Lecture 7 - Image / Video Mosaicking

Gidon Rosalki

2025-11-30

Once upon a time in 1900, people wanted a single large photo of a train. Since they could not mosaic photos together, they built a 500Kg camera, of size 2.5m by 1.5m, that needed 15 people to operate. These days, we can save the effort by mosaicking together many photos.

1 Global motion

1.1 Traditional mosaic construction

Traditionally, one computes the pairwise motion, so $M_{1\to 2}, M_{2\to 3}, \dots$, such as with Lucas Kanade with feature point matching. We select a reference frame, and scaling all images to that reference frame. This uses motion composition (matrix multiplication, so $M_{4\to 2} = M_{3\to 2}M_{4\to 3}$), and warping the frames to the reference frame, and then combine them into a single mosaic.

In order do this, we need global motion, where we can measure out motion with respect to the image. Here various changes (gradients) in several directions are needed in the search window. We have for example Cross Correlation Search, NCC, which is efficient for detection of translation (using pyramids), and has accuracy of around one pixel, but this is obviously rather limiting. There is also Lucas Kanade:

- Must have small motion
- Uses pyramids, and iteration within pyramid level
- Uses derivatives from the entire image
- Works better on a blurred image, so an image that is too sharp will not work. We can blur it, but this will limit us from the other direction.

How about feature point matching instead?:

- Use distinctive feature points that maximize sensing of motion (Harris corners, DOGs, etc.)
- A feature vector is computed for each selected point
- MOPS: a patch around the selected point, normalized for affine colour changes and for orientation.
- SIFT: Histogram of gradients around the selected point, normalized for orientation.

Finally we have RANSAC, where we suggest transformations by random selection, and validate with other points / the whole image. All of these have their advantages, and disadvantages. Often, Lucas Kanade does in fact return the best results.

In order to create these mosaics, what we originally did is to rotate a camera slightly between every image, with a new photograph with every new angle. Each image has about a 90% overlap with the previous image, so adds a new column to the overall image which may thus be built. This has the problems of the straight lines becoming conic sections:

Figure 1: Conic sections at Microsoft

These conic sections appear since we take these many images, and project them onto a cylinder. The projection of a straight line onto a cylinder will inevitably result in a conic section.

1.1.1 Pushbroom camera paradigm

Consider a satellite. Instead of looking at an entire plane, the satellite camera instead only records a single line, and merges it into a large image. This ensures that there is consistent viewing the 3D objects below it, as opposed to getting many vastly different images of the same object, from different angles.

1.1.2 Combining frames to mosaics

The first suggested method for how we do this is to align the images, and take the average / median pixel of all the overlapping strips. This has the issue of there being a need for perfect alignment *everywhere*, since otherwise there will be blur / noise added into the image. Instead what we do is cut and paste the centre strips, where we mark the centres of the frames, and cut at the midpoints between the centres. This way, we only need perfect alignment along the seams, and optimal seam may be computed using min cut or dynamic programming. We can then also use pyramid blending at the seams, to make the seams even more... seamless (I'll see myself out).

1.1.3 VideoBrush

Let us instead consider a video taken, where the camera translates sideways. We will stack the input images to create the space-time volume $f(x, y, t)$. We may then align the images by translation and rotation (using perspective homography), and cut & paste the centre strips. This makes a slice in space-time, where the strip width depends on the speed of the motion between frames. Fast motion will result in wider strips, where slower motion will result in narrower strips.

1.2 Translating camera

Let us instead consider translating the camera, but taking a full image every time. This results in seeing a whole image, from a different perspective, instead of merely a strip.

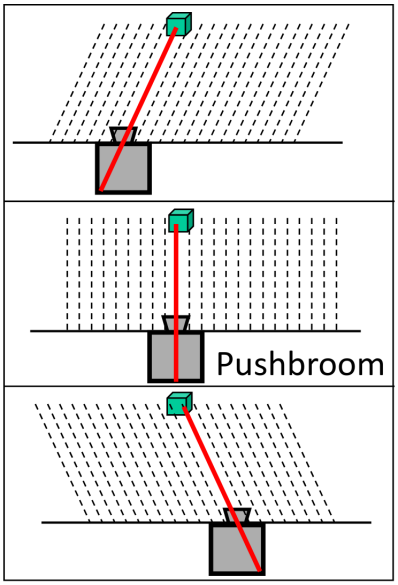

Figure 2:

As a result, if we make a panorama using pushbroom, using the strips from the leading edge of the frame will result in drastically different results as from the trailing edge. One can therefore use this to create a parallax image panorama (ex4).

1.2.1 Rotating camera

This can also be done from rotating cameras, however instead of rotating around the sensor, we rotate around some other axis. This naturally results in a different effect, but still mostly what we want.