MONTE CARLO OPTIMIZATION OF NON LOCAL MEANS DENOISING ALGORITHM

ROBU PETRU-RĂZVAN, VERZOTTI MATTEO-ALEXANDRU, VOAIDEȘ-NEGUSTOR ROBERT-IONUȚ

Abstract. Non-Local Means [1] is a powerful algorithm for image denoising, yet its high computational complexity limits real-time application. This paper replicates the Monte Carlo Non-Local Means Optimization [2], which utilizes random sampling to accelerate weight calculations without significantly degrading peak signal-to-noise ratio.

Keywords. Denoising, Monte Carlo, Optimization

1. Introduction

Image noise is a random variation of brightness or color information. In most real-life cases, it can be modeled as additive white Gaussian noise:

$$y = x + \eta \tag{1.1}$$

Where y is the noisy pixel, x the pure pixel and $\eta \sim \mathcal{N}(0, \sigma)$ is the noise.

1.1. Denoising Techniques.

Traditional denoising techniques, such as Gaussian smoothing, Median filtering or Local means, operate on the principle of locality, assuming that the true value of the pixel must be similar to the values of its neighbours. This is effective at removing noise, but it also leads to loss of fine textures and blurring of edges.

1.2. Non-Local Means.

The Non-Local Means estimates the true value of the pixel by computing a wieghted average of different patches, using a weight function that prioritizes pixels with similar structural patterns.

Given a noisy pixel, the surrounding patch $\mathbf{y}$ will be used to check for similarity with other patches. This object can be flattened into a d-pixel: $\mathbf{y} \in \mathbb{R}^d$.

The algorithm requires a set of patches $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ that are obtained from reference images.

NLM replaces the noisy pixel with the weighted average of pixels in the reference set:

$$z = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{1.2}$$

where x_i represents the center pixel in the patch $\mathbf{x_i}$, and the weight w_i measures the similarity between the match $\mathbf{y}$ and $\mathbf{x_i}$.

A standard choice for the weight function is:

$$w_i = e^{-||\mathbf{y}-\mathbf{x_i}||^2/(2h_r^2)} \tag{1.3}$$

Where h_r is a scalar parameter that controls the decay of the distribution: small values create a fast decay and ensure a focus on stricter similarity between patches, whereas larger values allow different patches to have a greater weight. And $||\cdot||$ is the Euclidian norm.

The downside of NLM is it high computational complexity. The most expensive step of this algorithms is computing the weights $\mathbf{w}_i$, which leads to $\mathcal{O}(mnd)$ operations, where m is the number of pixels to be denoised, n the number of patches and d the dimensions of the patch.

1.3. **Monte Carlo Non-Local Means (MCNLM).** This paper focuses on the implementation and effectiveness of a method developed in [2]. The main focus of the approach is to approximate 1.2 by selecting randomly k reference pixel and creating a k-subset of weights $\{w_i\}^k$. The computational complexity of this approach is $\mathcal{O}(mkd)$, which is significantly lower for $k \ll n$.

2. Monte Carlo Non-Local Means

In this section, we discuss the full implementation of the algorithm, the correctness of the approach and some other optimizations that can be added to the method.

2.1. **Sampling Patches.** There are two ways to extract reference patches. The first one is based on picking them from the original noisy, image, which is called *internal denoising*. The second method is based on having the patches taken from a large database of other images, which is called *external denoising*. The paper focuses on internal denoising.

The Monte Carlo sampling is done on the set $\mathcal{X} = \{x_1, \ldots, x_n\}$, considering each reference patch independent. The process is determined by a sequence of random variables $\{I_j\}_{j=1}^n$, where $I_j \sim \text{Bernoulli}(p_j)$. The weight will be sampled only if $I_j = 1$. The vector of these probabilities, $\mathbf{p} := [p_1, \ldots, p_n]^T$, is called the *sampling pattern* of the algorithm [2].

The main parameter of this algorithm is ξ, which is the expected value of the random variable which models the ratio between the number of the samples taken and the number of references:

$$S_n = \frac{1}{n}\sum_{j=1}^{n} I_j \tag{2.1}$$

which has:

$$\xi \overset{\text{def}}{=} \mathbb{E}[S_n] = \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[I_j] = \frac{1}{n}\sum_{j=1}^{n} p_j \tag{2.2}$$

So, an important requirement is that:

$$\sum_{j=1}^{n} = n\xi \tag{2.3}$$

S_n is called the *empirical sampling ratio* and ξ the average sampling ratio [2].

References

1. A. Buades, B. Coll, and J.-M. Morel, *A non-local algorithm for image denoising*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (San Diego, CA, USA), vol. 2, IEEE, 2005, pp. 60–65.
2. Stanley H. Chan, Todd Zickler, and Yue M. Lu, *Monte carlo non-local means: Random sampling for large-scale image filtering*, IEEE Transactions on Image Processing **23** (2014), no. 8, 3711–3725.