CAMERA-TO-CAMERA GEOMETRY ESTIMATION REQUIRING NO OVERLAP IN THEIR VISUAL FIELDS

Ding Yuan ♠*Ronald Chung*
Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong
e-mail: {dyuan, rchung}@mae.cuhk.edu.hk

ABSTRACT

Calibrating the relative geometry between cameras which would move against one another from time to time is an important problem in multi-camera system. Most of the existing calibration technologies are based on the cross-camera feature correspondences. This paper presents a new solution method. The method demands image data captured under a rigid motion of the camera pair, but unlike the existing motion correspondence-based calibration methods, it does not estimate optical flows nor motion correspondences explicitly. Instead it estimates the inter-camera geometry from the observations that are directly available from the two image streams – the monocular normal flows. Experimental results on real image data are shown to illustrate the feasibility of the solution.

Index Terms—Camera calibration, Extrinsic camera parameters, Active Vision

1. INTRODUCTION

Active vision system that allows each camera in a multitude of cameras to move independently of the others has the advantage that the visual coverage of the entire system could be re-shaped dynamically according to the need. An important problem in active vision is about calibrating the binocular geometry of any two cameras.

Many methods have been proposed on the subject. Some require certain specific objects appearing in the scene [4]. Such methods often constitute simpler solution mechanisms, but they are restricted to certain applications. There are methods that do not require the presence of calibration objects but the accessibility of cross-camera feature correspondences [3][6]. However, cross-camera correspondences require the cameras to have much in common in what they picture. In active vision, the visual fields of the cameras, due to their freely-moving nature, could have nothing in common.

A natural approach of tackling the problem is to conduct a rigid motion of the two cameras, establish motion correspondences in the respective image streams, estimate the camera motions **A** and **B** of the two cameras from the respective sets of motion correspondences, and recover the inter-camera geometry **X** from the composite transformation relation **AX**=**XB** (e.g., in [1]). However, establishing motion correspondences (i.e., full optical flow) is an ill-posed task due to the aperture problem, and requires the adoption of heuristics like scene- or flow- smoothness which usually are not applicable to everywhere in the scene.

This paper describes a new method of determining the relative geometry of two cameras. Different from all the methods mentioned above, the method does not assume presence of calibration objects or specific features in the imaged scene, nor does it impose restriction on the viewing directions of the cameras, thus allowing the visual fields of the cameras to have little or zero overlap. The estimation starts from the visual motion data acquired under a rigid motion of the cameras, and it estimates the inter-camera geometry from the observations that are directly available in the two image streams – the normal flows.

We assume that the intrinsic parameters of the cameras have been estimated by self-calibration methods like[5][6]. The focus of this work is the estimation of the camera-to-camera geometry.

2. FIELD MODELS USED FOR IMAGE DOMAIN

Fermüller and Aloimonos [2] proposed a few field descriptions of the image domain that is with regard to any chosen axial direction that passes through the optical center of the camera. The fields allow a camera's ego-motion to be estimated directly from normal optical flows. In this work we adopt the same field models to determine binocular geometry instead.

For any chosen axis **s**= [*A B C*] that passes through the optical center, on the image plane we can draw a series of conic sections generated by a family of cones centered at axis **s**. The s-coaxis vector field direction at each image position is the one that is perpendicular to the tangent of the conic section at that image point:

$$[M_x,M_y]=[(-A(y^2+f^2)+Bxy+Cxf),(Axy-B(x^2+f^2)+Cyf)] \quad (2.1)$$

♠ Corresponding author

where $[M_x, M_y]$ is the field vector assigned to image point $[x, y]$ and f is the focal length of the camera.

Suppose a camera undergoes a pure translation. For any particular choice of the s-axis, we have an s-coaxis field direction (Equation 2.1) at each image position. At any image position, the dot product between this field direction and the optical flow there allows the image position be labeled as either: '+' if the dot product is positive, or '-' if negative. By the distributions of the '+' and '-' labels, the image plane is divided into two regions: positive and negative, with the boundary being a 2nd order curve called the zero-boundary, as illustrated by Fig.1(a). This 2nd order curve is a function of the focus of expansion (FOE) of the optical flow, which precisely describes the translational direction of the camera. Fig.1(b) illustrates the positive-negative pattern generated in the same way when the camera takes a pure rotation. Different from the pattern Fig.1(a), the zero-boundary is now a straight line, which depends only on the ratios α/γ and β/γ where $\boldsymbol{\omega}=[\alpha\ \beta\ \gamma]$ is the rotation.

If the camera motion has both translation and rotation, the positive-negative pattern of the image space is the addition of the two patterns Fig.1(a) and Fig.1(b) in this manner: Positive+Positive=Positive, Negative+Negative= Negative, and Positve+Negative=Don't know (this zone depends on the structure of the scene), as illustrated by Fig. 1(c).

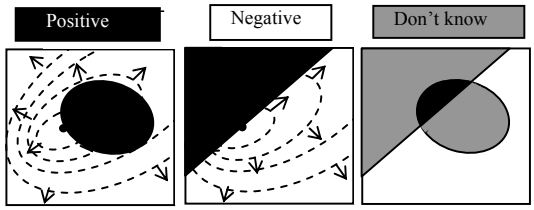


Fig.1 s-coaxis positive-negative patterns. (a) Camera undergoes pure translations. (b) Camera undergoes pure rotations. (c) Camera takes an arbitrary motion (including translation and rotation).

The difficulty is, only normal flows, the local gradients of the intensity information in the captured image sequences, can be computed directly. Fortunately, for any given s-axis, the positive-negative pattern can still be generated from the normal flows at least some image positions [2].

3. BINOCULAR GEOMETRY ESTIMATION

Suppose the cameras in our vision system move independently to track the object of interest, which results in a time-varying relative geometry of the cameras. At any instant that the relative geometry of the cameras is needed, the cameras can have their binocular geometry frozen to conduct a rigid motion of the camera pair, so as to collect two image streams for estimating the binocular geometry $(\mathbf{R}_x, \mathbf{t}_x)$. Here we use $(\mathbf{R}_A, \mathbf{t}_A)$ and $(\mathbf{R}_B, \mathbf{t}_B)$ to represent the motions of camera A and camera B respectively. The composite transformation relation **AX=XB** mentioned previously could be broken into:

$$\mathbf{R}_A\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_B \tag{3.1}$$

(or $\boldsymbol{\omega}_A = \mathbf{R}_x\boldsymbol{\omega}_B$ in vector form)

$$(\mathbf{R}_A - \mathbf{I})\mathbf{t}_x = \mathbf{R}_x\mathbf{t}_B - \mathbf{t}_A \tag{3.2}$$

where $\mathbf{R}_x$, $\mathbf{R}_A$, $\mathbf{R}_B$ are the rotational components in **X**, **A**, **B**; $\mathbf{t}_x$, $\mathbf{t}_A$, $\mathbf{t}_B$ are the translational vectors; and $\boldsymbol{\omega}_A$ and $\boldsymbol{\omega}_B$ are the rotational components of camera A and B in vector form.

Inspired by Fermüller and Aloimonos' work [2], we see the problem in the following light. Since the zero-boundaries (Fig.1) on the positive-negative patterns that carry information of the camera motions are what can be obtained directly from the image sequences without using any artificial constraints, we can estimate the binocular geometry by locating the zero-boundaries. Even so, locating zero-boundaries precisely is a great challenge. First of all, for any arbitrarily given s-axis, only a few image points, where their normal flows are exactly along the direction of the s-coaxis vector field, would be valid candidates and taken into account to generate the positive-negative pattern. Consequently sparse '+' and '-' candidates of the pattern causes the great difficulty in precisely locating the zero-boundaries. Furthermore, two "Don't know" regions appearing in the pattern Fig.1(c) result in more uncertainties in estimating zero-boundaries, when camera takes an arbitrary motion.

Aiming at the above two major problems, we propose our strategies. First, we apply more s-axes to make use of as many normal flows as possible to improve the precision of locating zero-boundaries. Also we hope to avoid dealing with patterns having "Don't know" regions. We try to obtain some simpler patterns (Fig.1 (a)&(b)) by applying specific motions to simplify the pattern analysis,

3.1. Estimation of $\mathbf{R}_x$

We let the camera pair undergo a specific rigid motion – a pure translation – so as to reduce the complexity in locating the zero-boundaries in the positive-negative patterns of the two image domains.

With rigid pure translation of the camera pair, the motion of each camera is also a pure translation. With this, for any arbitrarily chosen s-coaxis field, each camera will have in its image domain the positive-negative pattern like Fig. 1(a). The positive and negative regions are separated by a second order curve, and there is no "Don't know" area. The localization of the zero-boundary and in turn the determination of the direction $\tilde{\mathbf{t}}_A$ or $\tilde{\mathbf{t}}_B$ (unit vectors representing the respective FOEs of the two cameras) becomes trivial, as long as enough s-axes are used.

From (3.2) we have:

$$\tilde{\mathbf{t}}_A = \mathbf{R}_x \tilde{\mathbf{t}}_B \tag{3.3}$$

Then the rotational component $\mathbf{R}_x$ of the binocular geometry can be determined as:

$$\mathbf{R}_x = \mathbf{U}_t \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}_t \mathbf{V}_t^{\mathrm{T}}) \end{bmatrix} \mathbf{V}_t^{\mathrm{T}} \tag{3.4}$$

where $\mathbf{K}_t = \sum_{\alpha=1}^{N} \tilde{\mathbf{t}}_A \tilde{\mathbf{t}}_B^{\mathrm{T}}$,and $\mathbf{K}_t = \mathbf{U}_t \ \mathbf{S}_t \ \mathbf{V}_t$ by SVD (singular value decomposition). N represents the number of pure translations in different directions. It can be proved that two rigid pure translations (in different directions) are the minimum to obtain a unique solution i.e., min (N) =2.

3.2. Estimation of $\mathbf{t}_x$ up to Scale

Similar to the previous step of estimating the rotational component $\mathbf{R}_x$, we apply pure rigid rotations this time to the camera pair to compute the baseline $\mathbf{t}_x$ of the binocular geometry. However, only $\mathbf{t}_x$ up to arbitrary scale is to be obtained, unless certain metric measurement of the 3D world is available.

Suppose the camera pair has a pure rotation about an axis passing through the optical center of one camera, say the optical center of camera A. Then camera A undergoes only a pure rotation, while camera B rotates about an axis passing through the optical center of camera A, and at the same time translates along the direction tangential to the baseline. In this case Equation (3.2) can be rewritten as:

$$(\mathbf{R}_A - \mathbf{I})\mathbf{t}_x = \mathbf{R}_x \mathbf{t}_B \tag{3.5}$$

where rank($\mathbf{R}_A$- $\mathbf{I}$)=2, since it has a nonzero nullspace. We then rewrite (3.5) to a homogeneous system:

$$\hat{\mathbf{A}} \tilde{\mathbf{t}}_x = \mathbf{0} \tag{3.6}$$

where $\hat{\mathbf{A}}$, calculated from $\mathbf{R}_x$, $\mathbf{R}_A$ and $\tilde{\mathbf{t}}_B$,, is a 2×3 matrix with rank($\hat{\mathbf{A}}$)=1. $\tilde{\mathbf{t}}_x$ here is the normalized vector representing the direction of the baseline. Obviously two rotations are the minimum to obtain a unique $\tilde{\mathbf{t}}_x$, and it can be achieved by applying SVD to Equation (3.6).

Camera A has a simple positive-negative pattern without the "Don't know" area (like Fig. 1 (b)), in which the positive and negative patterns are divided simply by a straight line. We use the vector $\boldsymbol{\omega}_A$=[α_A β_A γ_A] to describe the rotation of camera A. The ratios α_A / γ_A and β_A / γ_A can be recovered from locating the straight zero-boundary. The third component γ_A have to be estimated by using the method named "detranslation" that is presented in [2].

For camera B, the positive-negative pattern (Fig.1(c)) generated from the normal flows is more complex, because it contains two "Don't know" areas. However, the rotational component $\boldsymbol{\omega}_B$ of camera B can be computed directly from Equation (3.1), with knowledge of $\mathbf{R}_x$ determined in the previous step. When applying the s-coaxis vector field, the two straight zero-boundaries on the positive-negative pattern are determinable from $\boldsymbol{\omega}_B$. As a result, the other 2nd order zero-boundary defined by FOE can be located, despite the presence of the two "Don't know" regions. The direction $\mathbf{t}_B$ of the translational component is determined during locating the 2nd order boundary. Finally, $\mathbf{t}_x$ (up to arbitrary scale) can be calculated from (3.6) once $\mathbf{R}_A$ and $\tilde{\mathbf{t}}_B$ are both computed from the positive-negative patterns.

4 .EXPERIMENTAL RESULTS

We have implemented the proposed method and tested it with both synthetic and real image data to investigate the performance.

4.1. Experimental Results on Synthetic Data

The experiments on synthetic data aim at investigating the accuracy and precision of the algorithm, because there is always ground truth to compare the experimental results with. Normal flows are the only input to our algorithm, just like in the case of real image experiments. We used image resolution of 101×101 in the synthetic data.

4.1.1. Estimation of $\mathbf{R}_x$

The normal flows are generated by assigning to each image point an arbitrary intensity gradient direction. Dot product between the gradient direction and the optical flow incurred from the assumed camera motion determines the normal flow precisely.

Below we show how we locate the zero-boundary. Given the first s-axis, for instance $\mathbf{s}$=[1 0 0], from the normal flows in the image frame, we got the first positive-negative pattern. We assumd the FOE to be somewhere in the image domain for simplicity. After investigating the pseudo FOEs 0.25 by 0.25 pixel, more than 1000 2nd order curves, determined from different pseudo FOEs, could well divide the pattern into two regions. Then we applied a second s-axis to examine if those pseudo FOEs that had good performance in the first pattern still perform well in this new pattern. We kept those that still had good performance in the next round under a new s-axis. We repeated this process, until all possible FOEs were located within a small area. Then the center of these possible FOEs was considered as the input for computing $\mathbf{R}_x$ using Equation (3.4). Experiments showed that the number of possible FOEs dramatically decreases as more s-axes were utilized.

We estimated the FOEs by locating the zero-boundaries for both camera A and B first, and the rotational component $\boldsymbol{\omega}_x$ of the binocular geometry was then estimated. The calculation result is shown in the Tab.1. The error is 0.7964° in direction, 1.2621% in length.

4.1.2. Estimation of $\mathbf{t}_x$ *up to Arbitrary Scale*

We assumed that the camera pair were rotated about an axis passing the optical center of camera A at two different given velocities. As above, the normal flows were generated to be the inputs. We located the zero boundaries on the positive-negative patters to estimate rotations $\boldsymbol{\omega}_A$ of camera A, using the algorithm named "detranslation" [2]. FOE of camera B $\tilde{\mathbf{t}}_B$ was obtained readily from the patterns. Finally we obtained $\mathbf{t}_x$ up to arbitrary scale using Equation (3.6).The result, shown in Tab.1, is a unit vector describing the direction of the baseline. The angle between the ground truth and the result is 2.0907°.

TABLE 1. ESTIMATION OF $\boldsymbol{\omega}_x$ AND $\mathbf{t}_x$ UP TO SCALE

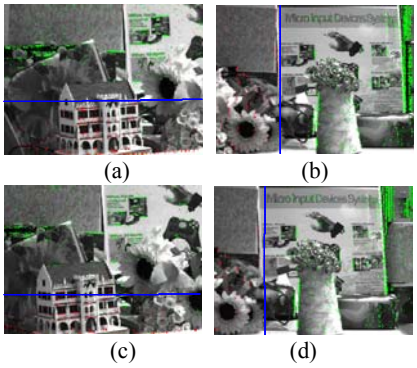
	Ground Truth	Experiment
$\boldsymbol{\omega}_x$	$[0.100\ \ 0.100\ -0.200]^T$	$[\ 0.097\ \ 0.204\ \ -0.203]^T$
$\mathbf{t}_x$	$[-700\ \ 20\ \ 80]^T$	$[-0.988\ \ 0.043\ \ 0.147]^T$

In this experiment, synthetic normal flows, computed from full optical flows by allocating to each pixel a random gradient direction, are more general than those calculated from the image sequences, because there is no assumption on the characteristics of the captured scene. The precision of the results well demonstrates the accuracy.

4.2. Experimental Results on Real Image Data

Here we only show results on the recovery of $\mathbf{R}_x(\boldsymbol{\omega}_x)$ due to limitation of page space. We moved the camera pair on a translational platform. Resolution of the image sequences is 640×480. We do not have any reference object in the scene. Moreover, there are almost no overlap in the two cameras' visual fields except a small portion of the background. Also there are a lot of occlusions and depth discontinuities. Estimating the binocular geometry of the cameras from such image data is an impossible task to the correspondence-based methods, and a difficult task to the motion correspondence-based methods.

We examined pseudo FOEs pixel by pixel in the image frames. No more than 132 s-axes were needed to precisely locate the locations of the possible FOEs. The zero-boundaries determined by the FOEs are shown in Fig 2.



(a) (b) (c) (d)

Fig 2. The zero-boundaries (blue curves) determined by estimated FOEs. Green dots represent negative candidates; red dots represent positive candidates. (a) Camera A, Motion 1, **s**=[0 1 0]; (b) Camera B, Motion 1, **s**=[1 0 0] ; (c) Camera A, Motion 2, **s**=[0 1 0]; (d) Camera B, Motion 2, **s**=[1 0 0].

As there is no ground truth available on the extrinsic parameters of the camera pair, we compare the results with those from a second independent experiment so as to examine the consistency of the solution. The binocular geometry was kept the same in the two experiments. Results of $\boldsymbol{\omega}_x$ in two individual experiments are shown in Tab.2.

TABLE 2. ESTIMATION OF THE ROTATIONAL COMPONENT $\boldsymbol{\omega}_x$ OF THE BINOCULAR GEOMETRY

Experiment 1	Experiment 2
$[0.1066\ \ -0.1328\ \ -3.0878]^T$	$[0.0879\ \ -0.3425\ \ -3.2002]^T$

We compared the two rotational vectors and calculated the error, which is 12.72% in length and 3.6675° in direction.

5. CONCLUSION AND FUTURE WORK

We have presented a method of determining the binocular geometry directly from monocular normal flows, which requires neither cross-camera correspondences nor full optical flow estimation. Our future work is to relax the requirement of the specific stereo-rig motions.

ACKNOWLEDGMENT

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4195/04E), and is affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

REFERENCES

[1] F. Dornaika and R. Chung, "Stereo geometry from 3D ego-motion streams , " *IEEE Trans. On Systems, Man, and Cybernetics: Part B, Cybernetics*, Vol. 33(2) April, 2003.

[2] C. Fermüller and Y. Aloimonos, "Qualitative egomotion," *Int' Journal of Computer Vision*, Vol.15, 7-29, 1995.

[3] R. Hartley, "An algorithm for self calibration from several views", *Proc. Conf. Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 908-912, June 1994.

[4] H., Malm and A. Heyden, "Stereo Head Calibration from a Planar Object," *Proc. Conf. Computer Vision and Pattern Recognition*, 657-662, Dec. 2001.

[5] S. J. Maybank and O. Faugeras, "A Theory of self-calibration of a moving camera," *Int' Journal of Computer Vision*, Vol. 8(2), 123-152, Aug. 1992.

[6] Z. Zhang, Q.-T. Luong, and O. Faugeras, "Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction," *IEEE Trans. on Robotics and Automation*, Vol. 12 (1), 103-113, February 1996.