Removing the projective distortion from a perspective image of a plane

H3Art
International School, Jinan University
Computer Science & Technology

Abstract

This report presents an analysis of the Direct Linear Transform (DLT) and normalized DLT methods for projective distortion correction in images. The study began with an evaluation using four feature points and extended to experiments with ten points to assess improvements in accuracy and robustness. The experiments were conducted using pairs of oblique and front-view images of various scenes, manually selecting feature points at prominent edges or corners. The results showed that increasing the number of points led to more stable homography estimations, and the normalized DLT method exhibited greater consistency in cases with unevenly distributed or scaled points. While both methods effectively addressed projective distortions, the normalized DLT was notably more resilient in handling challenging input conditions, making it the preferred choice for real-world image corrections.

1 Introduction

In digital image processing, the projective distortion occurs when the plane of the image sensor is not parallel to the plane of the subject being captured, leading to a warped representation where straight lines in the real world may appear curved or skewed in the image. Such distortions complicate tasks that require accurate measurements, alignment, or comparisons between images, making it essential to correct for these effects to recover the original geometric properties.

The primary objective of this report is to estimate a homography transformation between two-dimensional planes to remove projective distortion from perspective images. By applying homography, one can restore an image taken at an angle to its frontal, undistorted view, allowing for precise geometric interpretation and analysis.

To achieve this, we will implement the Direct Linear Transform (DLT) and normalized DLT algorithms. The DLT method is for estimating homography based on a set of corresponding points in the original and transformed images. The normalized DLT, an extension of the basic DLT, enhances numerical stability by normalizing point coordinates before computing the homography matrix. This additional step is particularly beneficial when dealing with images where the points are distributed unevenly or the scales differ significantly.

This assignment requires conducting experiments using a minimal set of four corresponding points as well as cases involving ten or more points. The minimal case allows us to explore the basic feasibility of the transformation, while the expanded case demonstrates robustness and accuracy improvements with more input data.

Through this work, we aim to showcase how homography can effectively correct projective distortion, thereby enhancing the utility of images in practical applications. The result is a visual demonstration of how these techniques restore original perspectives in images, facilitating tasks such as planar measurement, architectural analysis, and document scanning.

2 Methods

2.1 Overview of Approach

The goal of this project is to estimate the homography matrix H that maps points between two planes in a 2D space. This involves removing projective distortion to transform a perspective image into a fronto-parallel view. To achieve this, we implemented both the DLT and the normalized DLT algorithms. The process includes the manual or automatic selection of corresponding points between source and target images, construction of matrix A for homography estimation, and refinement of results using normalization techniques to ensure numerical stability.

2.2 DLT Method

The DLT method is a standard approach for estimating the homography matrix using a set of at least four corresponding points between two images. The goal is to find H such that for each point correspondence $(x_i, y_i) \to (x'_i, y'_i)$, the following equation holds:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix},$$

where H is a 3×3 matrix.

Expanding the homography equation leads to:

$$x'_i = h_{11}x_i + h_{12}y_i + h_{13}, \quad y'_i = h_{21}x_i + h_{22}y_i + h_{23}.$$

To make this system linear in terms of the elements of H, we rearrange the equations as follows:

$$-h_{11}x_i - h_{12}y_i - h_{13} + h_{31}x_ix'_i + h_{32}y_ix'_i + h_{33}x'_i = 0,$$
$$-h_{21}x_i - h_{22}y_i - h_{23} + h_{31}x_iy'_i + h_{32}y_iy'_i + h_{33}y'_i = 0.$$

Each correspondence contributes two rows to the matrix A:

$$A = \begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_ix'_i & y_ix'_i & x'_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_iy'_i & y_iy'_i & y'_i \end{bmatrix}_{2N \times 9},$$

where N is the number of point correspondences.

To solve $A\mathbf{h} = 0$, where $\mathbf{h}$ is a vector of the elements of H:

$$\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T,$$

we use Singular Value Decomposition (SVD) to decompose A as $A = U\Sigma V^T$. The solution $\mathbf{h}$ corresponds to the last column of V, which minimizes $\|A\mathbf{h}\|$.

Finally, we reshape $\mathbf{h}$ into a 3×3 matrix H and normalize it by setting $h_{33} = 1$:

$$H = \frac{H}{h_{33}}.$$

To implement the DLT, we take the following steps:

1. **Point Selection**: We begin by selecting $N \geq 4$ corresponding points in both images. This step can be done manually using the 'ginput' function or through automatic methods such as feature detection and matching (e.g., SURF or ORB).

2. **Construction of Matrix** A: For each corresponding point pair (x_i, y_i) and (x'_i, y'_i), we construct the matrix A as described above.

3. **SVD**: The solution to $A\mathbf{h} = 0$ is obtained by performing SVD on A. The homography vector $\mathbf{h}$ corresponds to the last column of V (right singular vectors), reshaped into a 3×3 matrix:

$$H = \text{reshape}(\mathbf{v}_9, 3, 3).$$

4. **Normalization of** H: To ensure consistency, the matrix H is normalized by dividing all elements by h_{33} (i.e., $H = H/h_{33}$).

2.3 Normalized DLT Method

The normalized DLT method enhances the numerical stability of the basic DLT by normalizing the point coordinates before constructing matrix A. Normalization involves translating and scaling points so that they are centered at the origin and have an average distance of $\sqrt{2}$ from the origin.

Given a set of points (x_i, y_i), we define the normalization transformation T as:

$$T = \begin{bmatrix} \frac{1}{\sigma_x} & 0 & -\frac{\mu_x}{\sigma_x} \\ 0 & \frac{1}{\sigma_y} & -\frac{\mu_y}{\sigma_y} \\ 0 & 0 & 1 \end{bmatrix},$$

where μ_x, μ_y are the mean coordinates and σ_x, σ_y are the standard deviations of x and y coordinates, respectively.

The points (x_i, y_i) are then transformed as:

$$\tilde{x}_i = T \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}.$$

The implementation of Normalized DLT is as follows:

1. **Normalize Source and Target Points**: Apply normalization transformations T and T' to source and target points, respectively.

2. **Construct Matrix A Using Normalized Points**: Construct A using the normalized points $(\tilde{x}_i, \tilde{y}_i)$ and $(\tilde{x}'_i, \tilde{y}'_i)$.

3. **Compute $H_{\text{normalized}}$**: Use SVD to find $H_{\text{normalized}}$.

4. **Denormalize the Homography**: Compute the final homography matrix H by denormalizing:

$$H = T'^{-1} H_{\text{normalized}} T.$$

2.4 Detailed Mathematical Differences

- **Standard DLT**: Constructs A directly from the original point coordinates and may suffer from numerical instability if the points are not well-distributed or vary greatly in scale.

- **Normalized DLT**: Applies a pre-normalization step to the point coordinates, ensuring the points have a similar scale and are centered. This significantly improves the condition number of the matrix A and leads to a more numerically stable solution.

3 Data

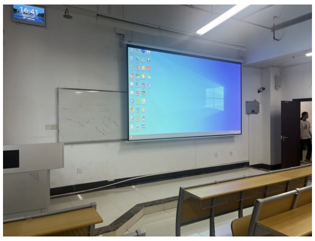
The data used for this project consists of paired photographs. All photos were taken personally using smartphone, ensuring control over the quality and conditions of the images. The source of the data is entirely self-collected from my immediate environment. Each pair includes one oblique-angled photo and one front-view photo of the same scene. These photos were captured to represent a variety of contexts found in daily life, such as items, buildings, and space view.

(a) Oblique view of laptop

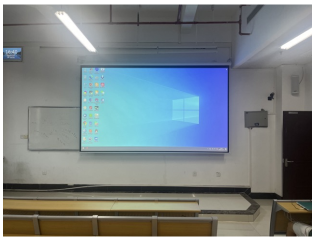
(b) Front view of laptop

(c) Oblique view of pillar

(d) Front view of pillar



(e) Oblique view of classroom



(f) Front view of classroom

(g) Oblique view of building

(h) Front view of building

Figure 1: Experimental photo pairing.

The project primarily focuses on working with multiple photo pairs. Although there is no fixed number of photo pairs set for this study, the experiment includes a minimum of one pair for detailed analysis. This approach allows for flexibility in scaling up the analysis with additional pairs if needed to validate the findings.

The raw images used in this project did not undergo any special preprocessing steps. This choice was made to maintain the authenticity of real-world scenarios and simplify the workflow. However, manual feature point selection will be performed on each image pair during the experiment to identify corresponding points, ensuring that the DLT and normalized DLT algorithms can be applied accurately.

4 Experimental results

To validate the effectiveness of the DLT and normalized DLT methods in solving the problem of projective distortion correction, a series of experiments were conducted using a simple strategy of marking the four corners of prominent quadrilateral objects in images. This approach was chosen for its simplicity and intuitive application, especially when dealing with images containing distinct rectangular or planar structures.

The figure 2 below illustrate the transformed images using the DLT method and those using the normalized DLT method:

(a) The DLT transformed result of laptop (b) The normalized DLT transformed result of laptop (c) The DLT transformed result of pillar (d) The normalized DLT transformed result of pillar

(e) The DLT transformed result of classroom (f) The normalized DLT transformed result of classroom (g) The DLT transformed result of building (h) The normalized DLT transformed result of building

Figure 2: Experiment results of Four-Corner Strategy.

For the Four-Corner Strategy, it has the following advantages:

1. **Ease of Implementation**: The manual marking of four corners in the images is straightforward and easy to understand, making it a suitable method for initial testing and demonstration.

2. **Computational Simplicity**: Four points represent the minimum number required for computing the homography matrix, allowing the DLT algorithm to proceed smoothly.

3. **Clear Visualization**: This method makes it easy to visually assess the transformation results, especially when working with objects that have clear boundaries.

However, there are also few limitaion when we only take the Four-Corner Strategy:

1. **Dependence on Point Precision**: The accuracy of the transformation is highly dependent on the precision of manually selected corner points. Any slight deviation in point selection can impact the final result.

2. **Applicability to Planar Objects**: This approach works best when the object of interest is strictly planar. If the object deviates from being perfectly planar, the transformation results may be suboptimal.

3. **Limited Complexity Handling**: The four-corner strategy may struggle to capture sufficient geometric information for complex or non-rectangular shapes, limiting its effectiveness in more intricate scenarios.

Initial results obtained from two sets of experiments using this four-corner marking strategy showed that the DLT method could produce acceptable transformations for basic projective corrections. However, when subtle perspective changes or less ideal point selections were involved, the difference between the DLT and normalized DLT results was not significant, or they failed to achieve the expected correction.

To extend the initial experiments, I adopted a strategy of selecting 10 corresponding points on each image pair to evaluate the impact on the transformation results. The purpose of this approach was to investigate whether using more points could improve the precision and robustness of both the DLT and normalized DLT transformations. For the selection of these 10 points, I primarily focused on choosing prominent edges or corners of objects in the images to achieve this strategy.

The results figure 3 from the eight image pairs transformed using both the DLT and normalized DLT methods with 10 selected points are shown in the figures provided.

(a) The DLT transformed result of laptop (b) The normalized DLT transformed result of laptop (c) The DLT transformed result of pillar (d) The normalized DLT transformed result of pillar

(e) The DLT transformed result of classroom (f) The normalized DLT transformed result of classroom (g) The DLT transformed result of building (h) The normalized DLT transformed result of building

Figure 3: Experiment results of 10 corresponding points selection.

Upon analysis, several observations were made:

1. **Improvement in Accuracy**: Compared to the previous experiments using only four points, the transformations obtained with 10 points showed a notable improvement in terms of alignment and reduced distortion. This suggests that increasing the number of corresponding points helps to better capture the underlying geometric relationships, leading to a more stable homography estimation.

2. **DLT vs. Normalized DLT Comparison**: When comparing the results of the DLT and normalized DLT methods using 10 points, there was some distinction in the outcomes. The normalized DLT transformations appeared slightly more consistent, especially in cases where the selected points varied greatly in scale or were not evenly distributed across the image. This aligns with the known advantage of normalized DLT in handling numerical stability more effectively.

3. **Scenarios of Minimal Difference**: In certain image pairs, the difference between the DLT and normalized DLT results was minimal. This occurred when the distribution of points was already balanced and the scaling differences were not significant. It demonstrates that the normalized DLT's benefits are more pronounced in cases where the input points have varied scales or are distributed unevenly.

Although the differences between DLT and normalized DLT were subtle in some cases, more noticeable improvements were observed under the following conditions:

- **Highly Skewed Points**: When the corresponding points were located at varying scales or distances from each other, the normalized DLT method consistently produced better-aligned results.
- **Larger Images**: For larger images where the point coordinates had a broader range, normalization played a crucial role in maintaining the accuracy of the transformation.

While the basic DLT method worked reasonably well with 10 points, the normalized DLT exhibited advantages in handling cases where the corresponding points were distributed unevenly or had significant scale differences. This reinforced the known theoretical benefit that normalization provides: reducing numerical instability and ensuring a better-conditioned matrix A.

In summary, while both DLT and normalized DLT methods can effectively correct projective distortions, the normalized DLT method's strength lies in its ability to handle more challenging point distributions and scale variations. This makes it a preferred approach when dealing with real-world images that might not have uniformly distributed corresponding points.

5 Conclusion

In this report, we implemented and analyzed the Direct Linear Transform (DLT) and normalized DLT methods for correcting projective distortion in images. The experiments demonstrated that using the minimum four points could yield basic transformation results, but expanding to ten feature points significantly improved the accuracy and robustness of the homography estimation. The comparison between DLT and normalized DLT revealed that while both methods effectively corrected distortions, normalized DLT performed better in handling unevenly distributed or varied-scale points due to its enhanced numerical stability. These findings support the use of normalized DLT for more reliable outcomes in practical applications involving real-world images.

6 Appendix: Source Code

```
clc;
clear;

% Read images
I1 = imread('source.jpg'); % Source image
I2 = imread('target.jpg'); % Target image

% Select corresponding points in the source and target images
numPoints = 4; % or 10
figure; imshow(I1);
[x1, y1] = ginput(numPoints);
figure; imshow(I2);
[x2, y2] = ginput(numPoints);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DLT method%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Construct the DLT matrix A
A = [];
for i = 1:numPoints
    A = [A;
        -x1(i) -y1(i) -1 0 0 0 x1(i)*x2(i) y1(i)*x2(i) x2(i);
        0 0 0 -x1(i) -y1(i) -1 x1(i)*y2(i) y1(i)*y2(i) y2(i)];
end

% Compute the homography matrix H using SVD
[~, ~, V] = svd(A);
H = reshape(V(:, end), 3, 3)'; % Use the last column of V as H
H = H / H(3, 3); % Normalize H

% Create projective2d object and apply the DLT transformation
H_projective = projective2d(H');
outputImageDLT = imwarp(I1, H_projective, 'OutputView', imref2d(size(I2)));

%%%%%%%%%%%%%%%%%%%%%%%%%Normalized DLT method%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define normalization function
normalizePoints = @(pts) deal(mean(pts, 1), std(pts(:)), (pts - mean(pts, 1)) / std(
    pts(:)));

% Normalize source and target points
[mean1, std1, normPts1] = normalizePoints([x1, y1]);
```

```
[mean2, std2, normPts2] = normalizePoints([x2, y2]);

% Construct normalization matrices T
T1 = [1/std1, 0, -mean1(1)/std1; 0, 1/std1, -mean1(2)/std1; 0, 0, 1];
T2 = [1/std2, 0, -mean2(1)/std2; 0, 1/std2, -mean2(2)/std2; 0, 0, 1];

% Construct the DLT matrix A (using normalized points)
A = [];
for i = 1:numPoints
    x1n = normPts1(i, 1);
    y1n = normPts1(i, 2);
    x2n = normPts2(i, 1);
    y2n = normPts2(i, 2);
    A = [A;
        -x1n -y1n -1 0 0 0 x1n*x2n y1n*x2n x2n;
        0 0 0 -x1n -y1n -1 x1n*y2n y1n*y2n y2n];
end

% Compute normalized H
[~, ~, V] = svd(A);
H_normalized = reshape(V(:, end), 3, 3)';
H_normalized = H_normalized / H_normalized(3, 3);

% Denormalize H
H_norm = inv(T2) * H_normalized * T1;
H_norm = H_norm / H_norm(3, 3);

% Create projective2d object and apply the normalized DLT transformation
H_projective_norm = projective2d(H_norm');
outputImageNormDLT = imwarp(I1, H_projective_norm, 'OutputView', imref2d(size(I2)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Result Display%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Display results
figure;
subplot(1, 4, 1), imshow(I1), title('Original Image');
subplot(1, 4, 2), imshow(I2), title('Target Image');
subplot(1, 4, 3), imshow(outputImageDLT), title('Transformed Image using DLT');
subplot(1, 4, 4), imshow(outputImageNormDLT), title('Transformed Image using Normalized DLT');
```