

Article

Distributed Consensus for Global Matrix Formation in the Principal Component Pursuit Scenario

Gustavo Suárez ^{1,*}  and Juan David Velásquez ² 

¹ Grupo de Investigación e Innovación en Energía, Institución Universitaria Pascual Bravo, Medellín 050034, Colombia

² Grupo de Investigación e Innovación Ambiental, Institución Universitaria Pascual Bravo, Medellín 050034, Colombia

* Correspondence: gustavo.suarez@pascualbravo.edu.co

Abstract: The aim behind principal component pursuit is to recover a low-rank matrix and a sparse matrix from a noisy signal which is the sum of both matrices. This optimization problem is a priori and non-convex and is useful in signal processing, data compression, image processing, machine learning, fluid dynamics, and more. Here, a distributed scheme described by a static undirected graph, where each agent only observes part of the noisy or corrupted matrix, is applied to achieve a consensus; then, a robust approach that can also handle missing values is applied using alternating directions to solve the convex relaxation problem, which actually solves the non-convex problem under some weak assumptions. Some examples of image recovery are shown, where the network of agents achieves consensus exponentially fast.

Keywords: consensus; image recovery; principal component pursuit; static graph



Citation: Suárez, G.; Velásquez, J.D. Distributed Consensus for Global Matrix Formation in the Principal Component Pursuit Scenario. *Appl. Sci.* **2024**, *14*, 3619. <https://doi.org/10.3390/app14093619>

Academic Editor: John D. Clayton

Received: 17 February 2024

Revised: 9 March 2024

Accepted: 15 March 2024

Published: 25 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Consensus algorithms [1] have been studied as powerful tools for distributed networks [2] in multiple applications [3] such as control [4], estimation [5] and the internet of vehicles [6]. They are based on a graph that describes the interconnections of the agents in the network, and the goal of the network is to achieve agreement upon some data, policy or action. Different approaches have been developed, including static, directed [7], undirected—meaning that the communication between two agents over an edge of the graph is bidirectional—and time-varying graphs [8].

Recent developments in consensus algorithms emphasize their adaptability and scalability in dynamic networks, facilitating robust, distributed decision-making processes [9,10]. These advancements have led to enhanced methodologies for reaching a consensus in networks with complex interconnections and varying topologies, demonstrating their effectiveness in a broader range of applications beyond traditional settings.

Moreover, the utility of PCA extends into the realm of big data and high-dimensional data analysis, where it plays a crucial role in dimensionality reduction, feature extraction, and data compression [11]. The ability to distill essential features from large datasets while preserving the integrity of the underlying structure has profound implications for computational efficiency and the interpretability of data in machine learning and data science.

There are many applications where PCA is relevant; some examples are video surveillance, face recognition [12], latent semantic indexing [13], ranking and collaborative filtering to anticipate tastes, image analysis including fluid dynamics [14], machine learning and food identification [15], among others. Examples of PCA applied to face image processing and fluid flows are shown in Figures 1 and 2. The video processing is similar, where each frame is re-shaped as a column vector, and they are stacked together in the new matrix, as shown in Figure 2.

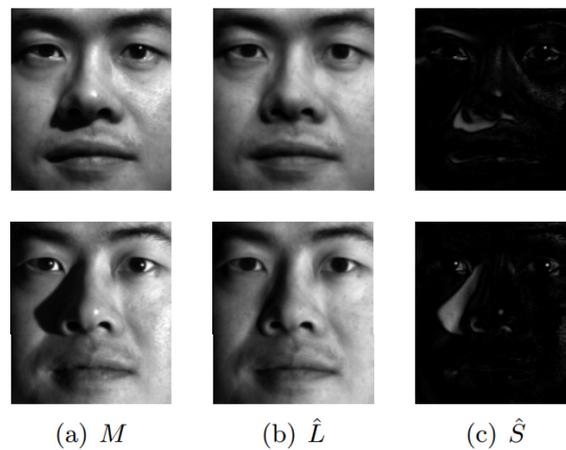


Figure 1. Removing shadows, peculiarities, and saturation from face images [16].

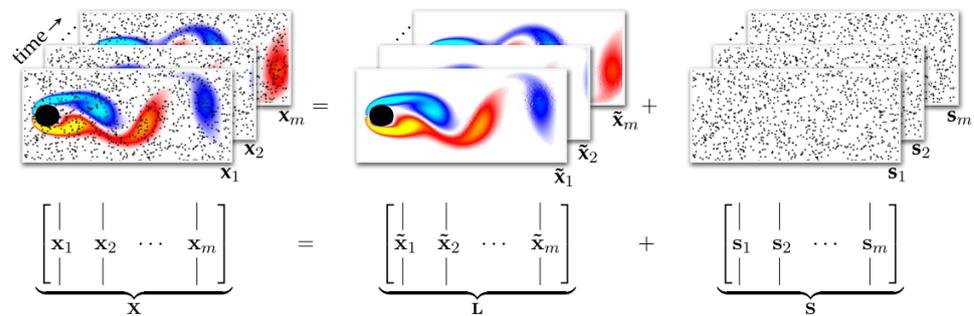


Figure 2. Schematic of PCA filtering applied to corrupt flow-field data [14].

Integrating consensus algorithms with PCA in a distributed setting is a novel approach to tackling the challenges of large-scale data analysis. By leveraging the strengths of consensus algorithms to achieve a unified view from multiple distributed agents, combined with the PCA’s capacity for dimensionality reduction, this methodology offers a compelling solution for decentralized data processing and analysis. The consensus-based PCA optimization framework enables distributed agents to collaboratively compute PCA decompositions, thereby overcoming limitations related to data privacy, storage, and computational bottlenecks [17–22].

In this article, an intersection of consensus algorithms and the optimization problem of principal component analysis (PCA) [23,24] is addressed. A consensus is reached along with a particular type of relaxation called convexification. It allows the network to solve the optimization with a low computational cost.

The PCA problem addresses the following case. Suppose it is given a large data matrix $M \in \mathcal{R}^{n \times n}$ with missing and noisy data, such that this matrix can be decomposed into a low-rank matrix $L \in \mathcal{R}^{n \times n}$ and a sparse one $S \in \mathcal{R}^{n \times n}$, where both matrix entries are of an arbitrary magnitude. The PCA question is how these components can be recovered with enough accuracy or even exactly recovered.

Now consider there are multiple agents, which can be sensors, where each agent measures part of a global matrix M , possibly with redundant data between agents. This aims for a local rule such that each sensor will find the same decomposition of the global M . This situation can emerge, for instance, in a vertical or horizontal arrangement of cameras, where they cover a bigger image than what can be covered by a single agent.

In scenarios where the cost of communication between agents is prohibitively high—such as in privacy-sensitive environments, conflict zones, or geographically remote areas where energy resources are scarce—it becomes imperative to minimize inter-agent communication while still achieving an accurate consensus on matrices L and S . This research introduces an innovative approach that significantly reduces the necessity for frequent

communication by optimizing the local computations at each agent. We propose a method where agents communicate only the essential information required to converge towards a unified decomposition of the global matrix M . This methodology leverages the inherent efficiency of local optimization steps, thus enabling the recovery of L and S with minimal communication overheads. Our approach is particularly suited for a distributed principal component analysis (PCA) under the constraints of a high communication cost or stringent privacy requirements.

We demonstrate through theoretical analysis and empirical validation that this strategy not only preserves the accuracy of matrix decomposition but also ensures a substantial reduction in communication costs. The feasibility and effectiveness of our method are supported by recent advancements in distributed optimization and consensus algorithms that emphasize communication efficiency [25,26].

More specifically, we propose an algorithm to recover a low-rank and a sparse matrix in a static undirected graph where each agent measures some components of the global corrupted matrix. The convexification process to relax the PCA into a computationally efficient problem is employed along with the Alternating Direction Method of Multipliers (ADMM) [27]. An application of mapping and image recovery is shown, demonstrating exponentially fast convergence.

Contributions

Mainly, other algorithms communicate during a complete PCA. This is undesirable in the context of very restrictive communication due to safety reasons or very high-cost communications. We prove and show how this does not have to be the case for a distributed PCA since the agents can communicate only until some agreement is reached on the global matrix formation. Then, each agent can optimize alone, minimizing the communication as much as possible. We present proof of the proposed algorithm along with a multi-camera application.

We also provide a simple but rigorous viewpoint to overcome the issue of overlapped regions of view or common measurement/knowledge of each agent in the network. We show how this is an issue for the classical consensus, showing over-prediction in the overlapped regions. We show this from the perspective of a distributed camera network application. We also show how to handle colored images, since most researchers use binary black and white scales for images.

We investigate the impact on the convergence of different network sizes and densities.

2. Problem Formulation

Suppose there are l agents represented by nodes in an undirected connected graph $G = \{V, E\}$, where $V = \{1, 2, \dots, l\}$ is the set of vertices, and the edges $E \subseteq V \times V$. There is an edge $(i, j) \in E$ if and only if agent i can communicate with j and vice versa. The neighborhood of agent i is defined as

$$\mathcal{N}_i := \{j \in V : (i, j) \in E\} \quad (1)$$

This graph has an associated adjacency matrix $A \in \mathcal{R}^{l \times l}$ and each agent i at iteration k has a state $x_i^k \in \mathcal{R}^{n \times m}$.

There is a global noise-corrupted matrix $M \in \mathcal{R}^{n \times m}$ that is the sum of a low-rank matrix $L \in \mathcal{R}^{n \times m}$ and a sparse matrix $S \in \mathcal{R}^{n \times m}$, (2).

$$M = L + S \quad (2)$$

Unlike the small noise term S in classical PCA, the entries in S can have an arbitrarily large magnitude. Hence, it is called robust PCA. Each agent can only observe part of M , and different agents can also have some coincident measurements. The distribution of what part of M every agent is measuring is irrelevant; they can have equal-row-distributed data

or a sparse distribution. The restriction is that each agent knows which matrix values are being measured by itself, such that there are zeros in the unobserved entries.

An algorithm is hypothesized such that every agent obtain the same estimation of L and S based only on its own measurements and the neighbor’s information, solving:

$$\min_{L,S} \text{rank}(L) + \|S\|_0 \quad \text{s.t.} \quad L + S = M \tag{3}$$

where $\|S\|_0$ counts the number of nonzero elements in S , quantifying how sparse it is. However, (3) is NP-hard [28] and neither the $\text{rank}(L)$ nor the $\|S\|_0$ terms are convex.

3. Algorithm

Because each agent measures only a part of M , it is expected to use a distributed consensus. The question is whether it is required to solve the optimization problem in a distributed way while the consensus is reached in the solution, or if there is another, more straightforward way to achieve this still only using local information.

The method studied and implemented here is to reach the first consensus based on local information to converge to the global matrix M . Since it is allowed that agents share redundant measurements, a filter is proposed such that the agent does not change its own initial measurements; however, note that this is still a local rule since each agent knows which data it measured. In other words, the algorithm updates a subset \mathcal{X}_i of the full matrix indices corresponding to the unobserved entries; hence, only $x_i^k(\mathcal{X}_i)$ is updated. In the results section, an example of why this step is required is presented. The only global shared parameter is the number of agents l .

Then, using convex relaxation in the optimization problem, together with alternate direction optimization, each agent can exactly reproduce the original global L and S under some weak assumptions. To solve (3), each agent performs convex relaxation [14], which optimally actually recovers the original solution under some assumptions, of the following form:

$$\min_{L,S} \|L\|_* + \beta\|S\|_1 \quad \text{s.t.} \quad L + S = M \tag{4}$$

where $\|\cdot\|_*$ is the nuclear norm, given by the sum of singular values, which is a proxy for the rank of the matrix, and $\|\cdot\|_1$ is the one-norm of the matrix viewed as a vector, given by the sum of the magnitudes of each entry in the matrix, which is a proxy for the $\|\cdot\|_0$ norm of a matrix. The hyperparameter $\beta = \frac{1}{\sqrt{\max(n,m)}}$ [16].

Here, the strategy of the augmented Lagrange multiplier is implemented:

$$\theta(L, S, Y) = \|L\|_* + \beta\|S\|_1 + \langle Y, M - L - S \rangle + \frac{\mu}{2} \|M - L - S\|_F^2 \tag{5}$$

A generic Lagrange multiplier would solve this by updating the Lagrange multiplier with $Y^{k+1} = Y^k + \mu(M - L^k - S^k)$ after finding L^k, S^k by minimizing $\theta(L, S, Y)$. However, here, we avoid solving a sequence of convex programs by first minimizing θ with respect to L , with S fixed, minimizing θ with respect to S with L fixed, and then updating the Lagrange multiplier matrix Y .

This method speeds up the computation since minimizing θ with respect to L and S separately is straightforward. In fact [16], the argument that minimizes θ with respect to S is $S_{\beta\mu}(M - L + \mu^{-1}Y)$, where $S_\tau[x] = \text{sgn}(x)\max(|x| - \tau, 0)$ is a shrinkage operator applied to each element of the matrix.

Similarly, let $D_\tau(X)$ denote the singular value thresholding operator given by $D_\tau(X) = US_\tau(\Sigma)V^*$, where $X = U\Sigma V^*$ is any singular value decomposition. Then, the argument that minimizes θ with respect to L is $D_\mu(M - S - \mu^{-1}Y)$. This strategy is summarized in Algorithm 1.

Algorithm 1 Consensus and Optimization

```

k ← 1
Converge ← 0
 $\mathcal{X}_i \leftarrow$  unobserved indices of agent  $i$ 
while Converge  $\neq$  1 do
  for  $i \in V$  do
     $U \leftarrow \sum_{j \in \mathcal{N}_i} A_{ij} x_j^k$  ▷ Consensus
     $x_i^{k+1}(\mathcal{X}_i) \leftarrow U(\mathcal{X}_i)$  ▷ Update only unobserved
     $k \leftarrow k + 1$ 
  end for
end while
for  $i \in V$  do ▷ Same optimization for each agent
   $S \leftarrow Y \leftarrow 0$ 
   $\mu \leftarrow 10\beta$  ▷ Any  $\mu > 0$ 
   $M \leftarrow l x_i^k$  ▷ Need to know number of nodes
  while iter < max do ▷ ADMM: update L and S
     $L^{k+1} \leftarrow D_\mu(M - S^k - \frac{1}{\mu} Y^k)$ 
     $S^{k+1} \leftarrow S_{\beta\mu}(M - L^{k+1} + \frac{1}{\mu} Y^k)$ 
     $Y^{k+1} \leftarrow Y^k + \mu(M - L^{k+1} - S^{k+1})$ 
  end while
end for

```

Additionally, some restrictions on A , L , and S are required for the method to converge and to solve the optimization problem.

Theorem 1. Consider a global matrix M that can be decomposed into low-rank L and sparse S matrices as $M = L + S$ and a static undirected connected graph of agents x_i described by an adjacency matrix A . Each agent measures part of the global matrix M in such a way that M is totally observed, with possibly overlapped entries. Algorithm 1 provides the steps such that at the end of the algorithm each agent knows L and S of the global M .

We outline the proof.

Proof. Given a network of agents with interactions described by a doubly stochastic matrix A , we assume the following properties for A :

1. $\lambda_1(A) = 1$ and $|\lambda_i| < 1$ for all $i > 1$, indicating A has a dominant eigenvalue of 1.
2. $A\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T A = \mathbf{1}^T$, showing A is doubly stochastic and the vector of ones, $\mathbf{1}$, is both a right and left eigenvector corresponding to the eigenvalue 1.

The consensus update rule for agent i is:

$$x_i^{k+1}(\mathcal{X}_i) = \sum_{j \in \mathcal{N}_i} A_{ij} x_j^k(\mathcal{X}_i) \tag{6}$$

To prove convergence, we utilize the eigenvalue decomposition of A , $A = V\Lambda V^{-1}$, and leverage the geometric decay of components associated with eigenvalues $|\lambda_i| < 1$. As $k \rightarrow \infty$, all components except those aligned with $\mathbf{1}$ diminish, leading to convergence towards an average state across the network.

The optimization problem is formulated as:

$$\min_{L,S} \|L\|_* + \beta \|S\|_1 \quad \text{s.t.} \quad L + S = M \tag{7}$$

and by introducing the augmented Lagrangian, the problem is rephrased as:

$$\theta(L, S, Y) = \|L\|_* + \beta\|S\|_1 + \langle Y, M - L - S \rangle + \frac{\mu}{2}\|M - L - S\|_F^2 \tag{8}$$

where $\|\cdot\|_*$ denotes the nuclear norm (sum of singular values), $\|\cdot\|_1$ denotes the l_1 norm (sum of absolute values of the matrix entries), $\langle \cdot, \cdot \rangle$ represents the inner product, and $\|\cdot\|_F^2$ is the Frobenius norm squared.

Update L with S and Y fixed.

Given S and Y , the update for L is obtained by minimizing $\theta(L, S, Y)$ with respect to L . This step involves solving:

$$L^{k+1} = \arg \min_L \left(\|L\|_* + \frac{\mu}{2}\|M - L - S^k + \mu^{-1}Y^k\|_F^2 \right) \tag{9}$$

This problem is a proximal operation that can be efficiently solved using the singular value thresholding (SVT) operator D_τ , defined as:

$$D_\tau(X) = U \cdot S_\tau(\Sigma) \cdot V^* \tag{10}$$

where $X = U\Sigma V^*$ is the singular value decomposition of X , and $S_\tau(\Sigma)$ applies a shrinkage operation to the singular values Σ , reducing each singular value σ_i by τ , but not to below zero. In this context, $\tau = \mu^{-1}$.

Update S with L and Y fixed.

After updating L , S is updated by minimizing $\theta(L, S, Y)$ with L^{k+1} and Y^k fixed, leading to:

$$S^{k+1} = \arg \min_S \left(\beta\|S\|_1 + \frac{\mu}{2}\|M - L^{k+1} - S + \mu^{-1}Y^k\|_F^2 \right) \tag{11}$$

This optimization step is akin to a soft thresholding operation on each element of the matrix $M - L^{k+1} + \mu^{-1}Y^k$ and can be efficiently solved using the shrinkage operator $S_{\beta\mu}$, which for a given matrix X and threshold τ , requires:

$$S_\tau[X]_{ij} = \text{sign}(X_{ij}) \max(|X_{ij}| - \tau, 0) \tag{12}$$

Update Y .

Finally, the Lagrange multiplier Y is updated to account for the discrepancy between M and the current approximation $L^{k+1} + S^{k+1}$:

$$Y^{k+1} = Y^k + \mu(M - L^{k+1} - S^{k+1}) \tag{13}$$

Convergence.

The iterative updates for L , S , and Y guarantee convergence to the global optimum under convex conditions. The SVT and shrinkage operators ensure that each step moves towards minimizing the objective function while adhering to the constraint $L + S = M$. Updating Y adjusts for the equality constraint violation, ensuring that as $k \rightarrow \infty$, $L^k + S^k \rightarrow M$, thus solving the original decomposition problem.

Finally, we need to prove that convex relaxation solves the original non-convex problem. It could be that there is not enough information to perfectly disentangle the low-rank and sparse components. To prevent this from happening, the assumption that the low-rank component L is not sparse is employed; this is called the incoherence condition [29]. Another identifiability issue arises if the sparse matrix is low-rank. Then, the following assumptions are required [16].

Write the singular value decomposition of $L \in \mathbb{R}^{n_1 \times n_2}$ as

$$L = U\Sigma V^* = \sum_{i=1}^r \sigma_i u_i v_i^*, \tag{14}$$

where r is the rank of the matrix, $\sigma_1, \dots, \sigma_r$ are the positive singular values, $U = [u_1, \dots, u_r]$, $V = [v_1, \dots, v_r]$ are the matrices of left- and right-singular vectors, and V^* is the conjugate transpose. Then, the incoherence condition with parameter μ states that

$$\max_i \|U^* e_i\|_2^2 \leq \frac{\mu r}{n_1}, \quad \max_i \|V^* e_i\|_2^2 \leq \frac{\mu r}{n_2}, \tag{15}$$

and

$$\|UV^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}. \tag{16}$$

Suppose L is $n \times n$ and obeys the incoherence condition and that the support set of S is uniformly distributed among all sets of cardinality m . Then, there is a numerical constant c such that with a probability of at least $1 - cn^{-10}$ (over the choice of support of S), the convex relaxation solution solves the nonconvex PCA problem exactly, given that:

$$\text{rank}(L) \leq \rho_r n \mu^{-1} (\log n)^{-2} \quad \text{and} \quad m \leq \rho_s n^2. \tag{17}$$

In the above, ρ_r and ρ_s are positive numerical constants. \square

4. Results

First, it is shown why it is important for each agent i to only update the unobserved entries \mathcal{X}_i . This step is required since redundant measurements between agents are allowed, which, in the case of surveillance, would mean a shared vision field, as in the example in Figure 3.

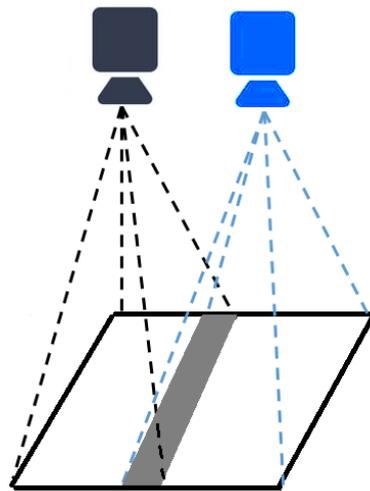


Figure 3. Two agents with redundant measurements in the overlapping field of view (gray region).

In this case, a showcase model is employed, updating all entries of each agent, with $M \in \mathcal{R}^{500 \times 500}$, with a low-rank component of 5 and a cardinality of the sparse component of 0.2 with random entries. Each agent will observe the same amount of rows, but adjacent agents share 20 rows. The result is shown in Figure 4. The highlighted rows in L are the overlapped field of view; hence, it is clear that it is required to update only the unobserved entries \mathcal{X}_i to avoid this overestimation, as stated in Algorithm 1.

Since it has already been explained why a local update is not performed for every entry of the matrix state, now, some results of the convergence of the consensus process and an example of image recovery are shown. Two cases, a and b , are tested with four agents. The graphs are shown in Figure 5, the respective adjacency matrices are shown in (6), and the convergence consensus is shown in Figure 6, where exponential convergence can be observed, which is a known property of this type of static graph.

$$A_a = \begin{bmatrix} 1/3 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \end{bmatrix}, A_b = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/4 & 5/12 & 1/3 \\ 0 & 1/4 & 1/3 & 5/12 \end{bmatrix} \quad (18)$$

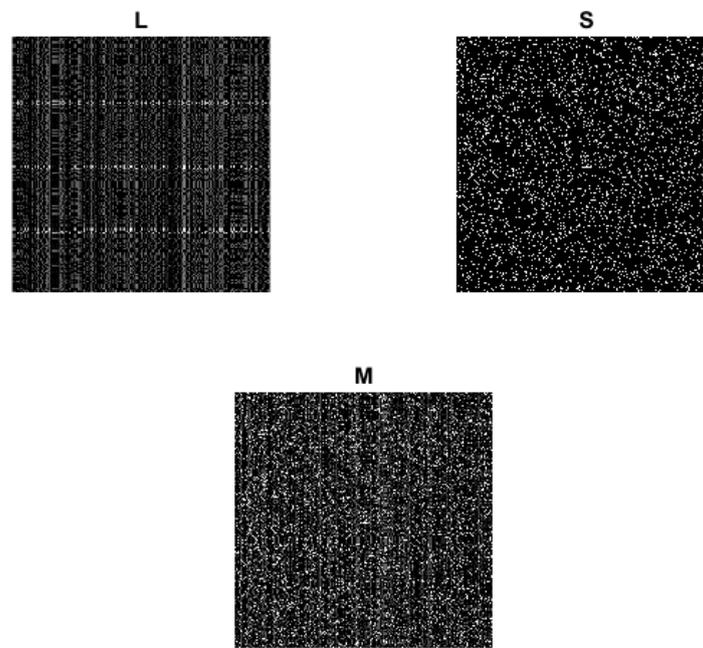


Figure 4. The agents reach consensus with overestimation (highlighted rows in L) in the overlapping regions in the field of view when the filter using \mathcal{X}_i is not applied.

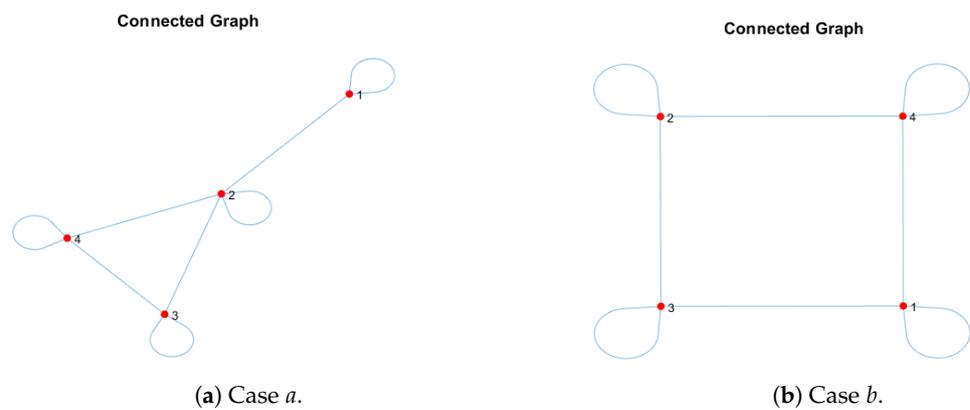


Figure 5. Connected graphs.

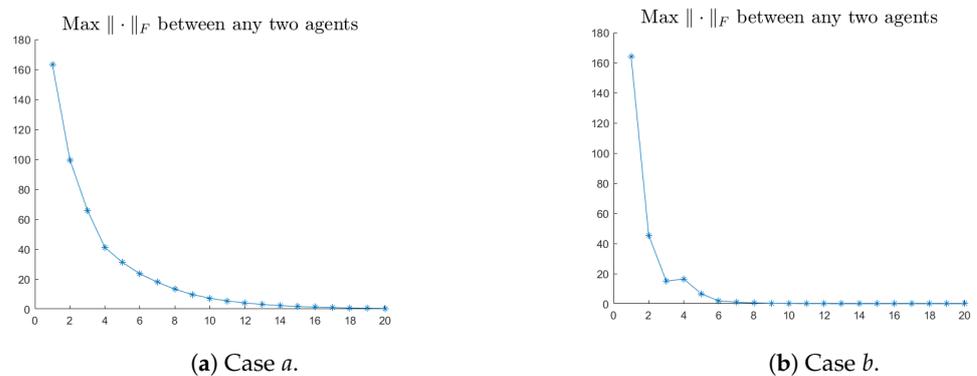


Figure 6. Convergence in consensus.

The first studied case involves decomposing a corrupted image of the moon's surface into a low-rank L and a sparse S matrix. Every agent knows part of the global M , and there is some amount of redundant values between the agent states, but every agent knows each of its own observed entries. After reaching a consensus, each agent runs the alternate direction optimization into the convex relaxation of (3), as described in Algorithm 1. The resulting low-rank image L is employed to replace the corrupted data in the original image M ; the results are shown in Figure 7.

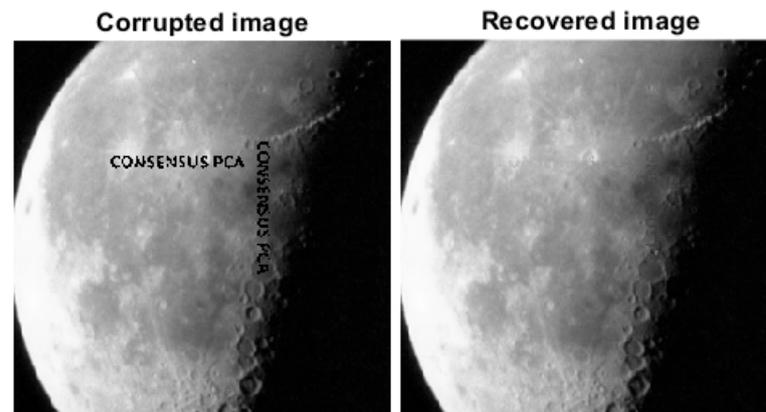


Figure 7. Moon image recovered by every agent, case a .

To deal with colored RGB images, the algorithm splits the original image $M \in \mathcal{R}^{n \times n \times 3}$ into three matrices, $M_R \in \mathcal{R}^{n \times n}$, $M_G \in \mathcal{R}^{n \times n}$, and $M_B \in \mathcal{R}^{n \times n}$, one for each color channel, and a consensus is reached and optimization is carried for each matrix. Finally, each low-rank matrix is employed to replace the corrupted data in each RGB channel, and then they are stacked into an RGB image. Further examples are shown in Figures 8 and 9.

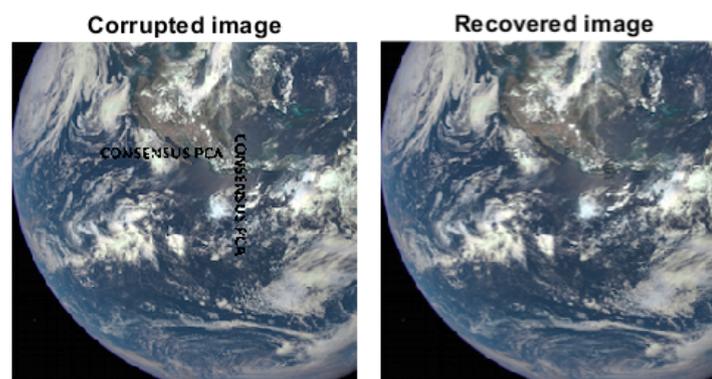


Figure 8. Earth image recovered by every agent.

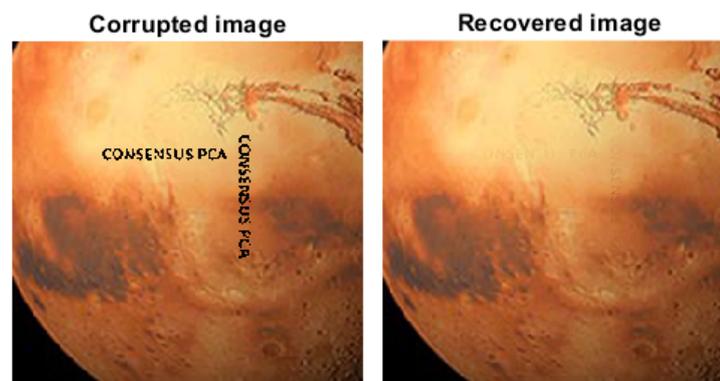


Figure 9. Mars image recovered by every agent.

Now, we investigate the impact of the size of the network. We designed random, strongly connected graphs with different number of nodes, and we ran the toy model of Figure 4. We compared the maximum Frobenius norms of any two agents in the network at step 20. The graphs for 20 and 100 agents are shown in Figure 10. The convergence results are shown in Table 1. The results show that the convergence speed is improved with the number of agents for a small network. In the case of 100 agents, the convergence is still higher than the smallest network of 4 agents, but worse than the other cases, probably caused by sparsity.

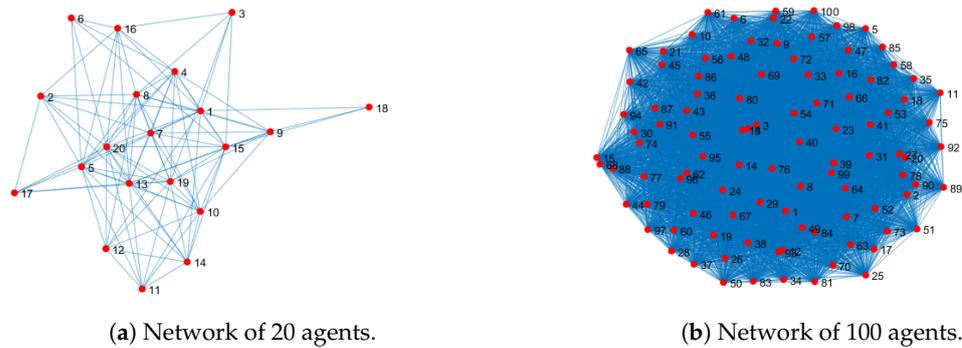


Figure 10. Random, strongly connected graphs.

Table 1. Maximum Frobenius norm between agents for different network sizes.

Number of Agents	Maximum Frobenius Norm between Any Two Agents at Step 20
4	0.28
5	0.001
6	0.0002
20	0.0008
100	0.04

Finally, we compare the convergence results after 20 iterations for a network with a fixed number of agents (nodes) with a different number of edges, meaning lower-density communication. We tested a network of 50 agents. The results are shown in Table 2. It can be seen that an overly dense network slows down the convergence.

Table 2. Maximum Frobenius norm between any two agents at step 20 for different network densities with a constant number of agents of 50.

Number of Edges	Max Norm
1051	0.059
851	0.055
651	0.060
451	0.044
251	0.027

5. Future Work

Under the present scheme, a global parameter must be known for each agent, namely, the number of agents in the network. Future research could involve how to exactly estimate this parameter using the consensus algorithm or finding a different consensus scheme that does not require this parameter.

Moreover, the network studied is static and undirected; hence, the main generalization of the method accounts for time-varying directed graphs. Further research could compare the time and energy costs of this methodology to the paired treatment of optimization and consensus.

6. Conclusions

An algorithm to recover a low-rank and a sparse matrix in an undirected static graph is proposed and validated. The consensus mechanism in static undirected graphs was coupled with a principal component pursuit problem, where there is a global noisy and corrupted matrix M and each agent knows a part of this matrix, with possible redundant observations between agents. Reaching a consensus, they agree on the global original M . To overcome the difficulties that are caused by redundant data, each agent updates only a subset of its own entries. This consensus scheme converges exponentially fast, given the properties of the connected graph described in the associated adjacency matrix A .

Each agent performs convex relaxation of the optimization problem, where an alternate direction method is employed to find the low-rank and sparse matrices of M . This alternate direction method speeds up the calculation since the sub-problems (when either L or S is fixed) are significantly more manageable than the coupled problem. Under some weak assumptions, these convexification and uncoupling processes recover the exact solution.

An application of distributed image recovery from a corrupted global matrix is shown, along with a method to handle colored images. A low-rank matrix is employed to recover the corrupted entries of the matrix found by the consensus.

A brief comparison of random, strongly connected networks of different sizes shows a faster convergence with a higher number of agents. However, for very dense networks, the performance declines. For a fixed number of agents, a lower edge density improves the convergence speed.

Author Contributions: Conceptualization, G.S. and J.D.V.; methodology, G.S. and J.D.V.; software, G.S.; validation, G.S. and J.D.V.; formal analysis, G.S. and J.D.V.; investigation, G.S. and J.D.V.; resources, G.S. and J.D.V.; data curation, J.D.V.; writing—original draft preparation, G.S. and J.D.V.; writing—review and editing, G.S. and J.D.V.; visualization, G.S.; supervision, G.S. and J.D.V.; project administration, J.D.V.; funding acquisition, J.D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Institución Universitaria Pascual Bravo.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Garin, F.; Schenato, L., A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms. In *Networked Control Systems*; Bemporad, A., Heemels, M., Johansson, M., Eds.; Springer: London, UK, 2010; pp. 75–107. [[CrossRef](#)]
2. Kuhn, F.; Moses, Y.; Oshman, R. Coordinated consensus in dynamic networks. In Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, San Jose, CA, USA, 6–8 June 2011; PODC '11, pp. 1–10. [[CrossRef](#)]
3. Kia, S.S.; Van Scoy, B.; Cortes, J.; Freeman, R.A.; Lynch, K.M.; Martinez, S. Tutorial on Dynamic Average Consensus: The Problem, Its Applications, and the Algorithms. *IEEE Control Syst. Mag.* **2019**, *39*, 40–72. [[CrossRef](#)]
4. Guan, J.C.; Ren, H.W.; Tan, G.L. Distributed Dynamic Event-Triggered Control to Leader-Following Consensus of Nonlinear Multi-Agent Systems with Directed Graphs. *Entropy* **2024**, *26*, 113. [[CrossRef](#)] [[PubMed](#)]
5. Guo, S.; Xie, L. Adaptive Neural Consensus of Unknown Non-Linear Multi-Agent Systems with Communication Noises under Markov Switching Topologies. *Mathematics* **2024**, *12*, 133. [[CrossRef](#)]
6. Du, Z.; Zhang, J.; Fu, Y.; Huang, M.; Liu, L.; Li, Y. A Scalable and Trust-Value-Based Consensus Algorithm for Internet of Vehicles. *Appl. Sci.* **2023**, *13*, 10663. [[CrossRef](#)]
7. Mustafa, A.; Islam, M.N.; Ahmed, S. Average Convergence for Directed & Undirected Graphs in Distributed Systems. *Comput. Syst. Sci. Eng.* **2021**, *37*, 399–413.
8. Kan, Z.; Yucelen, T.; Doucette, E.; Pasiliao, E. A finite-time consensus framework over time-varying graph topologies with temporal constraints. *J. Dyn. Syst. Meas. Control* **2017**, *139*, 071012. [[CrossRef](#)]

9. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and Cooperation in Networked Multi-Agent Systems. *Proc. IEEE* **2007**, *95*, 215–233. [[CrossRef](#)]
10. Xiao, L.; Boyd, S.; Kim, S.J. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.* **2007**, *67*, 33–46. [[CrossRef](#)]
11. Jolliffe, I.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [[CrossRef](#)]
12. Basri, R.; Jacobs, D. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 218–233. [[CrossRef](#)]
13. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407. [[CrossRef](#)]
14. Scherl, I.; Strom, B.; Shang, J.K.; Williams, O.; Polagye, B.L.; Brunton, S.L. Robust principal component analysis for modal decomposition of corrupt fluid flows. *Phys. Rev. Fluids* **2020**, *5*, 054401. [[CrossRef](#)]
15. Huang, L.; Liu, M.; Li, B.; Chitrakar, B.; Duan, X. Terahertz Spectroscopic Identification of Roast Degree and Variety of Coffee Beans. *Foods* **2024**, *13*, 389. [[CrossRef](#)] [[PubMed](#)]
16. Candes, E.J.; Li, X.; Ma, Y.; Wright, J. Robust Principal Component Analysis? *arXiv* **2009**, arXiv:0912.3599.
17. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [[CrossRef](#)]
18. Livani, M.A.; Abadi, M. Distributed PCA-based anomaly detection in wireless sensor networks. In Proceedings of the 2010 International Conference for Internet Technology and Secured Transactions, London, UK, 8–11 November 2010; pp. 1–8.
19. Schölkopf, B.; Platt, J.; Hofmann, T. In-Network PCA and Anomaly Detection. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*; MIT Press: Cambridge, MA, USA, 2007; pp. 617–624.
20. Borgne, Y.L.; Raybaud, S.; Bontempi, G. Distributed Principal Component Analysis for Wireless Sensor Networks. *arXiv* **2010**, arXiv:1003.1967.
21. Alimisis, F.; Davies, P.; Vandereycken, B.; Alistarh, D. Distributed Principal Component Analysis with Limited Communication. *arXiv* **2021**, arXiv:2110.14391.
22. Qu, Y.; Ostrouchov, G.; Samatova, N.; Geist, A. Principal Component Analysis for Dimension Reduction in Massive Distributed Data Sets. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan, 9–12 December 2002.
23. Eckart, C.; Young, G.M. The approximation of one matrix by another of lower rank. *Psychometrika* **1936**, *1*, 211–218. [[CrossRef](#)]
24. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic Decomposition by Basis Pursuit. *SIAM J. Sci. Comput.* **1998**, *20*, 33–61. [[CrossRef](#)]
25. Scaman, K.; Bach, F.; Bubeck, S.; Lee, Y.T.; Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. *arXiv* **2017**, arXiv:1702.08704.
26. Lan, G.; Lee, S.; Zhou, Y. Communication-Efficient Algorithms for Decentralized and Stochastic Optimization. *arXiv* **2017**, arXiv:1701.03961.
27. Wei, E.; Ozdaglar, A. Distributed Alternating Direction Method of Multipliers. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 5445–5450. [[CrossRef](#)]
28. Kang, Z.; Peng, C.; Cheng, Q. Robust PCA Via Nonconvex Rank Approximation. In Proceedings of the 2015 IEEE International Conference on Data Mining, Atlantic City, NJ, USA, 14–17 November 2015; pp. 211–220. [[CrossRef](#)]
29. Candes, E.J.; Recht, B. Exact Matrix Completion via Convex Optimization. *arXiv* **2008**, arXiv:0805.4471.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.