Optimization Project: Compressive Robust PCA

1. Problem Statement

The goal of this project is to perform **Background Subtraction** on video streams. Mathematically, a video can be represented as a matrix M, where each column is a flattened video frame. We model this matrix as the sum of two distinct components:

$$M = L + S$$

where:

- L is a **Low-Rank** matrix representing the static background (correlated across frames).
- S is a **Sparse** matrix representing the foreground objects (moving people, cars, noise).

To recover these components, we solve the **Robust Principal Component Analysis (RPCA)** problem:

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

where $\|\cdot\|_*$ is the Nuclear Norm (sum of singular values) encouraging low rank, and $\|\cdot\|_1$ is the L_1 norm encouraging sparsity.

2. Optimization Framework: ADMM

Since the objective function contains non-smooth terms (L_1 and Nuclear norms), Gradient Descent is not applicable. We utilize the **Alternating Direction Method of Multipliers (ADMM)**.

We form the Augmented Lagrangian:

$$\mathcal{L}(L, S, Y) = \|L\|_* + \lambda \|S\|_1 + \langle Y, M - L - S \rangle + \frac{\rho}{2} \|M - L - S\|_F^2$$

where Y is the dual variable and ρ is the penalty parameter. ADMM solves this by iterating three steps:

1. **L-Update (Background):** Minimizing for L leads to the **Singular Value Thresholding (SVT)** operator:

$$L^{k+1} = \mathcal{D}_{\frac{1}{\rho}} \left(M - S^k + \frac{1}{\rho} Y^k \right)$$

2. **S-Update (Foreground):** Minimizing for S leads to the **Soft Thresholding** operator:

$$S^{k+1} = \mathcal{S}_{\frac{\lambda}{\rho}} \left(M - L^{k+1} + \frac{1}{\rho} Y^k \right)$$

3. **Dual Update:**

$$Y^{k+1} = Y^k + \rho \big(M - L^{k+1} - S^{k+1} \big)$$

3. Compressive Online Modification

While standard ADMM guarantees the global optimum, it requires storing the entire video in memory (Batch Mode) and performing expensive SVDs. To achieve real-time performance on high-resolution video, we implement a **Compressive Online** approach.

3.1. Basis Factorization

We assume the low-rank background L lies in a subspace spanned by a basis U:

$$L = Uv$$

Instead of learning U iteratively via SVD, we use a **Robust Median Initialization**. Since the background is static for the majority of the time, the pixel-wise median of N frames provides a mathematically robust estimate of the subspace U.

3.2. Compressive Sensing Projection

For each incoming frame x (vectorized as m), we want to find the scaling coefficient v (representing lighting changes) that best fits the background basis U.

To accelerate computation, we apply a **Compressive Mask**. We only observe a subset of pixels Ω (where $|\Omega| \approx 0.1 \cdot d$). We solve the projection problem on this subset:

$$v = \arg\min_{v} \|P_\Omega(m) - v \cdot P_\Omega(u)\|_2^2$$

This has a closed-form solution:

$$v = \frac{\langle u_\Omega, m_\Omega \rangle}{\langle u_\Omega, u_\Omega \rangle + \varepsilon}$$

3.3. Sparse Reconstruction

Once v is computed, we reconstruct the full background $L = v \cdot U$ (recovering pixels we did not even sample). The foreground is extracted using a thresholding operator similar to the ADMM S-step, implemented via a shifted ReLU to act as a “Digital Gate”:

$$S = \mathrm{ReLU}(|M - L| - \tau)$$