1. Gebe dich nie auf

Pranaya Chowdary

***Abstract*— This documentation presents an investigation into the restoration of a distorted portrait by the application of convolution filters and template matching techniques. The project aims to recover the original content from a distorted image and accurately locate a template within a collage. Further there is a path planning element to the project as well that is achieved through the RRT-connect algorithm. By employing optimal image processing and path-finding techniques the project successfully achieves these objectives.**

The project demonstrates the effectiveness of image processing and template matching algorithms in solving complex visual problems. The outcomes of this project have broader implications in various fields, including computer vision, image restoration, and pattern recognition. This documentation provides a comprehensive overview of the problem statement, approach, and results, offering insights and potential directions for future research in computer vision and image processing.

I. INTRODUCTION

The problem statement requires an investigation into a distorted artwork and a collage containing a template. The objective is to recover the portrait from the distorted image and perform template matching to find the coordinates of the matched template in the collage. By applying a specific filter found from the distorted artwork to the portrait, and utilizing template matching techniques, the distorted artwork is successfully restored and accurate template matching is achieved. Finally, the coordinates are manipulated to obtain a password for a zip file, which contains the image of a maze. Finally using RRT-connect algorithm, the path from start to end points of the maze is found.

II. PROBLEM STATEMENT

The problem statement consists of four parts.

Part 1: Finding the Filter

The first part involves a distorted image of a π artwork. The digits of π in the image have been distorted and need to be restored. Each distorted digit is multiplied by π and converted to the greatest integer less than or equal to the result. These resulting numbers in decreasing row major order form a 2x2 filter that needs to be applied to the image.

Part 2: Recovering the Portrait

The filter obtained is to be applied on the corrupted artwork to obtain the original portrait, by applying any of the 3 operations: bitwise OR, bitwise AND, bitwise XOR. Element by element operations are performed and then the filter is moved by 2 pixels to the right and so on. The resulting image is the recovered portrait.

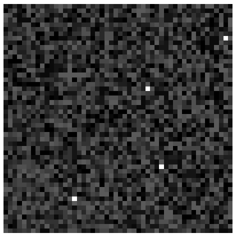

Fig. 1. distorted pi artwork

Part 3: Template Matching in Collage

The third part includes a collage image that contains the template of the famous portrait found in the previous step. The template needs to be scaled to 100x100 dimensions. Template matching is performed from scratch, without using built-in functions of OpenCV. The goal is to find the coordinates of the top left corner of the matched template in the image.

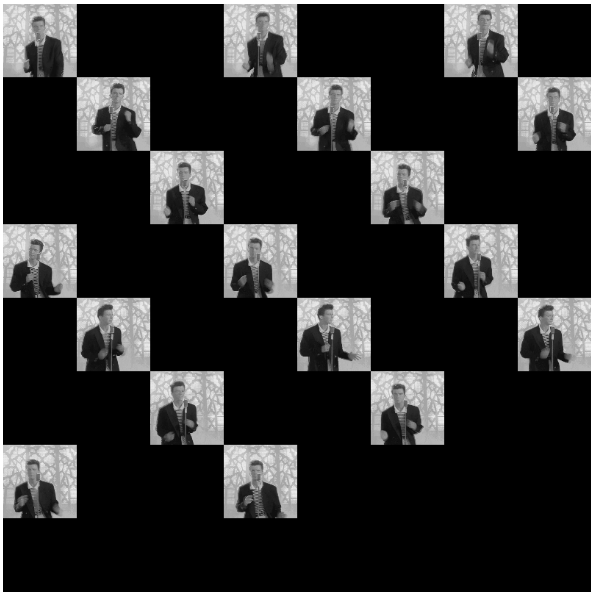

Fig. 2. Collage

Part 4: Path Finding using RRT Connect

The fourth part includes an image of a maze that contains start and end points along with obstacles, which needs to be solved using an RRT-Connect Algorithm.

III. RELATED WORK

There is extensive research into finding the most optimal ways to solve the problems that this project tries to solve as well. For the purpose of image restoration, there are a few research papers that use convolution filters and template matching in order to remove noise, blur or other artifacts that corrupt images. There is also research into utilising deep learning for image restoration.

There are a huge number of algorithms developed for path finding, but RRT based ones are gaining increasing popularity.

IV. INITIAL ATTEMPTS

Initially for the template matching algorithm, I had tried to use simple python lists instead of numpy arrays which resulted in the algorithm taking far too long to come up with a result. Using numpy array took away the need for a lot of loops and optimised the process.

V. FINAL APPROACH

Link to Google Colab Notebook

The step by step approach is as follows:

- All required dependencies are installed and imported and google drive is mounted.

I. Finding The Filter

- The given image has pixel values corresponding to the digits of pi multiplied by 10. The image is first extracted in grayscale and then flattened into a 1D numpy array.

```
pi_img = cv2.imread(pi_file ,0)
pi_arr = pi_img.flatten()
pi_arr = pi_arr/10
```

- 2500 digits of pi were extracted using the mpmath module.

```
import mpmath
mpmath.mp.dps = 2500
pi = 4* mpmath.atan(1)
```

- The image array found is now compared with the digits of pi to find the incorrect digits, and ergo the filter.

```
filter = []
for i in range(2500):
    digit = int(pi)%10
    if (digit != pi_arr[i]):
        filter.append(digit)
    pi *= 10
filter = np.array(filter)
```

- The digits of pi that were distorted can be each multiplied by 10*pi and converted respectively to the greatest integer less than or equal to them. The resulting numbers can be arranged in descending row major order and used as a 2x2 filter.

```
filter = filter*10*math.pi
for i in range(4):
    filter[i] = int(math.floor(
        filter[i]))
filter.sort()
filter = filter[::-1]
filter = filter.reshape(2,2)
```

And that gives us the filter:

```
[[282, 251]
 [ 94, 0]]
```

II. Recovering the Portrait

- The filter is applied onto the portrait and the three operations (bitwise OR, XOR and AND) are performed one by one to get three potential templates. The relevant one is saved for the next step of template matching.

Here is the final template.

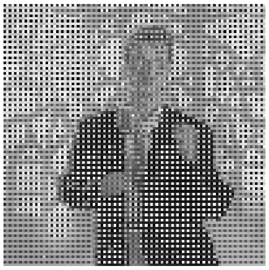

Fig. 3. Recovered Portrait

III. Template Matching

- Using the portrait, template matching is performed with collage using a function called Score Calculator and stored in results.

```
def ScoreCalculator(start_x , start_y):
  score = 0
  score += np.sum(abs(collage[start_x:
      start_x+100,start_y:start_y+100] -
      portrait))
  return score
results = []
for i in range(701):
  for j in range(701):
    results.append(ScoreCalculator(i,j))
```

- Using this, the coordinates of the top left vertex of the matched index is found, which comes out to be **(100,100)**

IV. Path Finding

- Using the coordinates, password to the zip file is extracted. Through which the image of a maze is found, which is to be solved using RRT-connect algorithm.
- The algorithm given in the following paper is followed to the maximum extent: RRT-Connect: An Efficient Approach to Single-Query Path Planning
- The algorithm consists of two trees starting from the start and end points moving towards randomly generated points throughout the maze while also extending towards each other.
- The exact code can be found in the google colab notebook linked above.

VI. RESULTS AND OBSERVATIONS

Comparing the recovered portrait in Fig.3 to the portrait found from the collage:

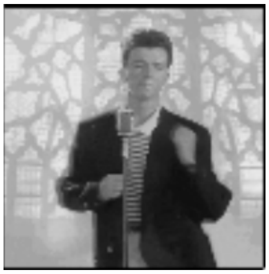

Fig. 4. Original Portrait

The similarity is undeniable, proving Picasso's guilt.

The final paths found after implementing RRT-Connect are here:

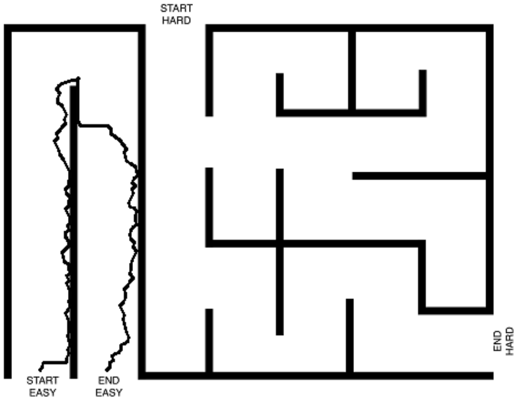

Fig. 5. Maze: Easy

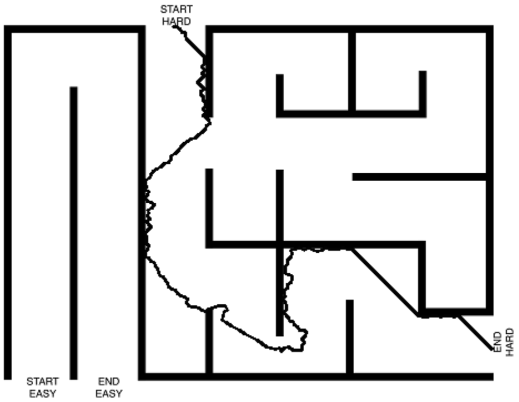

Fig. 6. Maze: Hard

I have manually implemented the filter operations over the portrait instead of utilising library functions, like filter2D that OpenCV provides. They might be able to provide a more optimised and faster solution than my current solution. The reason I have chosen to manually implement the filter function is to be able to better understand the inner workings of such functions.

RRT-Connect is a powerful algorithm for single-query path planning, known for its efficiency and ability to handle high-dimensional spaces. It offers completeness and adaptability, making it well-suited for dynamic environments. However, it may not always find optimal paths and there are other algorithms that perform better when global planning is required.

VII. FUTURE WORK

- Explore the integration of machine learning techniques, such as deep learning, to improve the accuracy and robustness of the image processing tasks. This can involve training models to handle image restoration, template matching, or path planning.
- Consider exploring alternative algorithms for template matching and path finding. There are various advanced algorithms available for these tasks, such as deep learning-based approaches.

CONCLUSION

This documentation presented a comprehensive overview of a multi-step project involving image processing, template matching, and path finding. It showcased the potential of these techniques in various domains and highlighted opportunities for future research and development. By continuing to refine the algorithms, optimize performance, and explore innovative approaches, further advancements can be made in these areas of study.

REFERENCES

[1] Kuffner, James J., and Steven M. LaValle. ”RRT-connect: An efficient approach to single-query path planning.” Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). Vol. 2. IEEE, 2000.

[2] https://docs.opencv.org/

[3] Baking 1000 Digits of Pi from 3 Small Lines of Python

[4] De Vylder, Jonas, et al. ”Image restoration using deep learning.” 25th Belgian-Dutch Conference on Machine Learning (Benelearn). 2016.