Binning: Understanding the Process

Mohtasim Hadi Rafi

Abstract

This project explores binning in image processing to lower signal noise. Throughout the effort, the foundations of binning, how it works and the related pros and cons were explored. The project involves generating a vector, simulating repeated measurements under consistent conditions with added randomized noise. Binning is applied at various levels, and the resulting signal-to-noise ratio (SNR) is computed. The experiments are replicated 25 times for reliability, and statistical analyses are performed on the SNR values. The impact of changing noise variance, while keeping signal mean constant, is also examined. Comparative analysis of SNR results across different noise and binning levels provides insights into the relationship between SNR and binning.

1 Introduction

Binning involves aggregating the charges of multiple pixels either horizontally, vertically, or both, creating a consolidated charge known as a ”binned pixel” or ”super pixel” [NKA+10]. This process typically includes adding the signals of adjacent pixels and converting the combined signal into digital values. The result is a reduction in noise and an enhancement of the signal-to-noise ratio (SNR). Binning is commonly used in various imaging applications, such as astronomy and microscopy, where improving SNR is crucial.

1.1 How it works

Let's consider a set of continuous data points, $A = \{x_1, x_2, x_3, \ldots, x_n\}$ where the range of the data would be $\text{Range} = \max(A) - \min(A)$ and Bin Width would be $\text{Bin Width} = \frac{\text{Range}}{\text{Number of Bins}}$. Now for k number of beans where i is the iteration count,

$$\text{Bin}_i : \min(A) + (i-1) \times \text{Bin Width} \leq x < \min(A) + i \times \text{Bin Width}$$

This process results in binned data, where each bin represents a group of values from the original data set.

1.2 Advantages of binning

Binning in image processing offers several advantages, depending on the specific application. Some common advantages includes-

- Improved Signal-to-Noise Ratio (SNR) and Faster Image Acquisition
- Increased Sensitivity
- Reduced Readout Noise
- Data Compression
- Enhanced Image Quality
- Adaptability to System Constraints

In image acquisition, binning is employed to boost the signal-to-noise ratio (SNR) in the readout signal. Thus the combined charges would overcome the read noise, even if the individual pixel values are small. This is achieved by adding noise to the binned pixel, which consolidates the signals from multiple individual pixels [ASS99, MEMA$^+$00]. Binning offers another advantage by enhancing the readout frame rate. By employing binning, the total frame rate of a system can be effectively increased. This is particularly useful for rapidly reading out highly binned low-resolution images when high speed is crucial [ZPF97, BCFS06].

1.3 Disadvantages of binning

While binning offers benefits such as improved signal-to-noise ratio (SNR) and increased frame rates, it also comes with certain disadvantages such as-

- Spatial Resolution Loss
- Color Information Loss
- Limited Flexibility

The enhanced noise performance is accompanied by a trade-off in spatial resolution. In color image sensors, binning is made complex by the existence of a color filter array (CFA). Data is usually acquired through a single CCD or CMOS sensor utilizing a CFA spatial sub-sampling process. This physical construction means that each pixel location measures only a single color. [JH12]

2 Code Implementation of Binning

2.1 Requirements

In this section binning will be implemented as per the stated requirements-

- Generate a vector of length 10240 (assuming you're repeatedly measuring the same signal under the same condition for 10240 times) with randomized noise added to each value. Reasonable scales of both signal (mean) and noise (variance) levels to be determined as needed to get enough precision level.
- Apply binning by 2, 4, 8, 16, 32, 64, 128, and 256 separately to the vector you just generated. Calculate the signal-over-noise ratio (SNR) in each case.
- Repeat previous steps at least 25 times to generate the replicated SNR results for each binning level and perform statistical analysis.
- Repeat previous steps by changing the noise (variance) level to 3 other values while keeping the mean constant.

```python
import numpy as np
import matplotlib.pyplot as plt
```

Listing 1: Importing necessary libraries

2.2 Implementation

The code implementation utilizes Python 3.11 as the programming language. The initial step involves importing essential libraries (listing 1). Then a function (listing 2) was developed to generate the specified vector. The function takes parameters for signal mean and noise variance, with the default vector length set to 10240 as per the specified requirement.

```python
def generate_vector(signal_mean, noise_variance, length=10240):
    signal = np.ones(length) * signal_mean
    noise = np.random.normal(0, np.sqrt(noise_variance), length)
    vector = signal + noise
    return vector
```

Listing 2: Generating Vector

The Signal-to-Noise Ratio (SNR) is commonly calculated using the following formula [Joh06]:

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{\text{Signal}}{\text{Noise}}\right)$$

The SNR calculation function can be written as shown in listing 3.

```python
def calculate_snr(binned_signal, axis=0, ddof=0):
    binned_signal = np.asanyarray(binned_signal)
    mean = binned_signal.mean(axis)
    standard_deviation = binned_signal.std(axis=axis, ddof=ddof)
    return 10*np.log10(abs(np.where(standard_deviation == 0, 0, mean/
    standard_deviation)))
```

Listing 3: Calculating SNR

The function (listing 3) takes signal and noise as inputs, divides the mean of the signal by the mean of the noise, and ultimately returns the result after applying a logarithmic transformation (specifically, 10*log10).

At this point, a separate function is required to execute binning and calculate the Signal-to-Noise Ratio (SNR) for each binning level. The function takes the vector to be binned and the binning factor as inputs. It begins by reshaping the vector, followed by the calculation of noise, achieved by subtracting the binning factor from the binned vector values. Ultimately, the function returns the computed SNR value using the method defined in Listing 3.

```python
def perform_binning_and_snr(signal, binning_factor):
    binned_signal = np.mean(signal.reshape(-1, binning_factor), axis=1)
    return calculate_snr(binned_signal)
```

Listing 4: Function to perform binning and calculate SNR for each binning level

The parameters for the experiment are defined in listing 5. The signal mean is established at 100.0. Three distinct noise levels (5.0, 50.0, and 500.0) were employed, with 25 replications and binning factors set to 2, 4, 8, 16, 32, 128, and 256 as specified. The experiment is executed using nested `for` loops, with the outer loop iterating through replications, the middle loop varying noise variances, and the inner loop calculating SNR for each binning factor based on a newly generated vector. The results are stored in a variable named `binning_factors`.

```python
signal_mean = 100.0
noise_variances = [5.0, 50.0, 500.0]  # Different noise levels
num_replications = 25
binning_factors = [2, 4, 8, 16, 32, 64, 128, 256]

# Results storage
snr_results = np.zeros((len(binning_factors), len(noise_variances), num_replications))

for rep in range(num_replications):
    for noise_var_idx, noise_variance in enumerate(noise_variances):
        vector = generate_vector(signal_mean, noise_variance)
        for binning_idx, binning_factor in enumerate(binning_factors):
            snr = perform_binning_and_snr(vector, binning_factor)
            snr_results[binning_idx, noise_var_idx, rep] = snr
```

Listing 5: Parameters and Main Function

Listing 6 is used to plot the SNR vs Binning Factor for different noise level.

```python
mean_snr = np.mean(snr_results, axis=2)
std_snr = np.std(snr_results, axis=2)

plt.figure(figsize=(10, 6))
for noise_var_idx, noise_variance in enumerate(noise_variances):
    plt.errorbar(binning_factors, mean_snr[:, noise_var_idx], yerr=std_snr[:,
    noise_var_idx],
                 label=f'Noise Variance = {noise_variance}')

plt.xscale('log', base=2)
plt.xlabel('Binning Factor')
plt.ylabel('SNR (dB)')
plt.title('SNR vs Binning Factor for Different Noise Levels')
plt.legend()
plt.grid(True)
plt.show()
```

Listing 6: Plotting the SNR vs Binning Factor

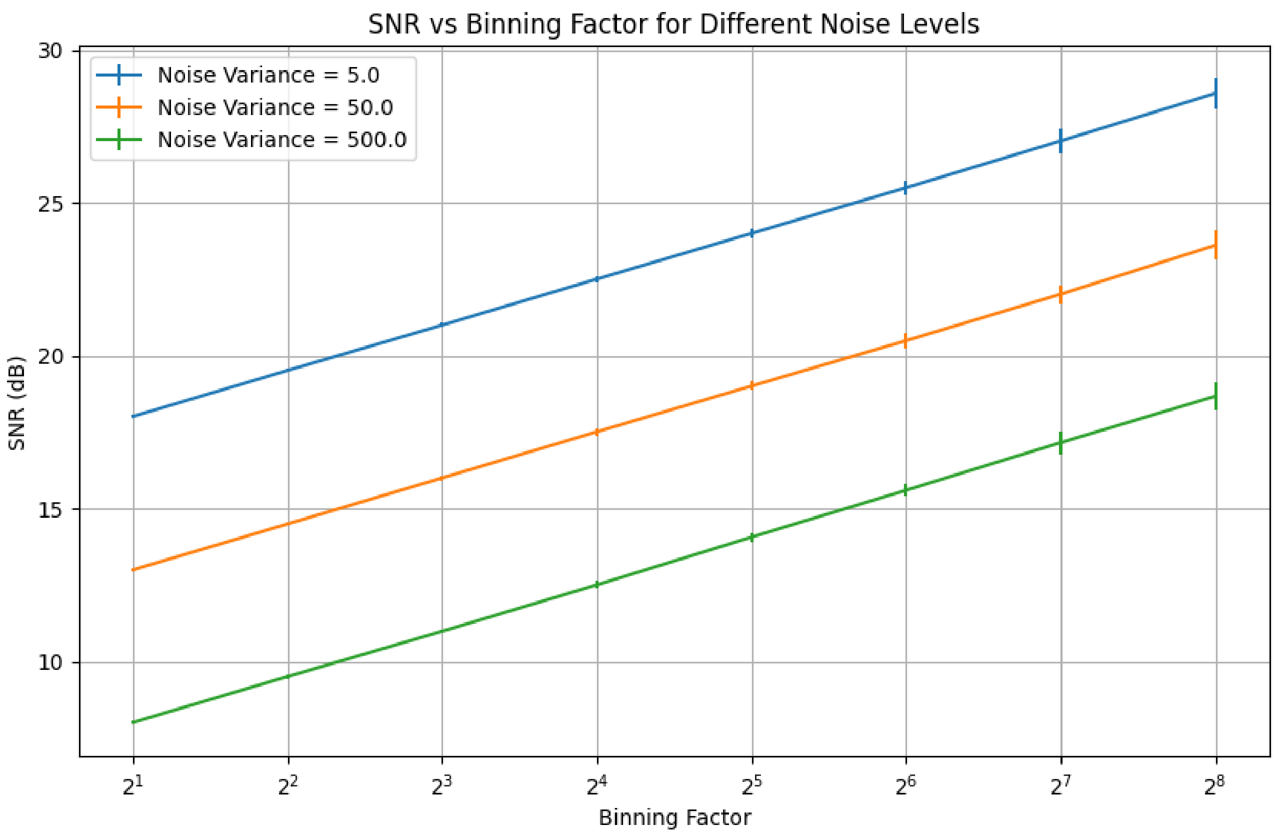

Fig. 1: Comparison of the SNR results between the different noise and binning levels

3 Result and Analysis

The experiment's outcomes were visualized and analyzed through the creation of Figure 1. In this graph, the blue line corresponds to a noise variance of 5.0, the orange line to a noise variance of 50.0, and the green line to a noise variance of 500.0. In summary, the analysis of the SNR values at different noise variance levels reveals the consistent trends:

- There seems to be variability in SNR values across different binning levels for each noise variance. This indicates that the choice of binning level has an impact on the observed signal-to-noise ratio. For each noise variance level (t.0, 50.0, 500.0) the SNR generally tends to increase as the binning level increases.
- Higher noise variance (e.g., 500.0) generally results in lower SNR across all binning levels compared to lower noise variance (e.g., 5.0). This suggests that higher levels of noise negatively impact the signal-to-noise ratio.

From the observations it can be stated that higher binning levels generally lead to higher SNR, indicating improved signal-to-noise ratio and higher noise levels result in lower SNR, making it more challenging to extract the signal. For a given noise level, increasing the binning level can enhance the ability to extract the signal from the noisy data. The choice of binning level depends on the trade-off between data resolution and the need to reduce noise impact.

4 Conclusion

This project investigates the code implementation of binning in image processing for noise reduction. Through experiments with varying noise levels and binning factors, the study reveals a direct relationship between noise reduction and increasing binning levels. Higher noise levels result in lower Signal-to-Noise Ratios (SNR), emphasizing the trade-off between noise reduction and data resolution. Advantages of binning, including improved SNR and faster image acquisition, are highlighted, but drawbacks such as spatial resolution loss and increased readout noise are acknowledged.

References

[ASS99] Til Aach, Ulrich W Schiebel, and Gerhard Spekowius. Digital image acquisition and processing in medical x-ray imaging. *Journal of Electronic Imaging*, 8(1):7–22, 1999.

[BCFS06] M Bigas, Enric Cabruja, Josep Forest, and Joaquim Salvi. Review of cmos image sensors. *Microelectronics journal*, 37(5):433–451, 2006.

[JH12] Xiaodan Jin and Keigo Hirakawa. Analysis and processing of pixel binning for color image sensor. *EURASIP Journal on Advances in Signal Processing*, 2012:1–15, 2012.

[Joh06] Don H Johnson. Signal-to-noise ratio. *Scholarpedia*, 1(12):2088, 2006.

[MEMA$^+$00] M Maolinbay, Y El-Mohri, LE Antonuk, K-W Jee, S Nassif, X Rong, and Q Zhao. Additive noise properties of active matrix flat-panel imagers. *Medical physics*, 27(8):1841–1854, 2000.

[NKA$^+$10] Humbat Nasibov, Alisher Kholmatov, Basak Akselli, Adalat Nasibov, and Sakir Baytaroglu. Performance analysis of the ccd pixel binning option in particle-image velocimetry measurements. *IEEE/ASME Transactions on Mechatronics*, 15(4):527–540, 2010.

[ZPF97] Zhimin Zhou, Bedabrata Pain, and Eric R Fossum. Frame-transfer cmos active pixel sensor with pixel binning. *IEEE Transactions on electron devices*, 44(10):1764–1768, 1997.