Image Super-Resolution using Bayesian Methods

Course Project for *CS 736: Medical Image Computing*, Spring 2021

Harshit Varma (190100055)
Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay
Mumbai, Maharashtra, India
harshitvarma@cse.iitb.ac.in

Gaurav P (190050037)
Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay
Mumbai, Maharashtra, India
gauravprakash@cse.iitb.ac.in

I. Introduction

A. Problem

From a set of low-resolution (LR) images of the same scene, we aim to estimate a single high-resolution (HR) image. This is also known as the 'Multi-Frame Super-Resolution' (MFSR) problem.

B. Motivation

Many vision applications such as satellite imaging, medical imaging, surveillance and forensic, and astronomical imaging require high-resolution (HR) image which usually exceed the abilities of available digital cameras. Using high-resolution cameras is usually limited by its high price, large size, or sensor manufacturing limitations. Thus, the use of image processing techniques to estimate the HR image from a set of low-quality LR images acts as a powerful and cheaper alternative [1].
The set of LR images could also be consecutive frames in a low-resolution video, and thus, MFSR approaches can also be applied to video super-resolution applications.

II. Background

Tipping et al. suggested a Bayesian approach that estimates the image registration and the PSF parameters by marginalizing the likelihood over the unknown HR image using a Gaussian prior, and then estimating the HR image by keeping the previously estimated parameters fixed.

Pickup et al. improved upon this approach by marginalizing over the registration parameters, which allowed the use of more suitable priors.

Our approach largely follows the approach suggested by Tipping et al.

III. Bayesian Formulation

Let $\{y_k \in \mathbb{R}^M\}_{k=1}^K$ be the set of LR images. Let $x \in \mathbb{R}^N$ be the corresponding HR image. We assume all images have been vectorized/unrolled and the corresponding intensities scaled to lie between $\left(-\frac{1}{2}, \frac{1}{2}\right)$.

A. Observation/Acquisition Model

The set of LR images are assumed to be obtained from the HR image by applying shifts, rotations and finally downsampling using a PSF.
This can be formulated as follows:

$$\begin{aligned} y_k &= W_k x + \epsilon_k \\ \epsilon_k(i) &\sim \mathcal{N}(0, \sigma^2) \\ W_{ij} &= \frac{W'_{ij}}{\sum_{j=1}^N W'_{ij}} \\ W'_{ij} &= \exp\left(-\frac{\|v_j - u_i\|^2}{\gamma^2}\right) \\ u_i &= R_k(v_i - \bar{v}) + \bar{v} + s_k \\ R_k &= \begin{pmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{pmatrix} \end{aligned}$$

- $W_k \in \mathbb{R}^{M \times N}$: The transformation matrix for the k^{th} image. Actually its a convolution operation where the kernel has maximum values at its centre and values decays as gaussian wrt to the distance between the centre pixel and the pixel of consideration.
- $\epsilon_k \in \mathbb{R}^M$: The additive Gaussian noise with variance σ^2. σ is fixed, and is not estimated from the image.
- $v_j \in \mathbb{R}^2$: The 2D-coordinate of the j^{th} pixel in the $n \times n$ HR image, e.g $v_0 = (0, 0)$
- $u_i \in \mathbb{R}^2$: 4 (the downscaling factor) times the 2D-coordinate of the i^{th} pixel in the $m \times m$ LR image
- $\gamma \in \mathbb{R}$: The PSF width parameter (to be estimated). It also represents the variance of the gaussian which models the PSF.
- $\bar{v} \in \mathbb{R}^2$: The coordinates of the HR image center. i.e $\left(\frac{N}{2}, \frac{N}{2}\right)$
- $s_k \in \mathbb{R}^2$: The shift for the k^{th} LR image (to be estimated)
- $\theta_k \in \mathbb{R}$: The rotation angle for the k^{th} LR image (to be estimated)

B. Likelihood

From the observation model, we get:

$$p(y_k | x, s_k, \theta_k, \gamma) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left(-\frac{1}{2\sigma^2}\|y_k - W_k x\|^2\right)$$

Thus, the likelihood will be:

$$\begin{aligned}\mathcal{L} &= p(\{y_k\}_{k=1}^K | x, \{s_k, \theta_k\}_{k=1}^K) \\ &= \prod_{k=1}^K p(y_k | x, s_k, \theta_k, \gamma) \\ \log \mathcal{L} &= \sum_{k=1}^K \log p(y_k | x, s_k, \theta_k, \gamma) \\ &= -\frac{KM}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{k=1}^K \|y_k - W_k x\|^2\end{aligned}$$

C. Prior

1) Gaussian Prior:

$$\begin{aligned}p(x) &= \frac{1}{(2\pi \det(Z_x))^{\frac{N}{2}}} \exp\left(-\frac{1}{2} x^T Z_x^{-1} x\right) \\ \log p(x) &= -\frac{N}{2} \log(2\pi \det(Z_x)) - \frac{1}{2} x^T Z_x^{-1} x\end{aligned}$$

Where $Z_x \in \mathbb{R}^{N \times N}$, and

$$Z_x(i,j) = A \cdot \exp\left(-\frac{\|v_i - v_j\|^2}{r^2}\right)$$

A, r are constants that model the prior strength and the correlation length scale respectively. These are fixed, and are not estimated. Thus, Z_x remains constant throughout, and thus can be pre-computed and re-used.

2) MRF Prior: We use a 4 neighbourhood system and a Huber penalty on cliques of size 2 for the prior.

$$\begin{aligned}p(x) &= \frac{1}{Z} \exp\left(-\frac{1}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}(i)} V(x_i - x_j)\right) \\ \log p(x) &= -\log Z - \frac{1}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}(i)} V(x_i - x_j) \\ V(u) &= \frac{1}{2}|u|^2 \text{ (for } |u| \le h) \\ &= h|u| - \frac{1}{2}h^2 \text{ (for } |u| > h)\end{aligned}$$

3) TV Prior:

$$\begin{aligned}p(x) &\propto \exp(-\alpha \text{TV}(x)) \\ \log p(x) &\propto -\alpha \text{TV}(x) \\ \text{TV}(x) &= \sum_{i=1}^n \sum_{j=1}^n \sqrt{(\Delta_{i,j}^h(x)^2 + \Delta_{i,j}^v(x)^2)} \\ \Delta_{i,j}^h(x) &= x_{i,j} - x_{i-1,j} \\ \Delta_{i,j}^v(x) &= x_{i,j} - x_{i,j-1}\end{aligned}$$

D. Posterior

$$\begin{aligned}\mathcal{P} &= p(x \mid \{y_k, s_k, \theta_k\}_{k=1}^K, \gamma) \\ &= \frac{p(x) \cdot p\left(\{y_k\}_{k=1}^K \mid x, \{s_k, \theta_k\}_{k=1}^K, \gamma\right)}{p\left(\{y_k\}_{k=1}^K \mid \{s_k, \theta_k\}_{k=1}^K, \gamma\right)}\end{aligned}$$

1) Using the Gaussian Prior: As the likelihood and the prior are both Gaussian, the posterior, which is a product of two Gaussians, also comes out to be a Gaussian with the covariance $\in \mathbb{R}^{N \times N}$ and the mean $\in \mathbb{R}^N$ as:

$$\begin{aligned}C &= \left(Z_x^{-1} + \frac{1}{\sigma^2} \sum_{k=1}^K W_k^T W_k\right)^{-1} \\ \mu &= \frac{C}{\sigma^2} \sum_{k=1}^K W_k^T y_k\end{aligned}$$

Note that in this case, μ is the MAP as well as the posterior-mean-estimate.

2) Using the MRF Prior: In this case, computing the marginal likelihood in closed form is difficult, unlike the previous case. Thus, we can't compute the posterior exactly. But, for estimating only x we don't require the marginal, as it's not a function of x. Thus we can ignore that.

$$\log \mathcal{P} \propto \log p(x) + \log \mathcal{L}$$

For more flexibility, we can weigh the terms by $\beta \in [0, 1]$

$$\log \mathcal{P} \propto (1 - \beta) \log p(x) + \beta \log \mathcal{L}$$

3) Using the TV Prior: For estimating only x we don't require the marginal, as it's not a function of x. Thus we can ignore that.

$$\log \mathcal{P} \propto \log p(x) + \log \mathcal{L}$$

E. Marginal Likelihood

Using the Gaussian prior, we can compute the marginal exactly, in closed form, since we know the posterior, prior and likelihood in closed form. It comes out to be:

$$\begin{aligned}&p(y \mid \{s_k, \theta_k\}_{k=1}^K, \gamma) = \mathcal{N}(0, Z_y) \\ &Z_y = \sigma^2 I + W Z_x W^T\end{aligned}$$

Where y, W are the stacked $\{y_k\}_{k=1}^K, \{W_k\}_{k=1}^K$ respectively. The log marginal likelihood comes out to be:

$$\log \mathcal{M} = \log p(\{y_k\}_{k=1}^K \mid \{s_k\}_{k=1}^K, \{\theta_k\}_{k=1}^K, \gamma)$$

$$= -\frac{1}{2}\Bigg(\frac{1}{\sigma^2} \sum_{k=1}^K \|y_k - W_k \mu\|^2 + \mu^T Z_x^{-1} \mu + \log(\det(Z_x)) - \log(\det(C)) + 2KM \log(\sigma)\Bigg)$$

IV. Estimation and Optimization

Let 'registration parameters' denote $\{s_k\}_{k=1}^K, \{\theta_k\}_{k=1}^K$ and γ. We first estimate these parameters by minimizing $-\log(\mathcal{M})$. Note that using the marginal here allows us to not worry about x. This step always uses the Gaussian prior for marginalization.

The marginal requires computing large matrices (and it's gradients) like C, which take up a lot of memory. To fix this, we extract smaller size patches from the image center and use these for estimating the registration parameters.

For estimating the HR image x, we fix the estimated registration parameters and minimize the negative log posterior (ignoring the marginal). Here, any relevant prior can be used (eg. TV, MRF).

The original approach by Tipping et al. uses 'Scaled Conjugate Gradients' for minimization. In our approach we use the AdamW optimizer [2] along with PyTorch's Automatic differentiation package [3] (https://pytorch.org/docs/stable/autograd.html) for performing the minimizations.

V. Experimental Setup

A. Hyperparameters

16 ($K = 16$) LR images of size 43×43 were used, which were scaled to be in $\left(-\frac{1}{2}, \frac{1}{2}\right)$. The upscale factor of 4 was used. σ was set to 0.01. γ used for generating the LR images was set to 2. Shifts used while generating the LR images were sampled uniformly from $(-0.65, 0.65)$ while the angles were sampled uniformly from $(-1.35, 1.35)$. Patches of size 15×15 were extracted from the center of the LR images to perform the registration parameters estimation.

$A = 0.04$ and $r = 1$ were used for the Gaussian prior. When using the MRF Prior for estimation of the HR image, $\beta = 0.005$ and $h = 0.055$ was used. When using the TV prior, $\alpha = 15$ was used.

Initial estimates for shifts and angles while estimation was set to 0, initial estimate for γ was the upscale factor ($= 4$) and the initial estimate for the HR image was a bicubic interpolated version of the LR image.

AdamW, with a learning rate of 0.05 was used throughout the estimation. The registration parameter optimization was run for 60 iterations while the HR image optimization was run for 100 iterations.

B. Libraries and Tool

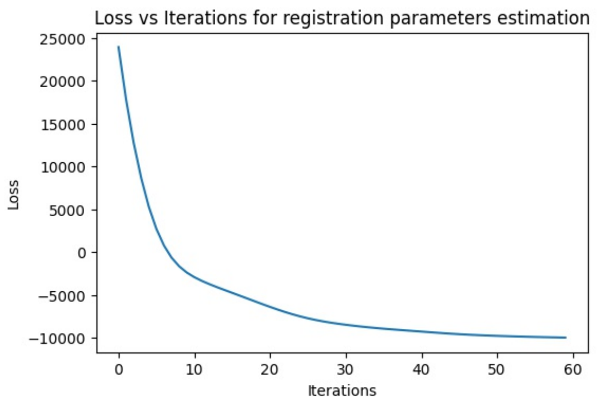
Google Colab (free version) was used to perform all the experiments. PyTorch is the central framework used for tensor processing.

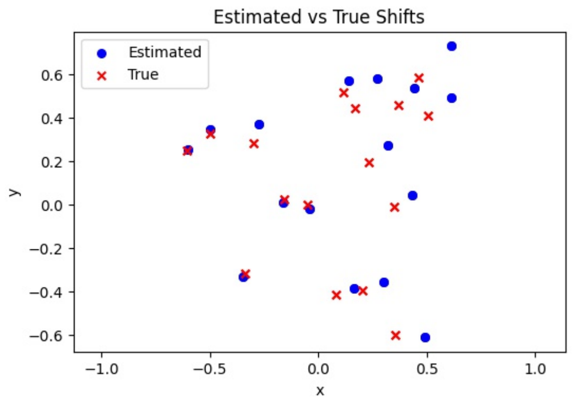
The GPUs automatically allotted by Colab kept varying between Tesla T4, Tesla P100-PCIE-16GB, and Tesla K80. While running the notebook, ensure that the allotted GPU capacity is atleast 15 GB, if not, then restart the Colab session and check whether a different GPU has been allotted.

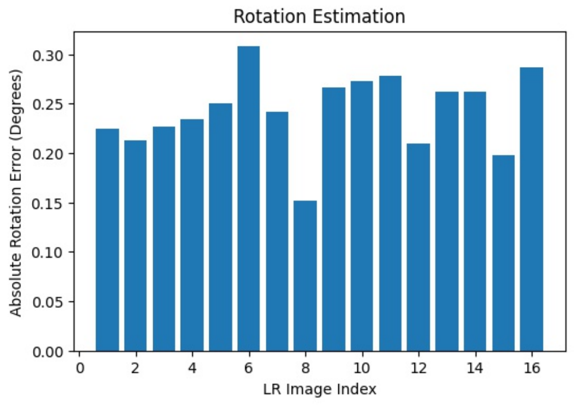
The time taken to run the entire Colab notebook, for the specified parameters, hyper-parameters and GPU is always < 5 minutes.

VI. Results

A. Registration Parameter Estimation

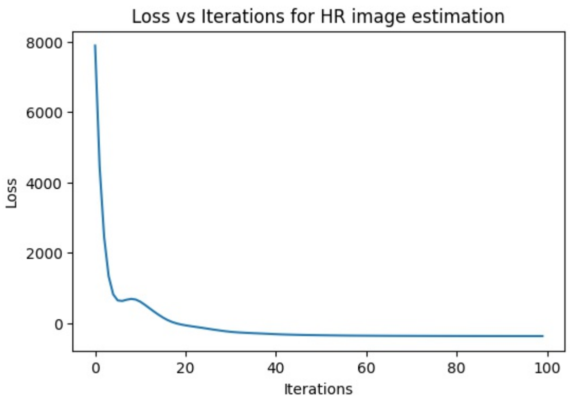






The absolute rotation error observed is almost the same for all the LR images. This maybe because of using the same initial learning rate for all registration parameters.
The estimated γ comes out to be 2.06, which is quite close to the true value of 2. The estimation of the HR image is quite sensitive to changes in the estimated γ, but not as sensitive to changes in the shifts and angles.

B. HR Image Estimation



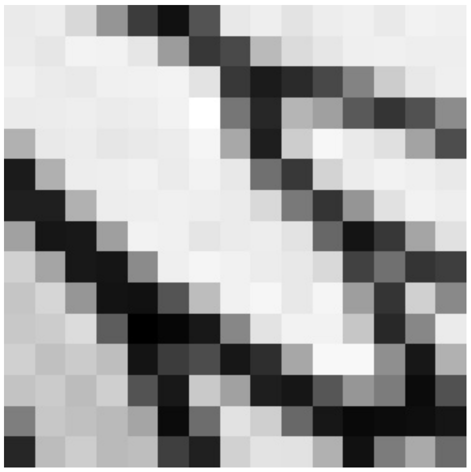


Fig. 3. Example of a patch used for registration parameters estimation

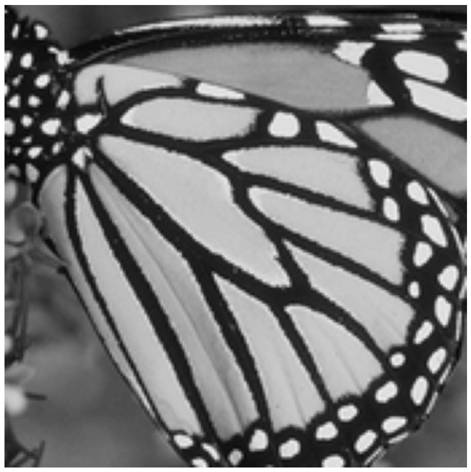


Fig. 1. The Original HR Image

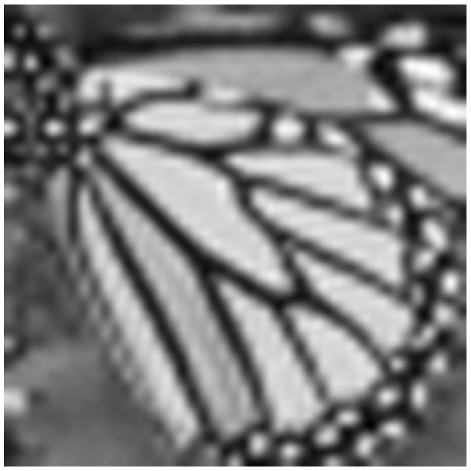


Fig. 4. Initial estimate for the SR image (PSNR = 15.57)

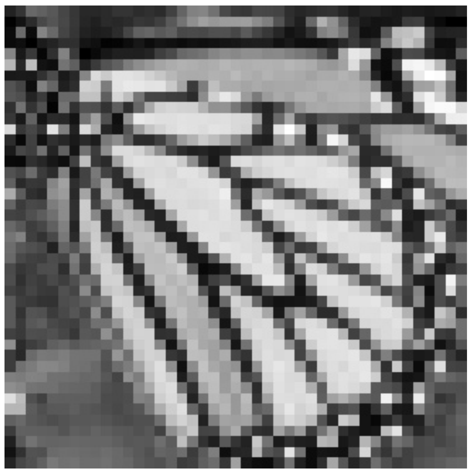


Fig. 2. One of the LR image

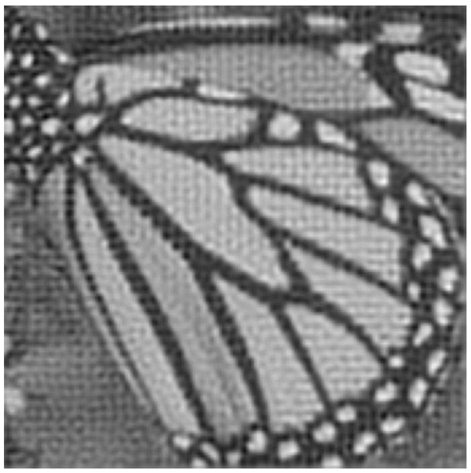


Fig. 5. Final SR image estimated using Gaussian Prior (PSNR = 18.62)

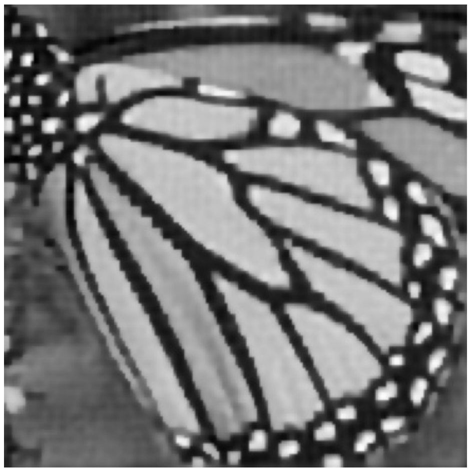


Fig. 6. Final SR image estimated using MRF Prior (PSNR = 22.05)

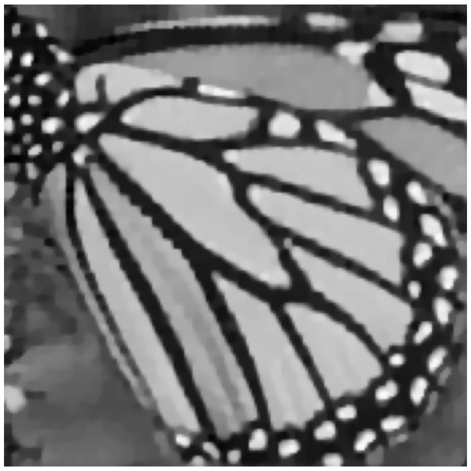


Fig. 7. Final SR image estimated using TV Prior (PSNR = 22.46)

As observed in figure 5, there some periodic artifacts in the image using a Gaussian prior. These may correspond to amplification of noise, non-optimal prior parameters, or some minor bug in the code. Although, these are not observed when estimation is performed using the MRF or the TV prior as observed in figure 6 and 7.

VII. Conclusion

In this report, we saw how the 'Multi-Frame Super-Resolution' problem can be formulated and solved in a Bayesian way. Using a Gaussian prior for computing the marginal likelihood allows us to estimate the registration parameters more accurately and gives us the freedom to use better priors (eg. MRF, TV) while estimating the HR image, which improves the visual quality of the estimated image by a lot.

References

[1] Mahmoud M. Khattab, Akram M Zeki, Ali A. Alwan, Ahmed S. Badawy, and Lalitha Saroja Thota. Multi-frame super-resolution: A survey. In *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–8, 2018. doi: 10.1109/ICCIC.2018.8782382.

[2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[3] Adam Paszke, S. Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[4] Lyndsey C Pickup, David P Capel, Stephen J Roberts, and Andrew Zisserman. Bayesian image super-resolution, continued. In *NIPS*, volume 19, 2006.

[5] Michael E Tipping, Christopher M Bishop, et al. Bayesian image super-resolution. *Advances in neural information processing systems*, pages 1303–1310, 2003.