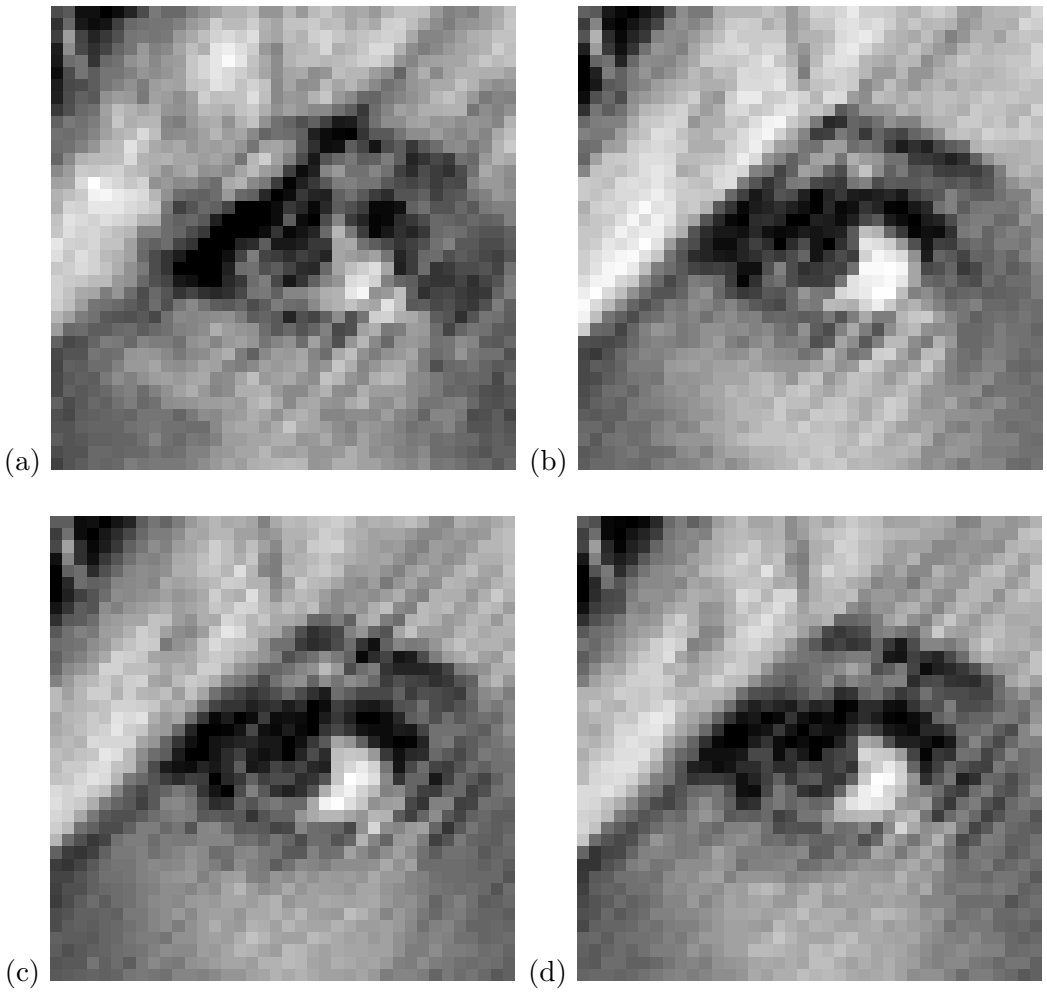

Figure 12: Typical images recovered by coupling the HIO ($\beta = 1$), HPR ($\beta = 1$), and ER algorithms for the noisy data corresponding to Figure 1(d) (SNR=34 dB) and the normalized zero-phase initial guess shown in Figure 1(c). (a) is reconstructed with 200 iterations of HIO followed by 200 iterations of ER. (b) is reconstructed with 200 iterations of HIO followed by 200 iterations of HPR. (c) is reconstructed with 200 iterations of HPR followed by 200 iterations of ER. For reference, the reconstruction obtained with 200 iterations of HPR is shown in (d).

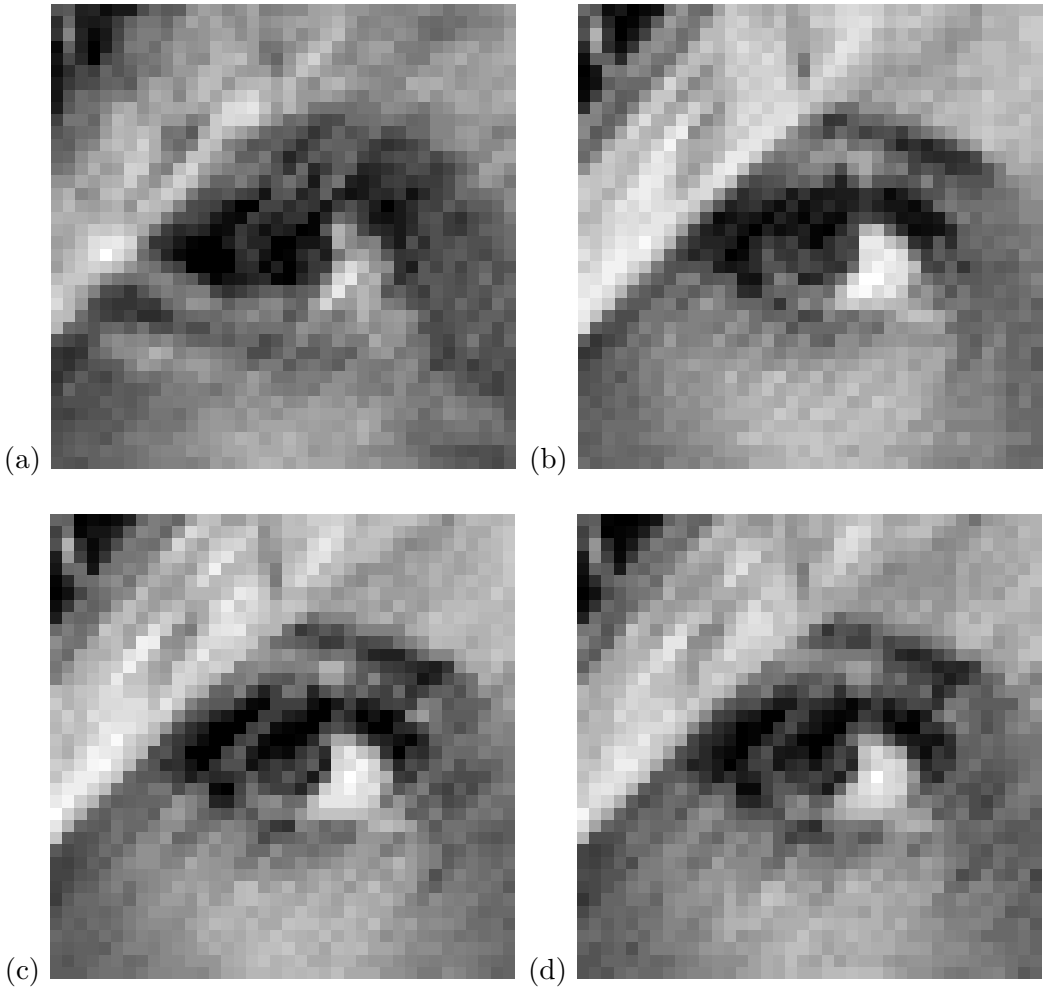

Figure 13: Typical images recovered by coupling the HIO ($\beta = 0.75$), HPR ($\beta = 0.75$), and ER algorithms for the noisy data corresponding to Figure 1(d) (SNR=34 dB) and the normalized zero-phase initial guess shown in Figure 1(c). (a) is reconstructed with 200 iterations of HIO followed by 200 iterations of ER. (b) is reconstructed with 200 iterations of HIO followed by 200 iterations of HPR. (c) is reconstructed with 200 iterations of HPR followed by 200 iterations of ER. For reference, the reconstruction obtained with 200 iterations of HPR is shown in (d).