SC4061/CZ4003

# Computer Vision Lab Assignment 2

**Submitted by**

J’sen Ong Jia Xuan

U2220457J

**College Of Computing And Data Science**

# Table of Contents

# Table of Figures

# 1. Introduction

This report presents the implementation and analysis of two fundamental computer vision techniques: **image segmentation** for document analysis and **stereo vision** for disparity map estimation. The experiments demonstrate the effectiveness of various thresholding algorithms in handling degraded document images and explore the relationship between local image structure and disparity estimation accuracy in stereo vision systems.

# 2. Image Segmentation

## 2.1 Experimental Setup

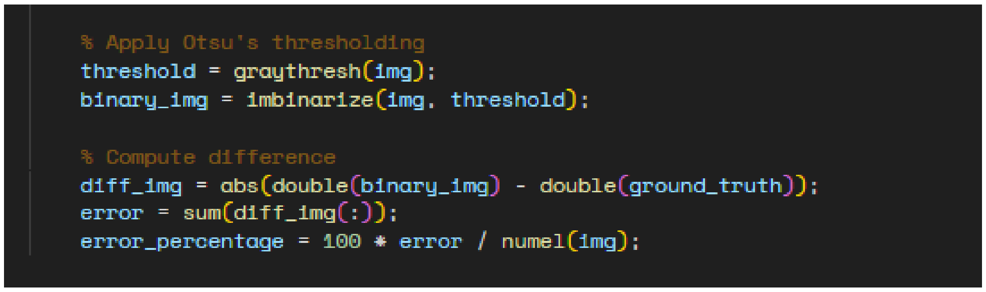
Four degraded document images were processed using two different thresholding approaches to segment text from background. Segmentation quality was evaluated by computing the sum of pixel-wise differences between the segmented binary images and their corresponding ground truth images.

## 2.2 Otsu's Global Thresholding

### 2.2.1 Methodology

Otsu's algorithm automatically determines an optimal global thresholding by maximizing the inter-class variance of pixel intensities. This method assumes a bimodal histogram distribution and computes a single threshold value applied uniformly across the entire image.

### 2.2.2 Implementation



```
% Apply Otsu's thresholding
threshold = graythresh(img);
binary_img = imbinarize(img, threshold);

% Compute difference
diff_img = abs(double(binary_img) - double(ground_truth));
error = sum(diff_img(:));
error_percentage = 100 * error / numel(img);
```

*Figure 1. Image Segmentation - Otsu's Global Thresholding code*

### 2.2.3 Results and Analysis

| Image | Otsu Threshold | Error (pixels) | Error (%) |
|---|---|---|---|
| **Document1.bmp** | 0.5451 | 27849 | 4.22% |
| **Document2.bmp** | 0.4392 | 9476 | 3.00% |
| **Document3.bmp** | 0.6902 | 179165 | 18.74% |
| **Document4.bmp** | 0.5961 | 134548 | 21.23% |

*Figure 2. Image Segmentation - Comparison of Otsu Threshold values*

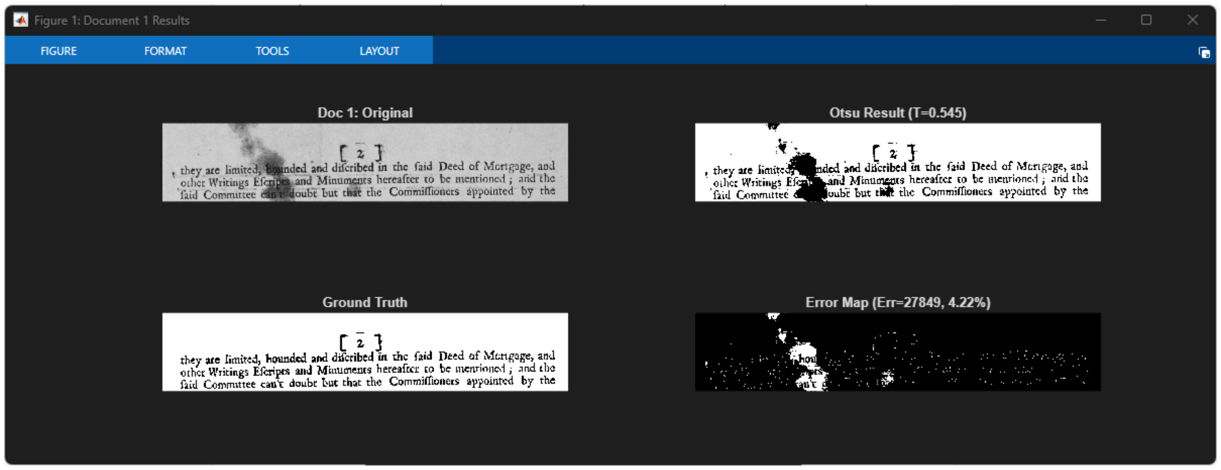


*Figure 3. Image Segmentation - Otsu's Global Thresholding results (Document01.bmp)*

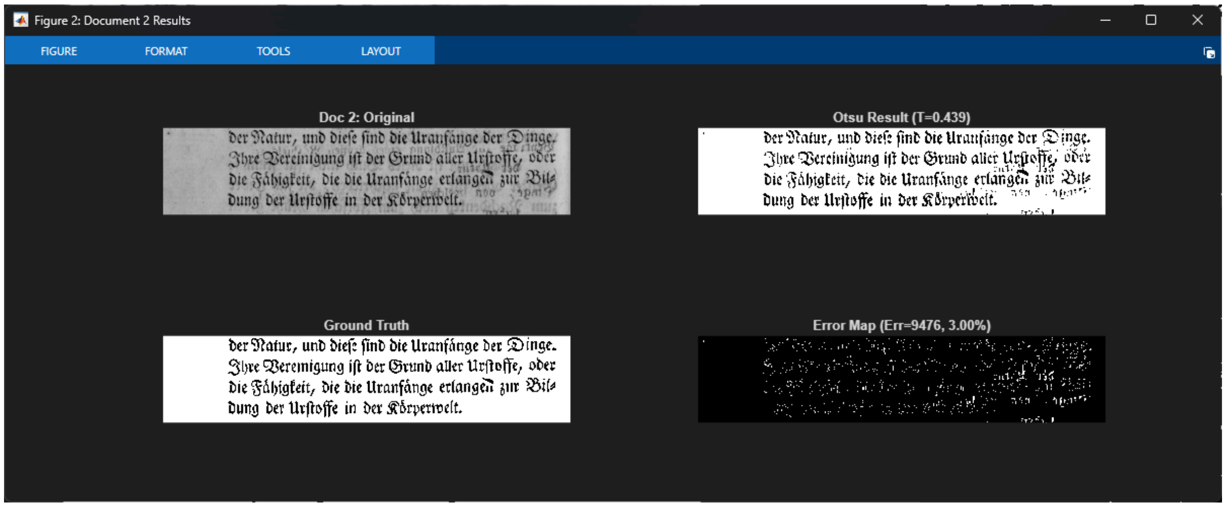


*Figure 4. Image Segmentation - Otsu's Global Thresholding results (Document02.bmp)*

### Document 1-2: Minimal Degradation

The first two documents exhibited relatively minor degradation, allowing Otsu's method to perform reasonably well:

- **Document 1 - Dark stain Challenge:** Contains a localized dark stain covering portions of the text. The stain appears darker than the text itself, obscuring the underlying characters. While this presents a challenging scenario for global thresholding, since the stain region requires a different threshold than the clean regions, the degradation was spatially limited. The majority of the document remained clean, allowing Otsu's algorithm to determine a threshold that captured most of the text. The error penalty remained relatively low because the stained region represented a small fraction of the total image area.

- **Document 2 - Background Tinting Challenge:** Features a uniformly tinted background rather than localized degradation. The primary challenge was distinguishing text (foreground) from the darker background. However, since the background darkening was relatively uniform across the image, the intensity distribution remained approximately bimodal. Otsu's global threshold could still separate the background and the text effectively, resulting in acceptable segmentation with minimal error.

For both documents 1 and 2, the degradation types did not fundamentally violate the bimodal histogram assumption underlying Otsu's method. The global threshold, while not optimal for every local region, still managed to achieve adequate overall segmentation performance as observed in Figure 3 and 4.

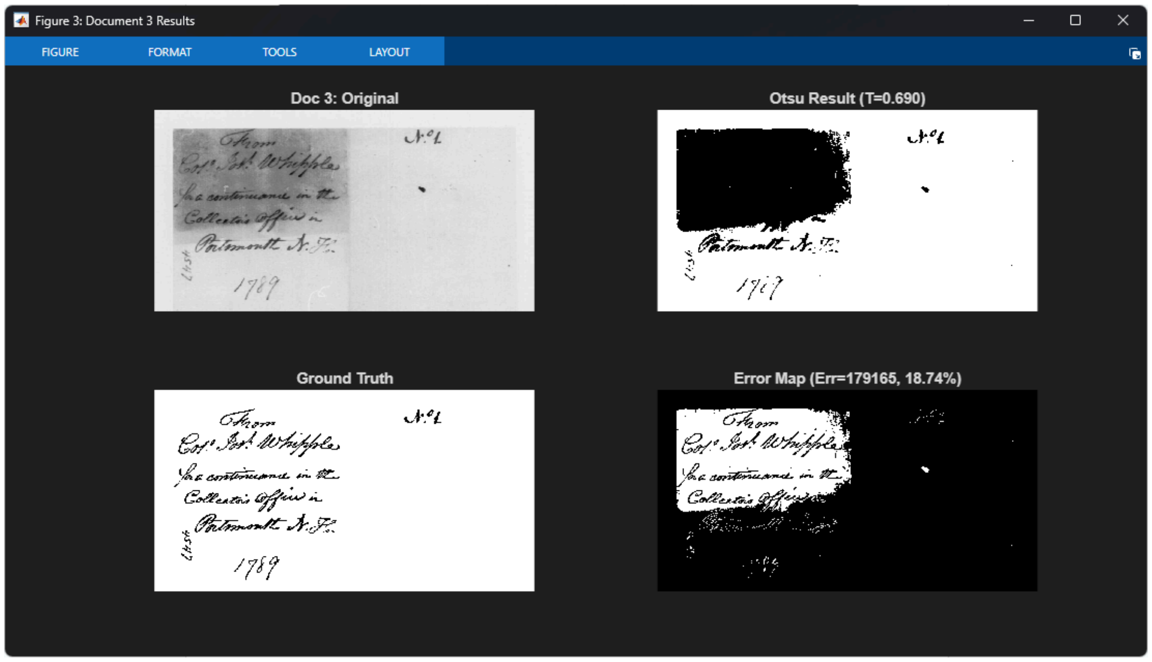


*Figure 5. Image Segmentation - Otsu's Global Thresholding results (Document03.bmp)*

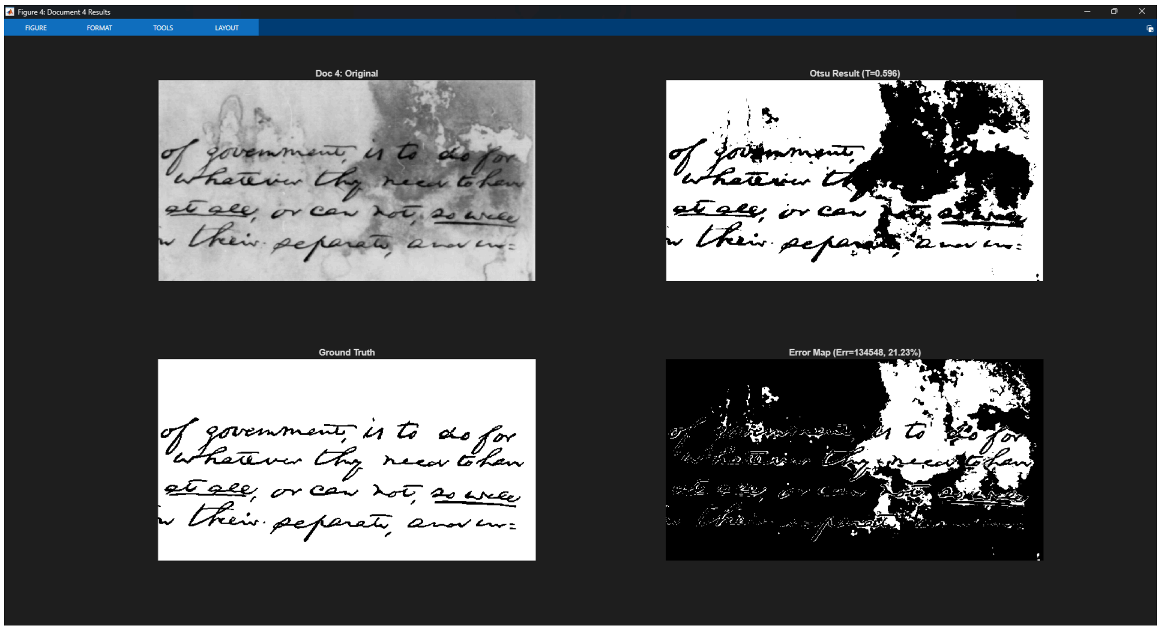


*Figure 6. Image Segmentation - Otsu's Global Thresholding results (Document04.bmp)*

**Document 3-4: Severe Degradation**

As observed in Figure 5 and 6, the latter two documents presented significantly more challenging conditions that exposed the fundamental limitations of global thresholding:

- **Document 3 - Large Dark Blotch:** This document features a substantial dark block covering a large portion of the upper-left region. The degradation is severe enough to completely obscure the underlying in the affected area. From the error map, we can observe that the entire stained region is misclassified as foreground (black), resulting in a large solid error area. This occurs because the stain's intensity falls below the global threshold, causing it to be incorrectly segmented as text. This misclassification causes the text within the stain region to be completely lost and cannot be recovered with any single global threshold since both the stain and the obscured text have similar intensities.
- **Document 4 - Multiple Severe Degradations:** This document shows the most complex degradation pattern, combining several challenges at once:
  - Large irregular stains with varying darkness levels
  - Non-uniform background intensity transitioning from lighter on the left to darker toward the right
  - Texture and noise distributed throughout the background
  - Partial text occlusion occurs in multiple regions across the image

Otsu's algorithm selected **T = 0.596**, which is lower than the threshold chosen for Document3.bmp. This lower value attempts to accommodate the overall darker appearance of the document. However, this single threshold value creates an impossible dilemma. Lowering the threshold would recover more text characters, particularly those in darker regions or those partially obscured by stains, but this same adjustment would include even more background noise and classify larger portions of the stain regions as foreground. Conversely, raising the threshold would produce a cleaner background with few false positives from stains and noise, but this approach would inevitably lose legitimate text characters, particularly those appearing in the darker regions of the document.

The **21.23%** error rate is the **highest** among all four documents and reflects the fundamental incompatibility between global thresholding and spatially varying degradations of this magnitude. When different regions of the same image require varying thresholds, no single global value can produce acceptable results across the entire document. This limitation demonstrates the clear need for localized thresholding methods that can adjust their behavior based on regional image characteristics, motivating the exploration of algorithms like Niblack's method in the following section.

## 2.3 Niblack's Local Thresholding with Bayesian Optimization

### 2.3.1 Methodology

Unlike Otsu's global thresholding explored in the previous section, which applies a single uniform threshold across the entire image. Nisblack's algorithm computes adaptive thresholds based on local statistics within a sliding window.

$$\mathbf{T(x,y) = \mu(x,y) + k \times \sigma(x,y)}$$

Where **$\mu(x,y)$** represents the local mean intensity, **$\sigma(x,y)$** represents the local standard deviation and **k** is a parameter controlling threshold sensitivity. For dark text on light backgrounds, **k** typically takes negative values.

This adaptive threshold computation fundamentally distinguishes Niblack's thresholding from global thresholding methods like Otsu's thresholding. It generates a spatially varying threshold map that adapts to the regional image characteristics. Where Otsu's method struggles with non-uniform illumination and localized degradation, Niblack's algorithm can adjust its behavior dynamically, applying different threshold values to different regions based on their local statistical properties.

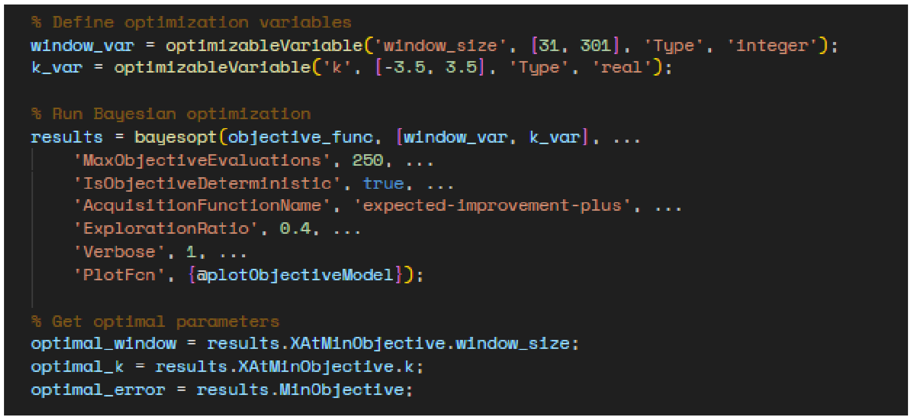
**Bayesian Optimization**

The effectiveness of Niblack's algorithm depends critically on two parameters: **window size** and the **k** value. The **window size** determines the spatial extent of the local neighborhood used for computing the local mean intensity and standard deviation, while **k** controls how aggressively the algorithm responds to local contrast variations. These parameters interact in complex ways, with optimal values varying significantly depending on the specific degradation characteristics of each document.

Traditional parameter tuning approaches to achieve the optimal **k** and **window size** face a significant computational challenge. Grid search would require exhaustive evaluation of all parameter combinations across the two-dimensional space. For example, testing 30 different **window sizes** and 70 **k** values would necessitate 2,100 evaluations per document. Since different documents exhibit different degradation characteristics, optimal value for different documents might differ, making this brute-force approach computationally prohibitive.

To address this optimization challenge, we employed Bayesian Optimization. This sequential model-based approach builds a probabilistic surrogate model of the objective function and determines which parameter combinations to evaluate next, balancing exploration of uncertain regions with exploitation of known promising areas. This method typically converges to near-optimal parameters within 150 to 250 evaluations and often discovers better parameter combinations that exhaustive grid search methods might miss.

### 2.3.2 Implementation



```matlab
% Define optimization variables
window_var = optimizableVariable('window_size', [31, 301], 'Type', 'integer');
k_var = optimizableVariable('k', [-3.5, 3.5], 'Type', 'real');

% Run Bayesian optimization
results = bayesopt(objective_func, [window_var, k_var], ...
    'MaxObjectiveEvaluations', 250, ...
    'IsObjectiveDeterministic', true, ...
    'AcquisitionFunctionName', 'expected-improvement-plus', ...
    'ExplorationRatio', 0.4, ...
    'Verbose', 1, ...
    'PlotFcn', {@plotObjectiveModel});

% Get optimal parameters
optimal_window = results.XAtMinObjective.window_size;
optimal_k = results.XAtMinObjective.k;
optimal_error = results.MinObjective;
```

*Figure 7. Image Segmentation - Niblack's Local Thresholding code*

### 2.3.3 Results and Analysis

In Figure 8, the 3D surface plots illustrate how **k** and **window size** impact segmentation accuracy for each document, with the z-axis representing the error sum relative to ground truth. The concentration of blue evaluation points in lower regions of each plot indicates parameter settings that achieve better performance, revealing the optimal parameter zones. These visualizations demonstrate how Bayesian Optimization systematically explores different parameter combinations, converging on regions that yield the best segmentation. The distinct surface landscape of each document emphasizes that optimal parameters vary depending on specific image characteristics and degradation patterns.

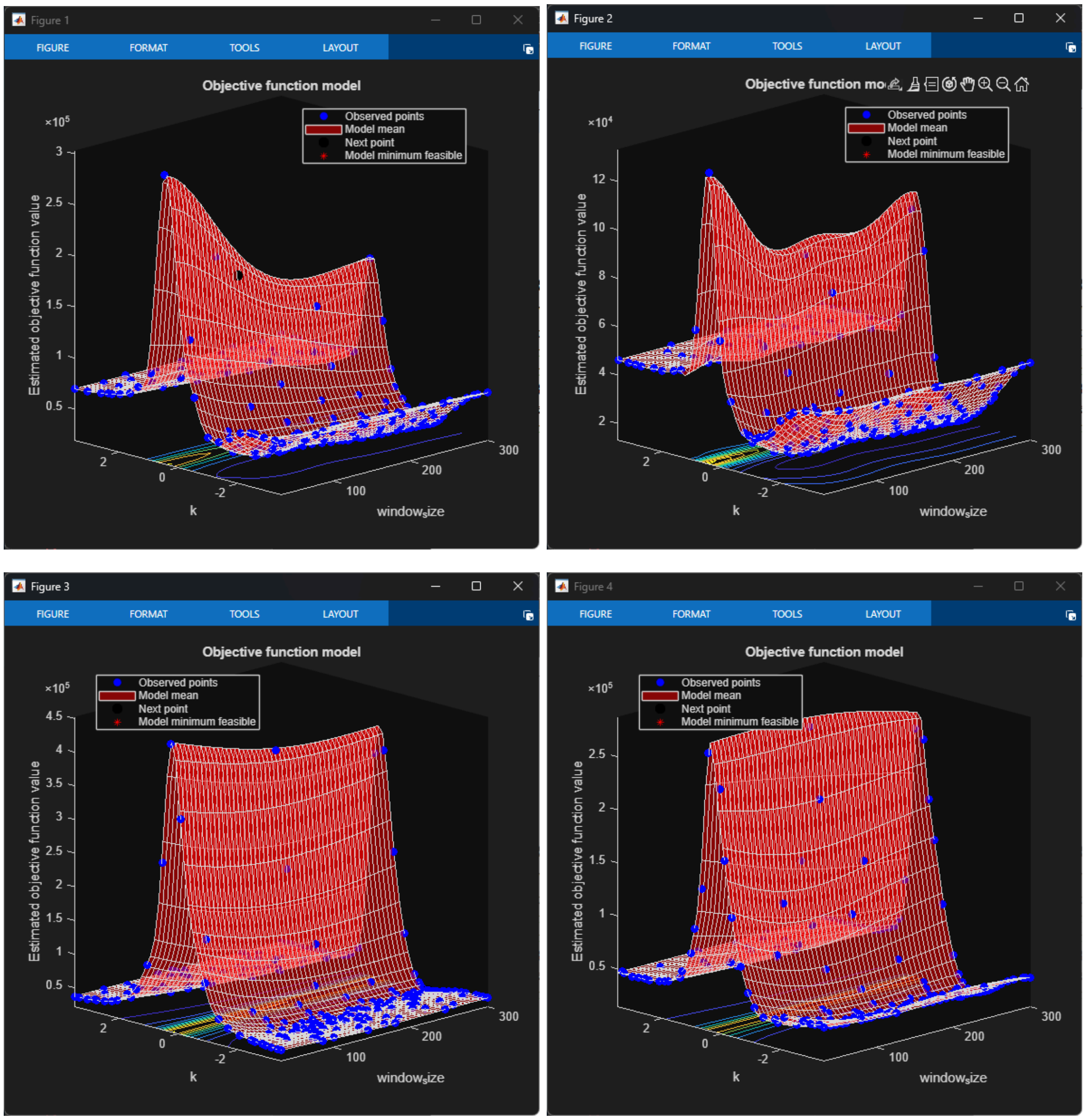


*Figure 8. Image Segmentation - Bayes Optimization Objective Function Surface Models for Documents 1-4*

| Image | Window Size | K Value | Niblack's Error (pixels) | Niblack's Error (%) | Otsu's Error (%) | Improvement |
|---|---|---|---|---|---|---|
| **Document1.bmp** | 301 | -1.3881 | 27849 | 2.92 | 4.22 | -1.30% |
| **Document2.bmp** | **301** | **-0.9709** | **9476** | **4.16** | **3.00** | **+1.16%** |
| **Document3.bmp** | 301 | -1.4838 | 179165 | 2.30 | 18.74 | -16.44% |
| **Document4.bmp** | **301** | **-1.5513** | **134548** | **2.25** | **21.23** | **-18.98%** |

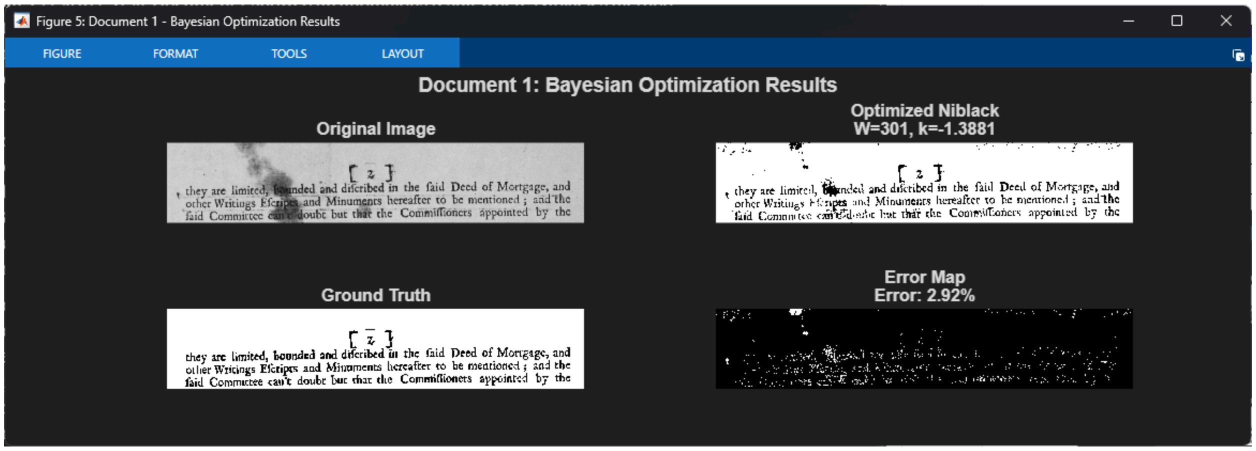
*Figure 9. Image Segmentation - Performance Comparison: Niblack's Local Thresholding with Optimal Parameters vs Otsu's Global Thresholding*

Document1.bmp and Document2.bmp, with their relatively minor degradations showed minimal performance differences between the two methods. Document 2 showed an anomalous result where Niblack's method actually performed slightly worse than Otsu's approach, with error increasing by **1.16%**. This likely occurred because Document 2's relatively uniform background tint was well-suited to global thresholding, while local thresholding may have introduced unnecessary noise sensitivity.

However, the true strength of local adaptive thresholding becomes evident in Document3.bmp and Document4.bmp. For Document3.bmp, which featured a large dark stain region, Niblack's method achieved a significant **16.44%** improvement, reducing the error rate from **18.74%** to **2.30%**. The algorithm successfully adapted to the spatially varying characteristics of the image, applying different thresholds to the stained region and clean text areas.

Document 4 's results validated the fundamental advantages of adaptive thresholding. With its complex combination of irregular stains, non-uniform background and multiple degradation types, Niblack's method reduced the error rate from **21.23%** to **2.25%**, an **18.98%** improvement over Otsu's thresholding algorithm. The spatially varying threshold map successfully distinguished text from stains in degraded regions while maintaining clean segmentation in less affected areas. The substantial performance gain directly validates that spatially varying degradation requires spatially adaptive thresholding solutions, where global thresholding algorithms commonly fall short in.

All four documents converged to the maximum window size of **301 pixels,** suggesting that larger windows yield more stable local statistics for threshold estimation. The optimal **k** values varied from **-0.9709** to **-1.5513**, with Documents 3 and 4 requiring more negative **k** values. This pattern suggests that degraded or non-uniform backgrounds benefit from larger neighborhoods and more aggressive threshold correction in adaptive thresholding.



Document 1: Bayesian Optimization Results

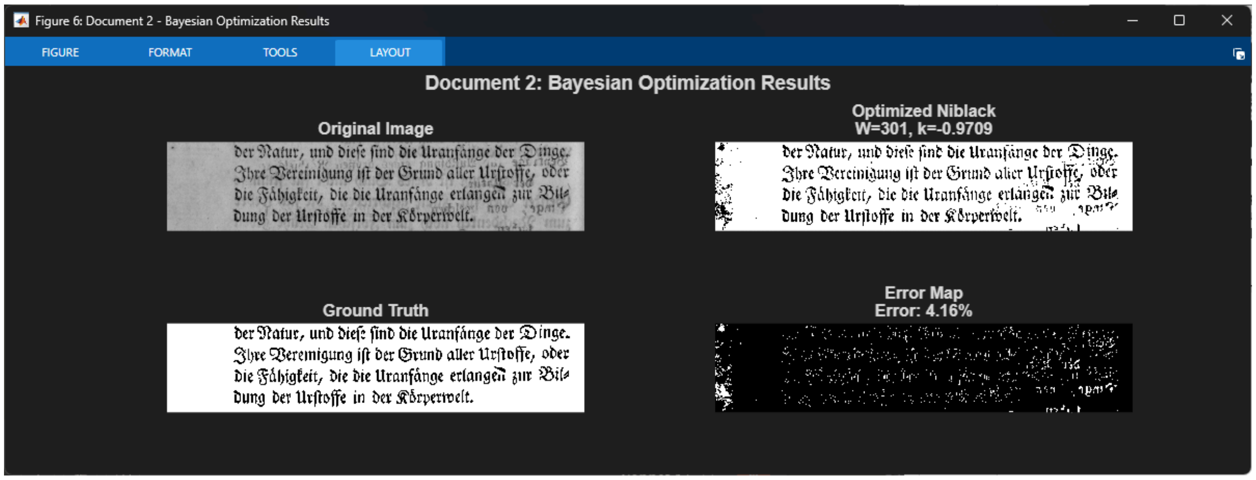
Original Image

Optimized Niblack
W=301, k=-1.3881

Ground Truth

Error Map
Error: 2.92%

*Figure 10. Image Segmentation - Niblack's Local Thresholding results (Document01.bmp)*



Document 2: Bayesian Optimization Results

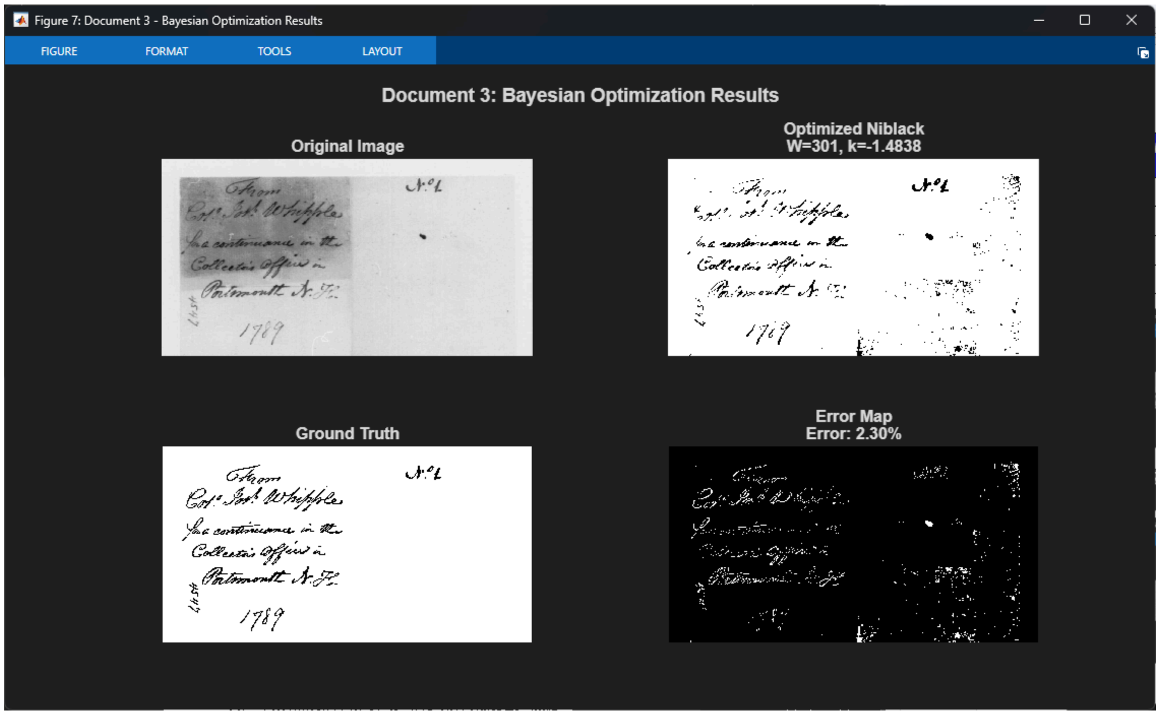
Original Image

Optimized Niblack
W=301, k=-0.9709

Ground Truth

Error Map
Error: 4.16%

*Figure 11. Image Segmentation - Niblack's Local Thresholding results (Document02.bmp)*

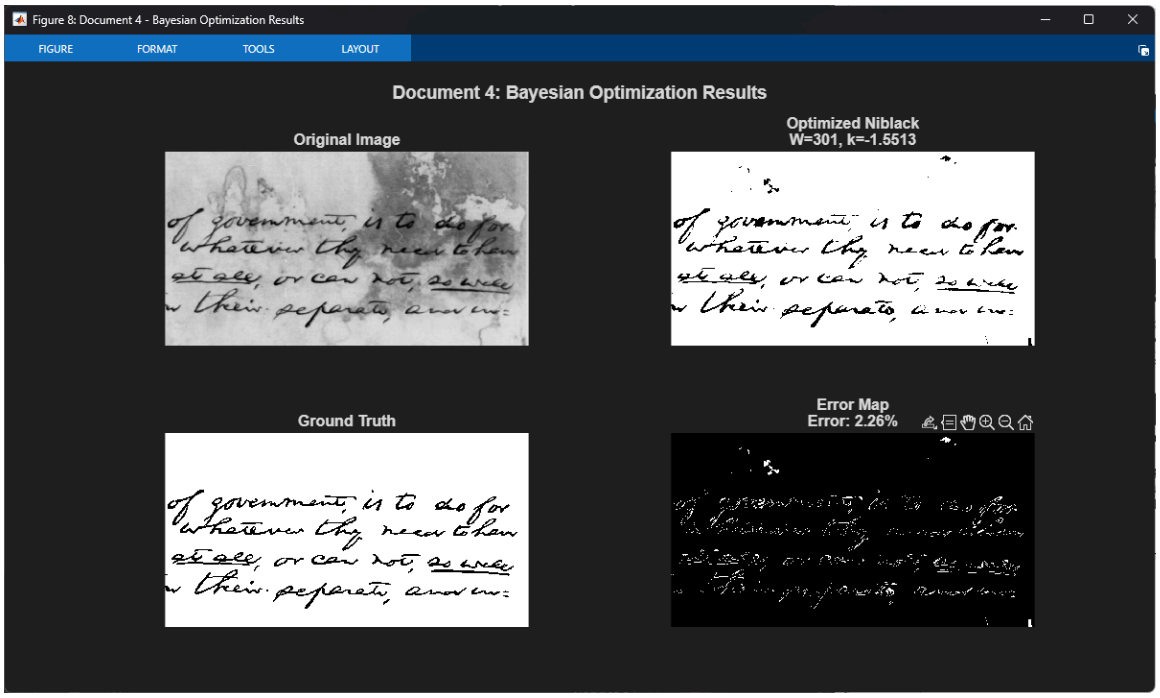


Document 3: Bayesian Optimization Results

Optimized Niblack
W=301, k=-1.4838

Error: 2.30%

Figure 12. Image Segmentation - Niblack's Local Thresholding results (Document03.bmp)



Document 4: Bayesian Optimization Results

Optimized Niblack
W=301, k=-1.5513

Error: 2.26%

Figure 13. Image Segmentation - Niblack's Local Thresholding results (Document04.bmp)

Despite the substantial improvements Niblack's algorithm has over Otsu's Algorithm, particularly for severely degraded documents, the method exhibits certain limitations. A notable issue is the tendency to introduce noise in relatively uniform background regions, where small variations in pixel intensity can trigger incorrect threshold adjustments that missclassify background pixels as foreground. This is clearly observed in Figure 11 for Document2.bmp, where the uniform background shows increased noise compared to Otsu's result.

Additionally, Niblack's method can be overly sensitive to the window size parameter in regions near text boundaries. Large windows that span both text and background produce averaged statistics that may not accurately represent either region, potentially causing boundary blurring or incomplete character segmentation.

These observations motivated the exploration of Sauvola's method in the following section, a modification of Niblack's algorithm designed to mitigate the background noise issue while retaining the underlying advantages of local thresholding.

## 2.4 Sauvola's Local Thresholding

### 2.4.1 Methodology

Sauvola's algorithm extends Niblack's local thresholding method to achieve stronger edge preservation and better discrimination between text and background in degraded or unevenly illuminated documents. The local threshold at each pixel is computed as:

$$\mathbf{T(x,y) = \mu(x,y) \times [1 + k \times (\sigma(x,y)/R - 1)]}$$

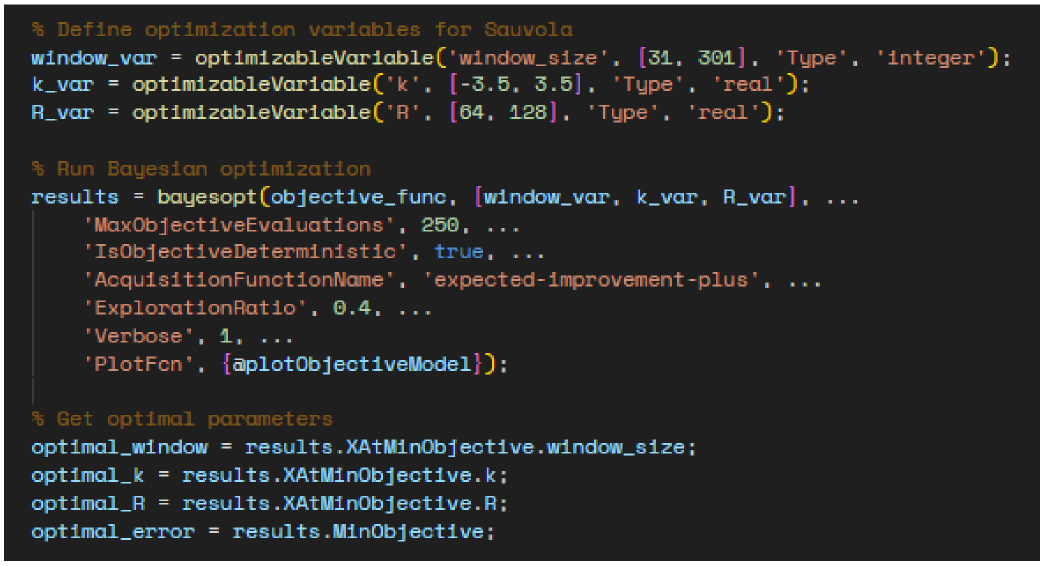
Where **$\mu(x,y)$** is the local mean intensity, **$\sigma(x,y)$** is the local standard deviation, k is a sensitivity parameter and **R** is the dynamic range parameter that acts as a sensitivity scaling factor.

While Sauvola's formula appears quite similar to Niblack's, they differ in several ways. Sauvola's equation is structured multiplicatively rather than additively, computing the threshold as a proportion of the local mean. The key distinction lies in the normalization of the standard deviation by the dynamic range **R**. By computing **$\sigma(x,y)/R$,** Sauvola creates a ratio between **0** and **1** that represents the relative local contrast rather than absolute standard deviation values. The **R** parameter controls how strongly the local standard deviation influences the final threshold, with larger **R** producing more conservative segmentation and smaller **R** values allowing more aggressive threshold adjustments.

This normalization helps the algorithm to adapt dynamically to different image regions. In uniform background areas where the local standard deviation **σ** is small, the ratio **σ/R** approaches **0**, making the bracketed term **[1 + k × (σ(x,y)/R - 1)]** approach **(1 - k)**, which keeps the threshold slightly below the local mean. This conservative adjustment reduces false positives from noise and compression effects. On the other hand, in high-contrast text regions where the local standard deviation **σ** is large, the ratio **σ/R** increases substantially, pushing the bracketed term above **1** and causing a more aggressive threshold adjustment away from the mean. This allows Sauvola's method to modulate its response less aggressively compared to Niblack's in uniform regions, making it more sensitive to local contrast variations while minimizing over-segmentation in background areas.

### 2.4.2 Implementation

Similar to Niblack's implementation, we employed Bayesian Optimization to identify optimal parameters. However, Sauvola's method introduces an additional parameter **R (dynamic range)** that requires optimization alongside **window size** and **k**:



```
% Define optimization variables for Sauvola
window_var = optimizableVariable('window_size', [31, 301], 'Type', 'integer');
k_var = optimizableVariable('k', [-3.5, 3.5], 'Type', 'real');
R_var = optimizableVariable('R', [64, 128], 'Type', 'real');

% Run Bayesian optimization
results = bayesopt(objective_func, [window_var, k_var, R_var], ...
    'MaxObjectiveEvaluations', 250, ...
    'IsObjectiveDeterministic', true, ...
    'AcquisitionFunctionName', 'expected-improvement-plus', ...
    'ExplorationRatio', 0.4, ...
    'Verbose', 1, ...
    'PlotFcn', {@plotObjectiveModel});

% Get optimal parameters
optimal_window = results.XAtMinObjective.window_size;
optimal_k = results.XAtMinObjective.k;
optimal_R = results.XAtMinObjective.R;
optimal_error = results.MinObjective;
```

*Figure 14. Image Segmentation - Sauvola's Local Thresholding code*

### 2.4.3 Results and Analysis

| Image | Window Size | K Value | R Value | Souvala's Error (%) | Niblack's Error (%) | Otsu's Error (%) | Improvement Vs Niblack |
|---|---|---|---|---|---|---|---|
| **Document1.bmp** | 42 | 0.5484 | 74.1491 | 1.33 | 2.92 | 4.22 | +1.59% |
| **Document2.bmp** | 44 | 0.5931 | 64.0219 | 2.57 | 4.16 | 3.00 | +1.59% |
| **Document3.bmp** | 31 | 0.1904 | 126.7684 | 1.11 | 2.30 | 18.74 | +1.19% |
| **Document4.bmp** | 31 | 0.3910 | 100.0403 | 1.55 | 2.25 | 21.23 | +0.70% |

*Figure 15. Image Segmentation - Performance Comparison: Souvala's Local Thresholding with Optimal Parameters vs Niblack's vs Otsu's Global Thresholding*

Sauvola's method demonstrated consistent improvements across all four documents, achieving further error reductions beyond Niblack's substantial gains.

Interestingly, Sauvola's Bayes Optimization converged to a smaller window **(31 to 44 pixels)** compared to Niblack's consistent maximum of **301 pixels**. This reduction occurs because Sauvola's normalization by **R** fundamentally solves the noise stability problem that forces Niblack to use large windows. In Niblack's method, small windows are problematic because noise causes large fluctuations in the raw standard deviation **σ,** which directly destabilize the threshold calculation. To achieve stable local statistics, Niblack must rely on very large windows to be able to average out those noise over extensive regions. However, Sauvola's computation of **σ/R** scales down noise-induced fluctuations proportionally. For example, a noise fluctuation of **±5** intensity units in **σ** becomes **±5/128 ≈ ±0.039** in the normalized ratio when **R = 128**, representing negligible impact on the overall threshold. This normalization allows Sauvola to use more localized neighborhoods without suffering from noise amplification, since stable threshold computation can be achieved even with small windows. With smaller window sizes, Sauvola can provide more spatially precise threshold computation, better preserving fine text details and sharper boundaries between text and background regions.

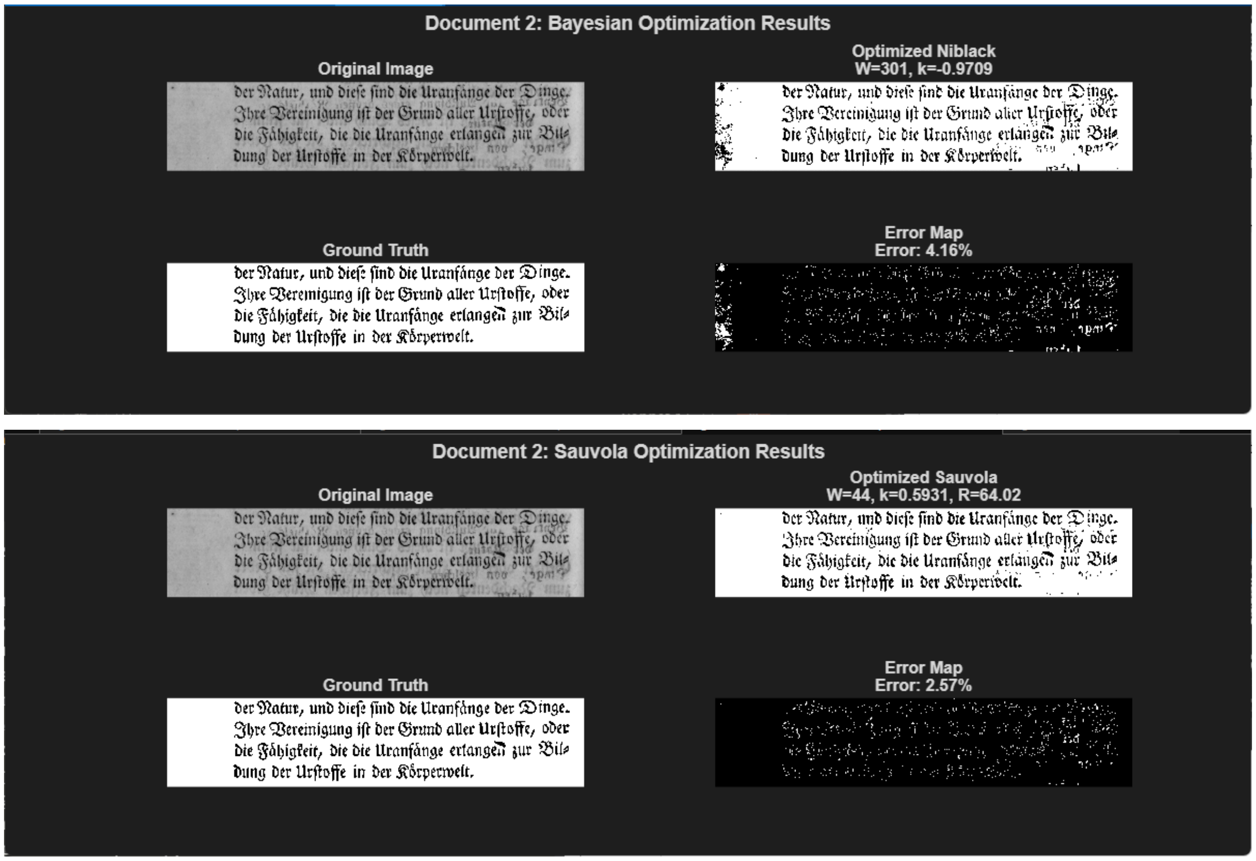


*Figure 16. Image Segmentation - Niblack's vs Souvala's Local Thresholding results (Document02.bmp)*

The comparative visualizations in Figure 16 demonstrate Sauvola's refinements over Niblack's method. We specifically analyzed Document2.bmp, since this was the only case where Niblack performed worse than Otsu's global thresholding. The differences become immediately apparent in the error maps.

Close examination of the degraded regions reveals that Sauvola achieves finer discrimination between text and background. While both methods successfully capture the main text structures, Sauvola's error map shows more precise boundaries around character strokes. This improvement stems from the smaller window sizes, which enable more spatially localized threshold decisions. Large windows that span both text and background produce averaged statistics that do not accurately reflect either region, leading to ambiguous threshold decisions. In contrast, Sauvola's smaller windows remain within more homogenous regions, enabling clearer distinctions that produce sharper character strokes and better preservation of fine text details.

# 3. 3D Stereo Vision

## 3.1 Introduction

In this section, we explore stereo vision techniques for estimating depth information from pairs of calibrated camera images. Stereo vision exploits the principle of binocular disparity, where the same 3D point appears at different horizontal positions in the left and right images. By identifying corresponding pixels between stereo pairs and measuring their horizontal displacement (disparity), we can reconstruct the relative depth of scene points.

The experiments focus on implementing and evaluating a **Sum of Squared Differences (SSD)** based on correspondence matching algorithm to compute disparity maps. Through these experiments, we investigate the fundamental relationship between local image structure and disparity estimation accuracy, revealing the role that texture, edges and feature geometry play in estimating depth information.

The algorithm operates on rectified stereo pairs, which are a prerequisite for efficient disparity computation. Rectification is a preprocessing step that geometrically transforms the left and right images so that corresponding points lie on the same horizontal scanline. This transformation simplifies the correspondence problem from a 2-dimensional search across the entire image to a 1-dimensional search along each horizontal scanline. Without rectification, corresponding points could appear at different vertical positions, requiring computationally expensive 2D searches. With rectification, the stereo pair behaves as if both cameras are perfectly aligned with parallel optical axes.

Given rectified images, the disparity at a point is defined as the horizontal displacement between corresponding pixels:

$$d(x_l, y_l) = x_l - \hat{x}_r$$

Where $x_l$ and $y_l$ are the relevant pixel coordinates in the left image and $\hat{x}_r$ is matched x-coordinate in the right image (on the same scanline), and **d** is the disparity value. Disparity is inversely proportional to depth: larger disparities indicate closer objects, while distant objects produce smaller disparities approaching zero.

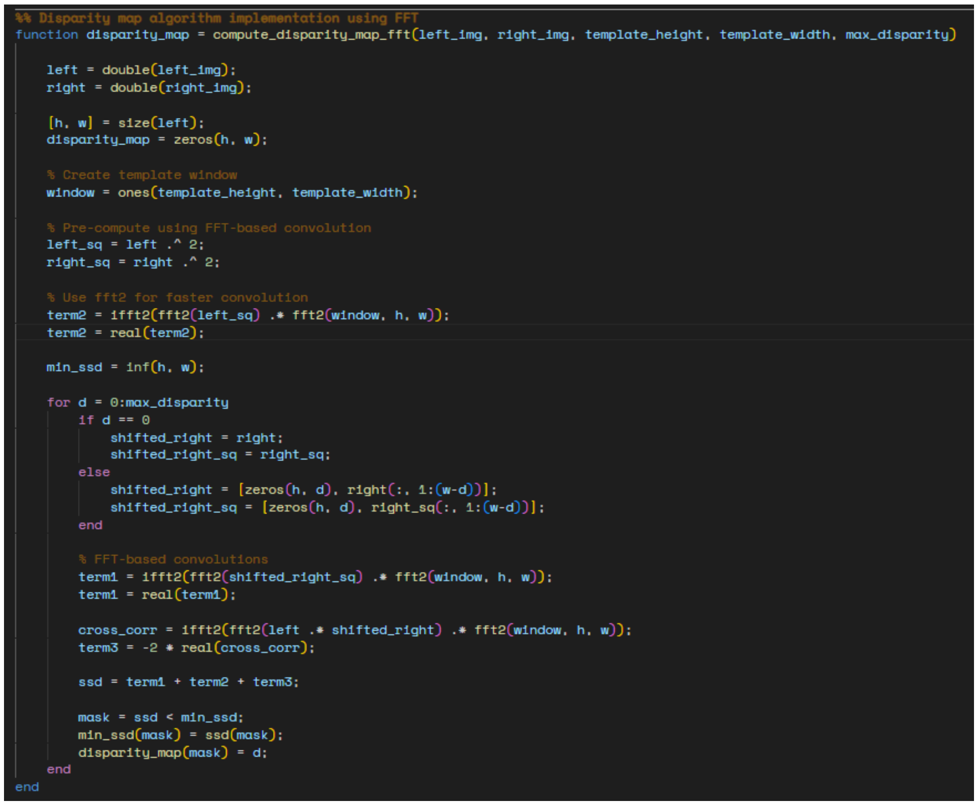
**Sum of Squared Differences (SSD) Matching**

To find corresponding pixels, we use the **SSD** metric, which measures the similarity between image patches. Direct evaluation of the SSD for every possible template position is computationally expensive. However, the SSD can be decomposed algebraically into three separable terms:

$$S(x, y) = \sum_{j=0}^{M}\sum_{k=0}^{N}\left(I(x+j, y+k)\right)^2 + \sum_{j=0}^{M}\sum_{k=0}^{N}\left(T(j,k)\right)^2 - 2\sum_{j=0}^{M}\sum_{k=0}^{N} I(x+j, y+k)\cdot T(j,k)$$

The second term is constant with respect to different parts of the image. First and third terms can be expressed as convolution which can be efficiently computed via **Fast Fourier Transforms (FFT)** operations.

## 3.2 Implementation



```
%% Disparity map algorithm implementation using FFT
function disparity_map = compute_disparity_map_fft(left_img, right_img, template_height, template_width, max_disparity)

    left = double(left_img);
    right = double(right_img);

    [h, w] = size(left);
    disparity_map = zeros(h, w);

    % Create template window
    window = ones(template_height, template_width);

    % Pre-compute using FFT-based convolution
    left_sq = left .^ 2;
    right_sq = right .^ 2;

    % Use fft2 for faster convolution
    term2 = ifft2(fft2(left_sq) .* fft2(window, h, w));
    term2 = real(term2);

    min_ssd = inf(h, w);

    for d = 0:max_disparity
        if d == 0
            shifted_right = right;
            shifted_right_sq = right_sq;
        else
            shifted_right = [zeros(h, d), right(:, 1:(w-d))];
            shifted_right_sq = [zeros(h, d), right_sq(:, 1:(w-d))];
        end

        % FFT-based convolutions
        term1 = ifft2(fft2(shifted_right_sq) .* fft2(window, h, w));
        term1 = real(term1);

        cross_corr = ifft2(fft2(left .* shifted_right) .* fft2(window, h, w));
        term3 = -2 * real(cross_corr);

        ssd = term1 + term2 + term3;

        mask = ssd < min_ssd;
        min_ssd(mask) = ssd(mask);
        disparity_map(mask) = d;
    end
end
```

*Figure 17. 3D Stereo Vision - Disparity Computation using FFT code*

## 3.3 Experimental Setup

We evaluated the disparity estimation algorithm on a synthetic stereo pair consisting of corridorl.jpg and corridorr.jpg. These images depict a corridor scene with well-defined geometric structures, textured surfaces and controlled lighting conditions. The nature of these images ensures perfect rectification, establishing a baseline to assess how the algorithm performs under ideal conditions before applying it to real-world images with practical challenges.

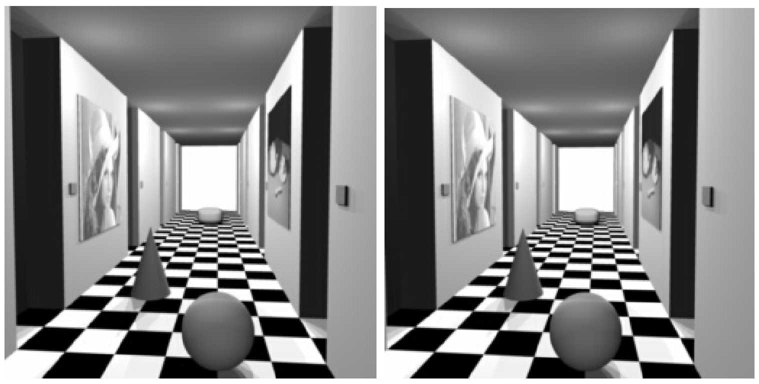


*Figure 18. 3D Stereo Vision - Synthetic Stereo Pair: Corridor Left and Right Images*

## 3.4 Results and Analysis

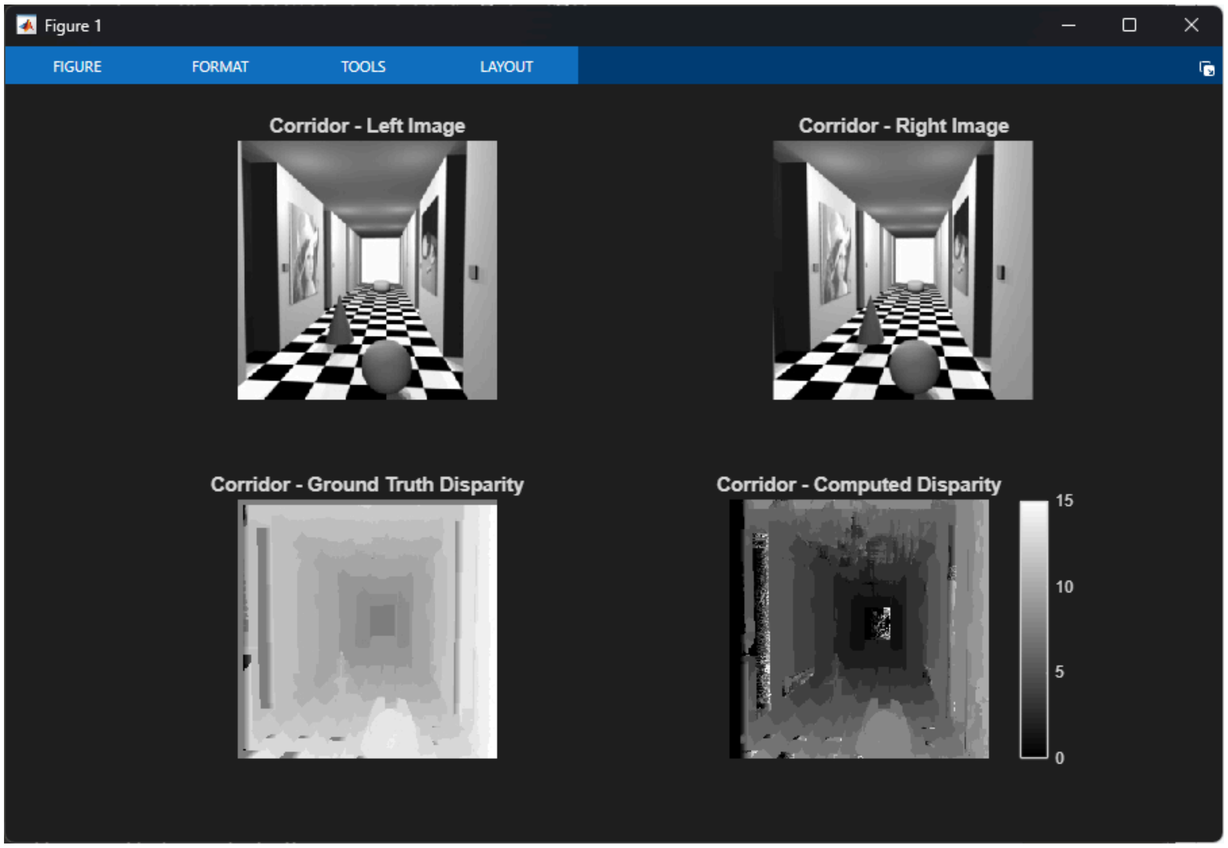


*Figure 19. 3D Stereo Vision - Synthetic Stereo Pair: Corridor Left and Right Images with Ground Truth and Computed Disparity Map*

As shown in Figure 19, the computed disparity successfully reconstructed the depth structure of the corridor scene. Demonstrating strong overall agreement with the ground truth disparity map.

The algorithm effectively captured the fundamental depth relationships within the scene. As we travel deeper into the corridor, disparity values progressively decrease, with regions transitioning into darker shades that correctly indicate increasing distance from the camera.The sphere and the cone in the foreground appears bright with high disparity values.

However, localized errors are visible at the end of the corridor, where speckles of white pixels appear in regions that should have uniformly low disparities. These bright noises falsely indicate that portions of the distant wall are closer than they actually are. This issue occurs primarily because the uniform appearance of the far wall provides insufficient discriminative features within the **11x11** template, leading to matching ambiguity. When multiple candidate positions produce similar **SSD** values, the algorithm occasionally selects incorrect high disparity matches, resulting in the visible noise. Despite these localized errors, the overall depth structure remains clearly interpretable across the majority of the scene.

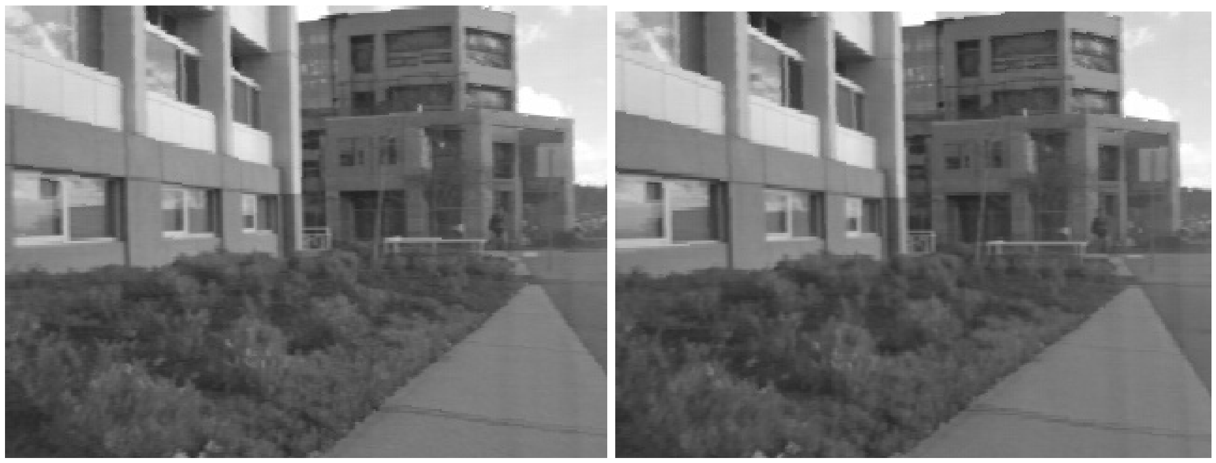


*Figure 20. 3D Stereo Vision - Real-world Stereo Pair: Triclopsi2 Left and Right Images*

Following the synthetic baseline evaluation, we applied the same disparity estimation algorithm to a real-world stereo pair, triclopsi2l.jpg and triclopsi2r.jpg. This outdoor scene features buildings, vegetation, pavement and sky. Unlike the synthetic corridor, this real-world capture introduces practical challenges including sensor noise and lighting variations.

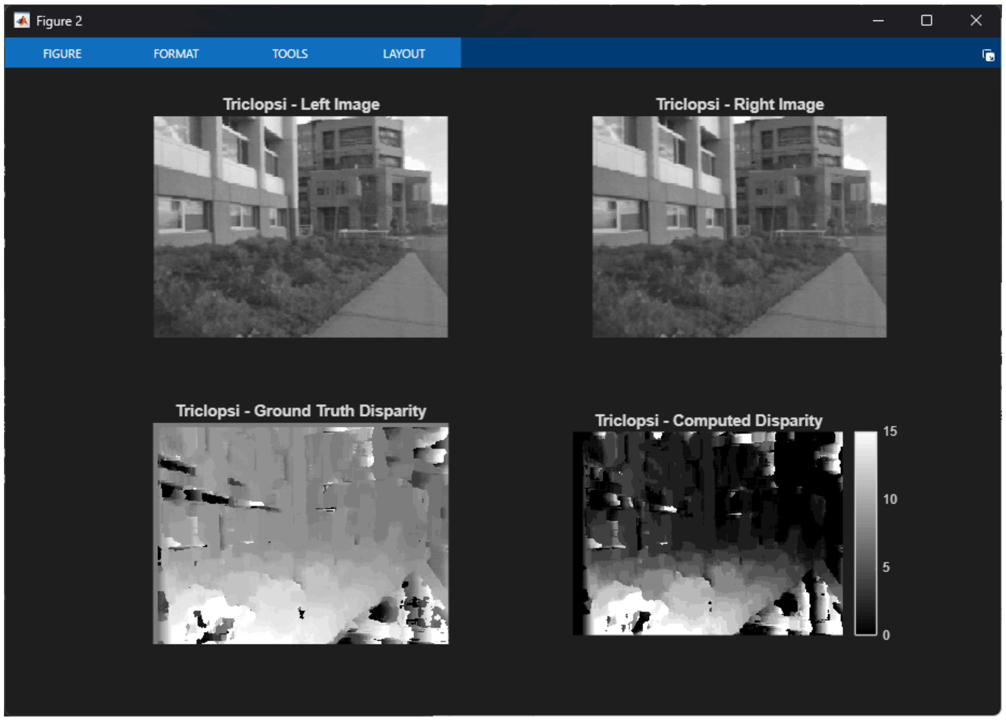


*Figure 21. 3D Stereo Vision - Real-world Stereo Pair: Triclopsi2 Left and Right Images with Ground Truth and Computed Disparity Map*

The computed disparity map partially reconstructed the depth structure but revealed significant limitations when applied to real-world imagery, as shown in Figure 21. While some scene elements are recognizable, the overall quality degraded substantially compared to the synthetic corridor results.

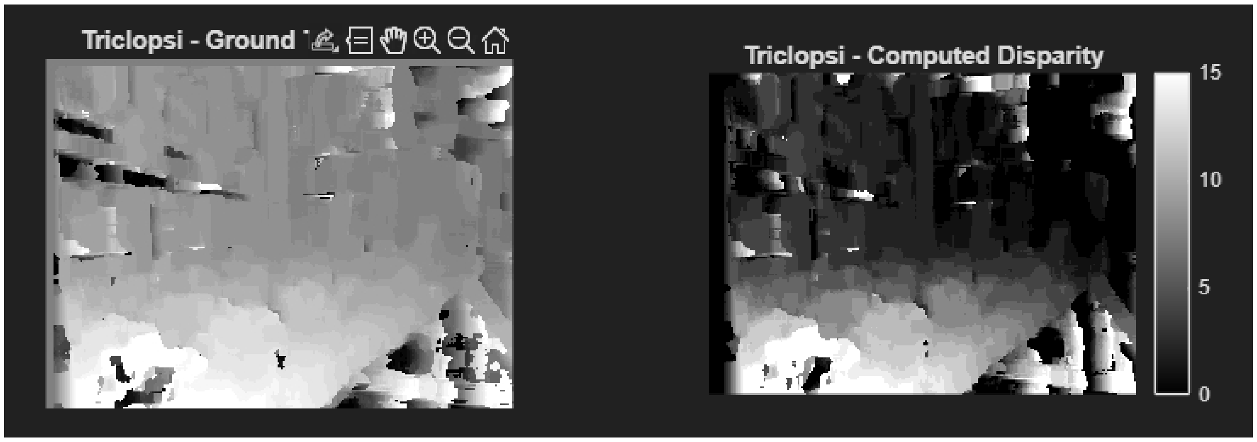


*Figure 21. 3D Stereo Vision - Real-world Stereo Pair: Triclopsi2 ‘s Ground Truth and Computed Disparity Map*

The bushes in the foreground are clearly outlined and exhibit a gradient darkening pattern as they extend deeper into the scene, correctly indicating increasing distance from the camera. This demonstrates that regions with rich texture enable reasonable disparity estimation. However, the algorithm struggles with building structures. Two distinct buildings exist in the scene: one closer to the camera positioned alongside the bushes and another at the far end. Despite their substantial separation in depth, both buildings exhibit similar disparity values in the computed map. Judging solely from the disparity representation, one might incorrectly conclude these are part of a single building at a fixed depth rather than two separate structures at different distances.This failure indicates the algorithm cannot reliably establish correspondence on relatively uniform area.

The background presents even more severe issues. The furthest building and sky region blend together in the disparity map, showing similar values despite representing fundamentally different scene elements at vastly different depths. The sky, being infinitely distant, should exhibit near-zero disparity, yet it shares values with the building structure because both regions lack sufficient texture for reliable matching. Only the textured vegetation demonstrates the expected gradual darkening with increasing depth, while any uniform appearance with large homogenous regions fails to produce any coherent depth estimates.

The difference in performance between the synthetic corridor.jpg and real triclopsi.jpg images highlights the gap between ideal and practical stereo vision. The corridor's deliberate design with strong patterns enabled near-perfect reconstruction. However, when applied to real-world scenes, the algorithm struggles significantly. The mixture of uniform architecture and textureless sky in the triclopsi.jpg image exposes the fundamental limitations of basic SSD matching, revealing that techniques successful under ideal conditions may fail when confronted with the complexity of practical environments.