EE 645 - Course Project Phase - III
Content Aware Rotation

S Deepak Narayanan, 16110142

Brief Overview and the method of solving the optimization function

First step is to extract and quantize all the line segments that we've found. I used the Line Segment Detector that the authors use and have put it as a part of my code. In my previous report, I had mentioned about how there was the need to solve a quadratic optimization function in the variables V and θ. This time around I was able to solve the quadratic equation by forming a quadratic in V, but rather representing the L_2 Norm of a vector x by $x^T x$. I basically take the energy function given below, differentiate it with respect to V for solving the first part of the optimization and then was able to obtain a linear system of equations of the form $(\alpha.MS + \beta.ML + \gamma.MB).V = b$, where b is the vector I obtained after differentiating the energy function for the boundary preservation in the first half of the optimization, and ML, MS, MB are the matrices formed during the optimizations from the Line, Shape and Boundary Preservation Energies respectively. I solved for V using the above equation and iteratively solved for θ. Solving for θ also was difficult, but even that eventually ended up in forming a matrix to solve and a gradient descent problem. Rather than the Gradient Descent approach, I am using what the authors proposed more like a numerical method than an exact solution, as this is based on checking for all values in a range, given you know the maximum and the minimum and solved the second part of the optimization. Solving the first part of Fix θ, solve V is briefly described here.

We know $V_q = QV$ for some Q that I had found earlier for shape preservation energy. Similarly for e_k, I found a P_k, such that $e_k = P_k V$. An example of how I solved the optimization problem is as follows: Take the shape preservation energy. That norm is the norm of a *vector* and not a matrix. This is the crux of the optimization. Write $V_q = QV$.

$$E_S(V) = \frac{1}{N}\sum_q ||(A_q(A_q^T A_q)^{-1}A_q^T - I)QV||^2$$

$$\implies \frac{1}{N}\sum_q (V^T((A_q(A_q^T A_q)^{-1}A_q^T - I)Q)^T(A_q(A_q^T A_q)^{-1}A_q^T - I)V)$$

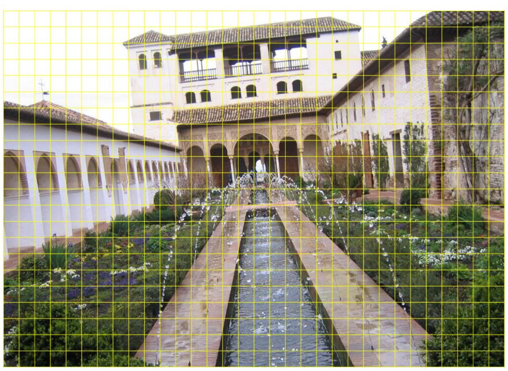
Now, call the inner two matrix product as S. Then we effectively have, $V^T SV$, the derivative of which, ,with respect to V is $V^T(S + S^T)$. But then, S is basically symmetric! This ensured that I got, finally, $2.V^T S$ as the derivative! Similar is the case with other matrices for boundary as well as line preservation energies. Therefore, we are left with the linear combination of the form mentioned we get it of the form $Ax = b$, which is solved for. Similar is the solution formation for the half quadratic splitting method that the authors have used for the other half of the optimization.

Results and Evaluation

I fix the size of each quad to be of size 30 x 30 approximately, because of the amount of time the function runs for. It becomes extremely long if I decrease the quad size and increase the number of quads. After performing the optimization, we have the warping. I did warping by using barycentric coordinates. Basically, I broke down each quad into two triangles, found the coordinate of each point inside the triangles using a linear combination of the triangle vertices, where each of the coefficient is at most 1. The most useful thing about these barycentric coordinates is that even in the warped image, these coefficients for the quads are the same for the corresponding points. This means, we initially compute these coefficients (denoted lamb in my implementation) and then, using these find the new points and then interpolate in those points of missing data (pixel values).

Image 1 - Angle to be rotated by - 5.8 degrees, My result and input image are here. For other images, please refer to the Readme.txt that I have attached with the submission.


The warped mesh is also displayed here.

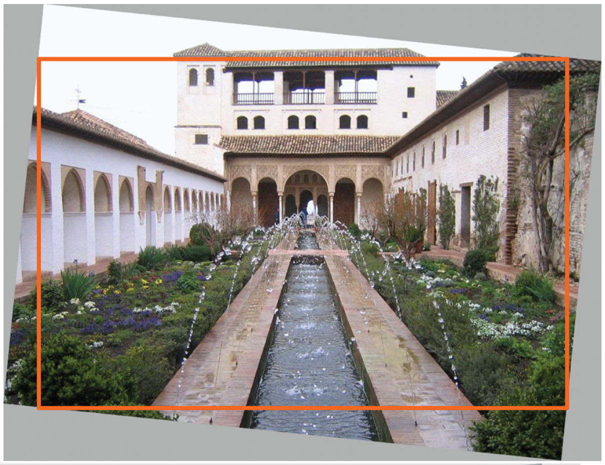
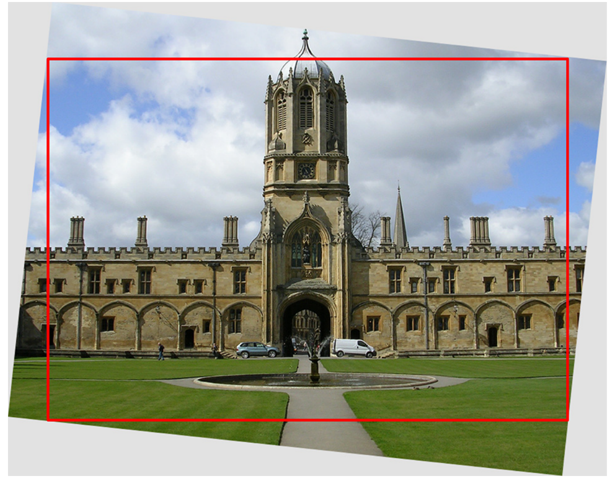



Also, when I tried rotating this following image by 16 degrees, there was very high level of distortion. The actual angle of rotation was only 6.1 degrees.

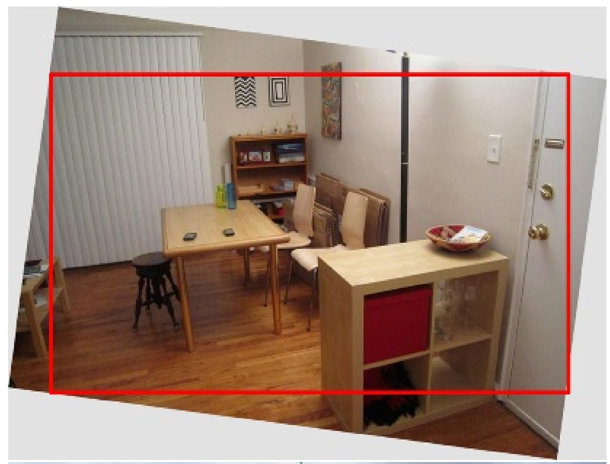

Content Aware Rotation has a very useful application here where it can generate visually appealing images like the one here. The rotation angle that I used was 8 degrees clockwise. An example is

My algorithm did not do well on a few test images that resulted in black regions. The rotation part seems alright, but the black regions I suspect are caused due to some numerical error, because I set all the improper values to 0. An example is this where the rotation angle is -6 degrees clockwise:


A few examples of what rotate and crop would look like versus, what I have obtained as result after performing content aware rotation is below

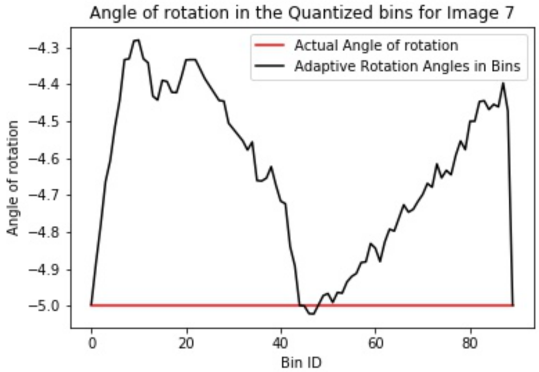
.

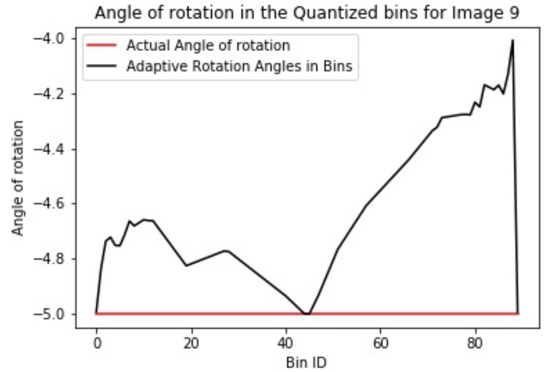



One more thing to note about the optimization function is how each bin is rotated. To create the perception of rotation, the authors constrain those bins that when rotated would consist of horizontal and vertical lines or line segments. They don't put any constraint on the other bins. This makes the algorithm adaptive. This ensures that we can distort the image a bit more on the areas where the other line segments have occurred. This adaptive rotation is achieved by setting δ_m in the $E_R(\theta)$ to 1000, for these bins and letting it to be zero for others. The only constraint that now directly influences the rotation of the secondary bins is the second term in $E_R(\theta)$, where $E_R(\theta) = \sum_m \delta_m(\theta_m - \Delta)^2 + \sum_m(\theta_m - \theta_{m+1})^2$. The second term being quadratic kind of slowly blows up the error rate and hence there is some level of continuity across the m^{th} and the $(m+1)^{th}$ bin. A few plots that were plotted for this adaptive rotation are shown here. The corresponding Image 7 and Image 9, are,





The plots seem to have reasonable amount of continuity and in the end are forced to be almost the same as the rotation angle due to the high cost incurred as bins 0,44,45,89 are the canonical bins and they are the ones that would, after rotation contain all the horizontal and vertical lines ((0,89) and (44,45) respectively).

Conclusion

- Content Aware Rotation can do particularly well in regions where there is a lot of white space or background. This is because of the rotation constraint occurring only on the canonical bins as the distortion caused by warping requires that there's not a lot of information everywhere on the image. Otherwise, for sure there'll be areas in the image where there'll be distorted content. Also to be noted is the fact that if a particular quad doesn't have a line, then it can, in essence undergo a lot of distortion
- It works well for only small angles of rotations, roughly upto 10 degrees in the datasets that I could test it on.
- In an ideal situation, we'd want most of the content in the middle, a lot of background and a little angle of rotation. This would ensure that Content Aware Rotation does a good job on the input image.

Acknowledgements

I would very sincerely like to thank Prof. Shanmuganathan Raman, his TAs Dipti and Gagan and Rajendra for their continuous help throughout the project.

References

1. K. He, H. Chang and J. Sun, "Content-Aware Rotation", *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, 2013, pp. 553-560. doi: 10.1109/ICCV.2013.74
2. S. Avidan and A. Shamir. "Seam carving for content-aware image resizing". In *SIGGRAPH, 2007.*

3. A. C. Gallagher. Using vanishing points to correct camera rotation in images. In *Computer and Robot Vision*, pages 460-467, 2005.

4. R. Von Gioi, J. Jakubowicz, J. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. TPAMI, pages 722732, 2010.

5. C.-H. Chang and Y.-Y. Chuang. A line-structure-preserving approach to image resizing. *In CVPR, 2012*.

6. G. Zhang, M. Cheng, S. Hu, and R. Martin. A shapepreserving approach to image resizing. *In Computer Graphics Forum, pages 18971906*. Wiley Online Library, 2009.

7. iRotate, https://github.com/yuchien302/iRotate

The initial proposal from phase 1 is attached here

Significance

Usually, pictures that are not carefully taken have some kind of tilt associated with them. This can lead to them looking tilted and not straight. The methods that are usually adapted to resolve this issue of a tilted image is rotation and cropping. A very important shortcoming of a method like cropping would include loss in data, which in turn might adversely affect the integrity of the image. Yet another issue is that even for very small angles of rotation, roughly around 5^o, there is a reduction in the area of a typical photo by roughly 20% [1]. Rotation being such an important component of image editing operations along with cropping and scaling has not had a lot of work trying to solve this fundamental problem described above.

Content Aware Rotation tries to solve this fundamentally important problem without cropping or scaling and uses a warping method to create the perception of rotation while keeping the image content inside a upright rectangle. This is motivated by human perception, which could very well be significant in other computer vision problems as the ultimate aim is to reach a level where we can match human level vision. Yet another significance of content aware rotation is that it is fast and can be used by common people.

Objectives

The main objective of this project is to implement the paper on Content Aware Rotation and compare the results that are obtained with regular cropping and rigid rotation. For doing the same, the authors of the paper come up with an optimization based method that "preserves the rotation of horizontal/vertical lines, maintains the completeness of the image content, and reduces the warping distortion" [1]. A Quad Mesh is optimized for warping, under various constraint that a few lines preserve orientation and few don't. Warping methods are used because they can preserve local shapes and straight lines [5]. This objective shall be achieved by performing various steps as described in methodology.

Methodology

- The algorithm to be described assumes that the angle of rotation of the input image is already given and that it is fixed or can be computed using an algorithm in [3].
- Firstly, there is line extraction and quantization that is done using [4]. Then the orientations of these lines are computed and the ones that are to preserve orientation are classified as canonical. These are identified by allocating to each line a bin, that is a measure of the orientation of the line. These canonical lines are in the canonical bins.
- An energy function is defined now, inspired from [5]. Various different constraints are imposed in this stage. These constraints cover the canonical lines, maintaining content completion by constraining the boundary vertexes and to minimize the local distortions so that shape is preserved. These are various factors that model the energy function. Each constrain to be modeled is an energy function in itself and the overall energy function is defined to be a linear combination of all of these. The aim is to minimize the *overall* energy function, which is a

function of the vertexes of the mesh grid and a quantity obtained by using the rotation angle of the input image.

- Optimization of the Energy Function: The authors propose an alternating algorithm. They optimize the energy by dividing the problem into two subproblems and then iteratively optimize each of the subproblems.
- Finally, Bilinear interpolation is done to deform the image to get the final image.

Expected Outcomes

- The rotated image after applying the algorithm is the expected output. This method performs well for small angles of rotation, which is usually the case for regular images.
- There will be comparisons that show where all Content Aware Rotation does well and where all it doesn't and a suitable reasoning for the same.
- The limitations of the method. For an instance it cannot do very well in a setup where there are too few operations that can be performed without getting noticed. Whenever rotation angle is large, it might have the above issue. These are in general the expected outcomes of Content Aware Rotation.

References

1. K. He, H. Chang and J. Sun, "Content-Aware Rotation", *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, 2013, pp. 553-560. doi: 10.1109/ICCV.2013.74
2. S. Avidan and A. Shamir. "Seam carving for content-aware image resizing". In *SIGGRAPH, 2007.*
3. A. C. Gallagher. Using vanishing points to correct camera rotation in images. In *Computer and Robot Vision*, pages 460-467, 2005.
4. R. Von Gioi, J. Jakubowicz, J. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. TPAMI, pages 722732, 2010.
5. C.-H. Chang and Y.-Y. Chuang. A line-structure-preserving approach to image resizing. *In CVPR, 2012.*
6. G. Zhang, M. Cheng, S. Hu, and R. Martin. A shapepreserving approach to image resizing. *In Computer Graphics Forum, pages 18971906.* Wiley Online Library, 2009.