A single-frame visual gyroscope

Georg Klein and Tom Drummond
{gswk2|twd20}@eng.cam.ac.uk
Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK

Abstract

Rapid camera rotations (e.g. camera shake) are a significant problem when real-time computer vision algorithms are applied to video from a handheld or head-mounted camera. Such camera motions cause image features to move large distances in the image and cause significant motion blur. Here we propose a very fast method of estimating the camera rotation *from a single frame* which does not require any detection, matching or extraction of feature points and can be used as a motion estimator to reduce the search range for feature matching algorithms that may be subsequently applied to the image. This method exploits the motion blur in the frame, using features which remain sharp to rapidly compute the axis of rotation of the camera, and using blurred features to estimate the magnitude of the camera's rotation.

1 Introduction

Real-time visual tracking is capable of following the pose of a camera relative to known surroundings. In the field of Augmented Reality (AR), visual tracking systems are often employed with head-mounted cameras to measure the pose of the user's head. Such tracking systems should ideally operate at high frame-rates, produce highly accurate results, and be robust to the fast camera rotations which a head-mounted camera can undergo. Satisfying these conditions simultaneously using purely visual sensing remains challenging today: in particular, many real-time systems are not robust to rapid camera rotations.

Real-time tracking systems commonly rely on salient features such as points, edges or texture patches. These features are tracked across frames to determine camera pose. Since searching a full image for features is computationally expensive, many systems achieve real-time performance by limiting their search range; features are assumed to lie near positions predicted by a motion model. Other systems cannot uniquely identify features and must rely on predicted feature locations to establish feature correspondences. Both of these approaches can fail under the large unpredictable image motions produced when a user's head rotates rapidly.

Tracking rapid rotations is often further complicated by motion blur, which can degrade an image sufficiently to make features undetectable. For example, a moderate camera rotation of 90 degrees per second exposed for 1/30th of a second with a focal length of 1000 pixels could blur image features across 50 pixels; this level of image blur would make the matching of all but the largest-scale image features very difficult. For these reasons, AR applications often employ additional sensors to provide robustness to rapid

rotations. Rate gyroscopes, which provide direct measurements of rotational velocity, are commonly used.

This paper proposes a vision-based alternative to the use of rate gyroscopes. We describe a novel algorithm which can compute rotational velocity (up to a sign ambiguity) from a single video frame. This is achieved by analysing the structure of the motion blur present in the image. The basis for the operation of the algorithm is the insight that in the presence of sufficient motion blur, the only sharp edges present in the image will be those parallel to the direction of blur; this allows the center of camera rotation to be computed rapidly, without the use of large-scale 2D image computations. Once the center of rotation is found, the magnitude of rotation can be quickly computed under some simplifying assumptions. Since the algorithm can process a 640×480 video frame in under 3ms, it is sufficiently rapid to be useful as an initialisation stage for other real-time tracking systems.

Section 2 of this paper provides more background information and describes related previous work. The algorithm is described in Section 3 and results are presented in Section 4. The algorithm is subject to a number of limitations, some of which (notably the ambiguity of sign) are unavoidable and others which are the result of speed vs. accuracy trade-offs; these are described in Section 5. Conclusions are presented in Section 6.

2 Background

In recent years, some real-time tracking systems which attempt to handle motion blur have been presented. Our earlier work [5] attempts to track heavily blurred image edges with a matched filter, but requires the use of rate gyroscopes to predict blur magnitude in the image. Claus and Fitzgibbon [1] produce a fiducial detector robust to moderate amounts of blur by including blurred images in the training set of a machine learning algorithm. Gordon and Lowe [3] obtain some resilience to motion blur by tracking features obtained by the scale-invariant feature transform (SIFT): this transform extracts features of many scales from the image, and hence makes use of large-scale features less affected by moderate amounts of blur.

Further, any system which can perform a localisation step at each frame (including [1] and [3]) may be considered robust to motion blur in that even if tracking fails during transient motions, tracking can be resumed once the camera rotation slows down. For this reason, many augmented reality applications employ fiducial-based tracking systems such as AR Toolkit, which is based on the work of Kato and Billinghurst [4]. The absence of pose estimates during rapid rotations may be acceptable for many applications.

In many cases motion blur can be avoided altogether. For example, the popular unibrain Fire-i camera used in many vision systems allows the electronic selection of a short exposure time. The selection of a very short exposure time minimises blur in captured images. Unfortunately this approach is not universally applicable: The use of low exposure times incurs noise penalties in low-light situations; further, many cameras (particularly ultra-compact modules attractive for AR or robotics applications) lack the facility of adjusting exposure time. Also, the reduction of exposure time still leaves the problem of tracking large unexpected image motions caused by rapid camera rotations. Rate gyroscopes, which offer robust, high-bandwidth measurements of rotational velocity remain commonly used in AR applications [8, 9].

Recent work similar in spirit to the contribution of this paper has been presented by Lin [6]. Motion blur in single images of moving vehicles is used to estimate the speed of vehicle motion. This estimate is used to de-blur the image, allowing blurred registration plates to be read. This method uses a Fourier transform to detect the image orientation of blur. Blur magnitude is estimated by analysing intensity ramps in scan-lines with an uniform background (i.e. a blue sky).

Rekleitis [7] uses motion blur to estimate optical flow in an image. Steerable filters applied to the Fourier transform of image patches are used to determine the orientation of local blur. Once orientation has been determined, the 2D spectrum is collapsed to obtain the patch's spectrum along the direction of blur; blur length is extracted using cepstral analysis. Run-time performance is limited by the cost of 128×128-pixel FFTs.

Favaro *et al*. [2] exploit both motion blur and distance-varying defocus present in images to reconstruct a scene's depth map, radiance and motion. The use of motion blur is an extension to the authors' previous work in the domain of shape from defocus. Blur is modelled as a diffusion process whose parameters are estimated by minimising the discrepancy between input images to the output of the hypothesised diffusion process. This approach is very different from our proposed method in that it attempts to determine the maximum of information from every pixel of two or more input images, whereas we attempt to measure only rotation from a single image in the shortest possible amount of time.

3 Method

This section describes our algorithm to estimate camera rotation from the motion blur present in a single video frame. The algorithm is designed for speed rather than accuracy; to this end, a simple model of motion blur is used. We consider rotation only and assume that the camera is not translating, or that translation does not contribute significantly to motion blur; further, the scene is assumed to be static.

During the frame's exposure, the camera is assumed to rotate with constant angular velocity about a center of rotation $\begin{pmatrix} x_c & y_c \end{pmatrix}^T$ in the image plane. Points d pixels away from this center will therefore be blurred across an arc of length θd, where θ is the angle the image rotates during exposure of the frame.

Considering projection by a standard pin-hole model with the camera center at the origin, the optical axis aligned with the z-axis and the image plane at $z = F$, the point about which the image rotates has coordinates

$$\boldsymbol{c} = \begin{pmatrix} x_c \\ y_c \\ F \end{pmatrix} \tag{1}$$

with all units in pixels. In 3D space, the camera rotates around an axis of rotation which passes through the origin and is described by the unit vector $\hat{\boldsymbol{a}}$. It follows that $\boldsymbol{c}$ is the projection of $\hat{\boldsymbol{a}}$ into the image plane. In the case that $\hat{\boldsymbol{a}}$ is parallel to the image plane (i.e. when the camera is purely panning), $\boldsymbol{c}$ is at infinity in the image plane, and the model turns arcs of blur in the image into straight lines.

Strictly speaking, the image locus swept by a point under camera rotation could be any conic section; the circular assumption corresponds to a very large focal length or a

spherical imaging surface. However, the circular assumption yields faster computation and introduces only small errors, particularly when using lenses which exhibit barrel distortion.

The algorithm operates in two stages: Section 3.1 demonstrates how the axis of rotation $\hat{\boldsymbol{a}}$ can be found. Once this has been calculated, the blur length is estimated in Section 3.2. Apart from the pixel aspect ratio, the method requires no precise knowledge of camera parameters.[1] Consequently, all quantities are calculated in pixel units. A conversion of these results to a 3D coordinates is straightforward if camera focal length and optic center are known. Knowledge of frame exposure time can then be used to obtain rotational velocity from the calculated blur length.

3.1 Axis of Rotation

To determine the axis of rotation in a blurred image we exploit the directional nature of motion blur. Any point in the image is blurred tangentially to a circle centered on $\boldsymbol{c}$, and not blurred in the perpendicular direction (radially towards $\boldsymbol{c}$). It follows that image edges emanating radially from $\boldsymbol{c}$ are corrupted by blur, while intensity edges in tangential directions are preserved. This is illustrated in Figure 1: Panel 1 contains a frame affected by motion blur. Panel 2 shows the results of a full-frame Canny edge extraction of this frame: edges parallel to the direction of blur dominate. Thus, the point $\boldsymbol{c}$ can be found as the point which is most perpendicular to all edges remaining in the blurred image.

To avoid the cost of full-frame edge extraction, the image is sparsely searched for edgels. This is done along a grid of vertical and horizontal lines spaced 10 pixels apart. The changes in intensity between adjacent pixels along these lines are computed: local maxima of instantaneous intensity change which exceed a threshold value are assumed to be edgels. Typically, between 100 and 600 edgels are found in this way, and the position $\boldsymbol{p} = \begin{pmatrix} x & y & F \end{pmatrix}^T$ of each edgel is recorded.

At each edgel site, the direction of the local image gradient is found using the Sobel operator. This yields the values G_x and G_y which describe local gradient in the x and y directions respectively. The vector $\boldsymbol{g} = \begin{pmatrix} G_x & G_y & 0 \end{pmatrix}^T$ then describes the direction of maximum gradient, i.e. the normal to any edge in the image. Panel 3 of Figure 1 shows edgel normals extracted from the video frame.

Each edgel describes a line $\boldsymbol{l} = \boldsymbol{p} + \lambda \boldsymbol{g}$ in the image plane along which the rotation center $\boldsymbol{c}$ is expected to lie. This line is more conveniently expressed as the intersection of the image plane with a plane $\mathcal{N}$ passing through the origin; this plane is parameterised by its unit normal vector $\hat{\boldsymbol{n}}$, given by

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{p} \times \boldsymbol{g}}{|\boldsymbol{p} \times \boldsymbol{g}|} \tag{2}$$

To find the image rotation center $\boldsymbol{c}$ we employ RANSAC followed by an optimisation of the consensus set. In the RANSAC stage, each hypothesis is formed by randomly selecting two edgels a and b. The image rotation center is the given by the intersection of lines $\boldsymbol{l}_a$ and $\boldsymbol{l}_b$. To handle rotation centers at infinity, this intersection is formulated in terms of the axis of rotation $\boldsymbol{a}$, which lies along the intersection of planes $\mathcal{N}_a$ and $\mathcal{N}_b$:

$$\boldsymbol{a} = \hat{\boldsymbol{n}}_a \times \hat{\boldsymbol{n}}_b \tag{3}$$

[1] The focal length F used in calculations can be very approximate. Here it is set to 640 pixels.

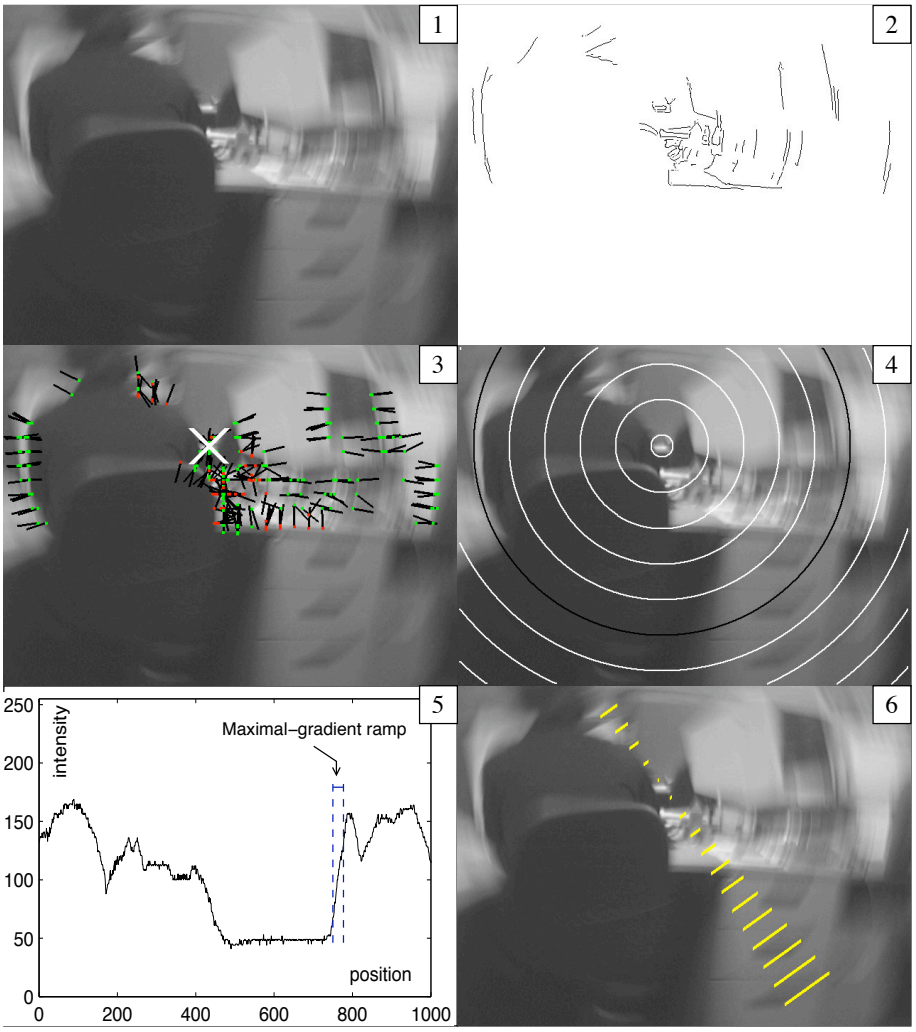

Figure 1: Operation of the Algorithm. (1) The input picture shows an office scene blurred by moderate camera rotation. (2) A Canny edge extraction (not used in the algorithm) illustrates the dominance of edges parallel to the direction of blur. (3) Edgels are extracted along grid lines and the orientation of maximal image gradient is indicated by black lines. The best intersection of the black lines is found using RANSAC and yields the center of rotation in the image (white $\times$). (4) Pixels along concentric circles around the rotation center are sampled. (5) Plot of the pixel intensity values sampled clockwise along the black circle. The highest-gradient intensity ramp is indicated: this is interpreted as the least-blurred feature on the circle and used to estimate blur length. (6) By combining estimated blur lengths from all circles, the overall angular blur (and hence the estimated camera rotation) is found. The estimated blur length is drawn in the image.

To evaluate consensus for each hypothesis, we sum angular error terms from all other edgels. For the ith edgel, θ_i is the angle at $\boldsymbol{p}_i$ in the image between the line $\boldsymbol{l}_i$ and the vector $\boldsymbol{c}-\boldsymbol{p}_i$. We approximate the square of the sine of this quantity for an error metric. In terms of the hypothesised axis of rotation $\boldsymbol{a}$, the error metric ε_i is

$$\varepsilon_i = \frac{|(\boldsymbol{a}\times\boldsymbol{p}_i)\times\hat{\boldsymbol{n}}_i|^2}{|\boldsymbol{a}\times\boldsymbol{p}_i|^2} \approx \sin^2(\theta_i). \tag{4}$$

The error metric is capped at a threshold value ε_{max} and the hypothesis with the lowest sum error is selected. The consensus set for this hypothesis is the set of N edgels whose error metric is lower than ε_{max}.

The winning hypothesis $\boldsymbol{a}$ is normalised and then optimised to minimise the sum-squared error $|\boldsymbol{\varepsilon}|^2$, where $\boldsymbol{\varepsilon}$ is the vector of error metrics for the consensus set. This is done using four Gauss-Newton iterations. At each iteration,

$$\boldsymbol{a}' = \boldsymbol{a} + \Delta\boldsymbol{a} \tag{5}$$

$$\Delta\boldsymbol{a} = (J^T J)^{-1} J^T \boldsymbol{\varepsilon} \tag{6}$$

Where J is the $N\times 3$ Jacobian matrix describing partial derivatives of $\boldsymbol{\varepsilon}$ with $\Delta\boldsymbol{a}$

$$J_{ij} = \frac{\partial\varepsilon_i}{\partial\Delta a_j} \tag{7}$$

and is found by differentiating Equation (4) w.r.t. $\Delta\boldsymbol{a}$:

$$J_{ij} = \frac{\partial}{\partial\Delta a_j}\left(\frac{|((\boldsymbol{a}+\Delta\boldsymbol{a})\times\boldsymbol{p}_i)\times\hat{\boldsymbol{n}}_i|^2}{|(\boldsymbol{a}+\Delta\boldsymbol{a})\times\boldsymbol{p}_i|^2}\right)\Bigg|_{\Delta\boldsymbol{a}=\boldsymbol{0}} \tag{8}$$

$$= \frac{\partial}{\partial\Delta a_j}\left(\frac{u}{v}\right) = \frac{vu'-uv'}{v^2} \tag{9}$$

$$\text{with}\quad u' = 2((\boldsymbol{a}\times\boldsymbol{p}_i)\times\boldsymbol{n}_i)\cdot((I_j\times\boldsymbol{p}_i)\times\boldsymbol{n}_i) \tag{10}$$

$$v' = 2(\boldsymbol{a}\times\boldsymbol{p}_i)\cdot(I_j\times\boldsymbol{p}_i) \tag{11}$$

where I_j is the jth column of the 3×3 identity matrix. Once $\boldsymbol{a}$ has been optimised, the image center of rotation $\boldsymbol{c}$ is found by extending $\boldsymbol{a}$ to intersect the image plane. The extracted rotation center is indicated in Panel 3 of Figure 1.

3.2 Blur magnitude

Where the axis of rotation has been determined by analysing image structure in the direction perpendicular to local blur, the magnitude of blur is determined by looking at pixels in the blurred direction. Pixels are sampled from $\boldsymbol{c}$-centered circles in the image using an incremental rasterisation algorithm. The circles are initialised at sparse radial intervals (typically 50 pixels apart) to maintain high run-time speed. This process is illustrated in Panel 4 of Figure 1. Each scanned circle produces a 1D signal of image intensity along the circle; this signal is assumed to have been convolved with a rectangular pulse of length d, which must be estimated. One such signal is shown in Panel 5 of Figure 1.

Rekleitis [7] estimates this blur length using cepstral analysis: convolution of the signal with a rectangular pulse produces a sinc-pulse envelope in the frequency domain.

The cepstrum (the inverse Fourier transform of the log power spectrum) is used to recover this envelope's fundamental frequency, which is proportional to the length of motion blur in the image. While this method is attractive in its conceptual elegance, it is unfortunately not applicable here. To make the results of the Fourier- and cepstral analysis resilient to noise, a large number of image samples are needed. Further, the minimum of the cepstrum becomes difficult to locate in images with large blur as camera noise starts to dominate.

Instead, we adopt an ad-hoc approach to blur length detection. Under the assumption that the axis of rotation has been correctly calculated and that the samples are therefore taken along the direction of blur, we notice that the blur length cannot exceed the length of the shortest intensity ramp which was produced by an intensity step in the scene. However, merely measuring minimum ramp length is unreliable, since two intensity steps of opposite sign in close proximity can produce arbitrarily short ramps in the image.

To avoid under-estimating blur length, we only consider ramps which span a large (50 greyscale levels or more) intensity change: These are assumed to have originated from large isolated intensity steps in the image. Under the further assumption that the largest intensity step in every scene spans approximately the same intensity increase, the gradient of the steepest ramp to span this change is then inversely proportional to the length of motion blur. This maximal-gradient ramp is found by a brute-force search in which the shortest distance to span the threshold intensity change is found. The maximal-gradient ramp thus found is illustrated in Panel 5 of Figure 1.

To combine the results of each circular scan, the maximal gradients of each scan are first scaled according to relative circle radius, and then combined with a p-norm (with $p = 5$) to provide some resilience to outliers. The inverse of this result provides the system's estimate for the magnitude of camera rotation during exposure time.

4 Results

This section describes the performance of the system for sequences where the system can operate correctly. There are many scenarios in which the system will always yield incorrect results. These cases are discussed in Section 5.

Figure 2: Rotation center placement results for four test scenes (un-blurred in top row.)

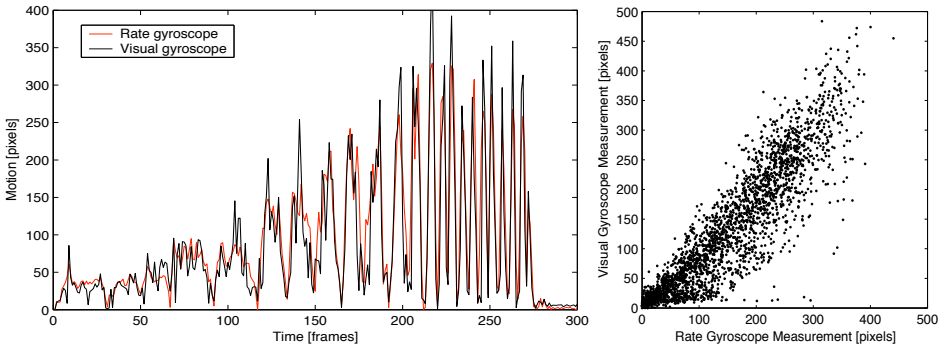

Figure 3: Blur magnitude results compared to a rate gyroscope (taken as ground truth.)

The visual gyroscope was implemented in C++ on a 2.4GHz Pentium-4 computer and tested on live video footage. Images are received from a fire-wire camera at 30Hz, are grey-scale and have a resolution of 640×480 pixels. The camera's exposure time was set to its maximum value of approximately 30ms.

Figure 2 shows results the algorithm's choice of rotation center for a number of different motions in four test sequences. The center-extraction section operates fairly reliably for scenes with motion blur of 10 pixels length or greater, even when these scenes contain very bright elements which make blur length detection problematic.

Figure 3 compares the output of the algorithm to the output of a rate gyroscope mounted to measure camera panning about the vertical axis. The left plot shows a series of camera shakes of increasing intensity recorded in an indoor setting (the same scene as used for Figure 1). Since the visual gyroscope produces measurements with a sign ambiguity, the results show the absolute value of horizontal image motion in pixel units. The plot on the right of Figure 3 compares 2000 samples of rotational velocity taken from the visual and rate gyroscopes. Ideally, the left plots should be identical, and the plot on the right should show a $y = x$ line.

The execution speed of the algorithm is largely limited by the floating-point computations required to determine the center of rotation (RANSAC and optimisation). The cost of these computations scales linearly with the number of edgels used; it is therefore possible to tune execution speed by varying the maximum number of edgels processed per frame. To reliably obtain an execution time of under 3ms per frame, the algorithm processes only the 300 strongest edgels of the 300-2000 edgels typically found in an image. Average computation times of different stages of the algorithm are shown in Table 1.

5 Limitations

This sections describes some known limitations of the proposed system.

1. **Sign ambiguity in blur magnitude:** Under the assumption of time-invariant illumination and an instantaneous shutter across the frame, there is no way of distinguishing the direction of time in a single frame. This information must be acquired elsewhere, e.g. by comparison with other frames or from a tracking system's motion model.

Process	Time [msec]
Edgel extraction	0.50
Best edgel selection	0.05
RANSAC	0.65
Optimisation	0.35
Circular sampling	0.15
Blur length search	0.40
Total	2.10

Table 1: Timing results on a 2.4GHz P4

2. **Intolerance to strobing lights:** Illumination is assumed to be of constant intensity throughout a frame's exposure. Some objects, such as CRT screens, do not satisfy this assumption, and produce sharp edges in otherwise blurred images. If the rest of the image is strongly blurred, these sharp edges can cause both the axis of rotation and blur magnitude to be mis-detected.

3. **Requirement of edges in the scene:** Scenes completely devoid of sharp edges cannot well be distinguished from heavily blurred images. In such cases the system can produce erroneous results. Further, if a scene's edges are all oriented in similar directions the system will frequently mis-detect the axis of rotation. For example, a diagonal pan across a chess-board is poorly handled, since the rotation axis detection stage lacks sharp diagonal edges.

4. **Requirement of motion blur:** The system over-estimates blur magnitude for images with small amounts of motion blur, or no blur at all. For scenes with no blur, the center of location is effectively random.

5. **Assumption of linear camera:** The intensity transfer function of the camera used is assumed to be linear (gamma=1.0). Scenes with bright light sources which saturate the camera sensor can be problematic, as their edges can produce high-gradient intensity ramps even under very large blur.

6. **Pure rotation assumption:** The algorithm assumes that motion blur is caused purely by camera-centered rotation and not by camera translation. In practice, when using a head-mounted camera, rotation does indeed contribute the largest component of motion blur, so this assumption is not unreasonable. There can however be cases in which translation is misinterpreted as rotation.

7. **Fixed rotation center assumption:** The algorithm assumes that the axis of rotation is fixed during exposure. Very rapid camera acceleration can however cause the axis of rotation to change during a frame. In these cases the paths traced by points in the image no longer form concentric circles, and the rotation center detection can fail.

6 Conclusions

This paper has presented a method for calculating camera rotation from a single blurred image. The use of sharp features remaining in the blurred image makes the method fast, permitting its use in combination with other real-time tracking systems.

While axis of rotation of the camera can computed fairly reliably, measurements of blur magnitude are currently fairly noisy. More accurate methods of estimating blur length, and the solutions to some of the limitations of the current system, are the subject of ongoing work.

References

[1] D. Claus and A. W. Fitzgibbon. Reliable fiducial detection in natural scenes. In *Proc. 8th European Conference on Computer Vision (ECCV'04)*, pages 469–480, Prague, May 2004.

[2] P. Favaro, M. Burger, and S. Soatto. Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *Proc. 8th European Conference on Computer Vision (ECCV'04)*, pages 257–269, Prague, May 2004.

[3] I. Gordon and D. Lowe. Scene modelling, recognition and tracking with invariant image features. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 110–119, Arlington, VA, Nov 2004.

[4] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. 2nd Int. Workshop on Augmented Reality (IWAR'99)*, pages 85–94, San Francisco, CA, Oct 1999.

[5] G. Klein and T. Drummond. Tightly integrated sensor fusion for robust visual tracking. In *Proc. British Machine Vision Conference (BMVC'02)*, volume 2, pages 787 –796, Cardiff, September 2002.

[6] H.Y. Lin. Vehicle speed detection and identification from a single motion blurred image. In *Proc. Seventh IEEE Workshop on Application of Computer Vision (WACV/MOTION'05)*, volume 1, pages 461–467, Breckenridge, CO, Jan 2005.

[7] I. Rekleitis. Steerable filters and cepstral analysis for optical flow calculation from a single blurred image. In *Vision Interface*, pages 159–166, Toronto, May 1996.

[8] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa. Accurate image overlay on see-through head-mounted displays using vision and accelerometers. In *Proc. IEEE Virtual Reality Conference (VR'00)*, pages 247–254, New Brunswick, NJ, March 2000.

[9] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Proc. IEEE Virtual Reality Conference (VR'01)*, pages 71–78, Yokohama, Japan, March 2001.