Estimating Camera Rotation Parameters from a Single Blurred Image

Giacomo BORACCHI
Vincenzo CAGLIOTI
Alberto DANESE

Dipartimento di Elettronica e Informazione
Politecnico di Milano

VISAPP 2008 3rd International Conference on Computer Vision Theory and Applications,

Rotational Blur: An Example

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera
- The blur is completely described by

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera

- The blur is completely described by
  - the **camera rotation axis**

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera

- The blur is completely described by
  - the **camera rotation axis**
  - the **angular speed**

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera

- The blur is completely described by
  - the **camera rotation axis**
  - the **angular speed**

- Dealing with such an images is not straightforward

Rotational Blur

- **Rotational blur** affects images acquired during a fast rotation of the camera

- The blur is completely described by
  - the **camera rotation axis**
  - the **angular speed**

- Dealing with such an images is not straightforward
  - Most of blur analysis and image restoration techniques are meant for spatially **invariant** blur while blur due to rotation is spatially **variant**

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.
- The algorithm is able to cope with blurs produced by a **generic** camera **rotation** (assuming the rotation axis through the viewpoint).

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.
- The algorithm is able to cope with blurs produced by a **generic** camera **rotation** (assuming the rotation axis through the viewpoint).
- The proposed algorithm is targeted to accuracy rather than computational efficiency

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.
- The algorithm is able to cope with blurs produced by a **generic** camera **rotation** (assuming the rotation axis through the viewpoint).
- The proposed algorithm is targeted to accuracy rather than computational efficiency
- **Accurate** estimates:

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.
- The algorithm is able to cope with blurs produced by a **generic** camera **rotation** (assuming the rotation axis through the viewpoint).
- The proposed algorithm is targeted to accuracy rather than computational efficiency
- **Accurate** estimates:
  - are essential for **deblurring**

Contribution

- We propose an algorithm for **estimating** the camera **rotation axis** and the **angular speed**, from a single blurred image.
- The algorithm is able to cope with blurs produced by a **generic** camera **rotation** (assuming the rotation axis through the viewpoint).
- The proposed algorithm is targeted to accuracy rather than computational efficiency
- **Accurate** estimates:
  - are essential for **deblurring**
  - can be used to infer the **camera egomotion**

Presentation Outline

- Related Works
- Image Formation Model
- The Algorithm
- Experiments
- Conclusions

Early Works

- Early works on roationally blurred images concern image restoration (Ribaric *et al* [2000])

Early Works

- Early works on roationally blurred images concern image restoration (Ribaric *et al* [2000])

- The key idea is that the blur in polar coordinates becomes shift invariant

Early Works

- Early works on roationally blurred images concern image restoration (Ribaric *et al* [2000])

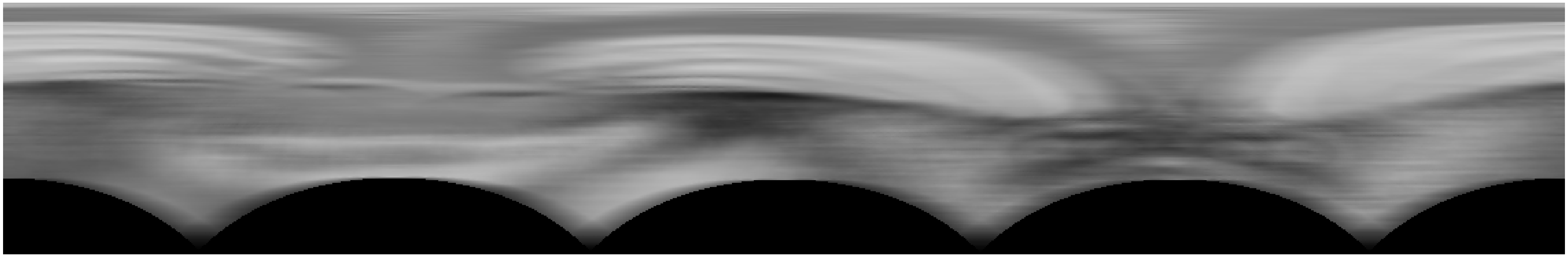
- The key idea is that the blur in polar coordinates becomes shift invariant

Space Variant Blur

Early Works

- Early works on roationally blurred images concern image restoration (Ribaric *et al* [2000])

- The key idea is that the blur in polar coordinates becomes shift invariant



Space Invariant Blur

Early Works

- This transform is possible only when the **rotation axis** is **orthogonal** to the image plane and the **intersection** between the axis and the image plane is **known**.

Early Works

- This transform is possible only when the **rotation axis** is **orthogonal** to the image plane and the **intersection** between the axis and the image plane is **known**.

- When the rotation axis is **not orthogonal** to the image plane, the blur on the image in polar coordinates **is not space invariant**.

Early Works

- This transform is possible only when the **rotation axis** is **orthogonal** to the image plane and the **intersection** between the axis and the image plane is **known**.

- When the rotation axis is **not orthogonal** to the image plane, the blur on the image in polar coordinates **is not space invariant**.

- In this case, it is **essential** to consider the **angle** between the **rotation axis** and the **image plane**.

Recent Works

- Klein and Drummond [2005]
  - Devised an algorithm for estimating the camera ego-motion from a rotationally blurred image.
  - The algorithm is meant as a visual gyroscope.
  - Targeted to efficiency rather than accuracy.
  - Handles correctly only rotation axis orthogonal to image plane.

Recent Works

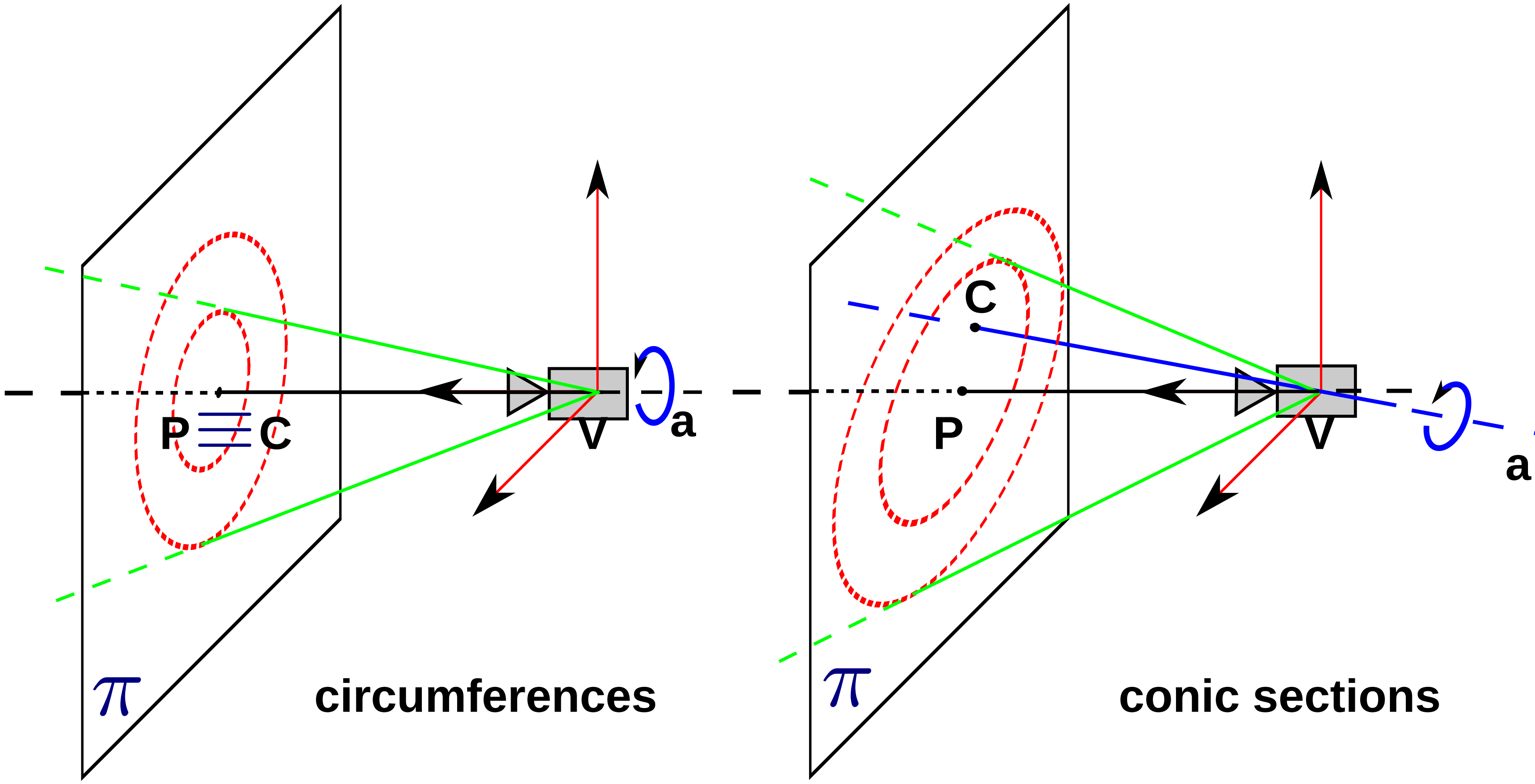
- Klein and Drummond [2005]
  - Devised an algorithm for estimating the camera ego-motion from a rotationally blurred image.
  - The algorithm is meant as a visual gyroscope.
  - Targeted to efficiency rather than accuracy.
  - Handles correctly only rotation axis orthogonal to image plane.
- Shan *et al* [2007]
  - Algorithm for both estimating rotation parameters and restoring a single rotational blurred image.
  - Assumes rotation axis orthogonal to the image plane and takes into account image translation also.
  - Exploits alpha matting techniques.
  - Relies on significant user interaction.

Presentation Outline

- Related Works
- Image Formation Model
- The Algorithm
- Experiments
- Conclusions

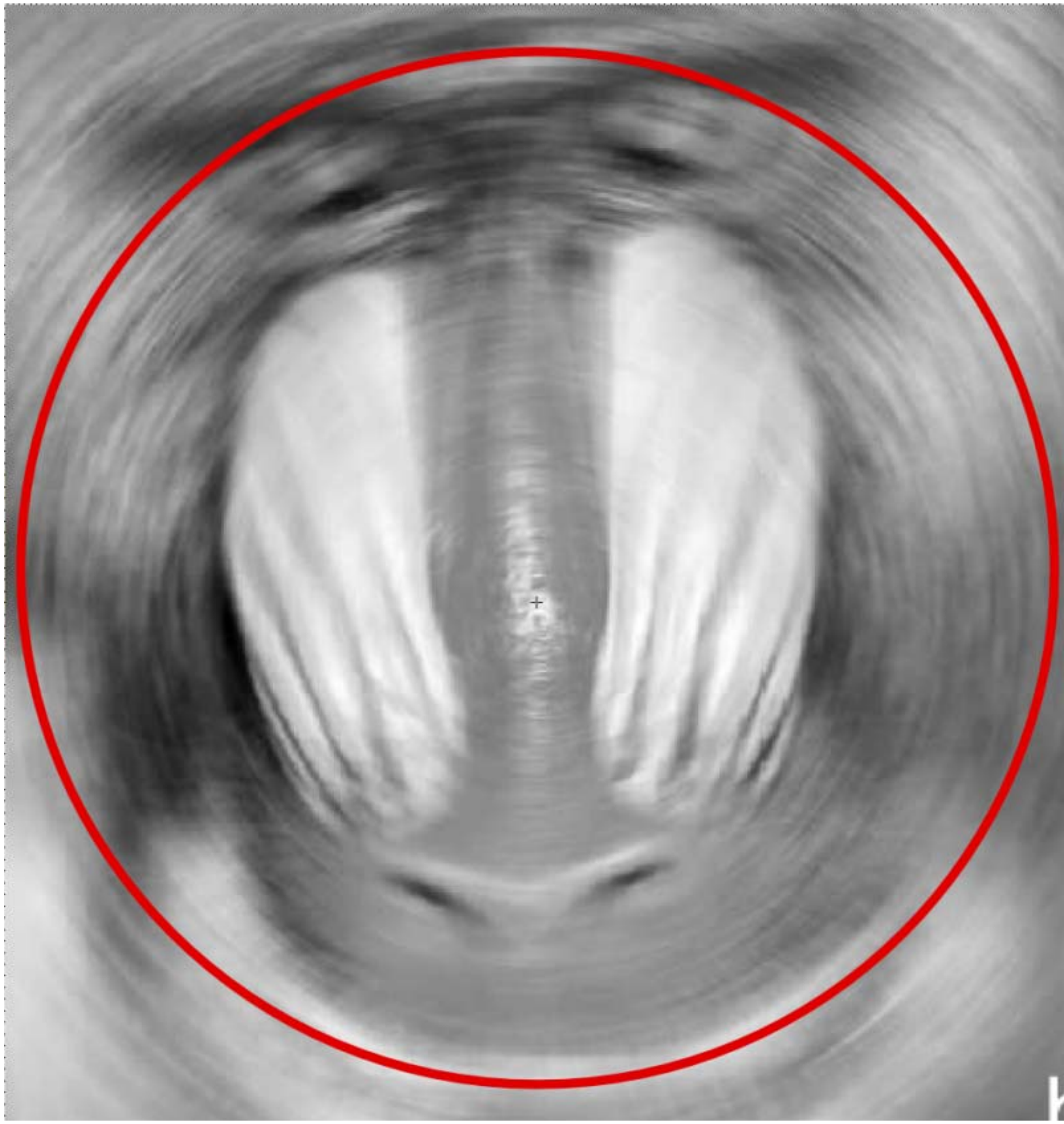
Blurring Path: definition

- A ***blurring path*** is defined as the set of image pixels that a viewing ray intersects during a camera rotation of 2π around the rotation axis ***a***



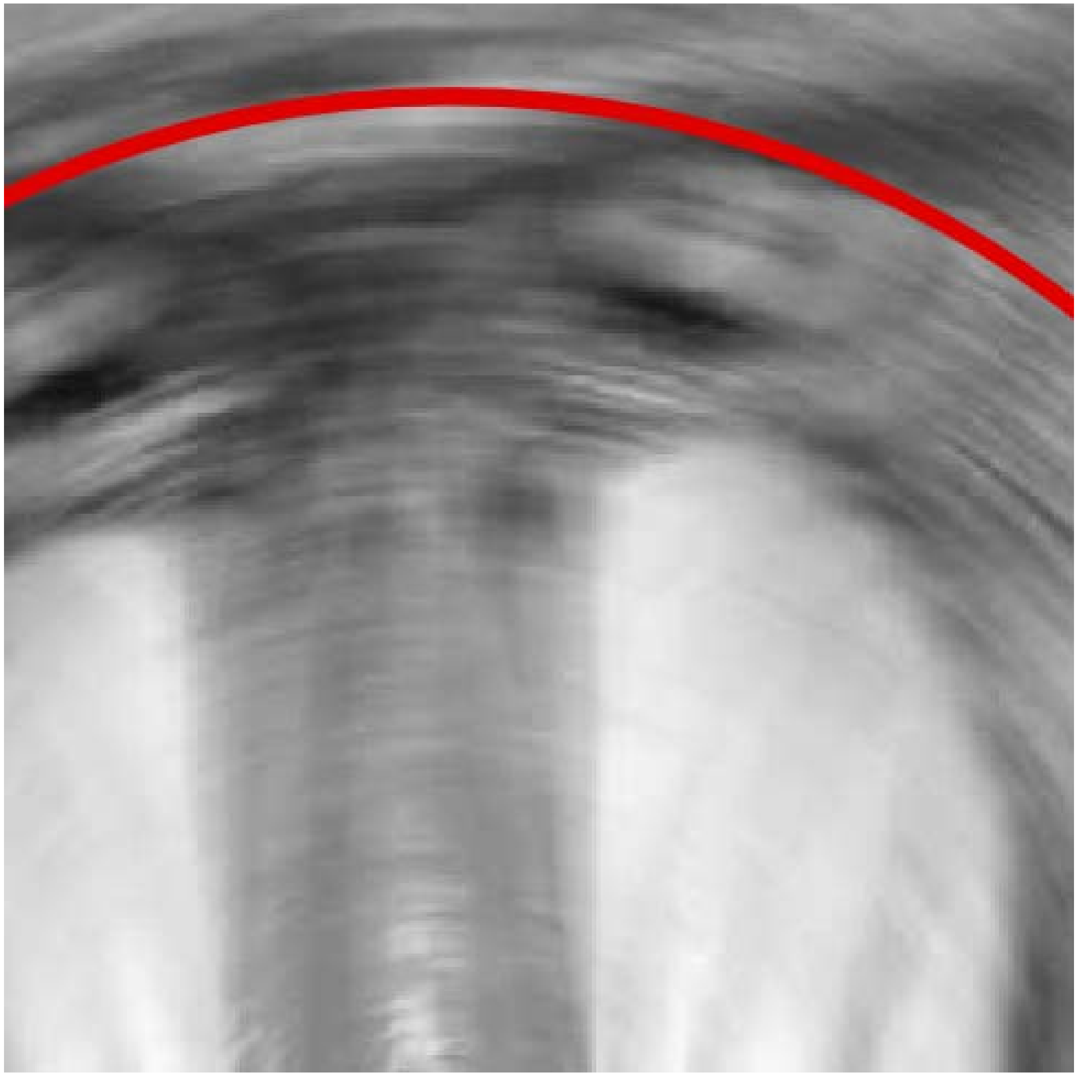
Blurring Path: an Illustrative Example

- Example of blurring paths obtained with a rotation orthogonal to the image plane



Blurring Path: an Illustrative Example

- Example of blurring paths obtained with a rotation orthogonal to the image plane



Blurring Path: an Illustrative Example

- Example of blurring paths obtained with a rotation orthogonal to the image plane

- General case, blurring paths are conic sections

Blurring Path: an Illustrative Example

- Example of blurring paths obtained with a rotation orthogonal to the image plane

- General case, blurring paths are conic sections

Rotational blur – Image Formation Model

- An image *z* degraded by space-variant blur is modelled as follows

$$z(\mathbf{x}) = \mathcal{K}\big(y\big)(\mathbf{x}) + \eta(\mathbf{x}) \qquad \mathbf{x} = (x, y) \in X$$

$$\mathcal{K}\big(y\big)(\mathbf{x}) = \int_X k(\mathbf{x}, s) y(s) ds$$

where $\mathcal{K}\big(y\big)$ represents the blurred and noise-free image, X the image domain and *k*(**x**,*s*) the PSF at a pixel **x**

Rotational blur – Image Formation Model

- An image *z* degraded by space-variant blur is modelled as follows

$$z(\mathbf{x}) = \mathcal{K}\big(y\big)(\mathbf{x}) + \eta(\mathbf{x}) \qquad \mathbf{x} = (x_1, x_2) \in X$$

$$\mathcal{K}\big(y\big)(\mathbf{x}) = \int_X k(\mathbf{x}, s) y(s) ds$$

where $\mathcal{K}\big(y\big)$ represents the blurred and noise-free image, X the image domain and *k*(**x**,*s*) the PSF at a pixel **x**

- For rotational blur:

$$k(\mathbf{x}, \bullet) = A_{\theta,e}(\bullet)$$

where $A_{\theta,e}$ represents an arc of conic section having tangent direction θ and extent *e*

Rotational blur – Image Formation Model

- Rotational Blur is thus:
  - ***Space-variant,*** as the Point Spread Functions are varying through the image plane
  - ***Parametric*** as these Point Spread Functions can be expressed as a function of the rotation axis and the angular speed of the camera.

Rotational blur – Image Formation Model

- Rotational Blur is thus:
  - ***Space-variant,*** as the Point Spread Functions are varying through the image plane
  - ***Parametric*** as these Point Spread Functions can be expressed as a function of the rotation axis and the angular speed of the camera.

- We are not interested in estimating the Point Spread Function at each image pixel, we estimate the rotation axis and the angular speed

Presentation Outline

- Related Works
- Image Formation Model
- The Algorithm
- Experiments
- Conclusions

Algorithm - Overview

- The algorithm is based on three steps:
  1. Local estimates of the directions **tangent** to the blurring paths
  2. Voting procedure for estimating the **rotation axis**
  3. Estimation of the **angular speed**

Algorithm - Overview

- The algorithm is based on three steps:
  1. Local estimates of the directions **tangent** to the blurring paths
  2. Voting procedure for estimating the **rotation axis**
  3. Estimation of the **angular speed**

- Assumptions:
  1. The camera is calibrated
  2. The camera viewpoint lies on the rotation axis
  3. The angular speed is constant during exposure

Local estimates of blurring paths tangent directions

- The blur **correlates** the image along the blurring paths

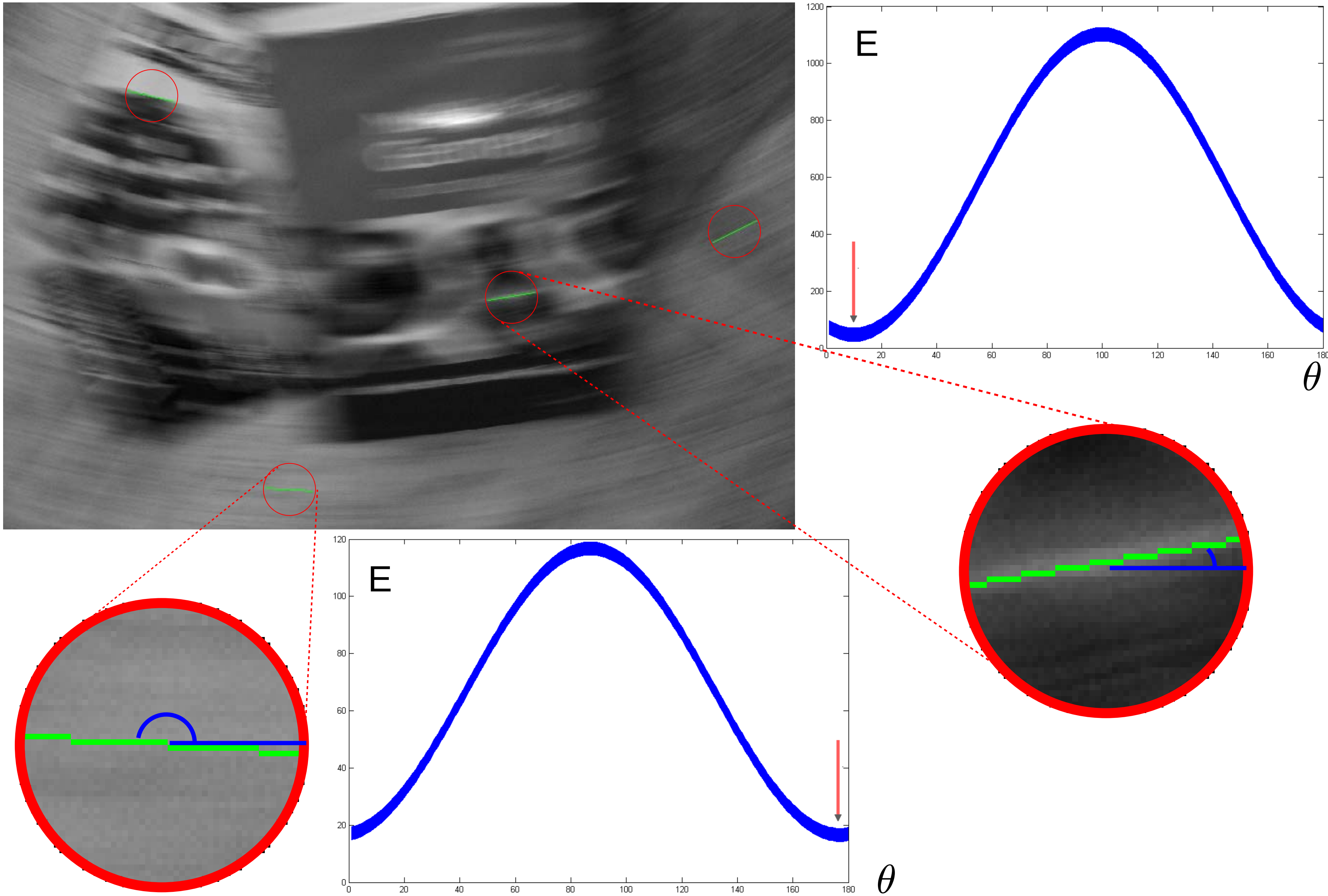
- Given U_i, a small image region, the response to a derivative filter d_θ has minimum energy when the derivative direction θ_i corresponds the blur tangent direction:

$$\theta_i = \arg\min_{\theta \in [0,\pi]} E(\theta_i) \qquad E(\theta_i) = \sum_{\mathbf{x}_j \in U_i} w_j \big((y \circledast d_\theta)(\mathbf{x}_j)\big)^2$$

where w_j represents a circular window with Gaussian weights and $\circledast$ the 2D convolution.

[Adaptation from Yitzhaky (1996)]

Local estimates of blurring paths tangent directions



Axis estimation – circular blurring paths

- Assume **circular** blurring paths centered in *C*, the intersection between the image plane and the rotation axis

Axis estimation – circular blurring paths

- Assume **circular** blurring paths centered in *C*, the intersection between the image plane and the rotation axis
- For each local estimate (data), the **possible** locations of *C* belong to the line orthogonal to the blurring path tangent

Axis estimation – circular blurring paths

- Assume **circular** blurring paths centered in *C*, the intersection between the image plane and the rotation axis
- For each local estimate (data), the **possible** locations of *C* belong to the line orthogonal to the blurring path tangent
- Following an Hough-like approach, a set of votes v_i is associated to each datum in a parameters space *P*

Axis estimation – circular blurring paths

- Assume **circular** blurring paths centered in *C*, the intersection between the image plane and the rotation axis
- For each local estimate (data), the **possible** locations of *C* belong to the line orthogonal to the blurring path tangent
- Following an Hough-like approach, a set of votes v_i is associated to each datum in a parameters space *P*
- After summing all the votes, *C* corrisponds to the parameter receiving the highest number of votes:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p} \in P} \mathcal{V}(\mathbf{p}), \;\; \mathcal{V}(\mathbf{p}) = \sum_{i=1}^{N} v_i(\mathbf{p})$$

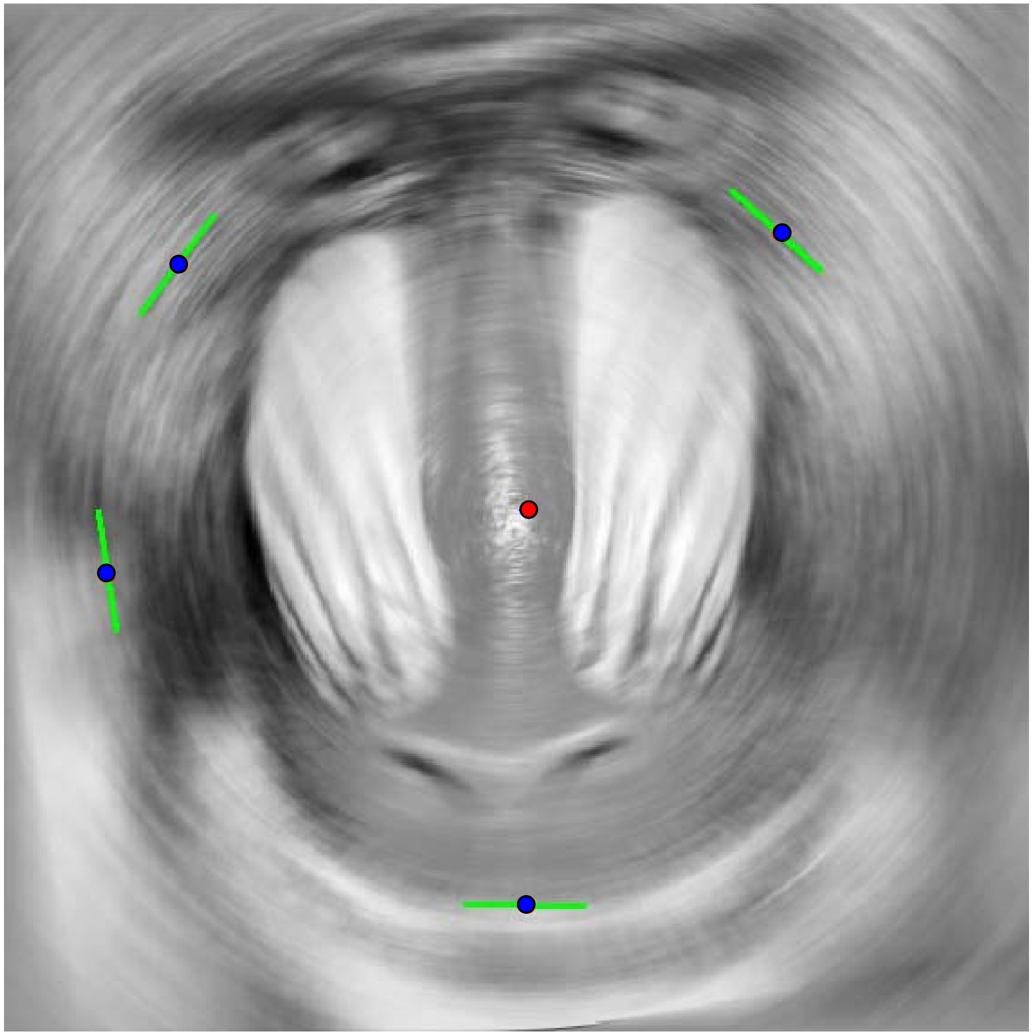
Axis estimation – circular blurring paths

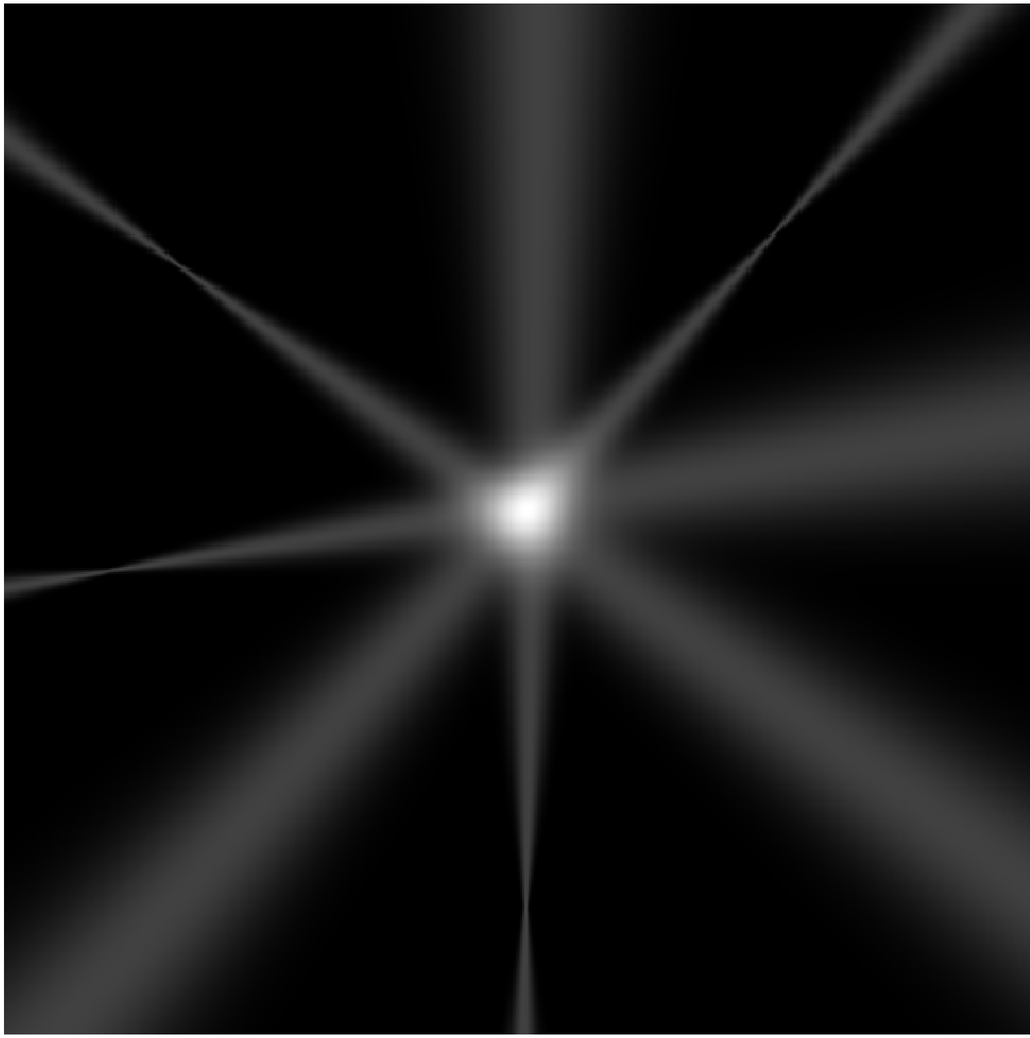
- Assume **circular** blurring paths centered in *C*, the intersection between the image plane and the rotation axis
- For each local estimate (data), the **possible** locations of *C* belong to the line orthogonal to the blurring path tangent
- Following an Hough-like approach, a set of votes v_i is associated to each datum in a parameters space *P*
- After summing all the votes, *C* corrisponds to the parameter receiving the highest number of votes:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}\in P} \mathcal{V}(\mathbf{p}), \;\; \mathcal{V}(\mathbf{p}) = \sum_{i=1}^{N} v_i(\mathbf{p})$$

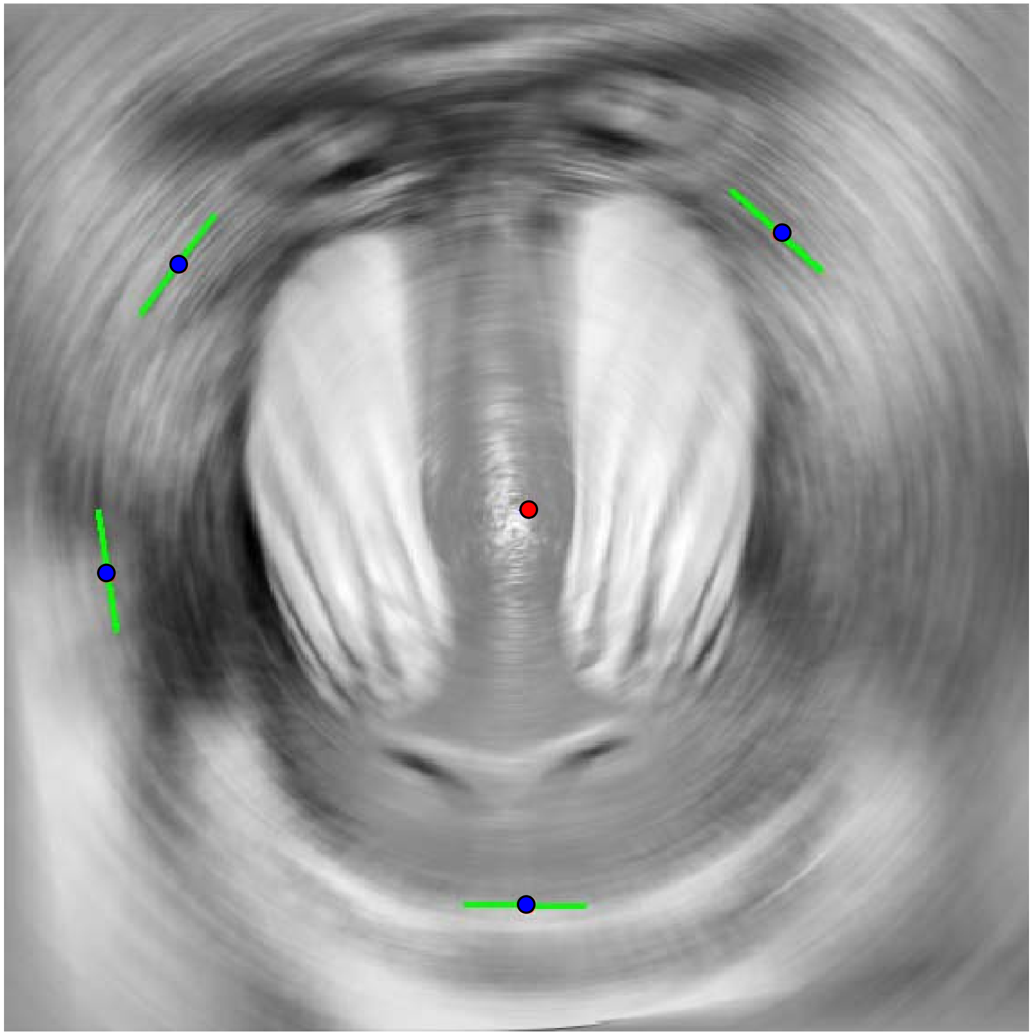
- The votes v_i are characterized by a Gaussian spread to consider the uncertainty in the blur tangent estimate

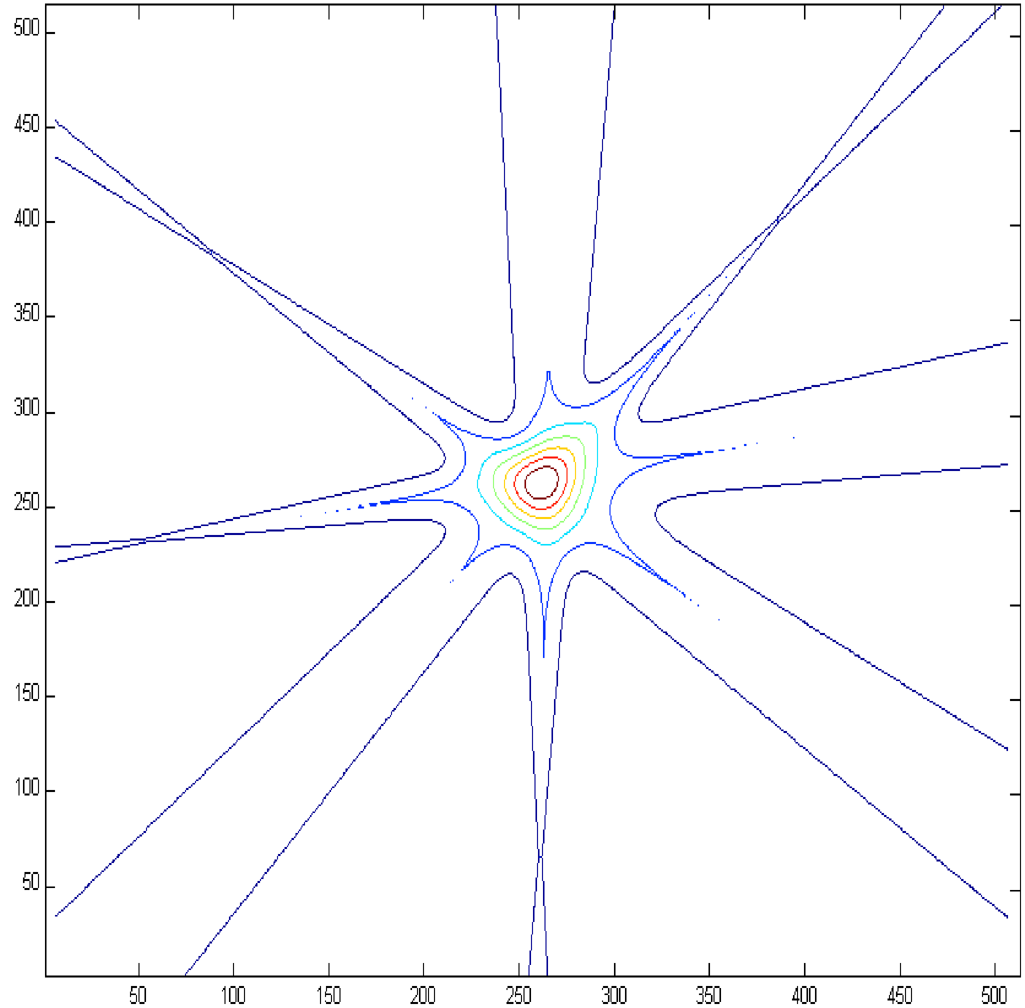
Axis estimation – circular blurring paths: an example





Axis estimation – circular blurring paths: an example

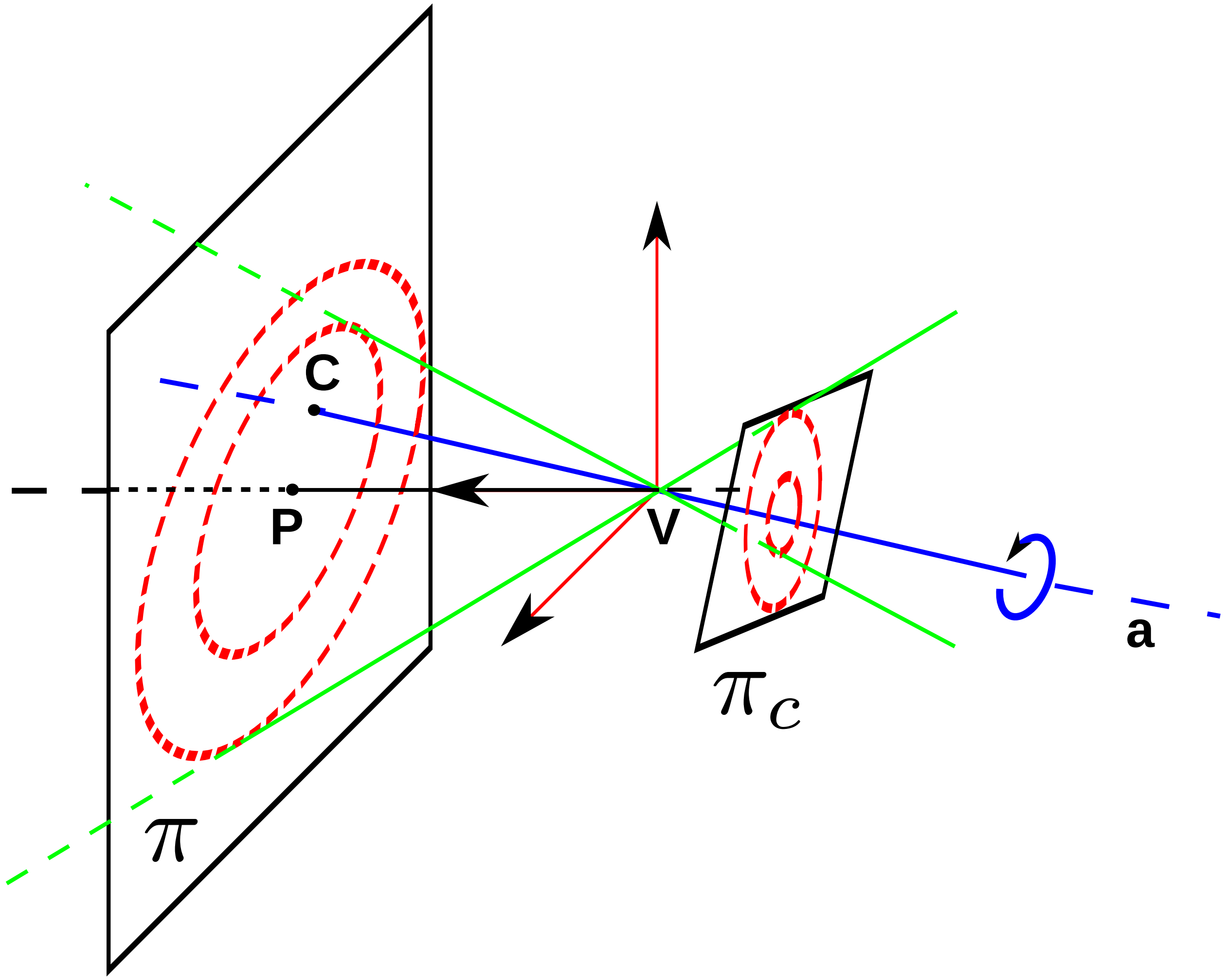




Axis estimation – conic blurring paths

- In the general case, the blurring paths are conic sections

- The projections of the blurring paths onto a plane orthogonal to the rotation axis are circumferences

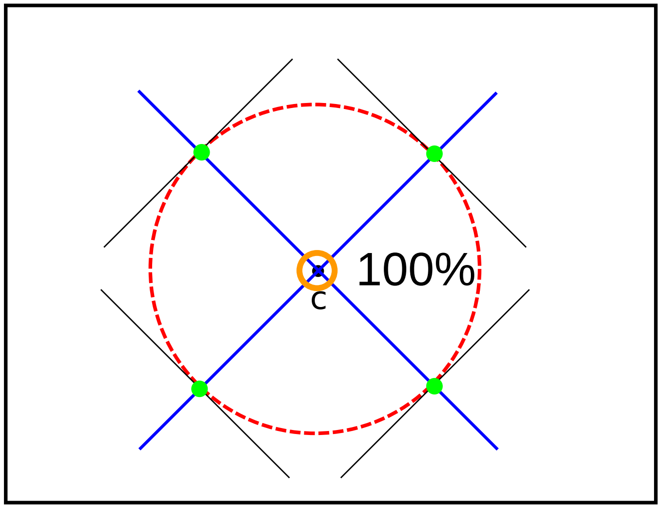


Axis estimation – conic blurring paths

- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the **ideal case all** the votes

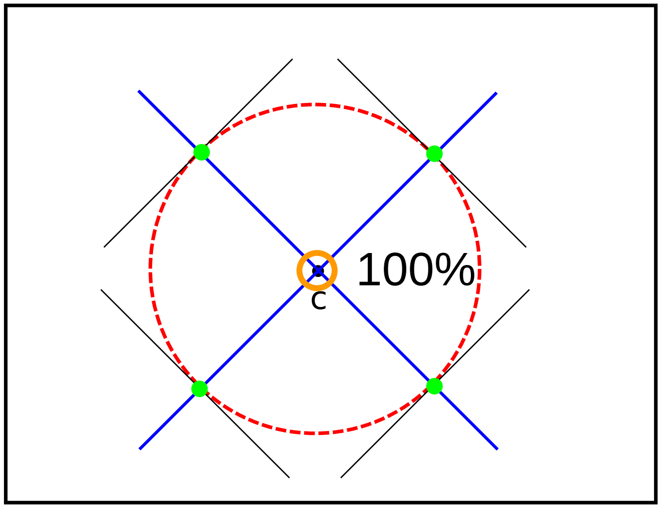
Axis estimation – conic blurring paths

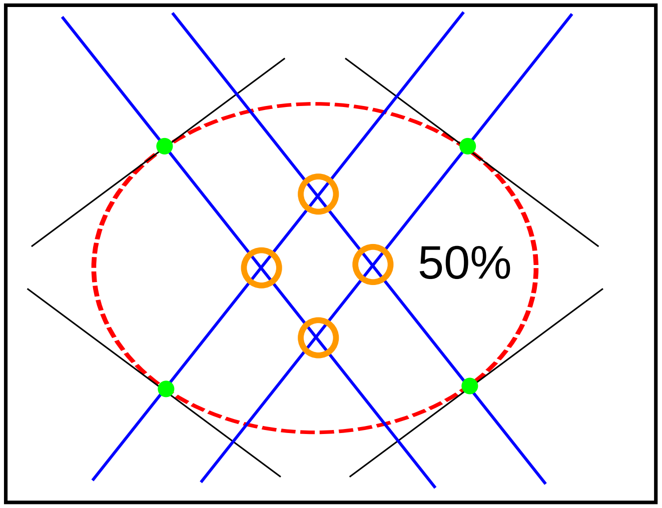
- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the **ideal case all** the votes



Axis estimation – conic blurring paths

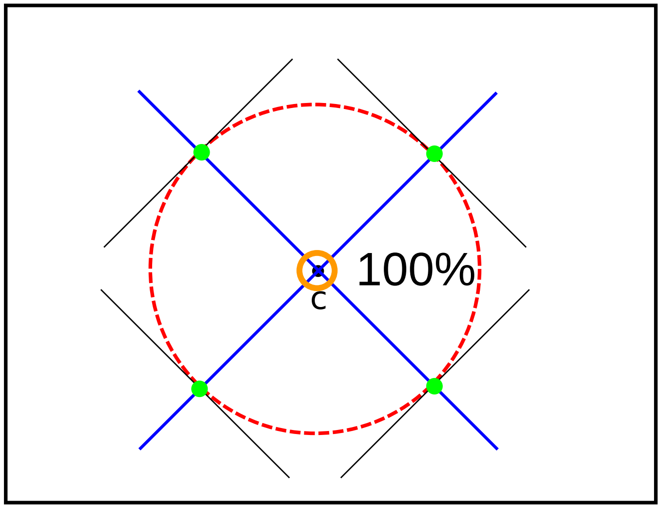
- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the **ideal case all** the votes

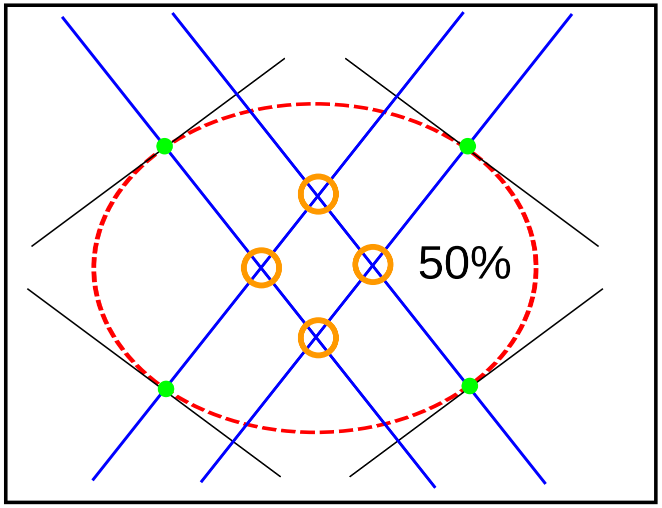




Axis estimation – conic blurring paths

- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the ideal case **all** the votes

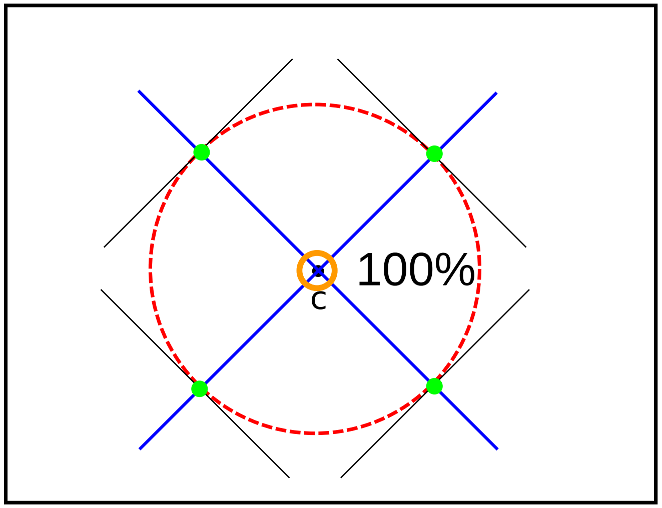


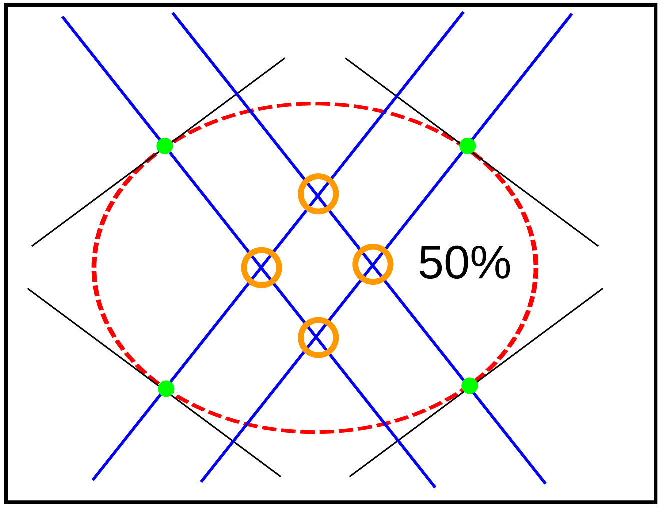


- This peculiarity is exploited for estimating the plane orthogonal to the rotation axis, among a set of candidates

Axis estimation – conic blurring paths

- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the **ideal case all** the votes

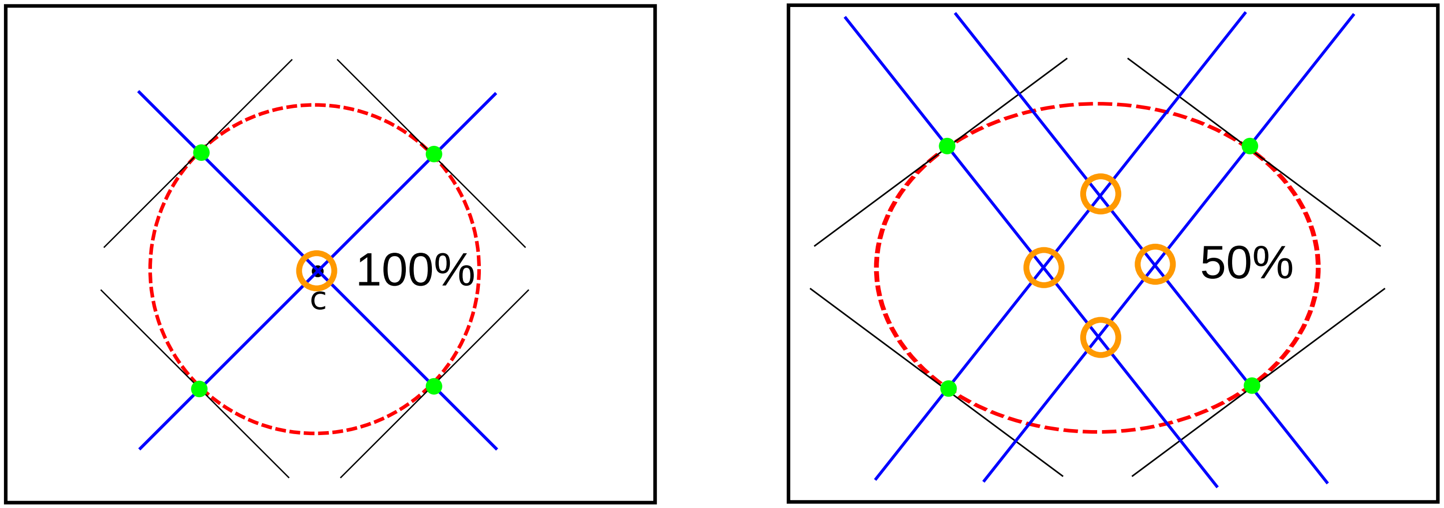




- This peculiarity is exploited for estimating the plane orthogonal to the rotation axis, among a set of candidates
- Each candidate plane is defined by two angles α and β

Axis estimation – conic blurring paths

- **Only** when the blurring paths are circular, the normals to the blur tangents cross in a **single** point (*C*), capable of collecting in the ideal case **all** the votes



- This peculiarity is exploited for estimating the plane orthogonal to the rotation axis, among a set of candidates
- Each candidate plane is defined by two angles α and β
- $M_{\alpha,\beta}$ defines the projection between π and $\pi_{\alpha,\beta}$

Axis estimation – conic blurring paths

- For each pair of parameters α, β we project the local estimates on the plane $\pi_{\alpha,\beta}$ using the transform $M_{\alpha,\beta}$

Axis estimation – conic blurring paths

- For each pair of parameters α, β we project the local estimates on the plane $\pi_{\alpha,\beta}$ using the transform $M_{\alpha,\beta}$
- We then compute $v_i^{\alpha,\beta}$, the votes corresponding to the *i-th* transformed estimate and thus

$$\mathcal{V}^{\alpha,\beta}(\mathbf{p}) = \sum_{i=1}^{N} v_i^{\alpha,\beta}(\mathbf{p})\,, \quad \hat{\mathbf{p}}_{\alpha,\beta} = \arg\max_{\mathbf{p}\in P} \mathcal{V}^{\alpha,\beta}(\mathbf{p})$$

Axis estimation – conic blurring paths

- For each pair of parameters α, β we project the local estimates on the plane $\pi_{\alpha,\beta}$ using the transform $M_{\alpha,\beta}$

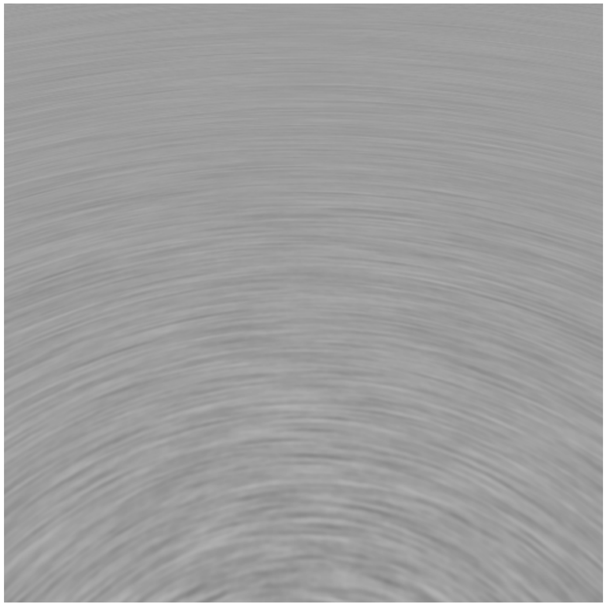
- We then compute $v_i^{\alpha,\beta}$, the votes corresponding to the *i-th* transformed estimate and thus

$$\mathcal{V}^{\alpha,\beta}(\mathbf{p}) = \sum_{i=1}^{N} v_i^{\alpha,\beta}(\mathbf{p})\,, \quad \hat{\mathbf{p}}_{\alpha,\beta} = \arg\max_{\mathbf{p}\in P} \mathcal{V}^{\alpha,\beta}(\mathbf{p})$$

- We compute the coordinates that reach the maximum of votes, similarly to the previous case

Axis estimation – conic blurring paths

- The value of the maximum votes corresponding to the **correct** α, β **parameters**, is **higher** than the others

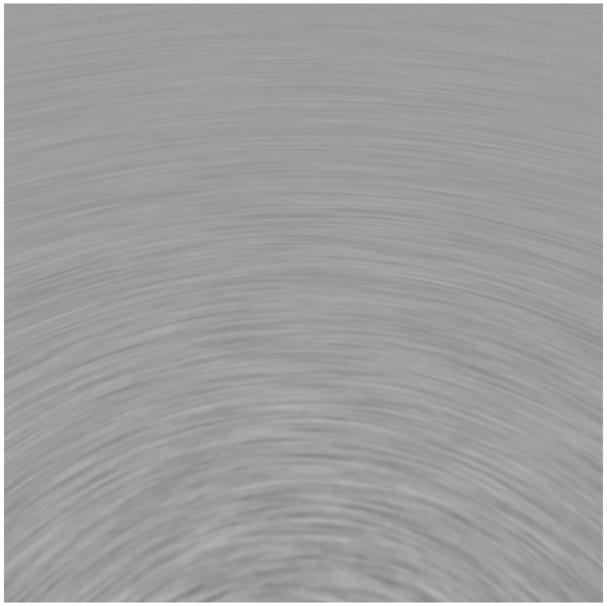


$(\alpha^* = 45°, \beta^* = 0°)$

Axis estimation – conic blurring paths

- The value of the maximum votes corresponding to the **correct** α, β **parameters**, is **higher** than the others

$(\alpha = 0°, \beta = 0°)$



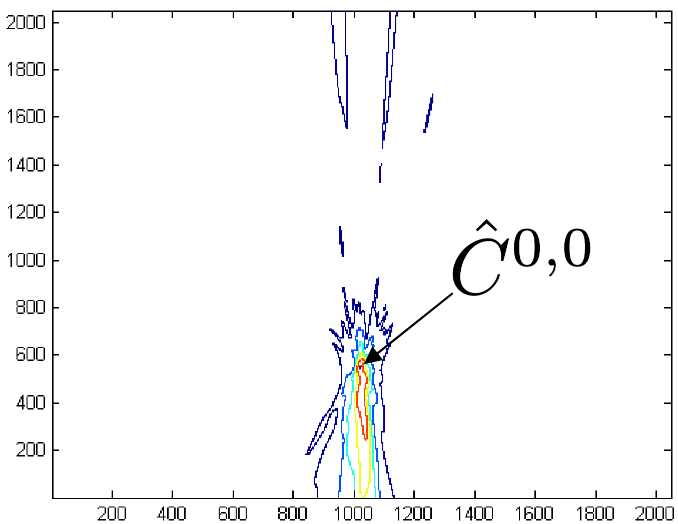
$(\alpha^* = 45°, \beta^* = 0°)$

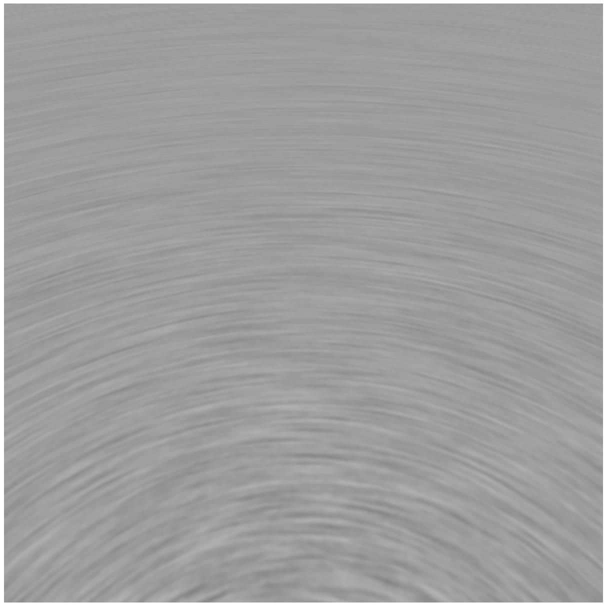
$(\alpha = 45°, \beta = 0°)$

Axis estimation – conic blurring paths

- The value of the maximum votes corresponding to the **correct** α, β **parameters**, is **higher** than the others

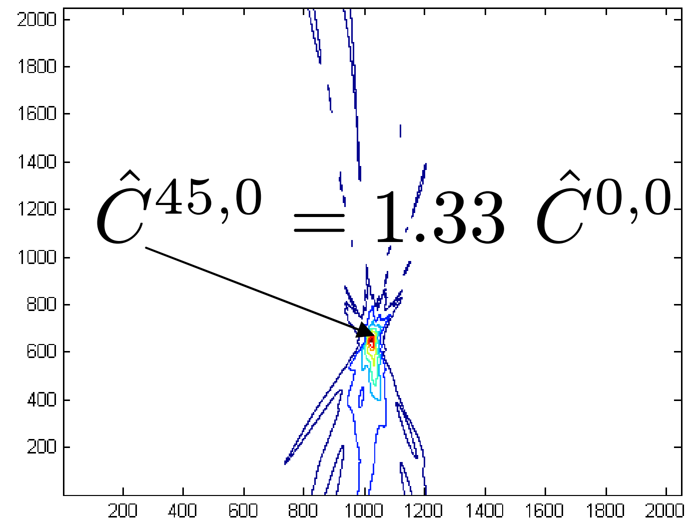
$(\alpha = 0°, \beta = 0°)$





$(\alpha^* = 45°, \beta^* = 0°)$

$(\alpha = 45°, \beta = 0°)$



Axis estimation – conic blurring paths

- Therefore the rotation axis is identified by $(\hat{\alpha}, \hat{\beta})$ satisfying the following relations

$$(\hat{\alpha}, \hat{\beta}) = \arg\max_{\alpha,\beta} \mathcal{V}^{\alpha,\beta}(\hat{\mathbf{p}}_{\alpha,\beta})$$

Angular speed estimation

- The image mapped by $M_{\hat{\alpha},\hat{\beta}}$ shows circular blurring paths

Angular speed estimation

- The image mapped by $M_{\hat{\alpha},\hat{\beta}}$ shows circular blurring paths
- In polar coordinates w.r.t. $M_{\hat{\alpha},\hat{\beta}}(\hat{C})$, the rotational blur becomes space-invariant, directed along lines $\rho = const$

Angular speed estimation

- The image mapped by $M_{\hat{\alpha},\hat{\beta}}$ shows circular blurring paths
- In polar coordinates w.r.t. $M_{\hat{\alpha},\hat{\beta}}(\hat{C})$, the rotational blur becomes space-invariant, directed along lines $\rho = const$
- The angular speed is thus proportional to the blur extent

Angular speed estimation

- The image mapped by $M_{\hat{\alpha},\hat{\beta}}$ shows circular blurring paths
- In polar coordinates w.r.t. $M_{\hat{\alpha},\hat{\beta}}(\hat{C})$, the rotational blur becomes space-invariant, directed along lines $\rho = const$
- The angular speed is thus proportional to the blur extent

- Several methods for space-invariant motion blur estimation can thus be employed.

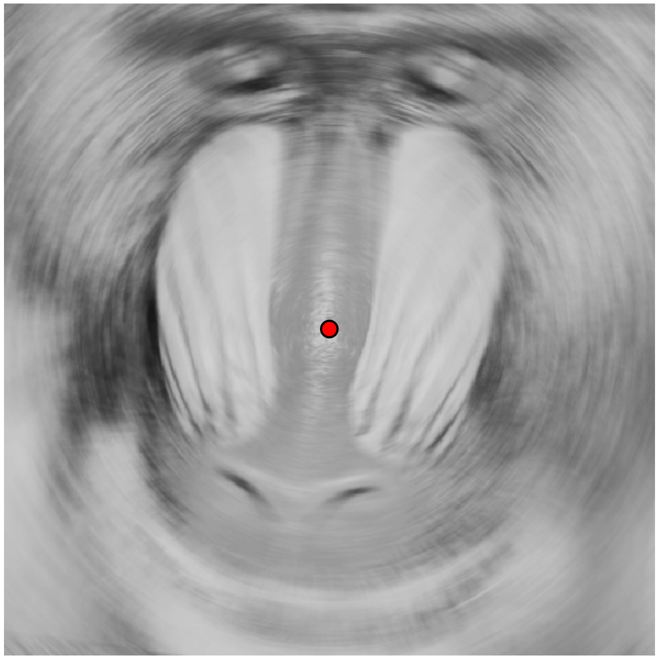
Angular speed estimation

- The image mapped by $M_{\hat{\alpha},\hat{\beta}}$ shows circular blurring paths
- In polar coordinates w.r.t. $M_{\hat{\alpha},\hat{\beta}}(\hat{C})$, the rotational blur becomes space-invariant, directed along lines $\rho = const$
- The angular speed is thus proportional to the blur extent

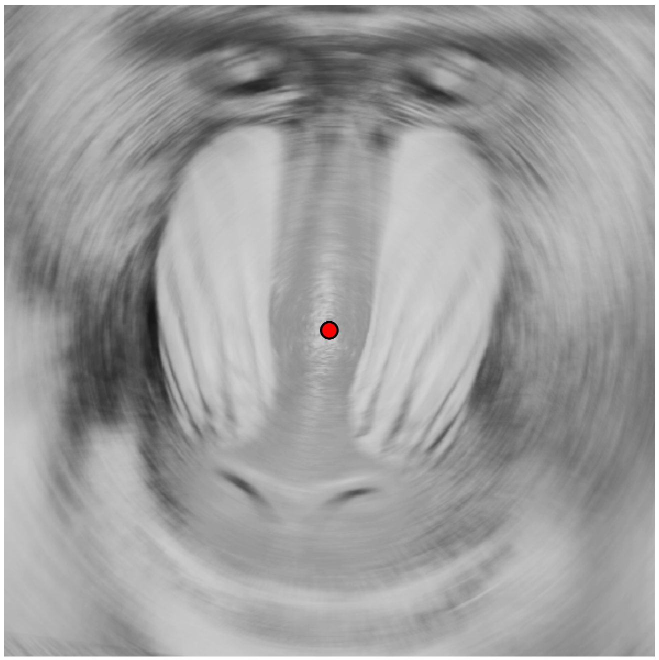
- Several methods for space-invariant motion blur estimation can thus be employed.
- An effective algorithm is based on the analysis of the autocorrelation of the derivatives along the blur direction [Yitzhaky (1996)]

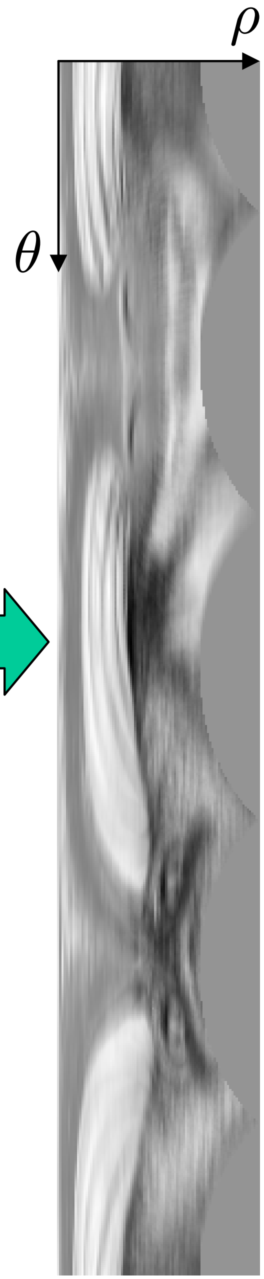
Angular speed estimation: an example

- Given a rotational blurred image we estimate the rotation axis and its intersection with the image plane.



Angular speed estimation: an example

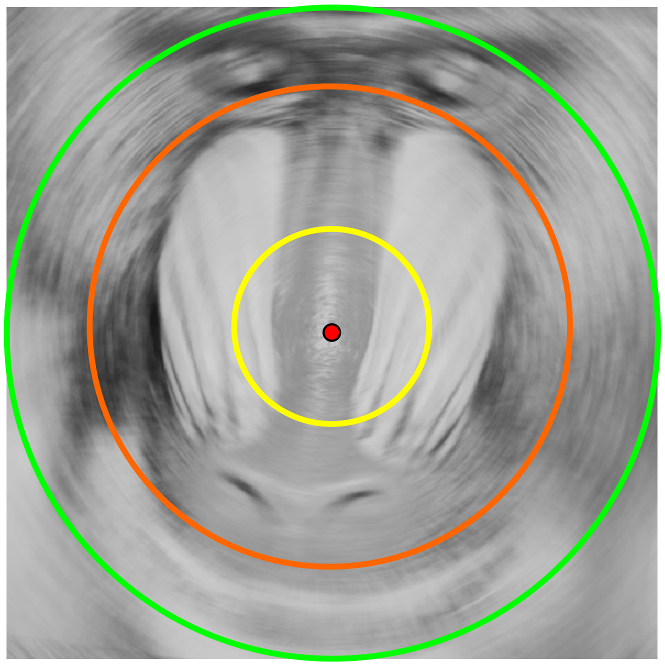


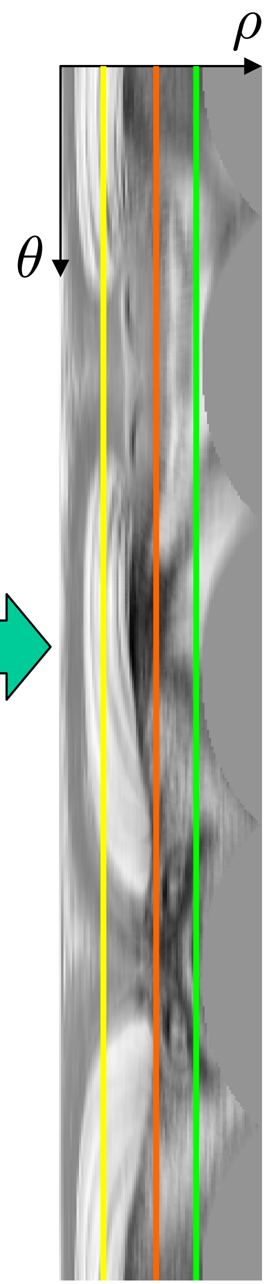


- We transform the image with $M_{\hat{\alpha},\hat{\beta}}$, so that the blurring paths becomes circumferences.

- The transformed image is mapped in polar coordinates

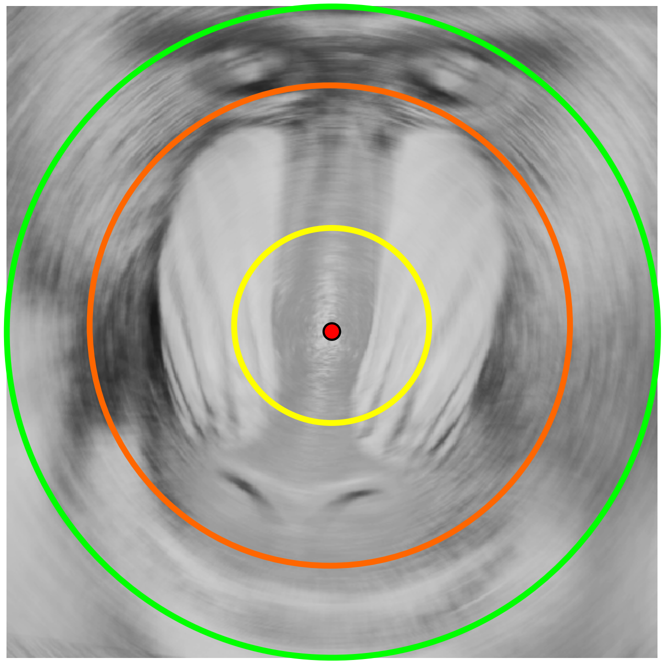
Angular speed estimation: an example

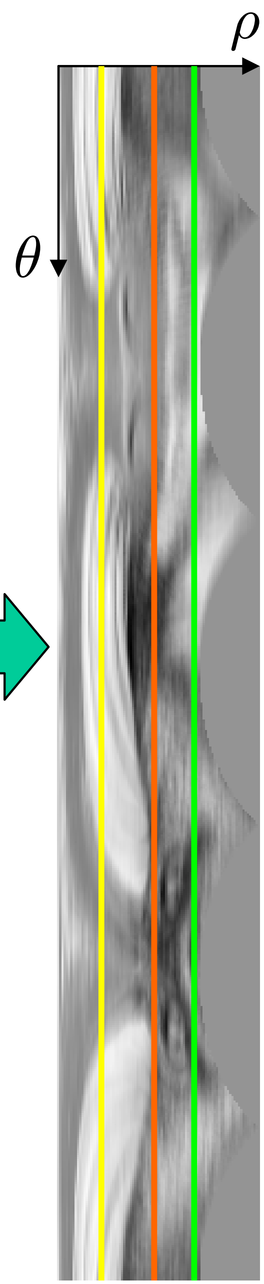


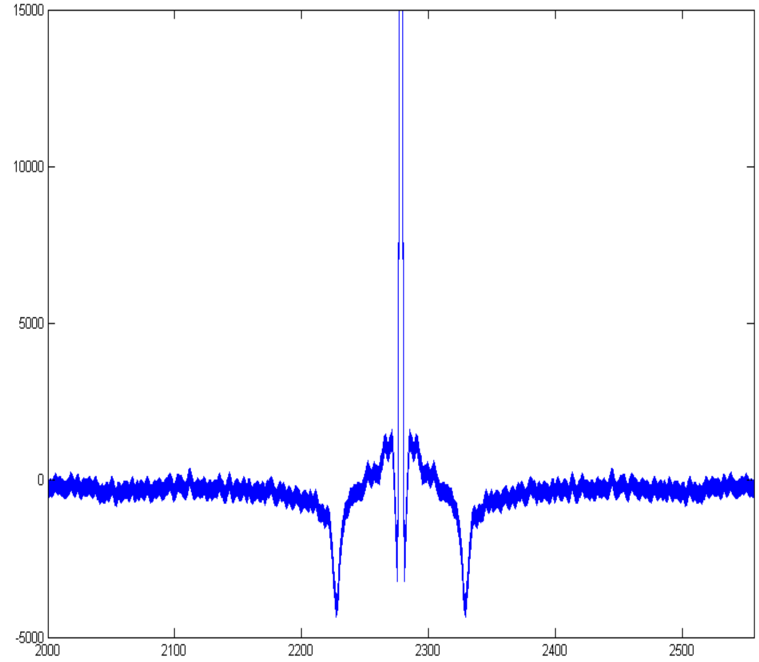


- We select a padding free area

Angular speed estimation: an example

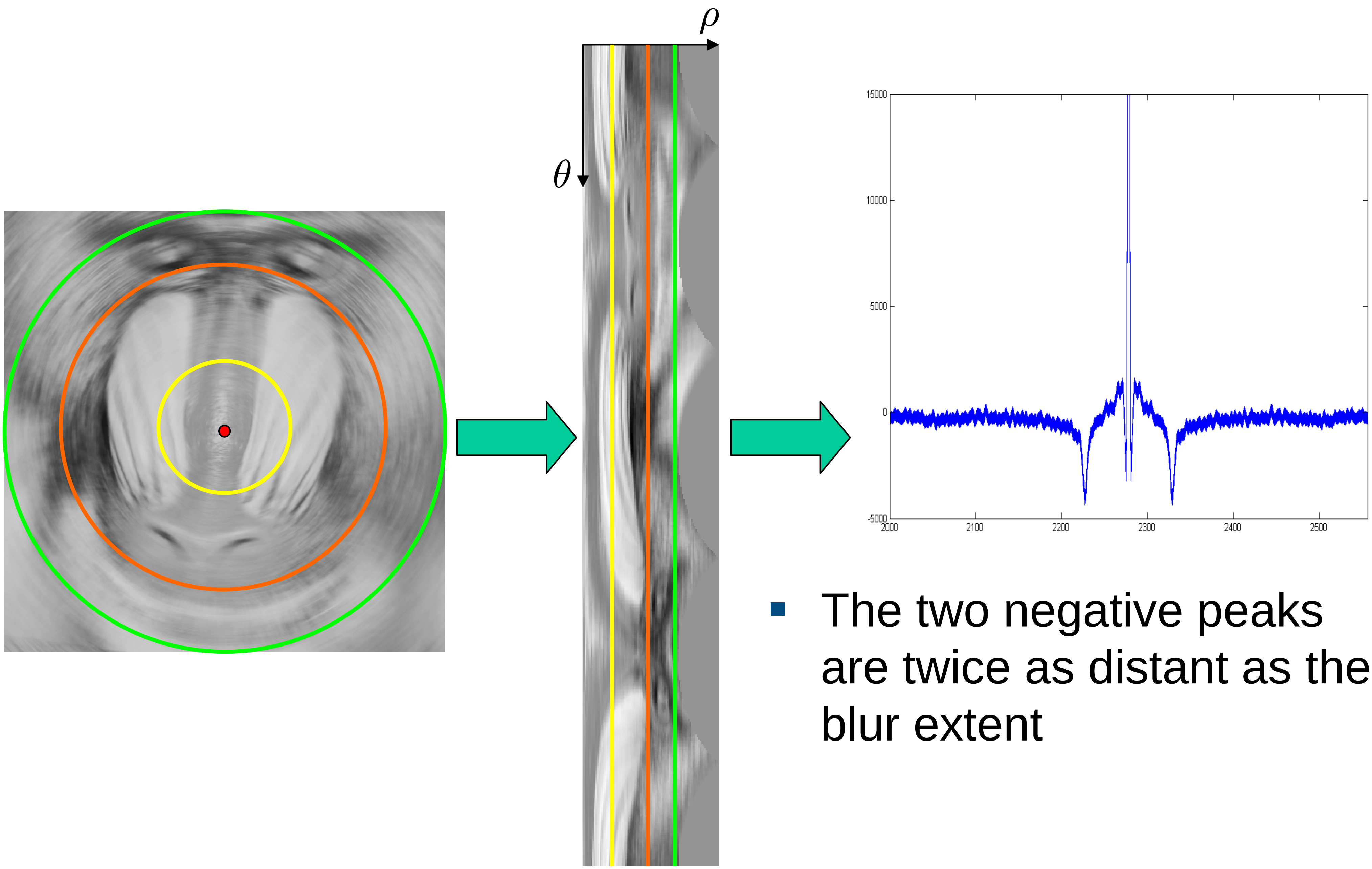






- We compute the auto correlation of the vertical derivatives

Angular speed estimation: an example



- The two negative peaks are twice as distant as the blur extent

Presentation Outline

- Related Works
- Image Formation Model
- The Algorithm
- Experiments
- Conclusions

Experiments Description

- Synthetic Images have been generated as follows
  1. We produce a planar tile of grayscale test images in a ray tracer environment (PovRay)
  2. We render several frames while rotating the camera
  3. The blurred image is given by the average of these frames
  4. We add Gaussian White Noise

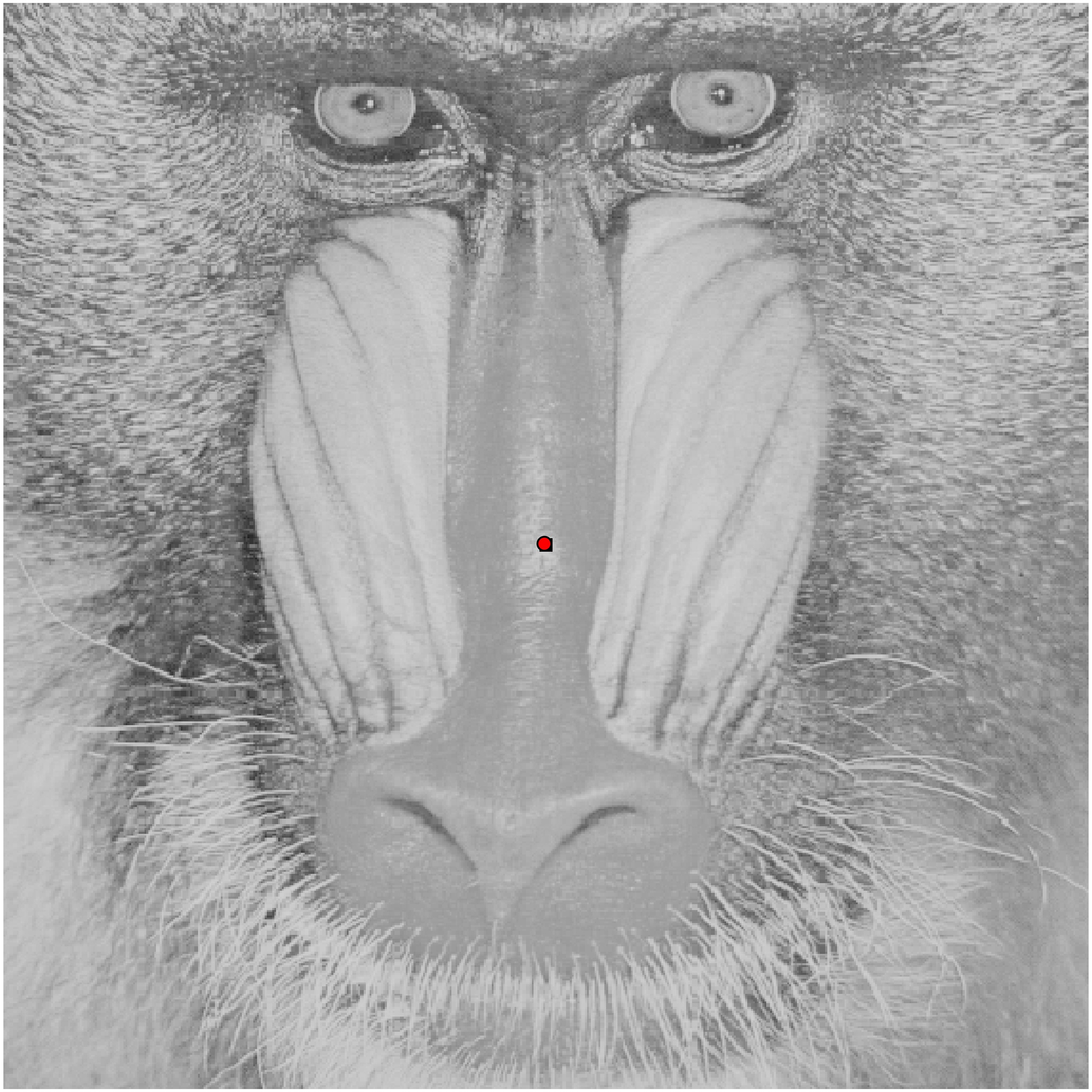
  5. We produced images with several angles between the image plane and the rotation axis.

Synthetic images – Mandrill (1)

$\alpha = 0°, \beta = 0°$

$\omega = 8°/s$

$T = 1s$

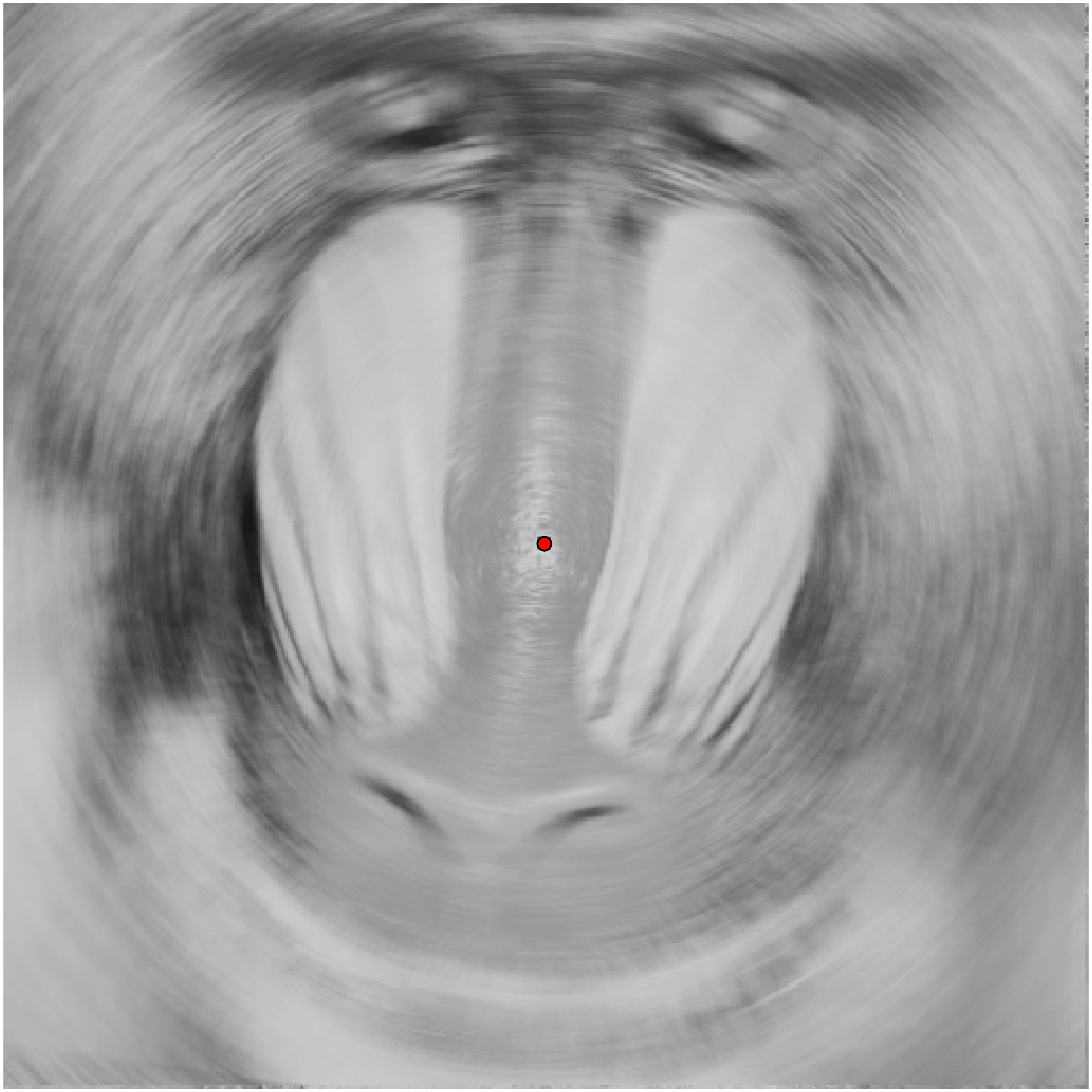


Synthetic images – Mandrill (1)

$\alpha = 0°, \beta = 0°$

$\omega = 8°/s$

$T = 1s$



Synthetic images – Boat

$\alpha = 20°, \beta = 0°$

$\omega = 6°/s$

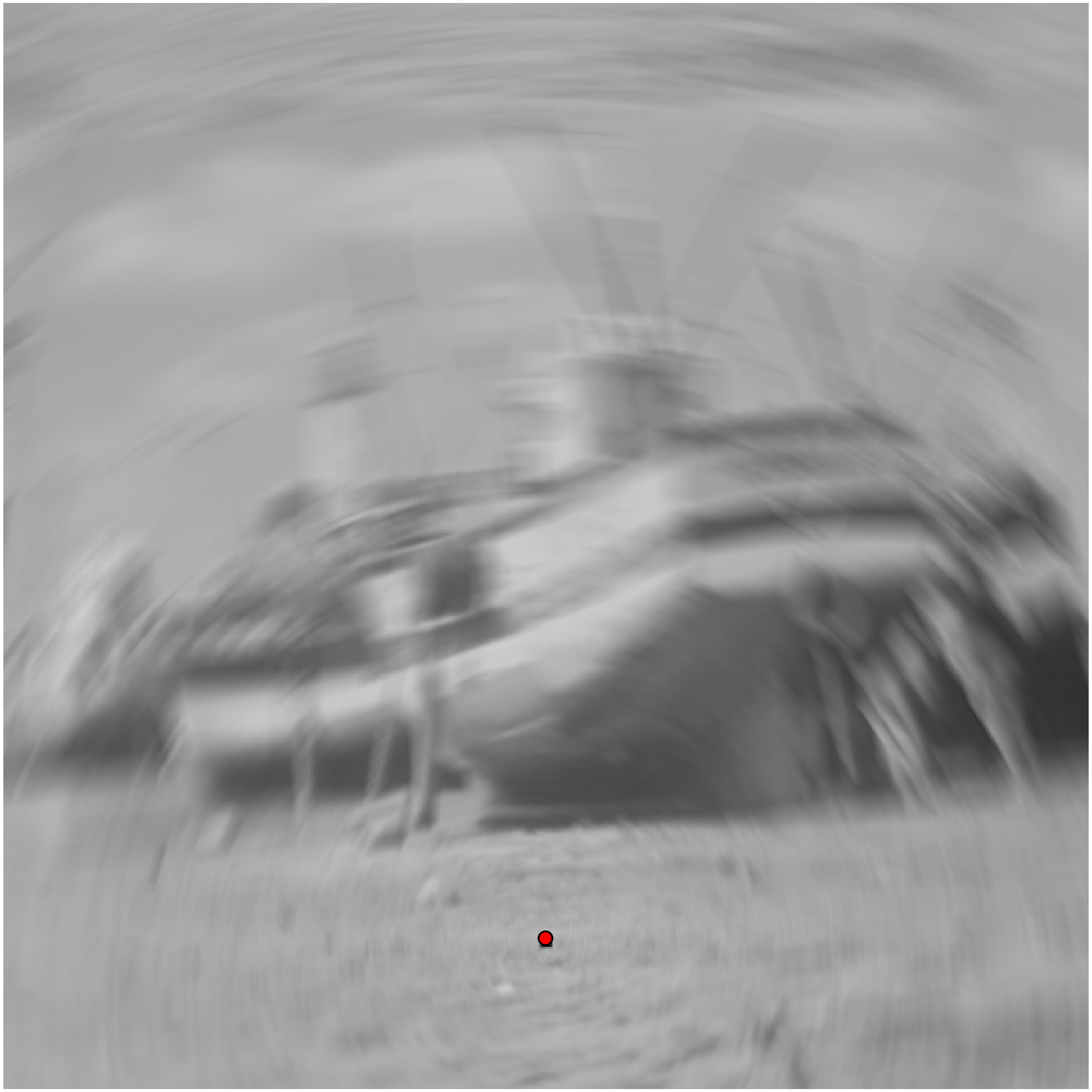
$T = 1s$

Synthetic images – Boat

$\alpha = 20°, \beta = 0°$

$\omega = 6°/s$

$T = 1s$

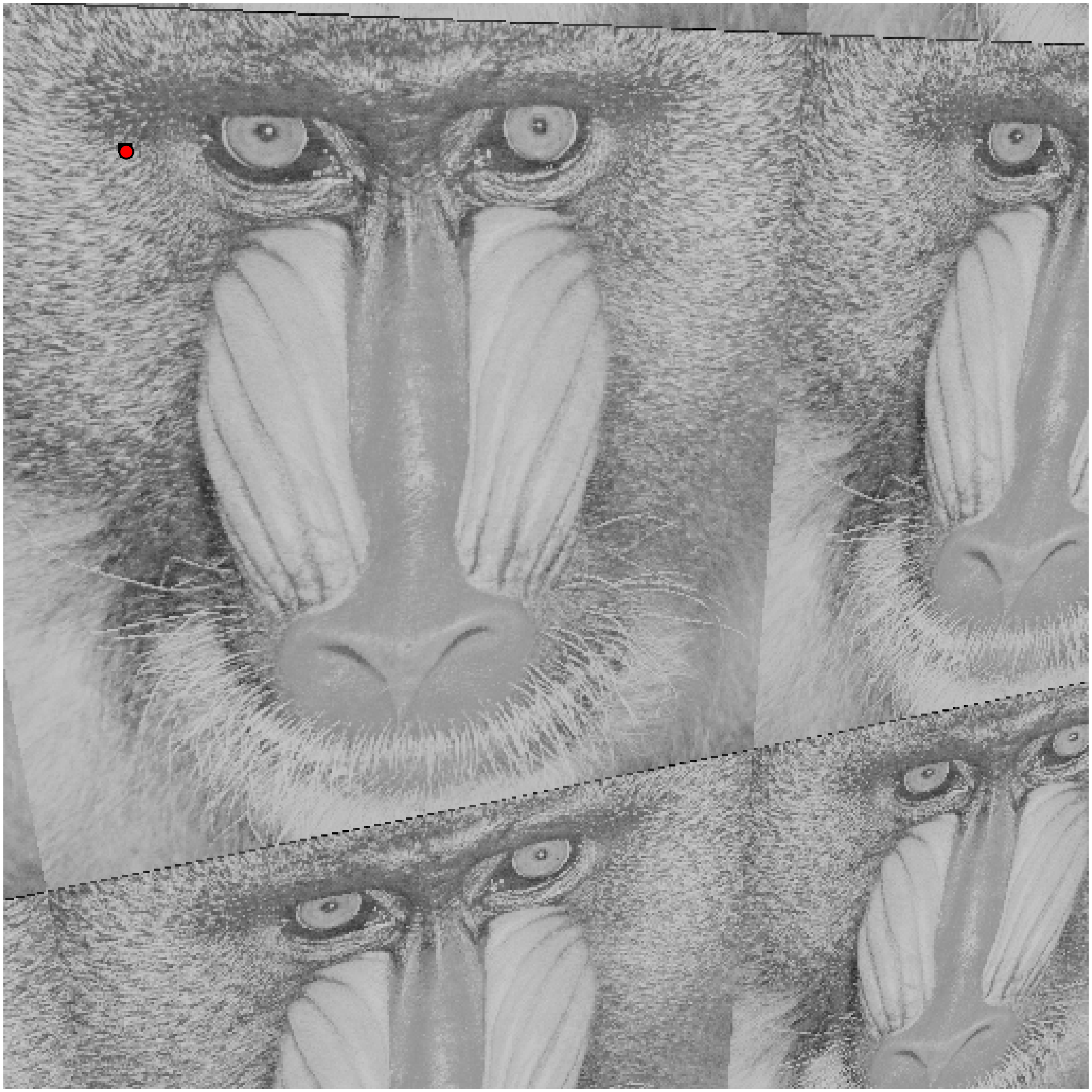


Synthetic images – Mandrill (2)

$\alpha = -20°, \beta = 20°$

$\omega = 8°/s$

$T = 1s$



Synthetic images – Mandrill (2)

$\alpha = -20°, \beta = 20°$

$\omega = 8°/s$

$T = 1s$

Synthetic images – Lena

$\alpha = 0°, \beta = -20°$

$\omega = 6°/s$

$T = 1s$

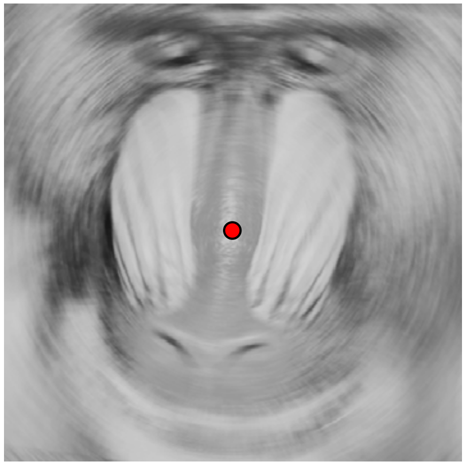
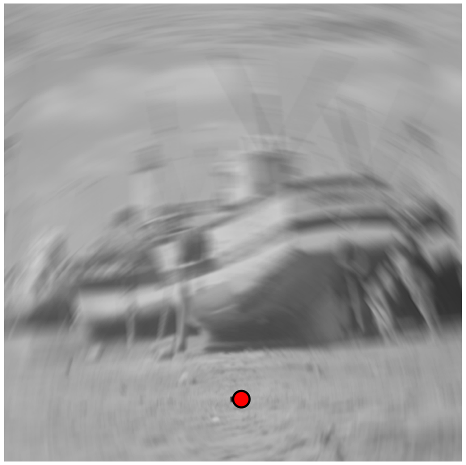
Synthetic images – Lena

$\alpha = 0°, \beta = -20°$

$\omega = 6°/s$

$T = 1s$

Experimental results - synthetic images

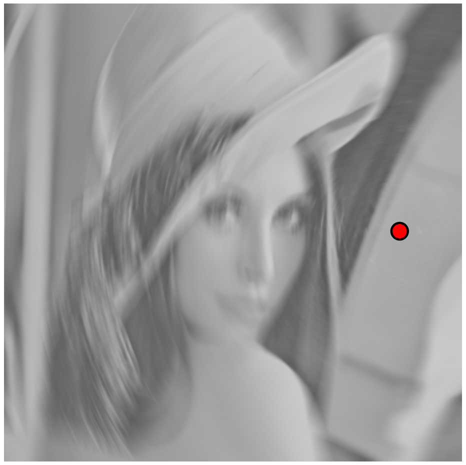
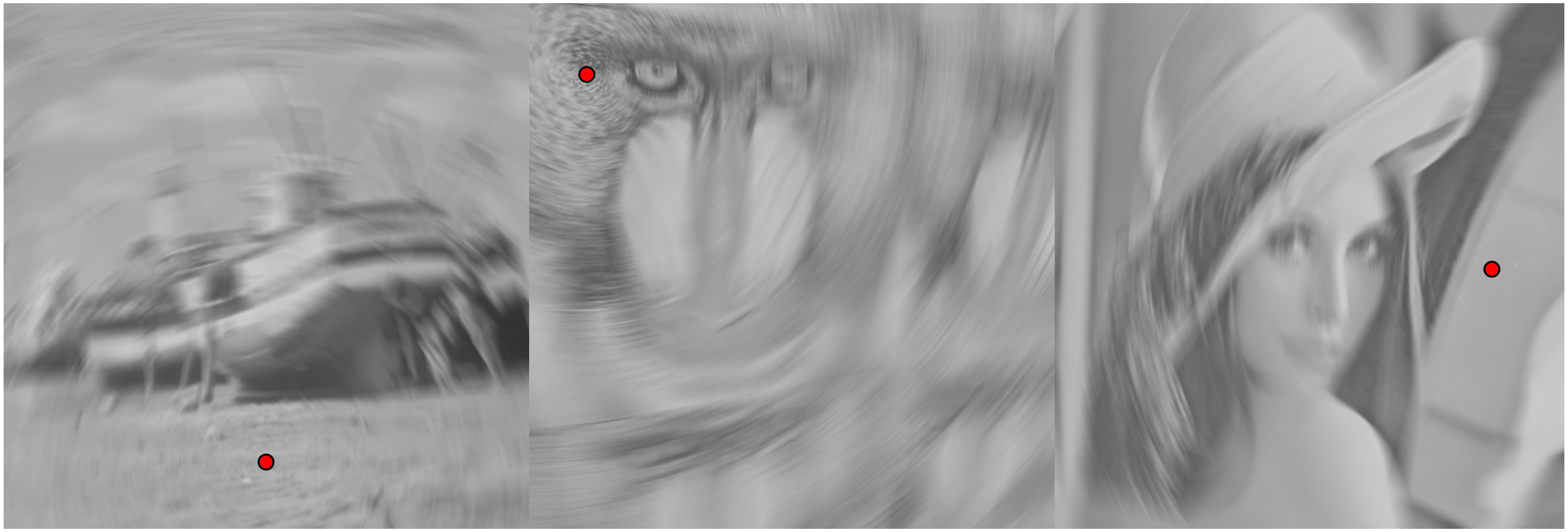
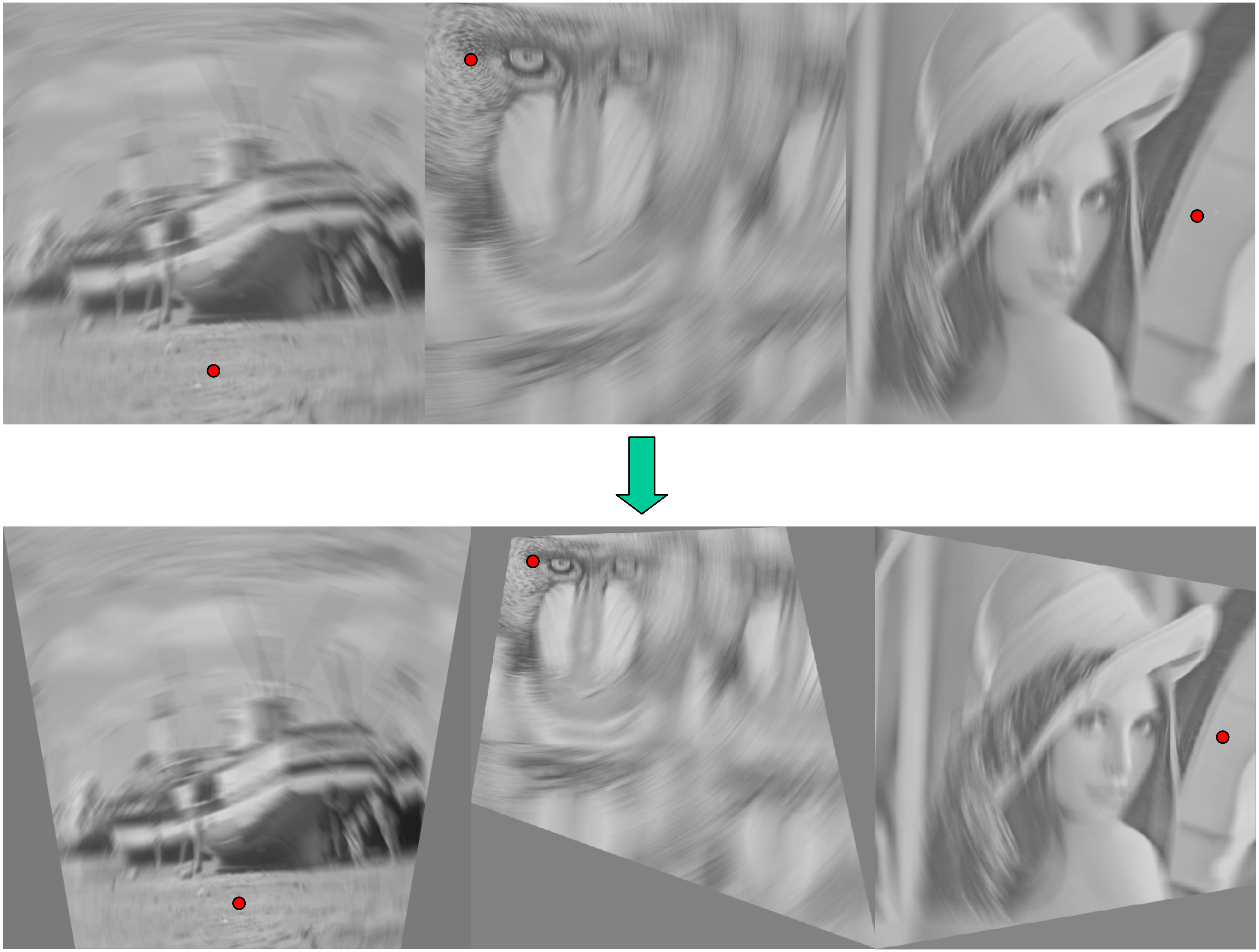


Image	σ_η	$\Delta(\alpha)$	$\Delta(\beta)$	$\Delta(\hat{C})$	$\Delta(\hat{\omega})$	$adv(\%)$	$\Delta(\hat{C}^{0,0})$	$\Delta(\hat{\omega}^{0,0})$
Mandrill1	0	0	0	1	0.05	26.37		
Mandrill1	0.5	0	0	2.27	0.04	20.84		
Mandrill1	1	0	0	3.53	0.08	13.71		
Boat	0	0	0	2.20	0.23	20.44	33.06	4.83
Boat	0.5	0	0	5.46	0.24	20.23	21.27	114.55
Boat	1	0	0	8.84	0.19	8.84	19.25	71.98
Mandrill2	0	0	0	1.00	0.09	5.66	7.07	0.96
Mandrill2	0.5	2	2	1.48	0.11	6.13	4.81	2.85
Mandrill2	1	4	4	1.17	0.26	5.25	4.41	2.29
Lena	0	0	0	3.00	0.08	11.01	12.08	0.60
Lena	0.5	0	0	3.88	0.20	14.06	33.64	64.94
Lena	1	0	4	5.23	0.48	6.00	29.43	62.58

Experimental results – synthetic images



Experimental results – synthetic images



Experiments Description

- Camera images have been acquired rotating the Canon EOS 400D, on a tripod

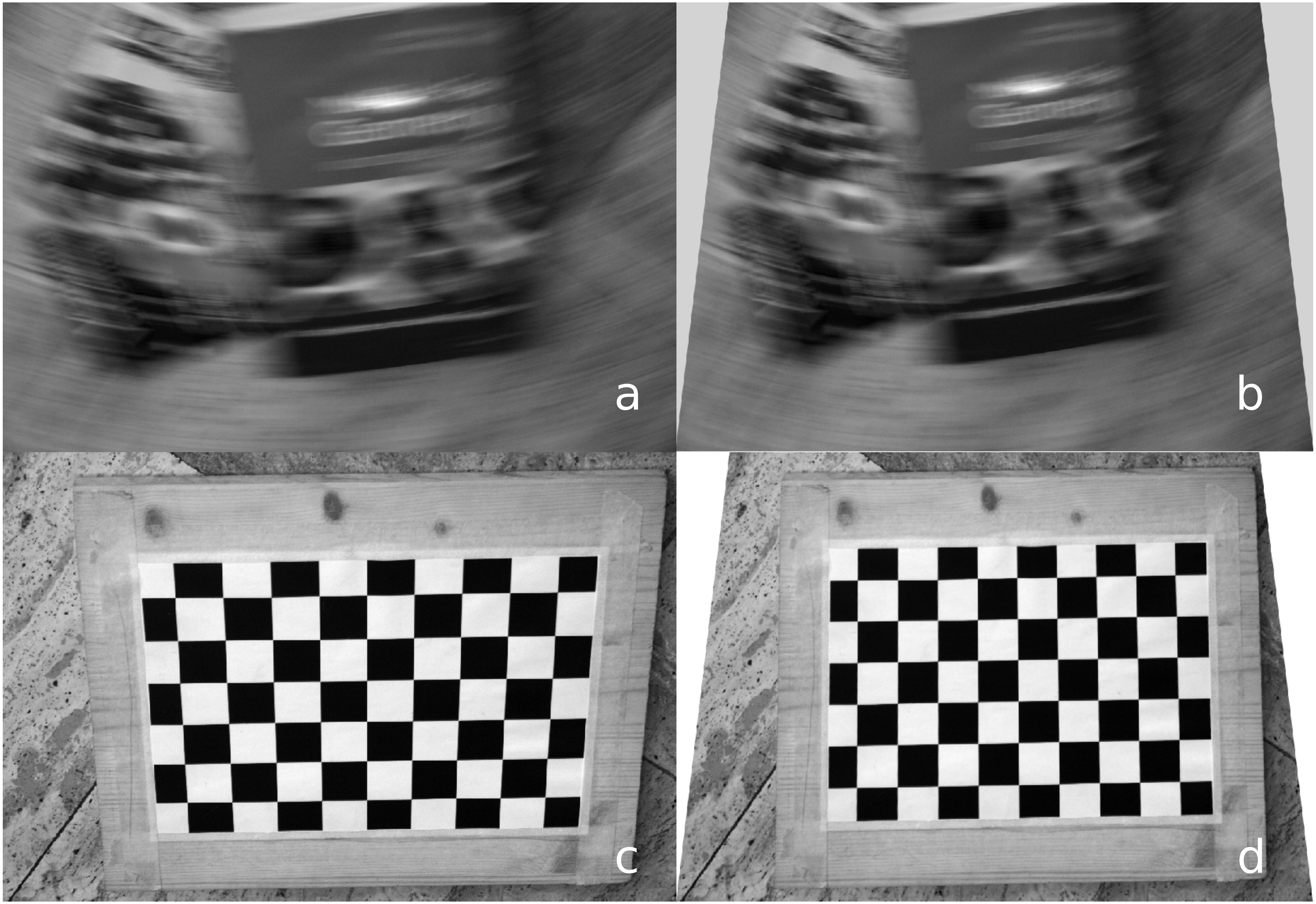
Experiments Description

- Camera images have been acquired rotating the Canon EOS 400D, on a tripod
- The rotation axis was orthogonal to the lab floor (where a checkerboard has been placed)

Experiments Description

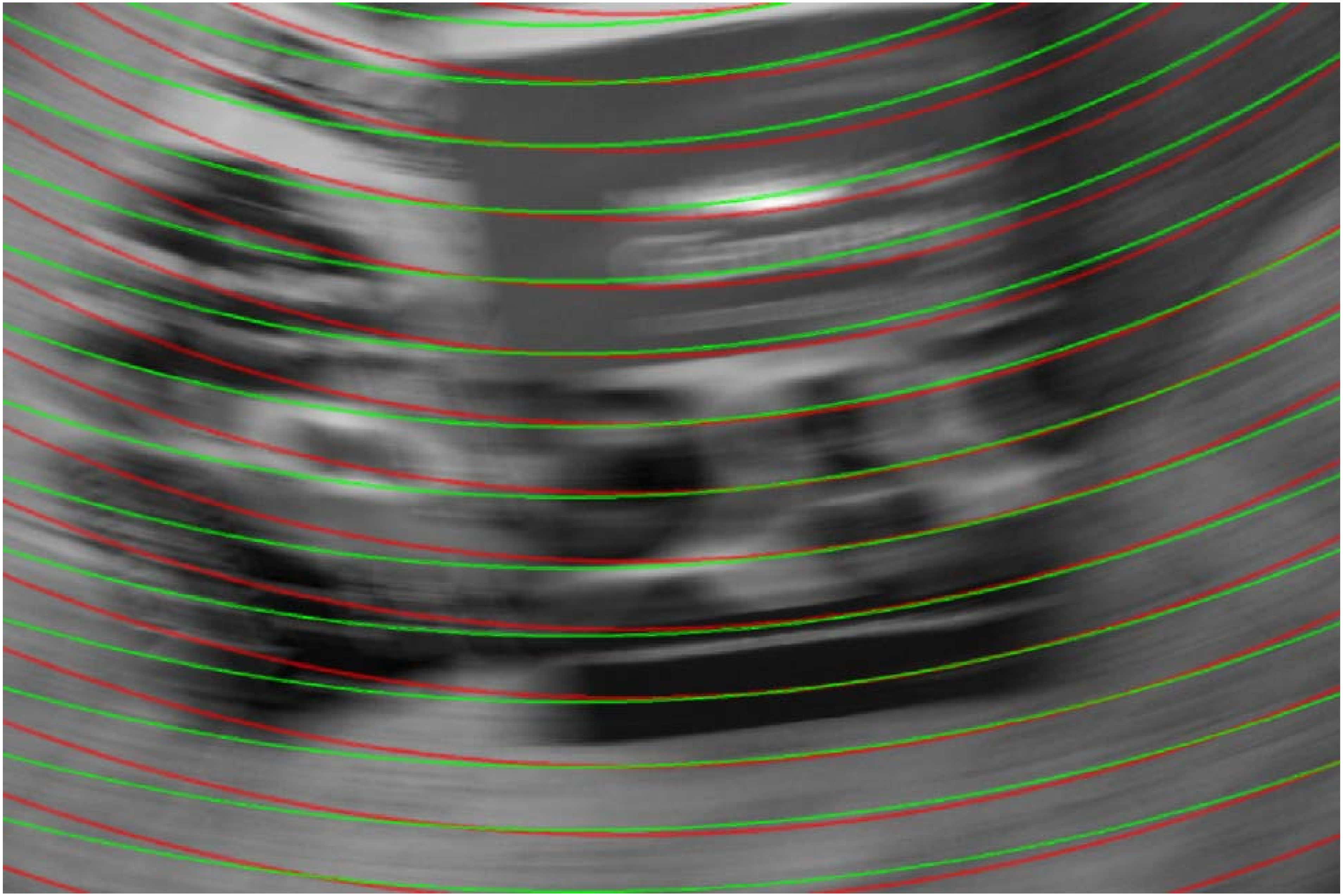
- Camera images have been acquired rotating the Canon EOS 400D, on a tripod
- The rotation axis was orthogonal to the lab floor (where a checkerboard has been placed)
- The *ground truth* on the orientation of the rotation axis w.r.t. the image plane has been obtained by rectifying the image of the **checkerboard**

Experimental results – camera images

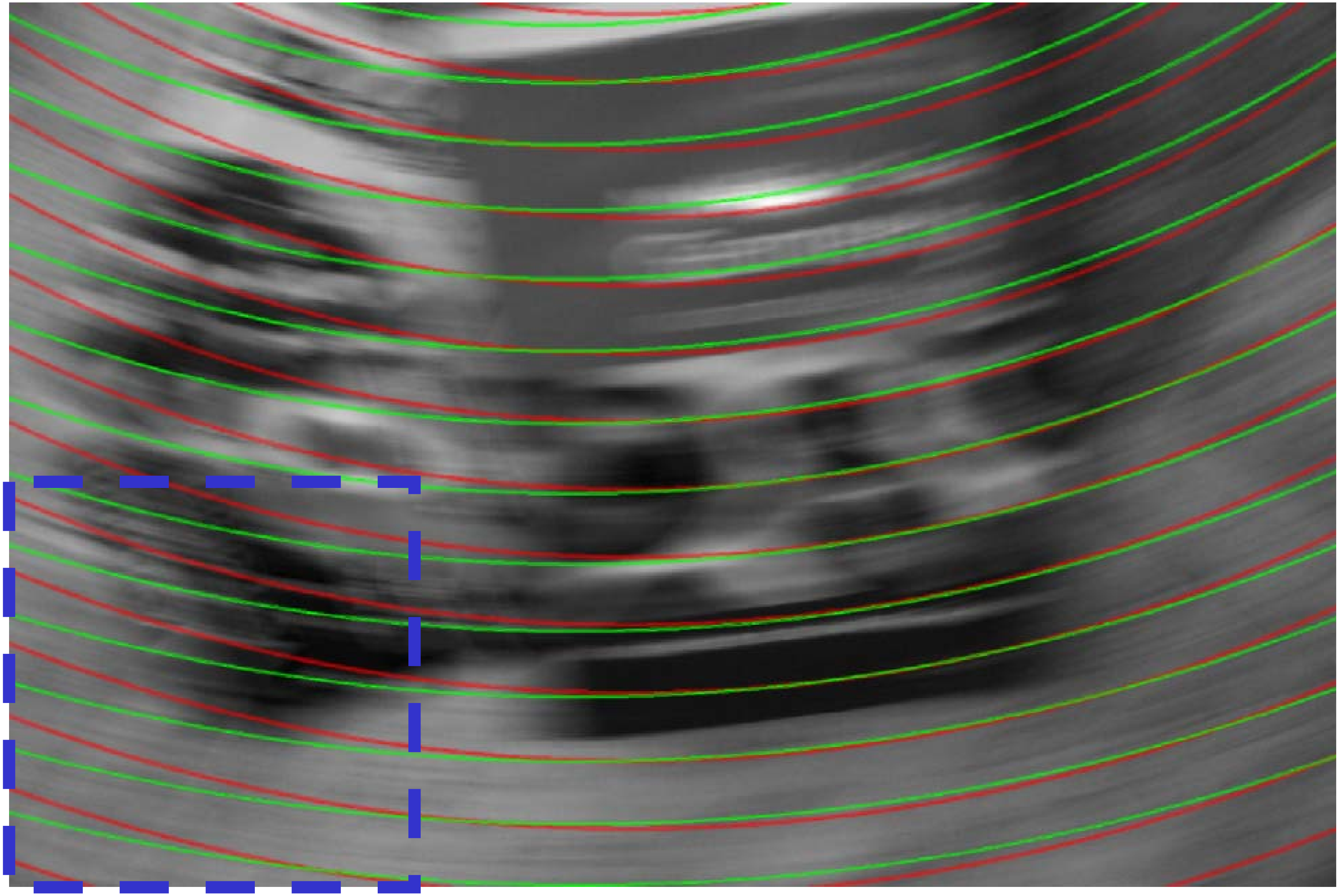


Experimental results – camera images

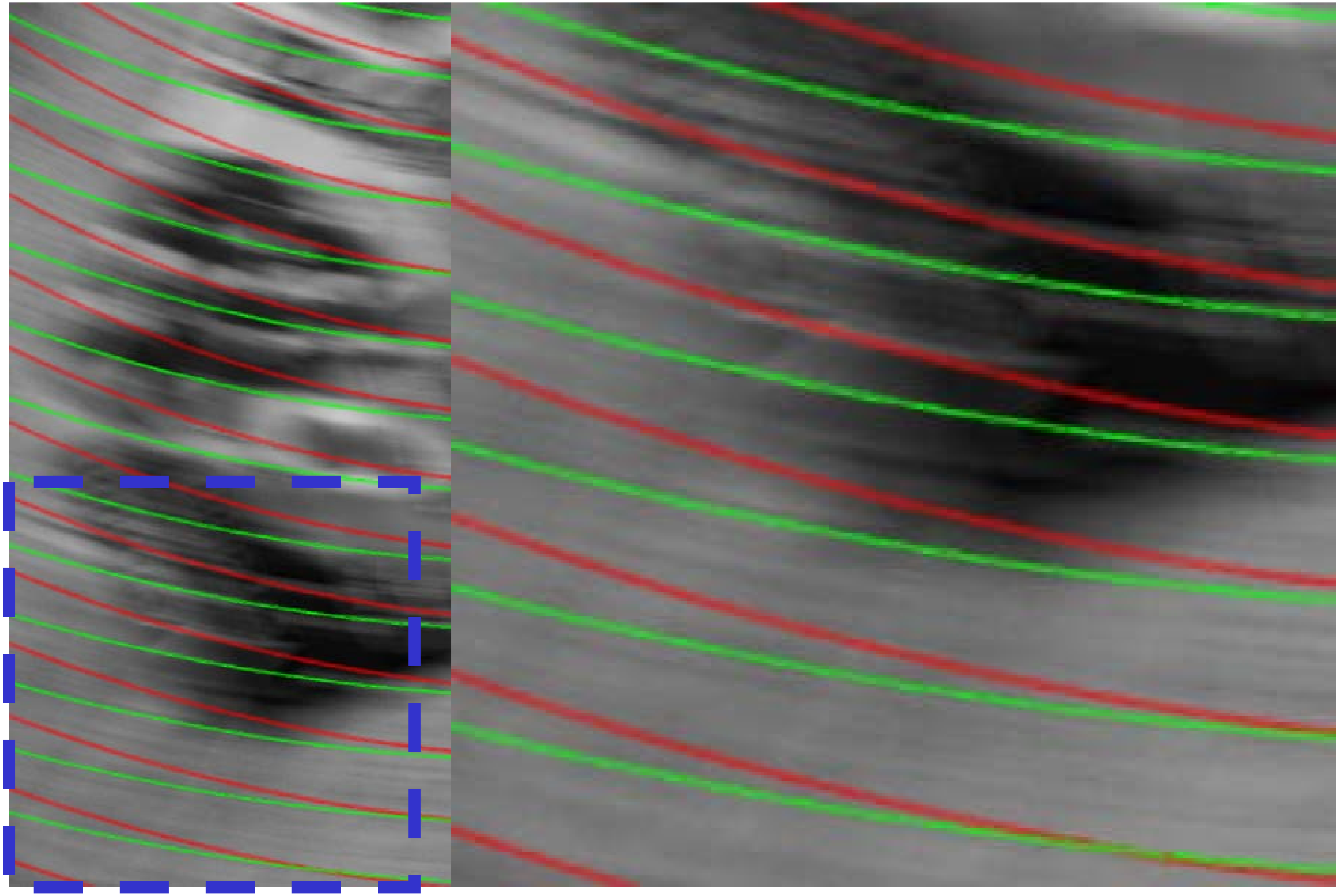
Experimental results – camera images



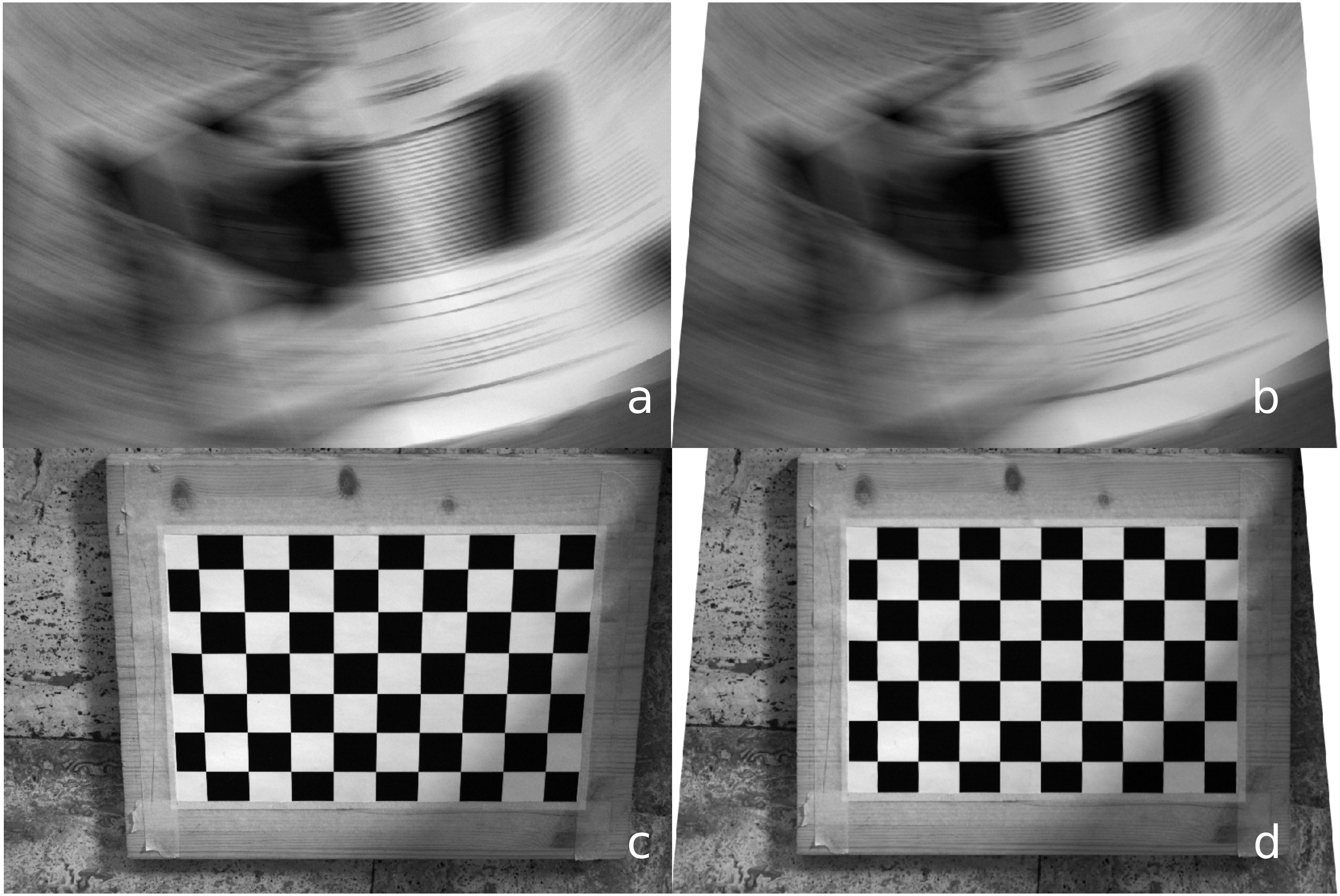
Experimental results – camera images



Experimental results – camera images



Experimental results – camera images



Presentation Outline

- Related Works
- Image Formation Model
- The Algorithm
- Experiments
- Conclusions

Conclusions and future works

- The proposed algorithm achieves **accurate** estimates of both the rotation axis and the angular speed,
  - In the trivial case, when the rotation axis **is orthogonal** to the image plane
  - In the most general case, when the rotation axis is **not orthogonal** to the image plane

Conclusions and future works

- The proposed algorithm achieves **accurate** estimates of both the rotation axis and the angular speed,
  - In the trivial case, when the rotation axis **is orthogonal** to the image plane
  - In the most general case, when the rotation axis is **not orthogonal** to the image plane

- From experimental evidence, it turns out that given a blurred image it is **important** to handle the **blurring paths as conic sections,**

Conclusions and future works

- Future developments concern:
  - Fast Implementation of the **voting procedure.**
  - Study of local blur estimator more **robust** to **noise**.
  - Study of an **ad hoc** algorithm for rotational blur **removal**
  - **Modeling** the effects of blur inversion on AWG noise in order to correctly use denoising algorithm after blur inversion

Thanks