Exposure Balancing and Extraction from Distorted Images

Avunuri Manideep, Damireddy Mohith Reddy, Satyam Kumar and Yash Raj

***Abstract*—This report presents a generous approach for extracting focused objects from skewed images. To address the tilt or other distortions in the image, homography using Direct Linear Transformation (DLT) is employed to achieve the desired perspective transformation. To enhance the visibility of the focused object, various exposure correction techniques, including adaptive gamma correction , are applied. Finally, the focused object is extracted from the image using the GrabCut algorithm, ensuring accurate isolation of the object .**

***Index Terms*—homography, perspective transformation, adaptive gamma correction, grab cut algorithm**

I. Introduction

In real-world scenarios, images are often distorted due to factors such as shaky hands, tilted viewpoints, or image processing failures. These distortions make it challenging to extract focused objects accurately from an image. Additionally, visibility issues arise when obstructions or poor lighting affect the clarity of the objects in the scene.

This problem is addressed by a straightforward yet effective three-step approach. The process begins with homography, a transformation matrix that defines the relationship between two perspectives of an image. This step accounts for distortions such as rotation, translation, or skew, aligning the image to a desired perspective, such as a bird's eye view.

Next, to enhance the visibility of the object, exposure correction methods like adaptive gamma correction are applied. These techniques improve the image's visual quality by adjusting the contrast and brightness, making the focused object more visible.

Finally, the focused object is extracted from the corrected image using the GrabCut algorithm. GrabCut is a graph-based segmentation technique that separates the foreground (the object) from the background. By iteratively refining the segmentation, it results in an accurate extraction of the object.

II. Methodology

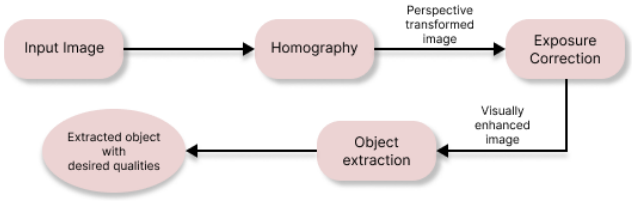

Fig. 1: Project Flow

A. Homogrpahy for Perpective Transformation

In our project we are establishing a relationship between two planes that is Source plane and the other is Target plane.This relationship is estimated through the homography matrix.We transform our perspective to our desired plane. This transformation can be achieved by applying homography matrix.

This Homography matrix is estimated by a method called Direct Linear Transformation(DLT) which is based on solving a system of linear equations.

1) Homography Matrix: A homography is represented by a 3×3 matrix, denoted as H, which transforms points in one plane to points in another. If a point (x, y) in the source plane is mapped to a point (x', y') in the target plane, the transformation can be expressed as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where H is the homography matrix and the coordinates are in homogeneous form. The homography matrix H is given by:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

This matrix contains 8 unknown parameters because the 9th parameter can be normalized to 1 by dividing the matrix by h_9.

2) Direct Linear Transformation (DLT): The DLT method is used to compute the homography matrix from point correspondences between two images. For each point correspondence (x_i, y_i) in the source plane and (x_i', y_i') in the target plane, we can set up a system of linear equations.

Let the source points be $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ and the target points be $\{(x_1', y_1'), (x_2', y_2'), \ldots, (x_N', y_N')\}$. Each pair of points contributes two equations, resulting in a system of $2N$ equations with 8 unknowns (since h_9 can be normalized).

The equations for a single point correspondence $(x_i, y_i) \leftrightarrow (x_i', y_i')$ are derived from the relationship:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

This gives two equations:

$$x_i' = h_1 x_i + h_2 y_i + h_3$$
$$y_i' = h_4 x_i + h_5 y_i + h_6$$
$$1 = h_7 x_i + h_8 y_i + h_9$$

For each correspondence, these equations can be written as a row in the matrix A, which is used to solve for h (the flattened homography matrix).

$$A = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1'x_1 & x_1'y_1 & x_1' \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & y_1'x_1 & y_1'y_1 & y_1' \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2'x_2 & x_2'y_2 & x_2' \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y_2'x_2 & y_2'y_2 & y_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

This matrix is then solved using Singular Value Decomposition (SVD).

3) Singular Value Decomposition (SVD): To solve the system of equations $A \cdot h = 0$, we use Singular Value Decomposition (SVD) on matrix A. The SVD of A is given by:

$$A = U \cdot \Sigma \cdot V^T$$

where: - U is the matrix of left singular vectors, - Σ is the diagonal matrix of singular values, - V^T is the matrix of right singular vectors.

The solution to $A \cdot h = 0$ is the last column of V (or equivalently the last row of V^T), corresponding to the smallest singular value. This is the homography vector h, which can be reshaped into a 3×3 matrix.

4) Computing the Homography Matrix: Once we obtain the vector h from the SVD, we reshape it into a 3×3 matrix:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

The final homography matrix is obtained by normalizing H so that $h_9 = 1$, which ensures that the matrix has the correct scale.

5) Warping: Warping is nothing but transforming the source plane. Once we have the Homography Matrix H we apply it on the image we get the transformed image. Many a times after applying H we get floating point coordinates of the transformed image. The pixel value at the floating points is evaluated by bilinear interpolation.

B. Exposure Correction Methods

Some standard exposure correction methods are implemented in this project. The implementation of the same are discussed below.

Fig. 2: Island name before homography

Fig. 3: Island name after homography

Fig. 4: IIT name before homography

Fig. 5: IIT name after homography

1) Histogram Equalization: In the histogram equalization function, a colour image is taken as input. Then, channel extraction is performed for each colour, i.e. red, green and blue, and the histogram is obtained for each colour. After obtaining the histogram, the cumulative sum is taken for each of the intensity levels present in the image and normalization of this cumulative sum is done. Now, this normalized sum is mapped to the respective intensity values, and the histogram equalization is performed with the help of the OpenCV library function.The probability density function (PDF) can be approximated as:

$$\text{PDF}(l) = \frac{n_l}{MN},$$

where n_l represents the number of pixels with intensity l, and MN is the total number of pixels in the image. The cumulative distribution function (CDF) is derived from the PDF and is defined as:

$$\text{CDF}(l) = \sum_{k=0}^{l} \text{PDF}(k).$$

Once the CDF of the digital image is calculated, the histogram equalization method utilizes the CDF as a transformation function given by:

$$T(l) = \text{CDF}(l) \cdot l_{\text{max}}.$$

2) Gamma Correction: In the gamma correction function, a colour image is taken as input along with a gamma value. First, the inverse of the gamma value is calculated to simplify the correction process. Then, a lookup table is created, which maps each possible intensity value (from 0 to 255) to its gamma-corrected value using the formula

$$\text{Corrected value} = \left(\frac{\text{Pixel value}}{255}\right)^{\gamma} \times 255$$

This table helps precompute the corrections for all intensity levels, making the process more efficient. Finally, the input image's pixel values are replaced with the corresponding values from the lookup table using the OpenCV LUT function, resulting in a gamma-corrected image.

3) Log Transformation: Log transformation is used for compressing high-intensity values and expanding low-intensity values. So, when dealing with images, it must be ensured that the output of this function lies within the input pixel intensity range. A constant c is calculated to ensure that:

$$c = \frac{255}{\log(1 + \text{max pixel value})}$$

At the conclusion, the log transformation is performed using:

$$\text{Final value} = c \cdot \log(1 + \text{pixel value})$$

4) Contrast Stretching: A grayscale or colour image is input into the contrast stretching function. First, the image is converted to a floating-point format for precise calculations. The minimum and maximum pixel intensity values in the image are then identified. Using these values, contrast stretching is performed by applying the formula:

$$\frac{\text{Pixel value} - \text{Minimum value}}{\text{Maximum value} - \text{Minimum value}} \times 255$$

This formula spreads the pixel intensities over the full range (0–255), enhancing the contrast by utilizing the entire intensity range. Finally, the transformed pixel values are clipped to ensure they lie within the valid range (0–255) and converted back to an 8-bit integer format. The result is a contrast-stretched image with enhanced visibility.

5) Adaptive Gamma Correction: Traditional gamma correction (TGC) and histogram equalization (THE) methods have limitations. As shown in Fig. 6, TGC produces uniform modifications using predefined parameters, while THE relies on histogram properties, often causing over-enhancement or under-enhancement. These issues, highlighted in Fig. 6(d), include decreased low intensities, excessively increased moderate intensities, and significantly reduced high intensities [**?**].

Adaptive gamma correction (AGC) dynamically adjusts the brightness and contrast of an image, enhancing dark regions while preserving details in brighter areas. It offers precise control through a weighting parameter.

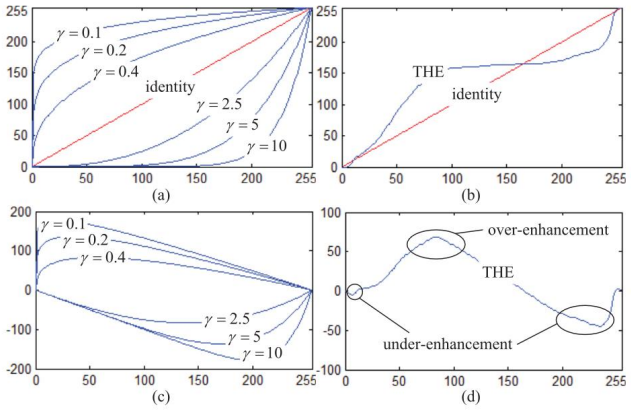

Fig. 6: Transformation curves illustrated by a)gamma correction and b)THE methods, with their corresponding intensity level modifications shown in (c) and (d).

The proposed adaptive gamma correction (AGC) is formulated as follows:

$$T(l) = l_{\max}\left(\frac{l}{l_{\max}}\right)^{\gamma} = l_{\max}\left(\frac{l}{l_{\max}}\right)^{1-\text{CDF}(l)}.$$

The AGC method can progressively enhance low intensities and prevent significant reduction of high intensities. Furthermore, the weighting distribution (WD) function is applied to slightly adjust the statistical histogram and mitigate adverse effects . The WD function is expressed as:

$$\text{PDF}_w(l) = \text{PDF}_{\max}\left(\frac{\text{PDF}(l) - \text{PDF}_{\min}}{\text{PDF}_{\max} - \text{PDF}_{\min}}\right)^{\alpha},$$

where α is the adjustment parameter, $\text{PDF}_{\max}$ is the maximum value of the statistical histogram's PDF, and $\text{PDF}_{\min}$ is the minimum value of the PDF. Now, the modified CDF is approximated as:

$$\text{CDF}_w(l) = \sum_{l=0}^{l_{\max}} \frac{\text{PDF}_w(l)}{\sum_{l=0}^{l_{\max}} \text{PDF}_w(l)}.$$

The sum of PDF_w is calculated as:

$$\sum \text{PDF}_w = \sum_{l=0}^{l_{\max}} \text{PDF}_w(l).$$

Finally, the gamma parameter based on the CDF is modified as:

$$\gamma = 1 - \text{CDF}_w(l).$$

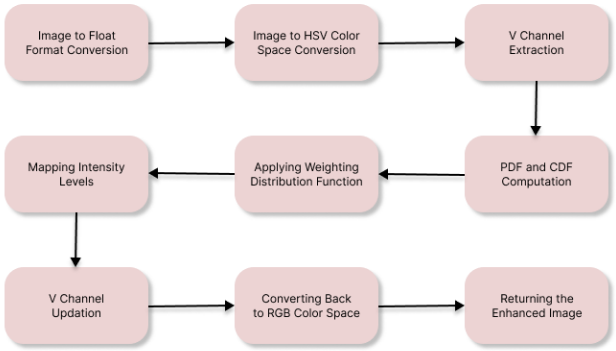

Fig. 7: Adaptive Gamma Correction

Fig. 8: Original Image

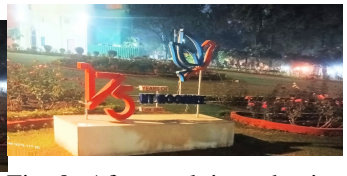

Fig. 9: After applying adaptive gamma correction

C. Object Extraction

We have made the object extraction using two different methods, the first one using grabcut algorithm and the second one using contour edge detection.

1) GrabCut Algorithm: GrabCut algorithm is an interactive image extraction technique that efficiently separates foreground from the background of the specified region. This algorithm uses classical graph cut segmentation method by incorporating Gaussian Mixture probabilistic model to improve performance and stability. This algorithm defines the area in the rectangle as a color distribution model using the Gaussian Mixture Model(GMM), where each pixel will be given a label to tell whether it is a foreground, background, or unknown.

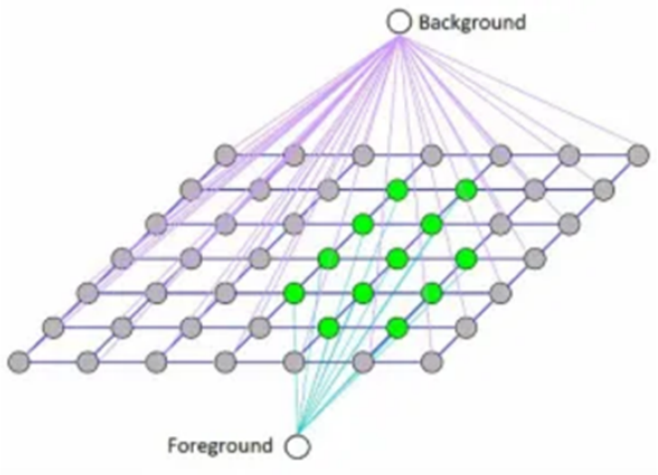

Fig. 10: Edges Connected with Source and Sink

Grabcut algorithm considers each pixel as a node and connects each of the pixel(node) in the image with two additionally introduced nodes named as the source node and the sink node. There are two types of edges in the graph

1) **T-links**: Connect each pixel node to the source or sink. The weight of a T-link represents the likelihood of a pixel belonging to the foreground or background:

$$w(x, \text{Source}) = -\log(P(x \mid \text{Foreground}))$$

$$w(x, \text{Sink}) = -\log(P(x \mid \text{Background}))$$

A higher weight to the source indicates a higher likelihood of the pixel being in the foreground. The probabilities $P(x \mid \text{Foreground})$ and $P(x \mid \text{Background})$ are derived from the GMM.

2) **N-links**: Connect neighboring pixel nodes (e.g., 4- or 8-connected neighbors) to enforce spatial smoothness. The weight of an N-link between two pixels is given by:

$$w(x, y) = \lambda \exp\left(-\frac{\|I_x - I_y\|^2}{2\sigma^2}\right)$$

where:

- I_x and I_y: Intensities (or color values) of pixels x and y.
- $\|I_x - I_y\|^2$: Squared intensity difference.
- σ: A parameter that scales the intensity difference.
- λ: A constant that controls the importance of the smoothness term.

Energy Minimization:-
The weights assigned to T-links and N-links are combined into an energy function, which the graph cut optimization minimizes:

$$E(L) = \sum_{x \in \text{pixels}} D(x, L_x) + \sum_{(x,y) \in \text{neighbors}} S(x, y, L_x, L_y)$$

where:

- L_x: Label of pixel x (foreground or background).
- $D(x, L_x)$: Data term (from T-links), based on the GMM.
- $S(x, y, L_x, L_y)$: Smoothness term (from N-links), based on intensity differences.

2) Canny Edge Detection: The process involves the following steps:

1) The input image is smoothened to remove noise.
2) The smoothened image is subtracted from the initial image to highlight the edges.
3) Canny edge detection is applied to identify all contours.

The given image is smoothened to remove any noise and the smoothened image is subtracted from the initial image to highlight the edges. Then after Canny edge detection is performed to show all the contours. To extract the object from the region of interest we need the contour that has the maximum area. But the problem is that all the edges are not connected for the main outer object so we were not able to extract the closed contour with maximum area. We

then dialated the obtained contours for connecting the edges connect which can be seen in the below image.

Now among all the detected contours the maximum area contours mask has been created after which bit wise and operation is been done for the original image and the final mask finally extracting the required object. Additionally a feature in which we can erase the extra unnecessary parts is also added to edit the image. Now, we can save our final image.

Fig. 11: Before object extraction

Fig. 12: After extraction

Fig. 13: Before object extraction

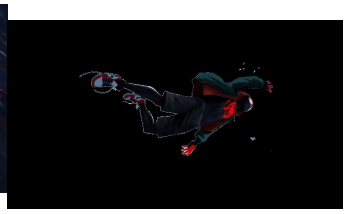

Fig. 14: After extraction

III. Experimentation and Results

Results are very amusing as you can see below 2 examples where image has been processed through out the image. We were able to correct distortions and transform the image to a desired perspective with the help of homography. Certainly, exposure correction methods have helped enhance the visibility of the image.Finally,The GrabCut Algorithm worked effectively in extracting the focused object from the image.

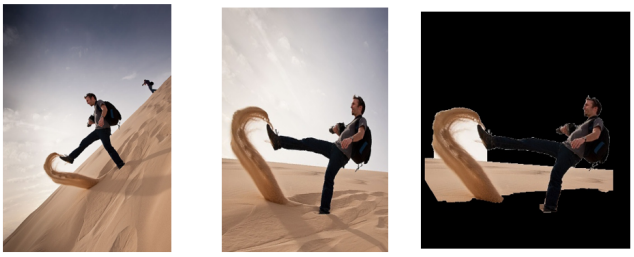

Fig. 15: Full processed image of Love desert

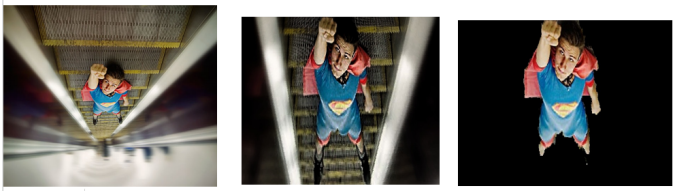

Fig. 16: Full processed image of a tilted man

IV. Conclusion

In Concluison, we were able to address the challange of extracting focused objects from distorted images.Our approach worked quite well with the images as shown in results. It provides a robust solution and handles real word image distortions and visibility issues.

References

[1] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2320–2331, Jun. 2013.

[2] J. Smith, "Estimating the homography matrix with the direct linear transform (DLT)," *Medium*, Jun. 2021. [Online]. Available: https://medium.com/@insight-in-plain-sight/estimating-the-homography-matrix-with-the-direct-linear-transform-dlt-ec6bbb82ee2b. [Accessed: Dec. 1, 2024].

[3] A. Author, "Image processing with Python: Image warping using homography matrix," *Medium*, Oct. 2020. [Online]. Available: https://medium.com/swlh/image-processing-with-python-image-warping-using-homography-matrix-22096734f09a. [Accessed: Dec. 1, 2024].

[4] Szeliski, R. (2010). "Computer Vision: Algorithms and Applications." Springer. Chapter on Graph Cuts for Segmentation covers the foundations of the method.

[5] Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing.* Pearson. Chapter on segmentation techniques includes references to energy minimization methods.

[6] Boykov, Y., & Kolmogorov, V. (2004). "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 26(9), 1124–1137.

Contributions

1) Avunuri Manideep:- Research on Homography and UI
2) Satyam Kumar:-Research on Homography and Exposure methods including Adaptive Gamma Correction
3) Damireddy Mohith Reddy:-Object extraction using Grab Cut algorithm, Canny Edge Detection and UI
4) Yash Raj:-Research on Exposure methods including Adaptive Gamma Correction and UI