Designing Constrained Projections for Compressed Sensing: Mean Errors and Anomalies with Coherence

Supplemental Material

Dhruv Shah[†], Alankar Kotwal[⋆], and Ajit Rajwade[‡]

[†]Dept. of Electrical Engineering, Indian Institute of Technology Bombay
[⋆]Robotics Institute, Carnegie Mellon University
[‡]Dept. of Computer Science & Engineering, Indian Institute of Technology Bombay

Reproducible Research

The author's implementation of algorithms, example usage, trained models and designed matrices can be found in the Github repository `PrieureDeSion/optimizing-scs-spc`.

4.3 Proposed Design Approach

Computing Gradients

For a single Gaussian component with covariance matrix $\mathbf{K}_{\epsilon,j}$, we have from (14)

$$\mathcal{M}_{\Phi,j} = \text{trace}\Big\{ \underbrace{\mathbf{\Sigma}_j - \mathbf{\Sigma}_j \mathbf{\Phi}^T (\mathbf{\Phi}\mathbf{\Sigma}_j\mathbf{\Phi}^T + \sigma_{\mathbf{n}}^2 \boldsymbol{I})^{-1} \mathbf{\Phi}\mathbf{\Sigma}_j}_{\mathbf{K}_{\epsilon,j}} \Big\}$$

For ease of representation, let's choose $\mathbf{B} = \mathbf{K}_{\epsilon,j}^{-1}$. We define function $g : \mathbb{R}^{m\times n} \to \mathbb{R}$ as $g(\mathbf{B}) = \text{trace}(\mathbf{B}^{-1})$, thus giving us

$$\mathcal{M}_{\Phi,j} = g(\mathbf{B})$$

Invoking the chain rule for matrix derivatives,

$$\begin{aligned}
\frac{\partial g(\mathbf{B})}{\partial \Phi_{\rho\omega}} &= \text{trace}\left[\Big(\frac{\partial g(\mathbf{B})}{\partial \mathbf{B}} \Big)^T \frac{\partial \mathbf{B}}{\Phi_{\rho\omega}} \right] \\
&= \text{trace}\left[-\mathbf{B}^{-2T} \frac{\partial \mathbf{B}}{\Phi_{\rho\omega}} \right] \\
&= -\text{trace}\left[\mathbf{B}^{-2T} \frac{1}{\sigma_{\mathbf{n}}^2} \Big(\mathbf{\Phi}^T \mathbf{J}^{\rho\omega} + \mathbf{J}^{\omega\rho}\mathbf{\Phi} \Big) \right] \\
&= -\frac{1}{\sigma_{\mathbf{n}}^2} \text{trace}\left[\mathbf{B}^{-2T} \Big(\mathbf{\Phi}^T \mathbf{J}^{\rho\omega} + \mathbf{J}^{\omega\rho}\mathbf{\Phi} \Big) \right] \\
&= -\frac{1}{\sigma_{\mathbf{n}}^2} \text{trace}\left[\mathbf{K}_{\epsilon,j}^2 \Big(\mathbf{\Phi}^T \mathbf{J}^{\rho\omega} + \mathbf{J}^{\omega\rho}\mathbf{\Phi} \Big) \right]
\end{aligned}$$

where $(\mathbf{J}^{\rho\omega})_{\chi\zeta} = \uparrow_{\rho\chi}\uparrow_{\omega\zeta}$ and $\uparrow_{(.)}$ is the Kronecker delta function. Thus, we get

$$\frac{\partial \mathcal{M}_{\Phi,j}}{\partial \Phi_{\rho\omega}} = -\frac{1}{\sigma_{\mathbf{n}}^2} \text{trace}\Big\{ \mathbf{K}_{\epsilon,j}^2 (\mathbf{\Phi}^T \mathbf{J}^{\rho\omega} + \mathbf{J}^{\omega\rho}\mathbf{\Phi}) \Big\}$$

Taking derivatives on both sides of (15) then gives

$$\frac{\partial \mathcal{M}_\Phi}{\partial \Phi_{\rho\omega}} = \sum_{j=1}^{c} \pi_j \cdot \frac{\partial \mathcal{M}_{\Phi,j}}{\partial \Phi_{\rho\omega}}$$

5. Evaluation – More Results

This section contains some additional results of the reconstruction of natural images from two different unseen datasets – the test set from the Berkeley Segmentation Data Set (BSDS500) and the INRIA Holidays Data Set. The measurements were taken according to the acquisition model described in the paper (see section 2), corrupted by 1% additive white Gaussian noise. The 4 different measurement-recovery setups are described below:

1. Measurements made using a $(m \times 256)$ matrix with entries drawn from a uniform random [0, 1] distribution. The underlying signal, assumed sparse in the DCT dictionary, is recovered by solving the BPDN problem (1) using the SPGL1 solver package.
2. Measurements made using a $(m \times 256)$ matrix optimized according to algorithm 1 (direction extension of the approach proposed in [7]). The underlying signal, assumed sparse in the DCT dictionary, is recovered by solving the BPDN problem (5) using the SPGL1 solver package.
3. Measurements made using a $(m \times 256)$ matrix with entries drawn from a uniform random [0, 1] distribution. The underlying signal is recovered under the SCS framework [21], using the PLE.
4. Measurements made using a $(m \times 256)$ matrix optimized according to the proposed algorithm (see Section 4). The underlying signal is recovered under the SCS framework [21], using the PLE.

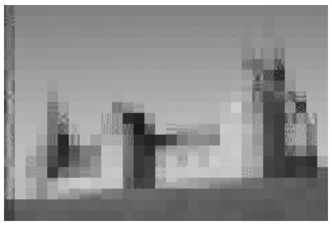
Note: For 3 & 4, a GMM with 100 components was trained on 20,000 natural image patches (16×16 each) from the BSDS500 training set. The learned GMM is available as `gmm_train/trained_model.mat`(46M).

For each image, we take measurements at four different measurement levels $m = \{32, 64, 96, 128\}$ and in each of the above setups. Reconstruction results for some images are given below.

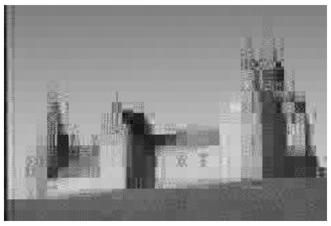
1. `bsds500/test/201080`

$m = 32$

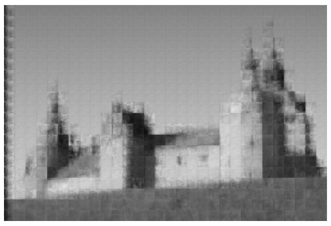
1



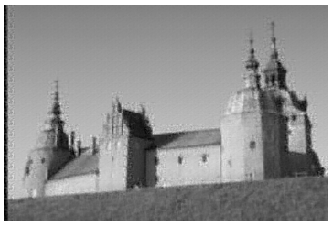
2



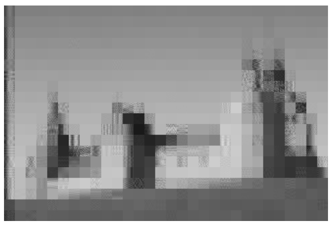
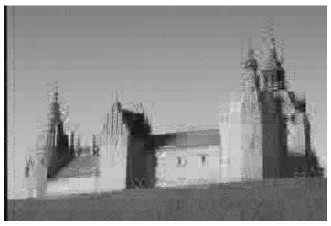
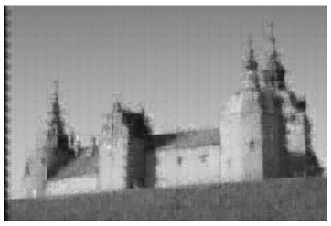
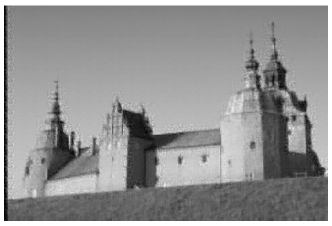
3



4

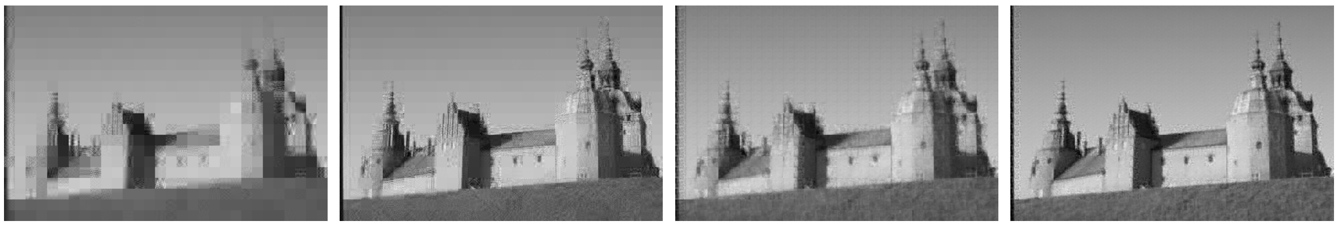


$m = 64$

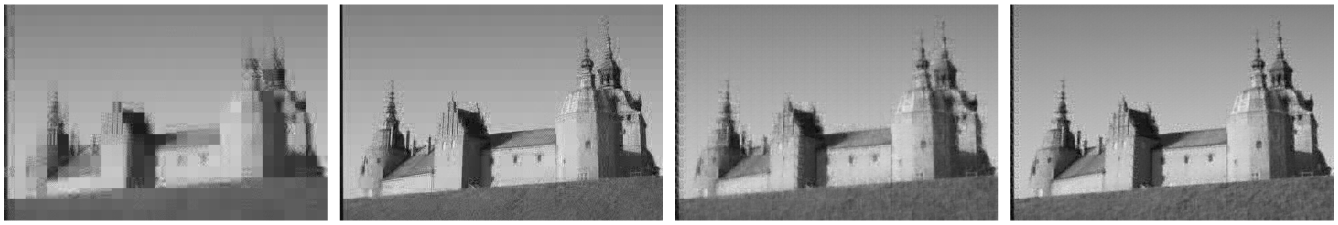





$m = 96$

1 2 3 4

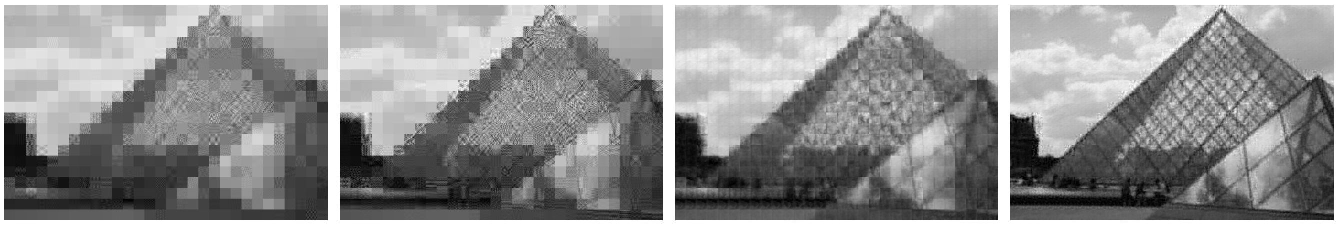


$m = 128$

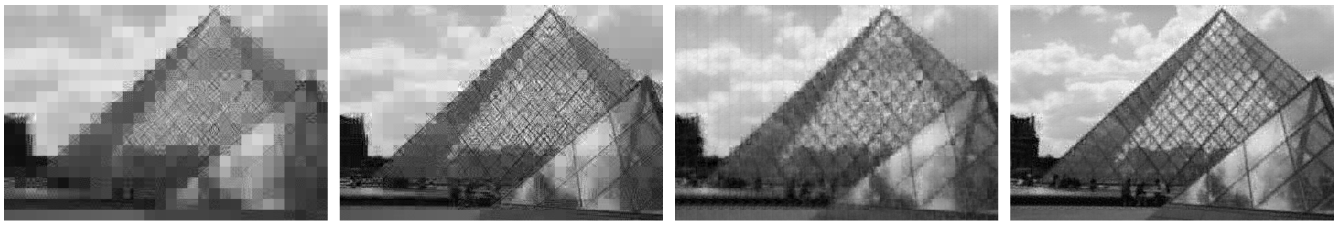


2. `bsds500/test/223060`

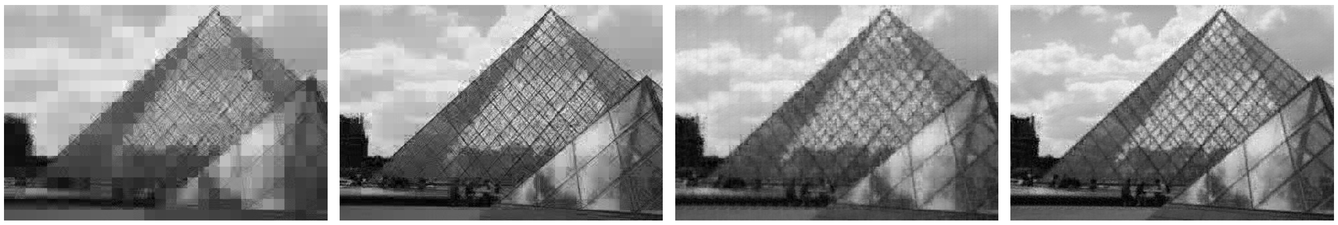
$m = 32$



$m = 64$

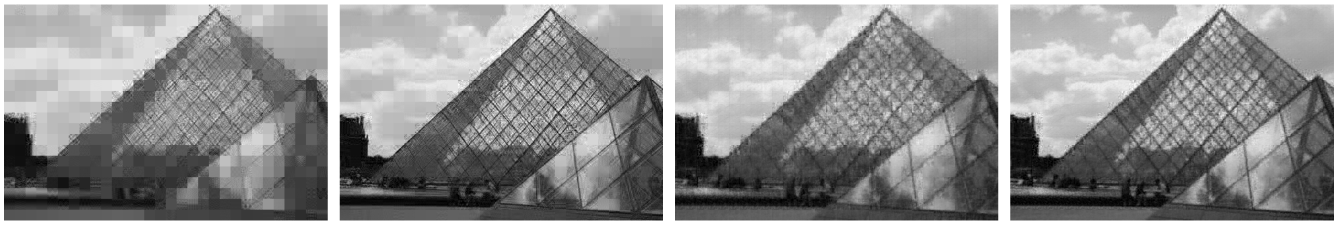


$m = 96$



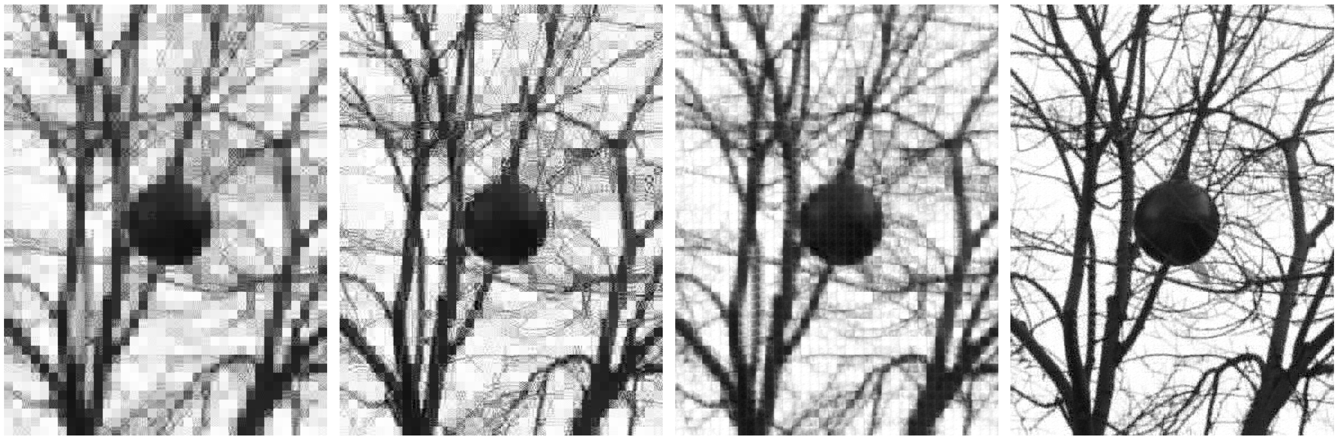
$m = 128$

1 2 3 4

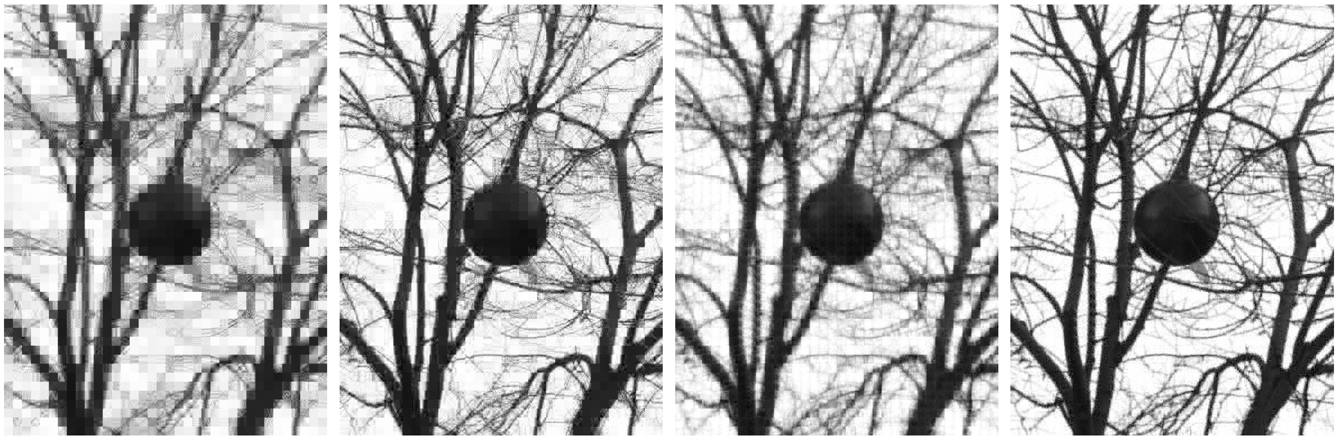


3. `inria_holidays/merry_winter0024`

$m = 32$

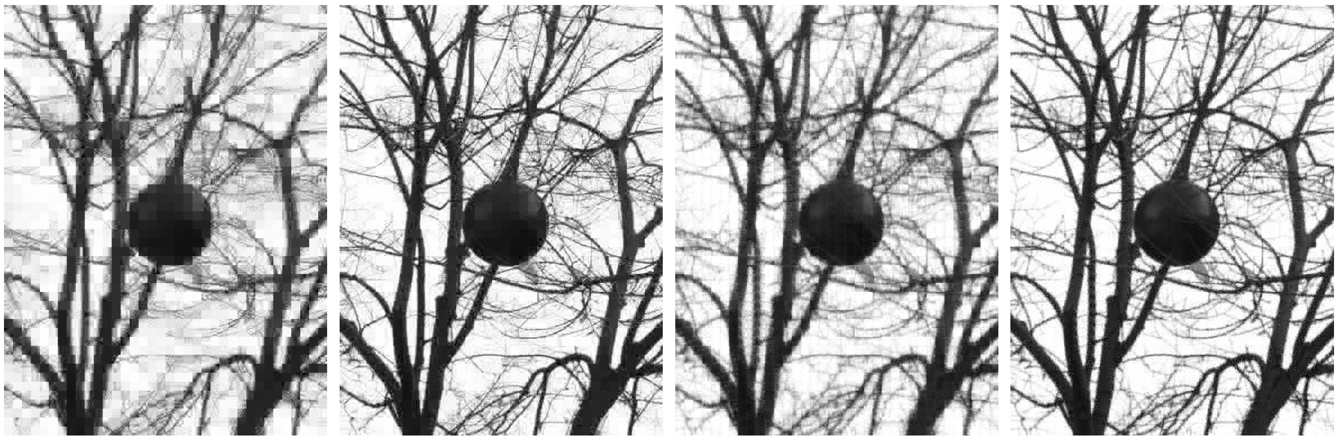


$m = 64$

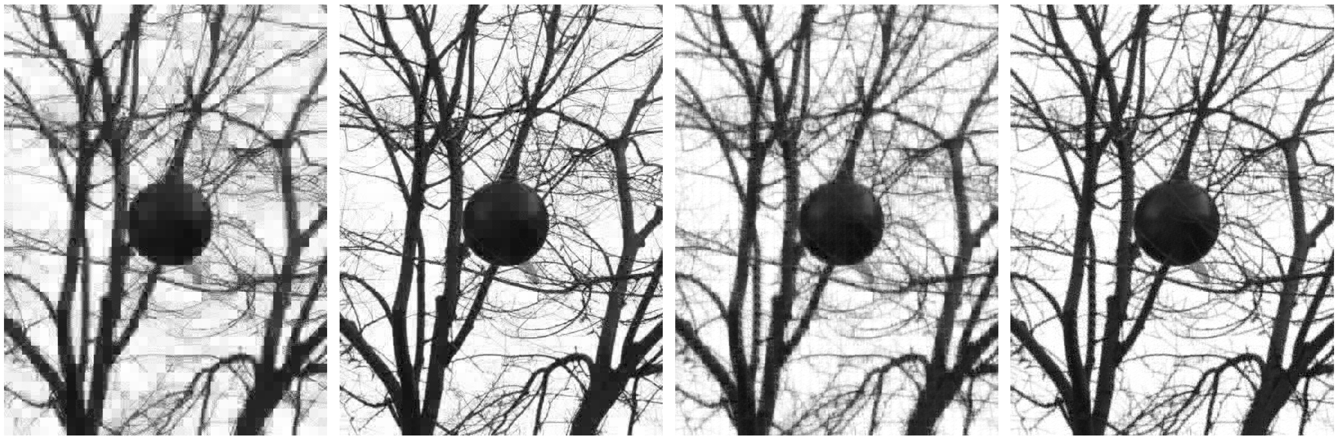


$m = 96$

1 2 3 4

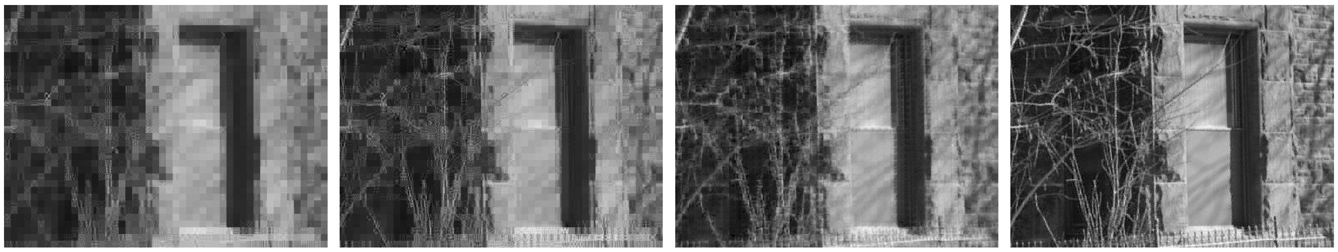


$m = 128$

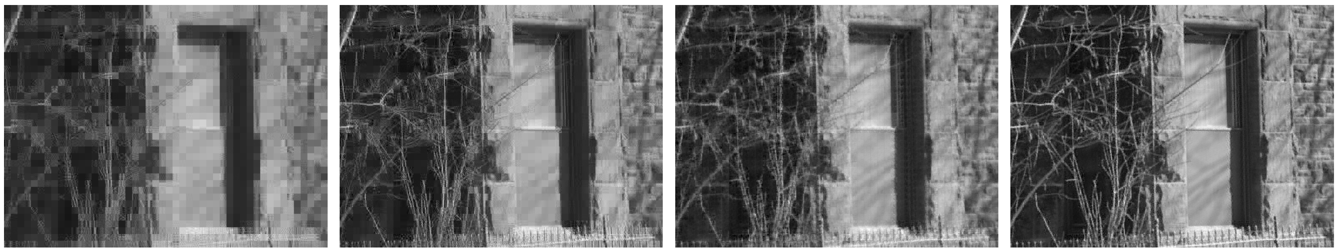


4. `inria_holidays/pippin_city65`

$m = 32$

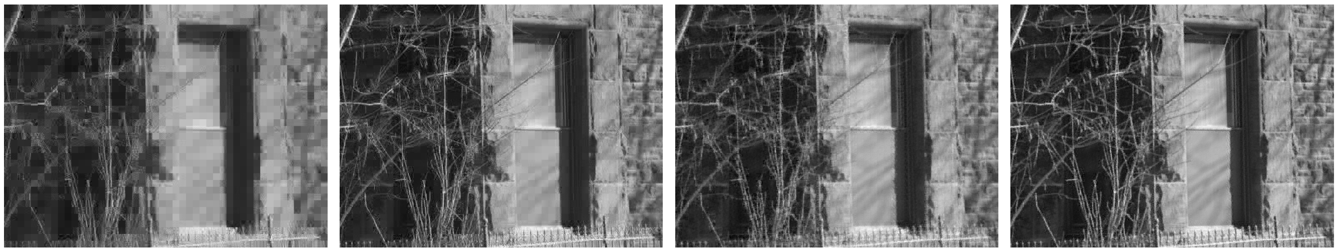


$m = 64$

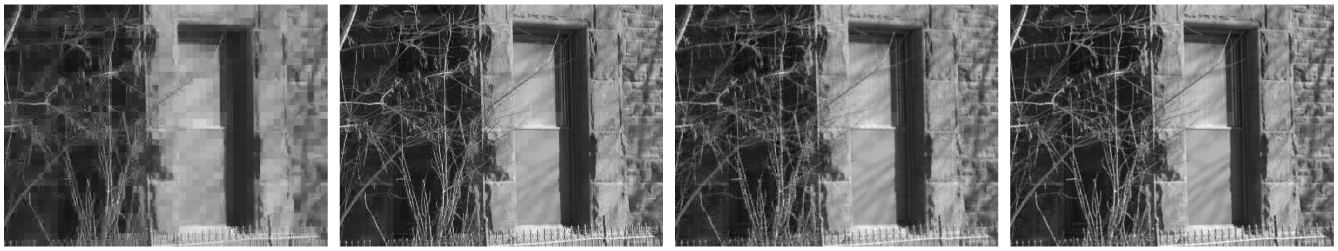


$m = 96$

1 2 3 4

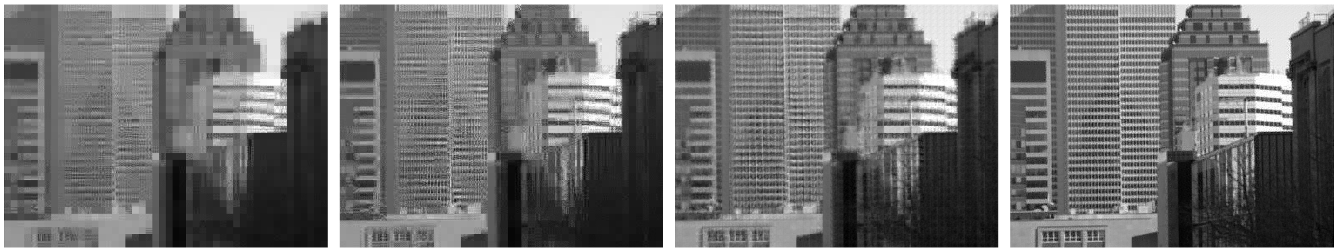


$m = 128$

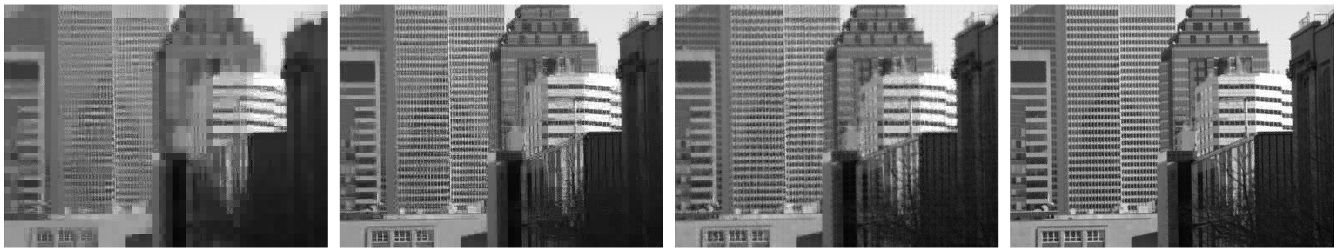


5. `inria_holidays/pippin_city69`

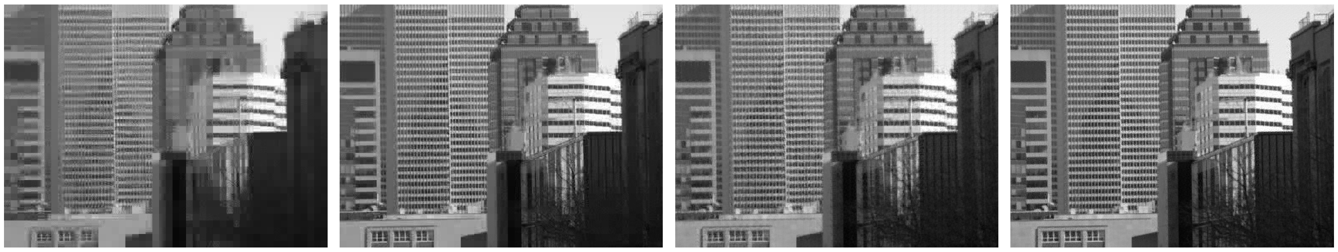
$m = 32$



$m = 64$

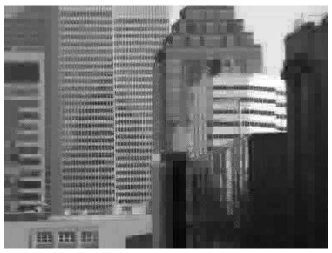


$m = 96$

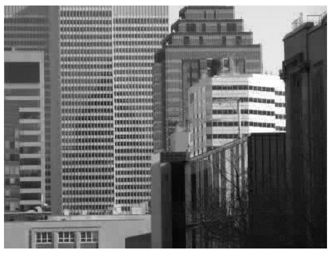


$m = 128$

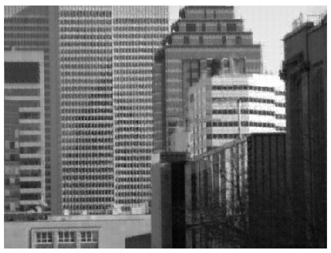
1



2



3



4