COREGISTRATION: MULTI-CAMERA IMAGE ALIGNMENT FOR AERIAL IMAGING

A major challenge with multi-camera systems is that regardless of how closely aligned the cameras are, they cannot have the exact same point of view and the images they generate do not cover exactly the same area. Aside from using a complex setup of mirrors and prisms, it is impossible for more than one camera to capture the same scene from the same point of view at the same time because both cameras cannot occupy the same physical space. However, a relationship through translation and rotation along various axes exists between the cameras and this allows them to be coregistered.

Coregistration takes places through an image transformation known as homography. This is when two images possess a common plane or surface in their respective image, such as the facade of a building, the cover of a book on a table, or a billboard in Times Square. A mathematical transformation is used to realign the coordinates of the common plane in the source image to match the coordinates in the destination image.

Homography is also used in televised sporting events to overlay information in parts of the scene such as the country that an athlete is representing in their respective swimming or running lane at the Olympics. Here, the four corners of the image to be overlaid are mapped to the coordinates in the video feed selected by the television production company to line up with the athlete's lane.

The homography transformation is only valid for the selected plane in both images. This works well for the Olympics example because the source image is being added to the destination image. However, when applied to real world images captured from two separate cameras, results may vary. Returning to the example of the facade of a building, it is possible to coregister two images taken from two different points of view based on the building's facade, but objects in front of the building such as trees or parked cars won't line up as they are not in the same plane as the front of the building.

One area where coregistration through homography is particularly useful is in aerial imaging with applications like precision agriculture such as NDVI (multispectral imaging using multiple cameras and bandpass filters), as well as increasing the aircraft's coverage rate through the use of multiple cameras. Since an aircraft flies at a significant distance from the ground, the entire scene of each camera can be considered as part of the same plane. This means that a homography transformation can be applied to the entire image to coregister images from multiple cameras. If more than two cameras are involved, the same process is be performed between a chosen camera and every other camera in the system until all the cameras are coregistered to the chosen camera's coordinate system.

Transformation through image homography sounds complicated, but there are computer vision tools that exist to simplify the coregistration of images. OpenCV has functions that are built into the standard open source package to simplify image coregistration. They calculate what is known as a transformation or warp matrix, a 3x3 matrix that performs the homography by multiplying the source image points with the matrix to convert them to the destination coordinate system. In this case, the warp matrix has eight parameters with the bottom right parameter (3, 3) set to 1 by convention.

Since the homography transformation matrix has eight parameters (or in this case, variables) a minimum of eight points are required to compute the warp matrix parameters. Specifically, four points from the source coordinate system and four corresponding points in the destination coordinate system are required. Since each point contains an (x, y) pair, the algorithm has enough variables to build a system of eight equations to solve for the eight parameters of the warp matrix.

Returning to the example of the Olympic lanes, the source coordinates would be the four corners of the image of the athlete's name and country's flag and the destination would be four corners in the athlete's lane. These eight known points would be sent to the OpenCV function `findHomography()` as seen below:

```
//Calculate the warpMatrix using findHomography() by passing
//vectors sourcePoints and destinationPoints
Mat warpMatrix = findHomography(sourcePoints, destinationPoints);
```

This computes the homography transformation matrix and saves it in the variable `warpMatrix`. The same function can be used for ground control points (GCP) with aerial imaging to coregister images from multiple cameras. The coordinates of all the GCPs in the source image would be compiled and paired with their corresponding GCPs in the destination image. Once a homography transformation matrix is found, it can be multiplied with the source image to coregister it with the destination image. This is completed using the function called `warpPerspective()` and can be seen in the example below. Completing both of these operations will ensure that the coordinates of any pixel in the source image will correspond to the same location in the destination image.

The minimum number of points required to compute the transformation matrix is eight. However, the function "`findHomography`" will accept more than four point pairs that are found in the same plane. If more than four point pairs are specified, a method can be specified to reduce the re-projection error.

See the opencv.org website for more details.

```
//Initialize the outputImage variable
Mat outputImage;
//Use warpPerspective to coregister the sourceImage using the
//warpMatrix and save it to outputImage
warpPerspective(sourceImage, outputImage, warpMatrix,
outputImage.size());
```

To use `warpPerspective()`, an output image must first be initialized. The function is then provided with the source image, the previously initialized output image, the homography transformation matrix, and the size of the output image. The source image is the image from which the source points were selected.

Upon successful completion, the `outputImage` variable will contain the source image that is coregistered with the destination image. The images can now be overlaid and further processing can be performed on the coregistered image set as their coordinate systems are now identical.

Further automation of this process is also possible through use of the OpenCV function `findTransformECC()` by specifying `MOTION_HOMOGRAPHY` as the motion type. This function requires either monochrome images or a single channel from color images to be used so that it may compute the homography transformation matrix without the need of specifying image coordinates in the source and destination images. It does so iteratively and with a relatively rapid convergence rate, typically requiring no more than 50 iterations depending on the specified tolerances. As can be seen below, `findTransformECC()` requires only source and destination images, an initialized transformation matrix, a motion type, and termination criteria.

```
//Initialize the warpMatrix to a 3x3 identity matrix
Mat warpMatrix = Mat:eye(3,3,CV_32F);
//Initialize the motionType to perform a Homography
int motionType = MOTION_HOMOGRAPHY;
//Set the termination criteria
int numIterations = 50;
double terminationEPS = 0.001;
TermCriteria criteria = (TermCriteria::COUNT+TermCriteria::EPS, numIterations,
terminationEPS);
//Calculate the warpMatrix by passing the sourceImage, the
//destinationImage, the motionType, and the termination criteria
findTransformECC(sourceImage, destinationImage, warpMatrix, motionType, criteria)
```

Once the warp matrix is calculated, the `warpPerspective()` function is called as in the previous example to coregister the `sourceImage` with the coordinate system of the `destinationImage`.

Regardless of the method that is used to determine the homography transformation matrix, the matrix can be reused for each subsequent set of images taken by the pair of cameras as long as their relative position to one another and their distance to the subject remains the same. In the case of aerial imaging, once the aircraft reaches its operational altitude it can begin capturing images. Upon its return, the warp matrix can be calculated once and then applied to each subsequent image from the source camera to coregister it with the destination camera's coordinate system.

If you would like to implement an aerial imaging or machine vision system that requires multiple cameras and image coregistration, reach out to our imaging experts at info@lumenera.com.They can help point you in the right direction in terms of camera selection and can assist you with integrating Lumenera's SDK into your software platform of choice.

ABOUT LUMENERA

Lumenera Corporation, a division of Roper Technologies, headquartered in Ottawa, Canada, is a leading developer and manufacturer of high performance digital cameras and custom imaging solutions. Lumenera cameras are used worldwide in a diverse range of industrial, scientific and security applications.

As a global market leader Lumenera provides an extensive range of high quality digital cameras with unique combinations of speed, resolution, and sensitivity to satisfy the demands of today's imaging applications. Lumenera also offers custom design services to OEM partners requiring specialized hardware and software features. Core competencies include digital bus technologies such as USB 3.0, USB 2.0, Ethernet, HDMI, and Gigabit Ethernet (GigE) as well as a complete command of digital imaging hardware and software built around CMOS and CCD based imagers. Our diversity provides our customers with the benefits of superior price-to-performance ratios and faster time-to-market.

Lumenera Corporation
7 Capella Court
Ottawa, ON
Canada, K2E 8A7

613-736-4077
info@lumenera.com
www.lumenera.com