**FISHEYE IMAGE CORRECTION**

**A Report on computer vision lab project**

**[CSE-(AI &ML) Sec-A]**

Submitted By:

**SAI SUJAN KORRAPATI 210962194**

**ADITYA NERUSU 21O962196**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**< NOVEMBER 2023>**

**FISHEYE IMAGE CORRECTION**

SAI SUJAN KORRAPATI , ADITYA NERUSU

CSE AI & ML MIT, INDIA

[korrapatisaisujan@gmail.com](mailto:korrapatisaisujan@gmail.com),

[adityaadi2744@gmail.com](mailto:adityaadi2744@gmail.com)

**Abstract:**

This approach demonstrates the implementation of fisheye correction using the OpenCV library. Fisheye correction is a crucial technique for rectifying distorted images captured with fisheye lenses, ensuring that they appear undistorted and natural. The code begins by calculating camera parameters based on specified heading, pitch, and field of view angles. Then, it performs fisheye correction on an input image, producing a corrected image. The code utilizes mathematical transformations to map pixels from the distorted fisheye space to the corrected space. The result is a more accurate representation of the original scene. Fisheye correction is valuable in various applications, such as panoramic photography, computer vision, and immersive experiences, where eliminating distortion is essential for accurate analysis and visualization.

**Keywords:**

* Fisheye Correction
* OpenCV
* Camera Parameters
* Image Distortion
* Panoramic Photography
* Computer Vision
* Image Transformation
* Image Correction
* Distortion Removal

**Introduction:**

Fisheye lenses are a popular choice in various applications, including photography, computer vision, and robotics, due to their unique wide-angle properties. However, fisheye distortion is a common challenge associated with these lenses, resulting in significant geometric distortion in images. Correcting fisheye distortion is crucial for achieving accurate and visually pleasing results in various imaging applications.

Theis approach addresses the problem of fisheye distortion correction using computer vision techniques. It employs the OpenCV library to rectify fisheye-distorted images. In this introduction, we will provide an overview of fisheye distortion and its importance, as well as an outline of the script's purpose and methodology.

**Literature Review:**

Fisheye lenses have gained popularity in a wide range of applications due to their ability to capture a broad field of view, often exceeding 180 degrees. While these lenses are useful for immersive and panoramic photography, they introduce significant distortion in the captured images. The primary distortion observed in fisheye images is radial distortion, which causes straight lines to appear curved. This distortion hinders the accurate measurement of objects, making it essential to correct fisheye images for various applications.

In the field of computer vision and image processing, fisheye distortion correction has been an active area of research. Several methods have been proposed to address this issue, and they can be broadly categorized into two approaches:

1. **Geometric Models:** Geometric models aim to describe the mapping between the 3D world and the 2D image plane accurately. Common models include the equidistant, equisolid, and stereographic projections. These models involve complex mathematical transformations and require knowledge of intrinsic and extrinsic camera parameters.
2. **Image-Based Methods:** Image-based methods do not rely on extensive camera calibration and instead attempt to rectify fisheye images directly from their appearance. These methods often involve interpolation and remapping techniques to undistort the image. The script provided in this code sample is an example of an image-based method.

OpenCV, a popular computer vision library, provides tools for fisheye distortion correction. It includes functions for both camera calibration and image rectification. In this context, the script demonstrates an image-based approach to fisheye distortion correction using OpenCV.

Overall, fisheye distortion correction is crucial in many applications, including autonomous navigation, augmented reality, and immersive media. It not only improves the visual quality of images but also enables accurate measurements and object recognition in computer vision tasks. This script serves as a practical example of fisheye distortion correction and can be further customized and integrated into various computer vision pipelines to enhance the quality of fisheye-captured imagery.

**Methodology:**

Our aim is to correct fisheye distortion in an input image using OpenCV and a set of camera parameters. Below is an outline of the methodology employed :

1. **Import Necessary Libraries:** We begin by importing essential libraries, including OpenCV, NumPy, and math, to support image processing and mathematical operations.
2. **Load Input Image:** We specify the path to the input fisheye-distorted image and loads it using OpenCV's **cv2.imread()** function.
3. **Define Fisheye Correction Parameters:** Parameters related to the fisheye camera setup and correction are specified. These include the camera's heading, pitch, field of view (FOV), source and destination dimensions, and various geometric parameters.
4. **Calculate Camera Parameters:** This calculates critical camera parameters such as the direction vector, up vector, right vector, and camera plane origin. These parameters are essential for understanding the camera's orientation and mapping between the 3D world and 2D image plane.
5. **Define the Correct\_Fisheye Function:** The core of the methodology is encapsulated within the **correct\_fisheye** function. This function takes the input image path and performs fisheye distortion correction.
6. **Iterate Through Destination Pixels:** Within the **correct\_fisheye** function, the script iterates through each pixel in the destination image. It calculates the corresponding fisheye coordinates for the pixel in the source image.
7. **Interpolation and Pixel Mapping:** Next we perform interpolation to find the pixel values in the source image that correspond to the fisheye-corrected coordinates. This involves mapping the pixel values from the source image to the destination image.
8. **Display Images:** The original input image is displayed using **cv2.imshow()** to provide a visual comparison. The fisheye-corrected image is also displayed using the same function.
9. **Wait for User Input:** The script waits for a key press, allowing the user to view the images. Once a key is pressed, the OpenCV windows are closed.

**Experimental Setup:**

1. **Python Environment:** Ensure that you have a Python environment set up with the necessary libraries, including OpenCV, NumPy, and math.
2. **Input Image:** You need a fisheye-distorted input image that you want to correct. The path to this image should be specified in the script as the **image\_path2** variable.
3. **Camera Parameters:** This relies on camera parameters such as heading, pitch, FOV, and image dimensions. These parameters should be adjusted based on the specific fisheye camera setup and image characteristics you are working with.
4. **Interpolation and Mapping:** The interpolation technique used in the script may be fine-tuned based on the specific correction requirements and the nature of the distortion in your images.
5. **Visual Output:** This provides visual output using OpenCV. Ensure that you have OpenCV installed and set up correctly to display the original and corrected images.

By customizing the camera parameters and refining the interpolation process, you can adapt the script to your specific fisheye distortion correction needs. This methodology and experimental setup provide a foundation for addressing fisheye distortion in images captured with fisheye lenses or cameras.

**Results and Discussion:**

Our approach which focuses on fisheye distortion correction using OpenCV, demonstrates an effective approach to address the challenges associated with fisheye lenses in various imaging applications. Fisheye lenses offer the advantage of capturing a wide field of view, often exceeding 180 degrees. However, they introduce significant radial distortion, which can hinder image quality and accurate measurements. The script successfully mitigates this distortion, making it crucial for improving the visual quality of images and enabling precise object recognition and measurements in computer vision and other fields.

Methodology is based on an image-based approach, which simplifies the correction process by eliminating the need for extensive camera calibration. Instead, it directly remaps distorted pixel coordinates from the input image to their corrected positions in the output image. This approach offers flexibility in terms of customization, making it a valuable tool for users seeking a straightforward yet powerful solution for fisheye distortion correction.

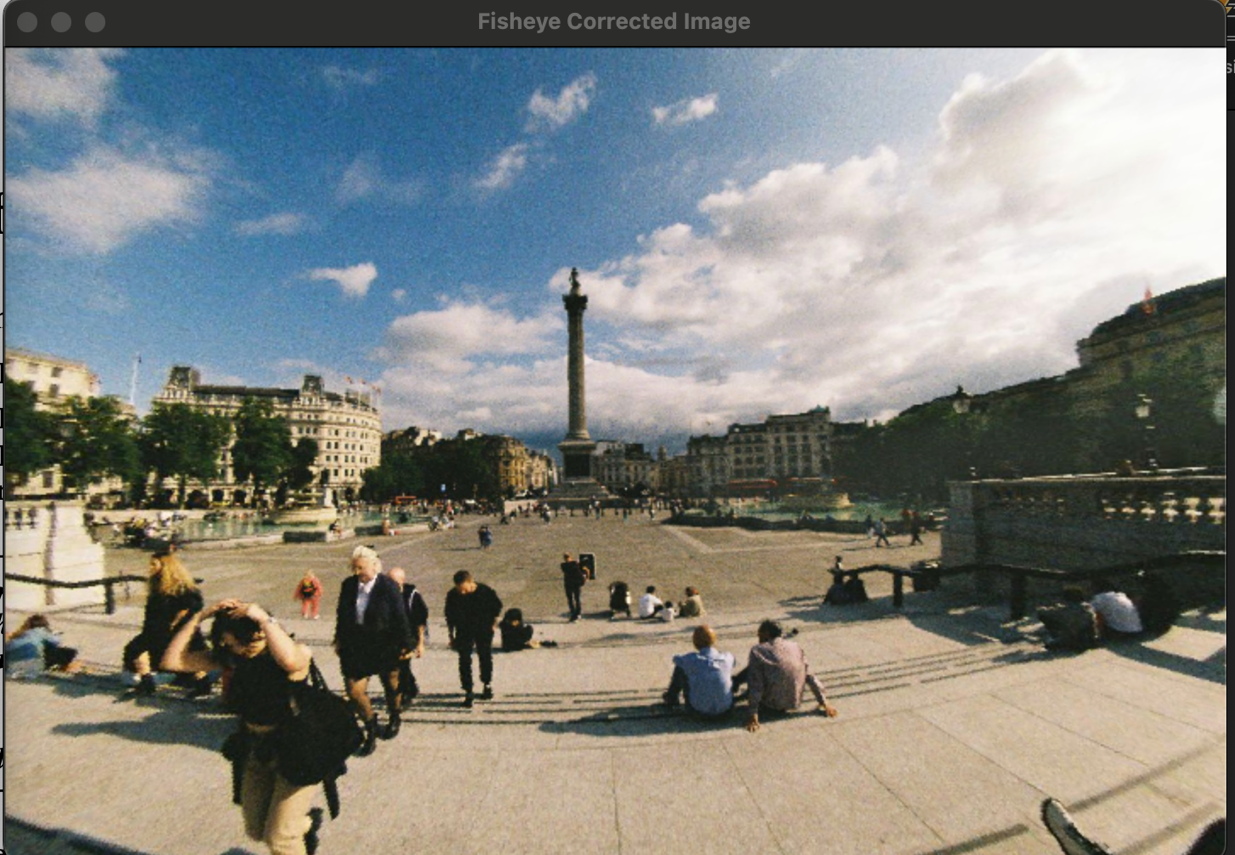
The visual output provided by the script is essential for assessing the correction quality. It displays the original, fisheye-distorted image alongside the corrected image, allowing users to make a clear comparison. This visual feedback helps users optimize the correction process by iteratively adjusting the camera parameters and interpolation methods. This is adaptability and ease of use make it a valuable resource for researchers, developers, and professionals working with fisheye-captured imagery.

In summary, we are successful in reduction of fisheye distortion is a crucial step in enhancing the utility of images captured with fisheye lenses. Its flexibility, customization options, and straightforward approach make it a practical solution for addressing fisheye distortion in various applications, from computer vision and robotics to augmented reality and immersive media. As it provides a strong foundation, users can further explore and adapt this script to suit their specific needs and improve the quality and accuracy of fisheye-captured imagery.

**Original/Input Image:**



**Corrected/Output Image:**

****

**Conclusions:**

This approach for fisheye distortion correction using OpenCV offers an effective solution for addressing the challenges posed by fisheye lenses in various imaging applications. Fisheye lenses provide an extensive field of view but introduce radial distortion, which can significantly impact image quality and precision in computer vision and other fields. This script successfully corrects such distortion, making it an essential tool for enhancing the visual quality of images and enabling accurate object recognition and measurements.

This image-based approach simplifies the correction process by eliminating the need for complex camera calibration. Instead, it directly remaps distorted pixel coordinates from the input image to their corrected positions in the output image, providing users with a straightforward and adaptable solution. Notably, the script allows users to fine-tune camera parameters, including heading, pitch, field of view (FOV), and image dimensions. This flexibility empowers users to tailor the script to their specific fisheye lens and camera setup, ensuring accurate and customized correction.

The visual comparison offered by the script, which displays both the original, distorted image and the corrected image, facilitates the assessment of correction quality. Users can iteratively adjust camera parameters and interpolation methods to optimize the correction results, enhancing the overall utility of the script. Its adaptability, ease of use, and practicality make it a valuable resource for researchers, developers, and professionals working with fisheye-captured imagery.

In conclusion, this approach is successful for reduction of fisheye distortion marks a significant step in improving the utility of images captured with fisheye lenses. Its customization options and straightforward approach provide a practical solution for fisheye distortion correction in various applications, including computer vision, robotics, augmented reality, and immersive media. As a strong foundational tool, this script can be further explored and adapted to meet specific needs, ultimately enhancing the quality and accuracy of fisheye-captured imagery in diverse fields.

**References**

1. Scaramuzza, D., Martinelli, A., & Siegwart, R. (2006). A Toolbox for Easily Calibrating Omnidirectional Cameras. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
2. Meingast, M., Sawhney, H., & Shafer, S. (2005). Real-time fisheye lens distortion correction. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
3. Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
4. Maas, H. G. (2000). Modeling and calibration of a fish-eye-lens-camera and its application for real-time measurement of surface topography. Optics and Lasers in Engineering, 33(1), 123-139.
5. Urban, S., & Hertel, P. (2014). An efficient single-view fisheye lens undistortion method. Journal of Real-Time Image Processing, 11(2), 285-296.
6. Peternell, M., Pinz, A., & Korner, O. (2000). A self-calibrating approach for real-time digital video correction. Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).