ESTIMATING CAMERA ROTATION PARAMETERS FROM A
BLURRED IMAGE

Giacomo Boracchi<sup>a,b</sup>, Vincenzo Caglioti<sup>a</sup> and Alberto Danese<sup>a</sup>
<sup>a</sup>Dipartimento di Elettronica e Informazione
Politecnico di Milano, Via Ponzio, 34/5 20133 Milano, Italy
<sup>b</sup>Tampere International Center for Signal Processing,
Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

Keywords: Blurred Image Analysis, Camera Rotation, Camera Ego-motion, Blur Estimation, Space Variant Blur.

Abstract: A fast rotation of the camera during the image acquisition results in a blurred image, which typically shows
curved smears. We propose a novel algorithm for estimating both the camera rotation axis and the camera
angular speed from a single blurred image. The algorithm is based on local analysis of the blur smears.
Contrary to the existing methods, we treat the more general case where the rotation axis can be not orthogonal
to the image plane, taking into account the perspective effects that in such case affect the smears.
The algorithm is validated in experiments with synthetic and real blurred images, providing accurate estimates.

1 INTRODUCTION

This paper concerns images corrupted by blur due to
a camera rotation or to a rotating object in the scene.
When the camera or the captured object are purely ro-
tating, the image blur is determined by only two fac-
tors: the camera rotation axis \(a\) and its angular speed \(\omega\). We present a novel algorithm for estimating both
\(a\) and \(\omega\), by analyzing the blur in a single image.

When the camera rotation axis and the angular
speed are known, the rotationally blurred image can
be restored by image coordinates transformation and
blur inversion. In broad terms, the image is trans-
formed from Cartesian to polar coordinates so that the
blur becomes space invariant and can be inverted using
a deconvolution based algorithm. Estimating cor-
rectly the camera rotation axis and its angular speed is therefore crucial for restoring these images as small
errors in the polar transformation are amplified by
blur inversion.

On the other hand, estimating \(a\) and \(\omega\) from a sin-
gle image can be also of interest for robotic applica-
tion as these describe the camera ego-motion.

Figure 1 shows an image acquired during camera
rotation. The shapes of the blur smears show that the
blur is space variant. Typically, these are assumed
arcs of circumferences, all having the same center.
However, this approach neglects the perspective ef-
fects that occur when the rotation axis is not orthog-
onal to the image plane. The proposed algorithm es-
timates the camera rotation axis in the most general
case when it is not necessarily orthogonal to the im-
age plane. To the best of our knowledge this issue has
never been correctly addressed before.

Besides the early works concerning rotational blur
modeling and restoration, Hong and Zhang (Hong and
Zhang, 2003) addressed the issue of both rotational
blur estimation and removal. Their method is based
on an image segmentation along circumferences to

---

*This work was partially supported by CIMO, the
Finnish Centre for International Mobility (fellowship num-
ber TM-07-4952)
estimate the blur and restore the image separately in these subsets. Recently, an algorithm for estimating the camera rotation from a single blurred image has been proposed (Klein and Drummond, 2005). The algorithm is meant as a visual gyroscope and it is targeted to an efficient implementation. In particular, this algorithm requires edges in the scene.

All the existing methods, concerning both image restoration and blur estimation, assume that the blur smears are arcs of circumferences having the same center. Therefore these methods are accurate only on images where the rotation axis is orthogonal to the image plane.

We present an algorithm for estimating the camera rotation axis and angular speed in the most general case, where the rotation axis is not orthogonal to the image plane. The proposed algorithm is mostly targeted to accuracy rather than efficiency and does not require the presence of edges in the scene.

2 PROBLEM FORMULATION

We propose an algorithm for estimating the camera rotation axis \( a \) and its angular speed \( \omega \) by analyzing a single blurred image acquired during camera rotation. We assume that the camera is calibrated, the rotation axis \( a \) passes through its viewpoint \( V \), i.e. \( V \in a \), and \( w \) is constant. Figure 2.a illustrates the situation typically considered in literature, where the rotation axis is perpendicular to image plane \( \pi \). The principal point \( P \) and the intersection between the image plane and the rotation axis \( C = \pi \cap a \) then coincide. Analogous blur is obtained when \( a \perp \pi \) and \( V \notin a \), but the capture scene is planar and parallel to \( \pi \) (Rribaric et al., 2000).

In this work we consider the most general situation, illustrated in Figure 2.b, where \( a \) is not orthogonal to \( \pi \) and the camera viewpoint \( V \in a \).

2.1 Image Blur

An image blur is defined as the set of image pixels that a viewing ray intersects during a camera rotation of \( 2\pi \) around axis \( a \). Figure 2 illustrates examples of blurring paths. In rotationally blurred images every pixel is merged with neighboring pixels from the same blurring path, see Figure 1. The blur is therefore space variant and can not be represented as a linear shift invariant system. We therefore model the rotational blur by an operator \( K \) on the original image \( y \) (Bertero and Boccacci, 1998) so that the observed blurred and noisy image \( z \) is

\[
z(x) = K(y)(x) + \eta(x) \quad x = (x_1, x_2) \in X,
\]

where \( \eta(x) \) is white Gaussian noise. The blur operator \( K \) can be written as

\[
K(y)(x) = \int_X k(x, s) y(x) ds.
\]

2.2 Line Segments from Blurring Paths

Other blurring effects, such as the out of focus blur, lenses aberrations and camera shake, are not considered.

3 THE ALGORITHM

The proposed algorithm is based on three steps: in the first step the lines tangent to blurring paths at some image pixels are estimated (Section 3.1). In the second step, these lines are used in a voting procedure for estimating the rotation axis \( a \) (Sections 3.2 and 3.3). The third step consists of the angular speed \( \omega \) estimation (Section 3.4).

3.1 Blur Tangent Direction Estimation

Image blur is analyzed within \( N \) image regions taken around selected pixels \( \{x_i\}_{i=1,...,N} \). There are no particular requirements in selecting \( \{x_i\}_{i=1,...,N} \), but avoiding smooth areas while covering uniformly the image. Therefore we take the local maxima of Harris corner measure (Harris and Stephens, 1988), or whenever these do not cover uniformly the image, we take \( \{x_i\}_{i=1,...,N} \) on a regular grid.
Blur is analyzed using the approach proposed by Yitzhaky et al. (Yitzhaky and Kopeika, 1996) for estimating the direction of blur “smears” by means of directional derivative filters. This method, proposed for space invariant blurs with PSF having vectorial support, assumes the image isotropic. The blur direction $\theta$ is estimated as the direction of the derivative filter $d_0$ having minimum energy response

$$\hat{\theta} = \arg \min_{\theta \in [0,\pi]} \left\{ \| (d_0 \otimes z) \|_1 \right\},$$

(4)

where $\otimes$ denotes the convolution and $\| (d_0 \otimes z) \|_1 = \sum_{x \in X} \| (d_0 \otimes z)(x) \|$ the $l^1$ norm.

Equation (4) is motivated by the fact that the blur removes all the details and attenuates edges of $y$ along blur direction. Therefore the blur direction can be determined by the directional derivative filter having minimum energy. This method can not be directly applied to rotationally blurred images, as the blur is not space invariant because in every pixel the circumference approximating the blurring path (i.e. the PSF) changes.

At $x_i$, the center of each region $U_i$, we estimate the direction $\theta_i$ of the line $l_i$ tangent to the blurring path in $x_i$, as

$$\theta_i = \arg \min_{\theta \in [0,\pi]} \sum_{x_j \in U_i} w_j \| (d_0 \otimes z)(x_j) \|^2,$$

(5)

being $w$ a window function rotationally symmetric with respect to the center. By using Gaussian distributed weights, it is possible to reduce the influence of pixels in Equations (5) with the distance from $x_i$. We adopted the 3 tap derivative filters presented in (Farid and Simoncelli, 2004) for blur analysis in Equation (5). These filters have been selected as they provide good accuracy and as they are separable. Experimentally the $l^2$ norm gave better results than the $l^1$ norm.

Figures 3 shows $\sum_{x_j \in l_i} w_j \| (d_0 \otimes z)(x_j) \|^2$ as a function of $\theta \in [0,\pi]$ within regions of the blurred image containing isotropic textures or edges. Regions containing edges, as pointed out in (Klein and Drummond, 2005), can be exploited for estimating the camera rotation: in $z$ only edges tangent to the blurring paths are preserved. Formula (5) gives accurate results also when $U_i$ contains a blurred edge, as the direction minimizing the derivatives energy is the edge direction, i.e. the blur tangent direction.

### 3.2 Voting Procedure for Circular Blurring Paths

When the camera optical axis and the rotation axis $a$ coincide, the blurring paths are circumferences centered in $C = \pi \cap a$, see Figure 2.a. Circular blurring paths are obtained also when $a$ is parallel to the optical axis and the scene is planar and parallel to the image plane (Ribaric et al., 2000; Hong and Zhang, 2003). In this case $C$ can be determined by a Generalized Hough Transform (Ballard, 1987).

The Generalized Hough Transform is a procedure for computing robust solution to a problem, given some input data. The procedure is developed by means of a parameters space $P$, which is the set of all the possible solutions. A vote is assigned to every parameter that satisfy a datum and then summed to the votes coming from the other data. After having considered all the data, the parameter that received the highest vote is taken as a solution.

In our case $P$ is a discrete grid of all the possible location for $C \in \pi$ and data are the pairs $(x_i, \theta_i)$ $i = 1,..,N$. Note that $C$ could be outside of the image grid $X$. Every data $(x_i, \theta_i)$ identifies a line $l_i$, the line tangent to the blurring path at $x_i$. The set of all the possible rotation centers $C$, given the line $l_i$, is the line perpendicular to $l_i$ and passing through $x_i$.

It is worth to take into account the root mean square error of each $\theta_i$, i.e.

$$\sigma_i = \sqrt{E[(\theta_i - \theta_i^*)^2]}$$

where $\theta_i^*$ represents the true tangent blur direction at $x_i$ and $E[\bullet]$ the mathematical expectation. Since
we can not directly compute $\sigma_i$, we approximate it with an indirect measurement: for example considering the amplitude of the area near $\theta_i$ in the energy function minimized in (5) or considering $\sigma_i$ proportional to $\sigma_0$ (1). Noise standard deviation is estimated using (Donoho and Johnstone, 1994). Given a datum $(\mathbf{x}_i, \theta_i)$, we assign a full vote to all the exact solutions and we spread smaller votes to the neighboring parameters, according to the errors in $\theta_i$.

Let now $p = (p_1, p_2)$ represent a coordinate system in the parameters space and assume $\theta_i = 0$ and $\mathbf{x}_i = p_i = (0, 0)$. Let now model the vote spread assuming that along the line $p_1 = 1$ the errors are distributed as $\sigma_i/\sqrt{2\pi} \cdot N(0, \sigma_i)$. We model the vote spread so that along line $p_1 = k$, the votes are still Gaussian distributed with a full vote at the exact solution $(k, 0)$ for and neighboring parameters the votes depend only on the angular distance from $\theta_i$, see Figure 4. Therefore the following weight function is used for distributing the votes in the parameter space (when $\mathbf{x}_i = p_i = (0, 0)$ and $\theta_i = 0$),

$$v_i(p_1, p_2) = e^{-\frac{p_1^2}{2\sigma_i^2}} \cdot \frac{1}{\sqrt{2\pi} \cdot \sigma_i}, \quad (7)$$

The votes weight function $v_i$, associated to other data $(\mathbf{x}_i, \theta_i)$, correspond to Equation (7) opportunely rotated and translated. When all pairs $(\mathbf{x}_i, \theta_i)$ $i = 1, \ldots, N$ have been considered, the parameter that received the highest vote is taken as the solution, i.e.

$$p = \arg \max_{p \in P} \sum_{i=1}^{N} v_i(p), \quad \text{being} \quad \sum_{i=1}^{N} v_i(p) = \arg \max \sum_{i=1}^{N} v_i(p). \quad (8)$$

The coordinates of $C = \pi \cap a$ are determined from $p$.

### 3.3 Conic Section Blurring Paths

Assuming circular blurring paths reduces the complexity load but gives inaccurate solutions whenever $a$ is not perpendicular to $\pi$. We present an algorithm for estimating $a$ and $\pi$ when $V \in a$ and $a$ is in a general position w.r.t. $\pi$. In particular, if we call $C_a$ a plane perpendicular to $a$, $C_a$ is obtained by two rotations of $\alpha$ and $\beta$ from $\pi$. We do not consider $V \notin a$ as in this case the blur would depend on the scene depth.

Voting in the parameters space show at a glance what happens assuming circular blurring paths when $a$ is not orthogonal to $\pi$. Figure 5.a shows a blurred image produced when the plane orthogonal to $a$ forms angles $\alpha' = 45^\circ$ and $\beta' = 0^\circ$ with $\pi$. If we treat the blurring paths as circunferences, the votes in the parameters space do not point out a clear solution, as shown in Figure 5.b and 5.c.

Directions $\theta_i$ obtained from (5) represent the blurring paths tangent direction, even when the blurring paths are conic sections. But the blurring paths themselves are not circunferences, thus lines perpendicular to these tangent lines do not cross at the same point.

From basic 3D geometry considerations, and as pointed out in (Klein and Drummond, 2005), it follows that the blurring paths are circunferences on an ideal spherical sensor $S$, Figure 2.b. Then, if we project the image from $\pi$ on $S$ surface, the blurring paths become circunferences. Each of these circunferences belongs to a plane and all these planes have the same normal: the rotation axis $a$. Let now consider one of these planes, $\pi_C$, tangent to the sphere. The projections of the blurring paths on $\pi_C$ are circunferences, Figure 2.b.

The plane $\pi$ and the plane $\pi_C$ are related by a projective transformation determined by two parameters, namely $(\alpha, \beta)$, the angles between the two planes. Define the map $M_{\alpha, \beta}: \pi \rightarrow \pi_{\alpha, \beta}$ as the projection from $V$ between $\pi$ and $\pi_{\alpha, \beta}$, which is the plane tangent to $S$, forming angles $(\alpha, \beta)$ with $\pi$ (Rothwell et al., 1992). We search for $(\alpha, \beta)$ that project the blurring paths into circunferences, by modifying the voting procedure of Section 3.2.

There is no need to transform the whole image with $M_{\alpha, \beta}$ as each $I_i$, the line tangent to the blurring path at $\mathbf{x}_i$, can be directly mapped via $M_{\alpha, \beta}$. Let $v_{i, \alpha, \beta}$ be the weight function (7) associated to data $(\mathbf{x}_i, \theta_i)$ $i = 1, \ldots, N$ mapped via $M_{\alpha, \beta}$. The parameters pair identifying the plane $\pi_C$ is estimated as

$$(\hat{\alpha}, \hat{\beta}) = \arg \max_{\alpha, \beta} \sum_{i=1}^{N} v_{i, \alpha, \beta}(p_i), \quad (9)$$

being

$$\hat{p}_{\alpha, \beta} = \arg \max_{p \in P} \sum_{i=1}^{N} v_{i, \alpha, \beta}(p), \quad \sum_{i=1}^{N} v_{i, \alpha, \beta}(p) = \sum_{i=1}^{N} v_{i, \alpha, \beta}(p_i). \quad (10)$$

Figure 5.d and 5.e represent the votes in case the data have been transformed according to the correctly estimated parameters $\alpha = 45^\circ, \beta = 0^\circ$. These votes are much more concentrated than votes in Figure 5.b and 5.c.

Once $\hat{\alpha}$ and $\hat{\beta}$ have been estimated, the camera rotation axis $a$ is determined and it is possible to map the image $z$ to $M_{\alpha, \beta}(z)$. As said before, in $M_{\alpha, \beta}(z)$ the blurring paths are circunferences centered at $M_{\alpha, \beta}(C) = \pi_C \cap a$ and it is therefore possible to transform $M_{\alpha, \beta}(z)$ in polar coordinates for estimating the angular speed.

### 3.4 Angular Speed Estimation

Once $C$ has been determined, it is possible to transform $M_{\alpha, \beta}(z)$ (the image projected on $\pi_C$) on a polar
of planar tiles of test images. Blurred images are obtained by rotating the camera in front of the scene. Synthetic images of Figure 6 have been generated with a raytracer software (http://www.povray.org/). We estimate the blur size assuming a Gaussian PSF with a radius of 0.5 pixels around the center of the blurred image. The PSF extent, opportunely scaled to a restricted image area, avoiding lines which contain several pixels of padding introduced by the polar lattice resolution, di-vided by the exposure time gives the camera angular speed.

4 EXPERIMENTS

The algorithm has been validated both on synthetic and camera images. Synthetic images of Figure 6 have been generated with a raytracer software (http://www.povray.org/) rotating the camera in front of planar tiles of test images. Blurred images are obtained averaging all the rendered frames, according to Equation (1). Ten frames (512x512 pixels, grayscale 0-255) are rendered per each rotation degree. The blurring paths tangent directions are estimated in 121 equally spaced regions having a 10 pixel radius, using formula (5).

Table 1 shows \( \psi^\alpha \beta (\hat{\mathbf{p}}_{\alpha \beta}) \) (the value of the maximum vote obtained with \((\alpha, \beta)\)) as a percentage w.r.t \( \psi^\alpha \beta (\mathbf{p}_{0,0}) \) (the maximum vote obtained with \((\hat{\alpha}, \hat{\beta})\)). Here \((\hat{\alpha}, \hat{\beta})\) coincides with \((\alpha^*, \beta^*)\), the ground truth. Table 2 shows the results at a second iteration considering a refinement around \((\hat{\alpha}, \hat{\beta})\).

Table 3 shows results obtained on synthetic images of Figure 6. Each of them has been tested adding white gaussian noise with standard deviation 0, 0.5 and 1 and considering \(\alpha\) and \(\beta\) in \(A = B = \{-40^\circ, -20^\circ, 0^\circ, 20^\circ, 40^\circ\}\). Algorithm performances are evaluated with the following metrics:

\[
\Delta(\hat{\alpha}|\alpha) = |\hat{\alpha} - \alpha^*|, \quad \Delta(\hat{\beta}|\beta) = |\hat{\beta} - \beta^*|.
\]

\(\Delta(C)\) and \(\Delta(\hat{\omega})\) represent the absolute error between the ground truth and the estimated values of \(C = \pi/\alpha\) and \(\omega\), respectively.
Figure 7: Boat, Mandrill and Lena rectified with the corresponding estimated \((\hat{\alpha}, \hat{\beta})\). Intersection between the image plane and the rotation axis is marked with a red circle.

The effectiveness of our algorithm is evaluated as

\[
adv = \frac{\gamma^{\hat{\alpha}, \hat{\beta}}(\hat{\theta}_{\alpha, \beta}) - \gamma^{\alpha_*, \beta_*}(\hat{\theta}_{\alpha, \beta})}{\gamma^{\alpha_*, \beta_*}(\hat{\theta}_{\alpha, \beta})},
\]

being \(\gamma^{\alpha_*, \beta_*}(\hat{\theta}_{\alpha, \beta})\) the maximum vote obtained among other parameters \((\alpha, \beta)\). The higher this ratio, the better. Finally, \(\Delta(\alpha, \beta)\) and \(\Delta'(\alpha, \beta)\) are the corresponding errors obtained assuming circular blurring paths. Results for noisy images represent the average over ten different noise realizations.

Results reported in Table 3 show that our algorithm can cope with a reasonable amount of noise, obtaining regularly better results than the circular blur assumption. This is more evident in the estimation of the angular speed, which lacks physical meaning when the rotation axis is not correctly identified. Figure 7 shows blurred images of Figure 6 transformed with the corresponding \(M_{\alpha, \beta}\).

Camera images have been captured rotating a Canon EOS 400D camera on a tripod, assuring that \(a\) is orthogonal to the floor. The ground truth \(\alpha^*\) and \(\beta^*\), can be then computed rectifying still images of a checkerboard on the floor. Figures 8.a and 10.a show the downsampled RAW converted in grayscale used to test our algorithm. The blurring path tangent directions are estimated on 187 uniformly spaced regions, having 10 pixel radius.

Tables 4 and 5 show the results of the execution of two iterations of the algorithm on Figure 8.a. The solution obtained is \(\hat{\alpha} = -30^\circ\) and \(\hat{\beta} = 0^\circ\), which is acceptable as the ground truth, obtained from the checkerboard, is \((-27^\circ, 0^\circ)\). Figure 9 points out the differences between the blurring paths estimated with the circular approximation (in red) and the conic section paths estimated by our method (in green). As clearly seen from the detail, the blur is correctly interpreted by the green blurring paths. Figure 10 shows results on another camera image, having \(\alpha^* = -20^\circ\) and \(\beta^* = 0^\circ\). After two iterations, the algorithm converges exactly to the correct solution. Figures 8 and 10 show the blurred images and the checkerboard images rectified with the estimated \((\hat{\alpha}, \hat{\beta})\).

5 CONCLUSIONS

We described a novel algorithm for estimating the camera rotation axis and the angular speed from a single blurred image. The algorithm provides accurate...
Table 3: Algorithm performances on synthetic images. When $\sigma_\eta > 0$, averages over 10 noise realizations.

<table>
<thead>
<tr>
<th>Image</th>
<th>$\sigma_\eta$</th>
<th>$\Delta(\alpha)$</th>
<th>$\Delta(\beta)$</th>
<th>$\Delta(C)$</th>
<th>$\Delta(\hat{C})$</th>
<th>$\Delta(\hat{\omega})$</th>
<th>$\Delta(C^{\text{adv}})$</th>
<th>$\Delta(\hat{C}^{\text{adv}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boat</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.20</td>
<td>0.23</td>
<td>0.24</td>
<td>20.44</td>
<td>33.06</td>
</tr>
<tr>
<td>Boat</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>5.46</td>
<td>0.24</td>
<td>0.24</td>
<td>20.23</td>
<td>21.27</td>
</tr>
<tr>
<td>Boat</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8.84</td>
<td>0.19</td>
<td>0.19</td>
<td>8.84</td>
<td>19.25</td>
</tr>
<tr>
<td>Mandrill</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
<td>0.09</td>
<td>0.09</td>
<td>5.66</td>
<td>7.07</td>
</tr>
<tr>
<td>Mandrill</td>
<td>0.5</td>
<td>2</td>
<td>2</td>
<td>1.48</td>
<td>0.11</td>
<td>0.11</td>
<td>6.13</td>
<td>4.31</td>
</tr>
<tr>
<td>Mandrill</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1.17</td>
<td>0.26</td>
<td>0.26</td>
<td>5.26</td>
<td>4.41</td>
</tr>
<tr>
<td>Lena</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.00</td>
<td>0.08</td>
<td>0.08</td>
<td>11.01</td>
<td>12.08</td>
</tr>
<tr>
<td>Lena</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>3.88</td>
<td>0.20</td>
<td>0.20</td>
<td>14.06</td>
<td>33.64</td>
</tr>
<tr>
<td>Lena</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>5.23</td>
<td>0.48</td>
<td>0.48</td>
<td>6.00</td>
<td>29.43</td>
</tr>
</tbody>
</table>

Table 4: Camera Image. Highest votes corresponding to $(\alpha, \beta)$ in the parameters space, expressed as a percentage with respect to the maximum vote.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>-40</th>
<th>-20</th>
<th>0</th>
<th>20</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>-40</td>
<td>63</td>
<td>88</td>
<td>100</td>
<td>88</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>-20</td>
<td>62</td>
<td>77</td>
<td>78</td>
<td>70</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>67</td>
<td>74</td>
<td>80</td>
<td>70</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>62</td>
<td>81</td>
<td>85</td>
<td>65</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>71</td>
<td>73</td>
<td>82</td>
<td>78</td>
<td>77</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Camera Image. Refinement around $(\hat{\alpha}, \hat{\beta})$ from Table 4.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>-10</th>
<th>0</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>-50</td>
<td>82</td>
<td>76</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>-40</td>
<td>81</td>
<td>96</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>-30</td>
<td>85</td>
<td>100</td>
<td>83</td>
<td></td>
</tr>
</tbody>
</table>

estimates also in the most challenging cases, when the rotation axis is not orthogonal to the image plane. To the best of the authors’ knowledge, none of the existing methods handles these cases correctly since said methods assume circular blurring paths. We have shown how this assumption produces inaccurate estimates when the rotation axis is not orthogonal to the image plane, while our algorithm is more accurate.

The algorithm is targeted to accuracy rather than efficiency. Accuracy in the estimation of these parameters is a primary issue in restoring such images as the deblurring is typically based on a coordinate transformation and a deconvolution, which are highly sensitive to errors.

Ongoing works concern the design of a more noise-robust method for blur analysis on image regions and the implementation of a faster voting procedure.

REFERENCES


http://www.povray.org/.


