

Supplementary Material

From Bows to Arrows: Rolling Shutter Rectification of Urban Scenes

In this supplementary document, we show how RS distortions could affect geometric inference through vanishing point analysis. We then provide an analysis of the performance of our method in the presence of curve breaks and in the presence of natural curves as well. We also provide additional insights for a rectification example that we showed in the main paper. To show the effectiveness of our method, we also show additional RS rectification examples.

1. Geometric Analysis

To show how the RS distortions could affect geometric inference, we show an analysis based on the vanishing point. Parallel lines in a scene should converge in the image to a common vanishing point [12]. The RS effect disturbs the straightness of lines, and hence affects vanishing point estimation. We use this estimation error to quantify rectification. To avoid making all the lines parallel, we removed the zeroth degree parameter (i.e. the DC value) in (4) during r_x and r_y estimation. To estimate the vanishing point, we first fit lines to vertically oriented curves (of RS and rectified images, separately) and determine the point which minimizes the orthogonal distance to all these lines. As there is no ground truth image, we use this minimum distance as our vanishing point estimation error. The lower this distance, the better is the rectification.

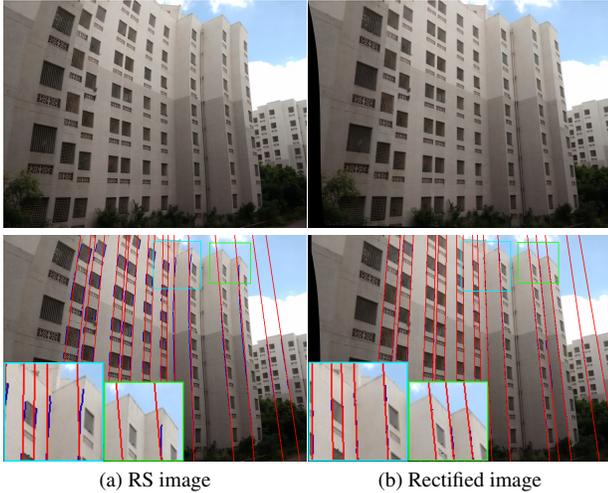


Figure 9. Vanishing point estimation.

Fig. 9(a) shows an RS image of a building with curved vertical edges. The fitted lines to these curves are marked in red. Correspondingly, Fig. 9(b) shows our rectification output with rectified curves and fitted lines. The fitted lines after rectification align more closely with the vertical edges

as compared to those before rectification (compare zoomed-in patches). This fitting error due to curvature leads to a resultant error distance of 12.1 pixels for the RS image, but a lower value of 5.1 pixels for the rectified image during vanishing point estimation. This shows that the RS effect must be handled correctly to study scene geometry, and an RS rectification method such as ours will be pivotal.

2. Presence of Curve Breaks

Since our motion estimation method uses curves as features, we conduct an experiment in which we break the curves to create discontinuities and test the performance.

For this experiment, we consider an image of size 816x612 in which there are ten vertical lines spaced equally apart (Fig. 10(a)). We then introduced an r_y motion to transform these lines into curves (Fig. 10(b)). In those curves, we introduced breaks by removing pixels from a block of rows of length b in the middle (Fig. 10(c)). After this process, the number of RS curves becomes 20; the top and bottom broken parts of each of the original curves are treated as two curves. Using these discontinuous curves, we estimate the camera motion r_y (fixing r_y of the top row as 0) and then rectify the broken curves. The rectified image is shown in Fig. 10(d). It is clear that these additional discontinuities in the curves do not significantly affect our RS rectification method.

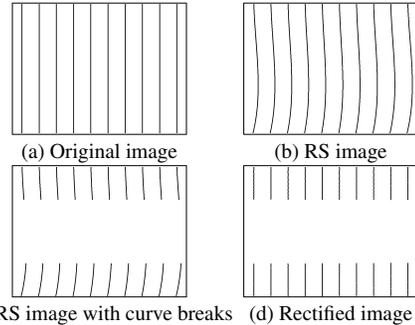


Figure 10. RS rectification in the presence of curve breaks.

Further, we vary the break-length b from 0 to 550 (total curve length is 612), and study the performance both in terms of the motion estimate and the rectified output. In addition, this experiment will reveal how well our method identifies the association between the top and bottom broken parts of curves. Fig. 11 shows our rectification for break lengths $b = 200$ and 400 (corresponding to 67.3% and 34.6% of total information, respectively, for 612 rows). The RS curves are shown in red and the rectified curves are shown in blue. The rectification of curves as lines is correct, and there is no association-disconnect between the broken parts too; both the top and bottom curve segments are corrected back to the same original line. The estimated motion trajectories for different break-lengths (for $b = 0$ to 550)

are shown as dotted paths in Fig. 12. The ground truth trajectory is shown as continuous green path in Fig. 12. The estimated trajectory closely follows the ground-truth trajectory till $b = 400$. This indicates that our method performs very well, identifying correct curve segment associations, even when only 34.6% of the curve information is present.

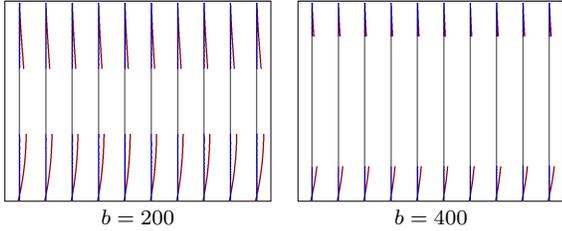


Figure 11. *RS rectification in the presence of curve breaks* (best viewed in PDF). The gray color denotes original lines, the red color denotes RS curves, the blue color denotes rectified curves. Here, b is the break-length of curves.

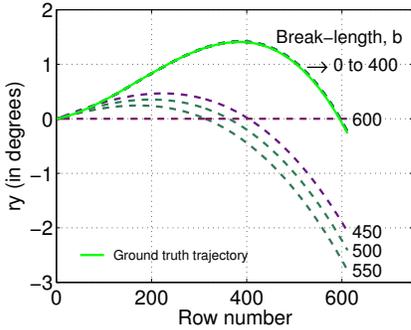


Figure 12. Motion estimation in the presence of curve breaks.

A further increase in the break length (which may not be observable in real scenarios though) causes our method to disassociate the top and bottom curve segments. Fig. 13 shows our rectification for break length $b = 500$. The available amount of information is only 18.3%. The rectification is proper in which all the curve segments are transformed as lines. With no availability of information in a large number of rows in the middle, our estimated motion deviates from the ground-truth for $b > 400$ as can be seen in Fig. 12. Since we fix the r_y of the top row to be 0, the estimated path follows the ground-truth for the first few rows rectifying the curve segments at the top as same lines as in the original image. It is important to note that the curvature of the estimated trajectory for the last few rows is very similar to that of the ground-truth trajectory. This is, in fact, the reason that the curve segments at the bottom (red curves in Fig. 13) do get rectified as lines (blue lines in Fig. 13). Due to the unavailability of information in most of the middle rows, our method fits a different camera path from that of the ground-truth, and hence the rectified lines have shifted a little from the original lines (gray lines in Fig. 13).

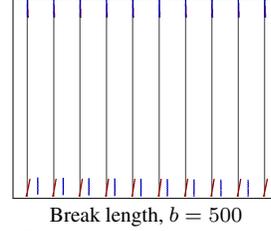


Figure 13. *RS rectification in the presence of curve breaks* (best viewed in PDF).

Though long curves are favored as inputs for our motion estimation method, as we have discussed in this section, our method performs very well even in the presence of curve breaks correcting them as lines and associates the broken curve parts correctly even when only 34.6% of the information is available.

3. Presence of Natural Curves

A scene naturally containing many curves might pose issues during optimization, since our cost formulation primarily assumes that the RS curves correspond to lines in the original scene. In the main paper, we showed an example of a scene containing a tree. Our method performed very well, since there were a number of other straight lines that influenced the cost. In this section, we analyze the effect of such natural curves in our method under synthetic conditions.

We consider a total of ten lines and curves in the scene. The natural curves are generated without conforming to any property mutually. The RS effect causes all the ten structures to become curves. We then rectify the RS image and test its performance based on the error in the rectified curves that correspond to straight lines in the original scene. We study this behavior by increasing the number of natural curves $n = 0$ to 9. The first case $n = 0$ is the no-natural-curve case, and the last case $n = 9$ is the all-but-one-are-curves case. The RS and rectified images (for $n = 0, 2, 4, 6$) are shown in Fig. 14. As n increases, the performance of RS rectification decreases. In Fig. 14(a), there are no natural curves, and hence the rectification is perfect. The rectification performance of valid lines is visually good till $n = 4$ (Fig. 14(c)). In (d), there is a clear visual deviation of the rectified line from the original line.

Fig. 15 shows a quantitative analysis. In the left, we show the variation of average absolute angular error during motion estimation with respect to n , and in the right, we show the variation of mean squared error deviation of the rectified curves from original lines. Both the angular error and the rectification error exhibit an increasing behavior with increasing n . Note that the rectification error is under three pixels till $n = 4$. This shows that our method is robust to the presence of natural curves to a good extent.

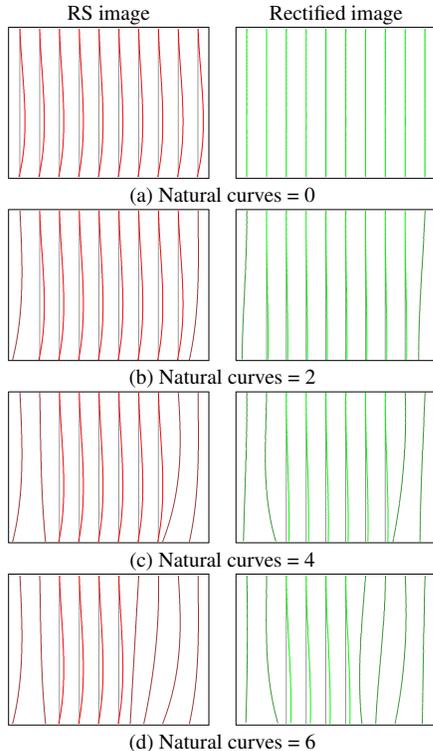


Figure 14. *RS rectification in the presence of natural curves* (best viewed in PDF). The red color denotes RS curves, and the green color denotes rectified curves. Dark and light color variations correspond to natural curve and straight line, respectively. The gray color indicates vertical lines in the original undistorted image.

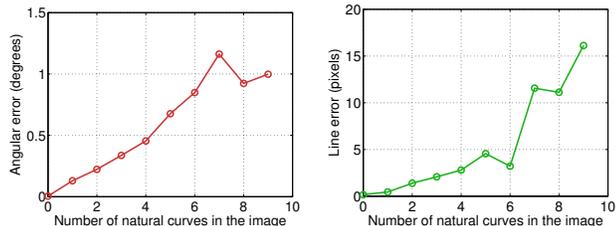


Figure 15. Motion estimation in the presence of natural curves.

4. Additional Insights

Due to space constraints, we were not able to discuss more about our motion estimation in the main paper. In Fig. 16, we show the RS and rectified images overlaid with curves and lines (repeated from Fig. 4 of the main paper). As already noted, all curves are transformed into lines. In addition, we would like to point out that the lines that are already straight in the RS image are not disturbed during rectification. The green lines closer to the image center in Fig. 16(a) are, in fact, not curved significantly due to camera motion; only the long green lines near the periphery are bent. Our cost formulation implicitly considers these aspects and estimates a camera motion that straightens only

the curves leaving the already-straight lines in the RS image intact. In this example, the camera motion consists predominantly of inplane rotation r_z , and the effect of which increases when going away from the center. Adhering to this behavior, our optimization results in the correct motion trajectory which transforms both curves and lines in the RS image as lines in the rectified image. Hence, it is important to detect and exploit lines also (in addition to curves) in the RS image, without which an incorrect motion might be estimated that could disturb lines that are already straight in the RS image. The rectified outputs for this RS image over iterations during optimization is included as a video.

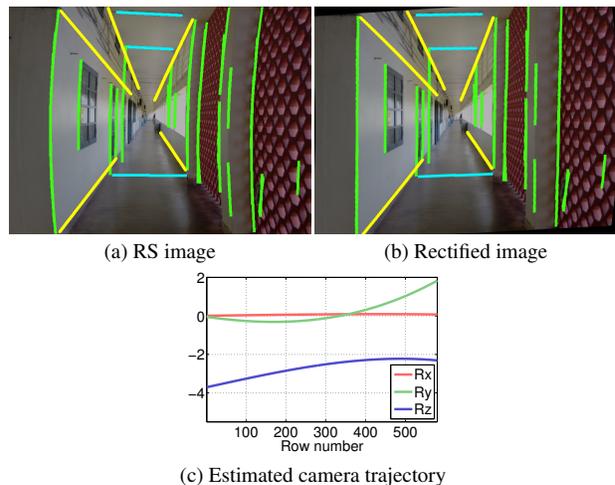


Figure 16. RS rectification and motion estimation.

5. Additional Examples

In this section, we show some more single image RS rectification outputs (in addition to the examples in the main paper). In the examples shown in Fig. 17, we can observe that the RS effect has been rectified by our method successfully. In the last example in Fig. 17, the RS effect of the pillars are rectified correctly, but a small residual RS effect is visible along the gray wall edge in the right-part of the image. This may be due to the presence of small translatory motion (which is depth-dependent) that we do not account for in this work.

RS images



Rectified images



Figure 17. More examples of our RS rectification.