COURSE PROJECT REPORT

CSL7360 - COMPUTER VISION

PROPAGATED IMAGE FILTERING

SUBMITTED BY

Suparni Maitra; Roll no: M25CSA029
Suraj Kumar; Roll no: M25CSA030
Vidhan Savaliya: Roll No: M25CSA031

November 2025
Indian Institute of Technology Jodhpur

Colab link:
https://colab.research.google.com/drive/1R4_xfeiY2iyzHC6My4ZGRKaNH7zof_4O?usp=sharing
Github link:

AIM OF THE PROJECT

The primary objective of this project was to implement and analyze the image filtering techniques proposed in Chang and Wang's research paper "Propagated Image Filtering" (CVPR 2015). The project involves implementing classic filters, including bilateral, guided and geodesic filters as baselines, followed by the implementation of the proposed propagation filter, to compare its performance and advantages. Experiments were conducted on various computer vision tasks, including image denoising, image smoothing, flash/no flash denoising and image fusion. For each task, the performance of all the filters was evaluated and compared using the PSNR, to validate the effectiveness of the propagation filter in preserving image details while accomplishing the task.

CODE EXPLANATIONS

A. Implementing Bilateral Filter

- Defined a PSNR function to measure image quality improvement after filtering.
- Initialized Bilateral filter parameters including spatial radius and range, sigma and kernel size.
- Computed a spatial Gaussian kernel based on pixel distances within the neighbourhood.
- For each pixel, extracted its local window from the padded image and computed the range Gaussian based on the colour difference between the center pixel and its neighbours.
- Spatial and range Gaussians were multiplied to form combined weights, and the weighted average of pixel intensities was calculated to produce filtered value.
- The filter was applied across the entire image to obtain the result.
- PSNR value was computed and displayed, along with visual comparison between original, noisy and filtered images.

B. Implementing Guided Filter

- The image was loaded and converted to grayscale.
- Two inputs are taken, the image to be filtered, and the guidance image that controls smoothing near edges.
- For each pixel, a local window was extracted from both input and guidance images, and mean and variance of guidance window was computed.
- Pixel weights were determined based on intensity similarity between each neighbour and window mean- higher weight for similar pixels.
- Weights were normalized and the weighted average was calculated for input pixels to obtain filtered value.
- The result is a locally edge-aware smoothed image with reduced noise.

C. Implementing Geodesic Filter

- Input and guidance images were converted to float32 and padded to handle borders.
- A spatial Gaussian kernel was precomputed based on given radius and sigma_space.
- For each pixel, a local patch was extracted from both the image and the guidance image.
- The colour distance from each pixel in guidance patch to center guidance pixel was computed, and colour distances were converted to colour weights via a Gaussian.
- Spatial and colour weights were multiplied to form combined weights for the patch then normalized.
- The filtered pixel was computed as the weighted sum of the corresponding colour channels in the image patch.
- The result was clipped and converted back to uint8 before returning.

D. Implementing Propagation Filter

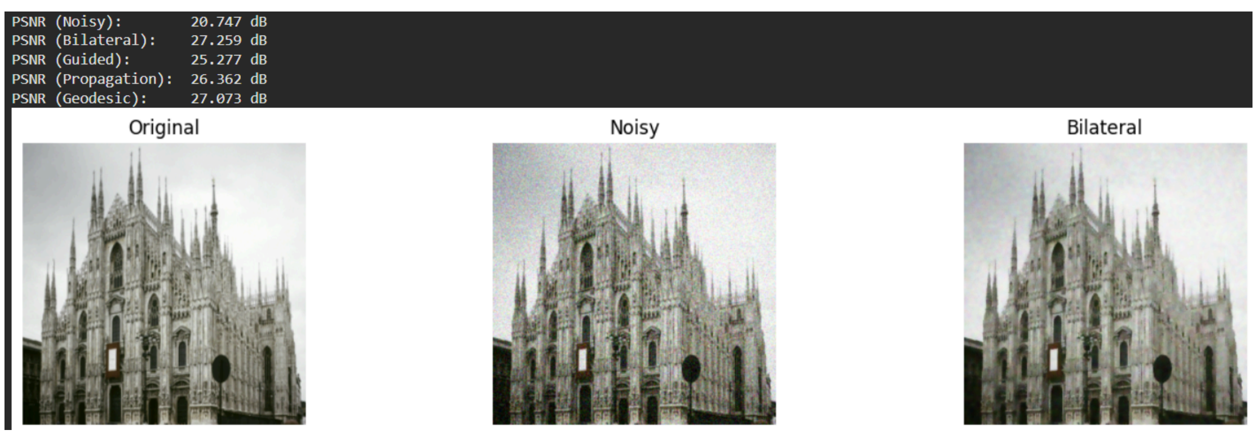
- Input image was padded so local windows are valid at borders.
- A Gaussian G(delta) was defined that maps photometric differences to weights.
- For each pixel in the image, the center pixel was extracted and iterated over a local (2w+2) x (2w-1) neighbourhood.

- For each neighbour, two distances were computed- L2 distance between center and neighbour, and L2 distance between the neighbour and its predecessor along the path.
- The distances were converted to probabilities via D=G(diff_adj) and R=G(diff_center), then combine weight=D*R. The weighted colour values and sum of weights were calculated for the neighbourhood.
- The total weight was normalized to produce the filtered value.
- The resultant image is a context aware smoothing that preserves edges by downweighting neighbours not connected via photometric propagation.

EXPERIMENTS CONDUCTED

A. Denoising Images using Filtering

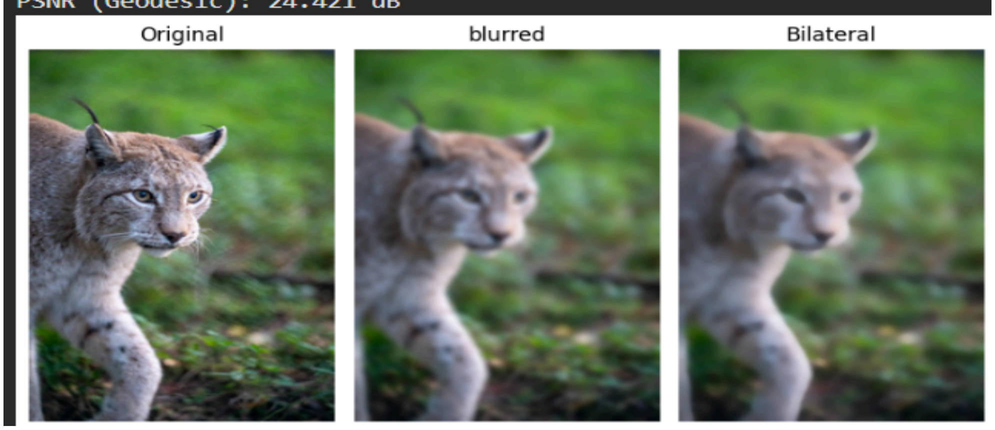
- The image was loaded and resized, and then Gaussian noise was added to simulate a noisy capture.
- The image was denoised with 4 methods: Bilateral, Guided, Geodesic, Propagation filters, handling normalization and value conversions where needed.
- The PSNR values were computed and printed for noisy and each denoised result.
- The original, noisy images and the four filtered outputs were displayed for visual comparison.



B. Image Smoothening

- The image was loaded and a blurred version was created using a Gaussian kernel.
- A PSNR helper was defined that compares images in the [0,1] range.
- The Bilateral filter and the Geodesic filter were run on the blurred colour image.
- The guided filter was run on a grayscale version of the image and converted back to RGB for display.
- The propagation filter was run on the blurred image after converting it to float32, and the result was converted back to uint8.
- The PSNR was computed and printed between the original image and each filtered result to quantify the smoothing quality.
- The original, blurred and the four filtered outputs were displayed for visual comparison.

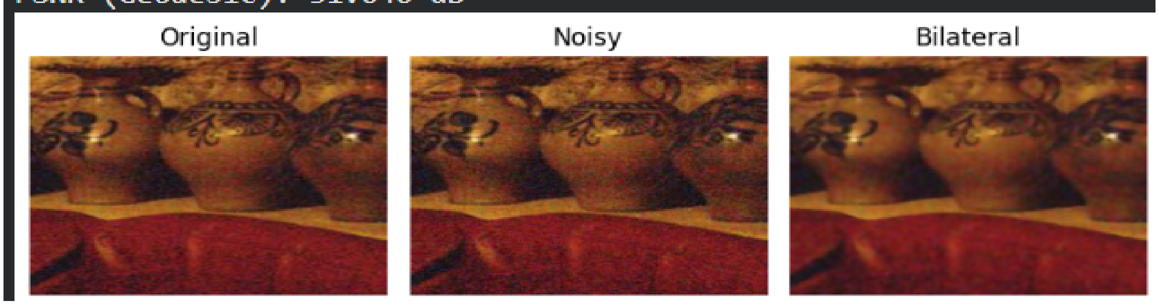
```
PSNR (Bilateral): 26.046 dB
PSNR (Guided): 20.144 dB
PSNR (Propagation): 26.868 dB
PSNR (Geodesic): 24.421 dB
```

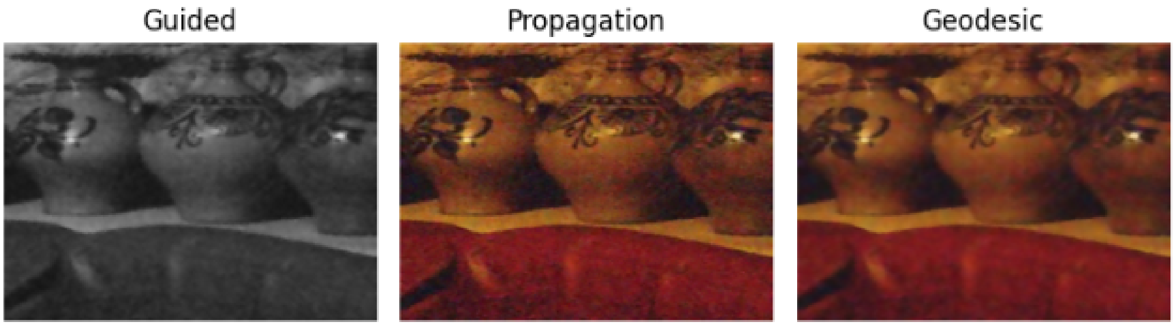


C. Flash/No Flash Denoising

- The no-flash image was loaded and resized for speed.
- Synthetic Gaussian noise was added to the no-flash image to simulate a noisy capture.
- The noisy image was denoised with all four filters- Bilateral, Guided, geodesic and Propagation.
- The outputs were scaled and converted where needed so all results were comparable.
- The PSNR was computed and printed between the clean no-flash image and each filtered result.
- The original, noisy and four filtered images were displayed for visual comparison.

```
PSNR (Noisy): 24.771 dB
PSNR (Bilateral): 31.44 dB
PSNR (Guided): 17.167 dB
PSNR (Propagation): 30.537 dB
PSNR (Geodesic): 31.046 dB
```

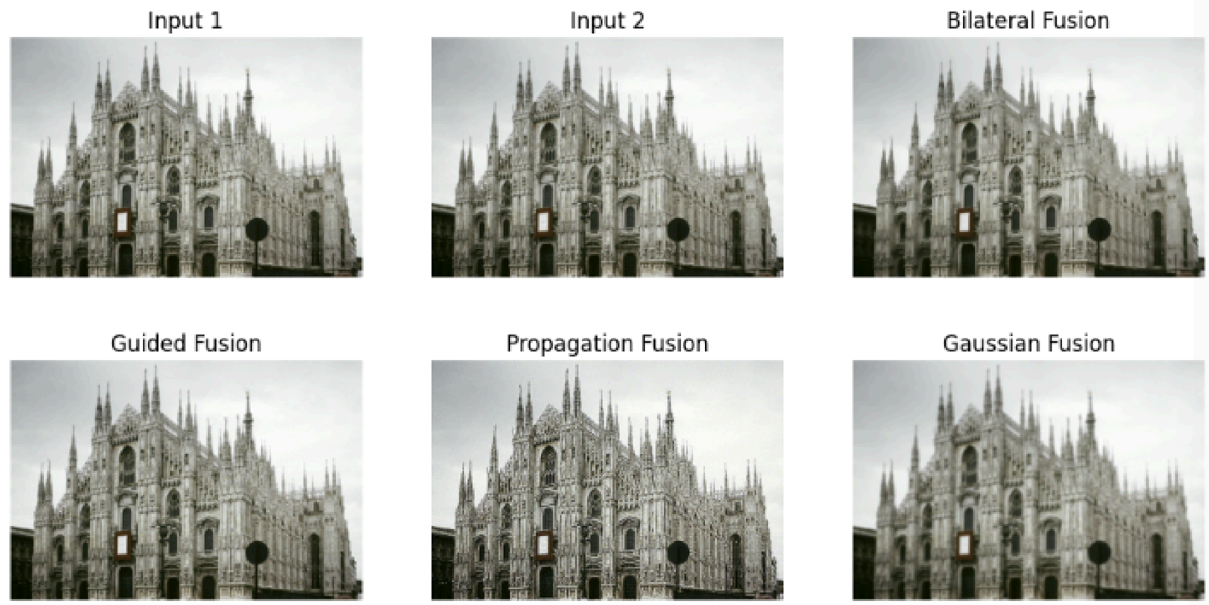




D. Image Fusion

- The bilateral, guided, geodesic and propagation filters were applied on the images, along with a baseline Gaussian blur.
- A saliency weight map per input was built using the absolute Laplacian, smoothed by a Gaussian.
- The two weight maps were normalized.
- Each input was preprocessed with the corresponding filter, and the fused image was computed as a per pixel weighted sum.
- The fused inputs and results were displayed side by side for visual comparison along with the PSNR values to quantify the accuracy.

```
PSNR Comparison:
Bilateral    : 30.806623221641512
Guided       : 51.79096904011991
Propagation  : 17.748997593258874
```



OBSERVATIONS

- In image denoising, the propagation filter, bilateral and geodesic filters performed very well with high PSNR values around 27, showing their strong noise suppression while preserving the image edges. The guided filter was slightly less effective for high noise levels, producing lower PSNR.
- In image smoothing, the propagation filter delivered the best smoothing performance at 26.86dB, effectively removing minor texture noise while maintaining the image details. The guided and geodesic filters followed closely while the bilateral filter had the lowest PSNR.
- In Flash/No Flash denoising, all the filters significantly improved the PSNR. The bilateral (31.44dB) and geodesic(31.04dB) filters performed the best, while the propagation filter had similar results (30.53dB). The guided filter performed slightly weaker in this case.
- In image fusion, the guided filter achieved the highest PSNR, indicating the smoothest transition across focus regions. Bilateral fusion also performed well, while the propagation fusion was less stable.

CONCLUSIONS

- The propagation filter, inspired by Chang and Wang (CVPR 2015), consistently produced balanced results, effectively preserving edges while reducing noise.
- The bilateral and geodesic filters performed best in denoising and flash/no-flash tasks, confirming their strength in local edge-aware smoothing, with the propagation filter also producing comparable results.
- The guided filter excelled in image fusion, where its local linear model effectively merged focus regions with minimal artifacts.
- These results validate the theoretical claims of the research paper, showing that propagated filtering provides a general, flexible framework that performs competitively with traditional filters across multiple image enhancement applications.