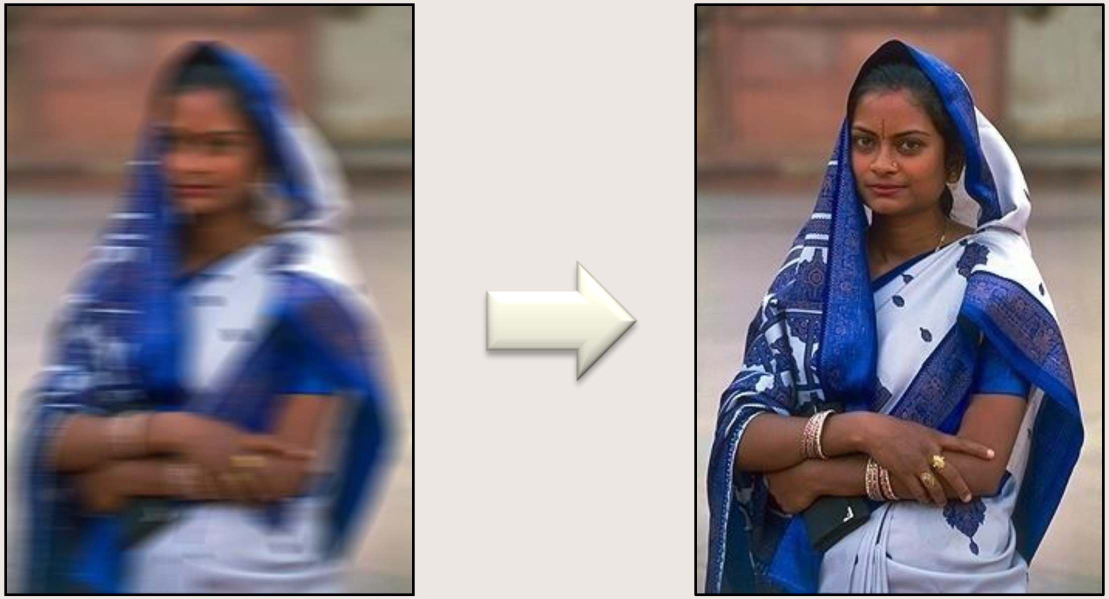
Recent advances in deblurring and image stabilization

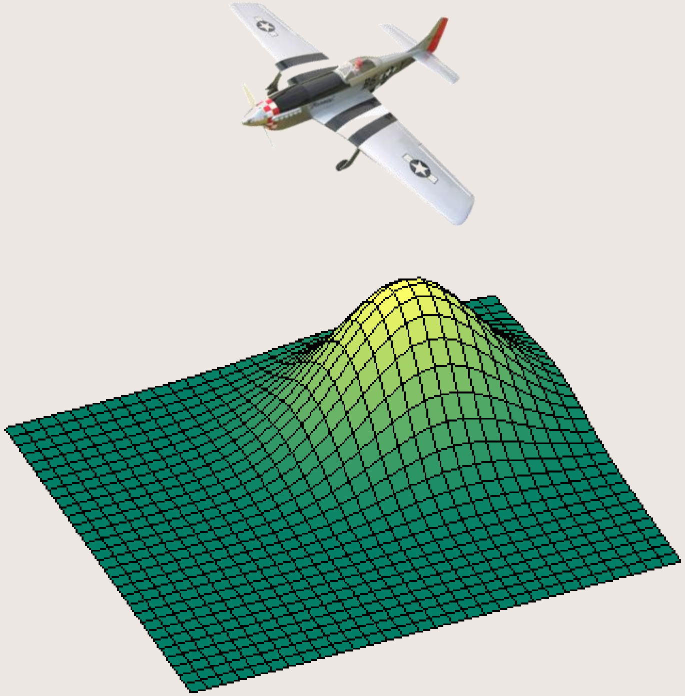
Michal Šorel
Academy of Sciences of the Czech Republic

Camera shake stabilization



- Alternative to OIS (optical image stabilization) systems
- Should work even for subject motion

Remote sensing example

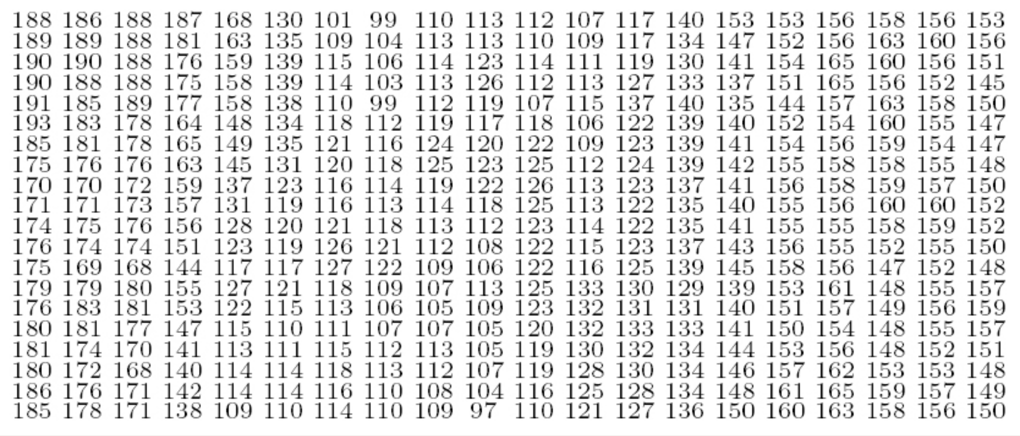


Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

What is an image?

- Rectangular grid of pixels
- Image is a matrix **M × N** for greyscale

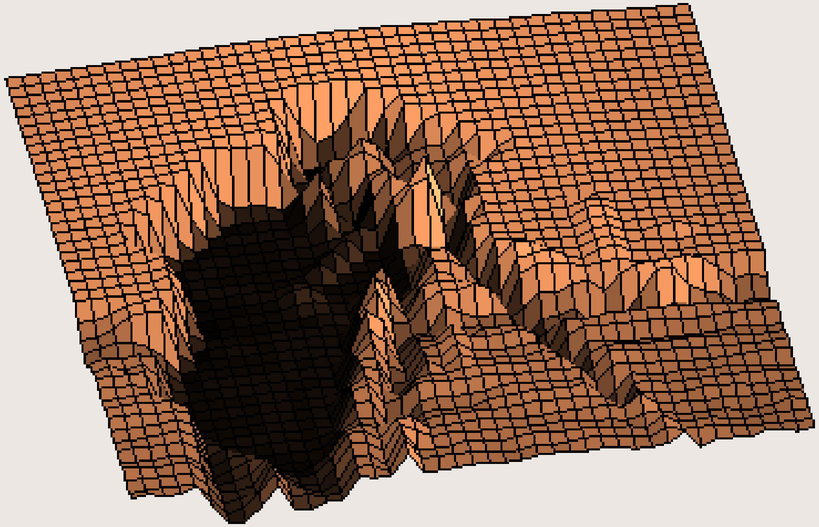


188	186	188	187	168	130	101	99	110	113	112	107	117	140	153	153	156	158	156	153
189	189	188	181	163	135	109	104	113	113	110	109	117	134	147	152	156	163	160	156
190	190	188	176	159	139	115	106	114	123	114	111	119	130	141	154	165	160	156	151
190	188	188	175	158	139	114	103	113	126	112	113	127	133	137	151	165	156	152	145
191	185	189	177	158	138	110	99	112	119	107	115	137	140	135	144	157	163	158	150
193	183	178	164	148	134	118	112	119	117	118	106	122	139	140	152	154	160	155	147
185	181	178	165	149	135	121	116	124	120	122	109	123	139	141	154	156	159	154	147
175	176	176	163	145	131	120	118	125	123	125	112	124	139	142	155	158	158	155	148
170	170	172	159	137	123	116	114	119	122	126	113	123	137	141	156	158	159	157	150
171	171	173	157	131	119	116	113	114	118	125	113	122	135	140	155	156	160	160	152
174	175	176	156	128	120	121	118	113	112	123	114	122	135	141	155	155	158	159	152
176	174	174	151	123	119	126	121	112	108	122	115	123	137	143	156	155	152	155	150
175	169	168	144	117	117	127	122	109	106	122	116	125	139	145	158	156	147	152	148
179	179	180	155	127	121	118	109	107	113	125	133	130	129	139	153	161	148	155	157
176	183	181	153	122	115	113	106	105	109	123	132	131	131	140	151	157	149	156	159
180	181	177	147	115	110	111	107	107	105	120	132	133	133	141	150	154	148	155	157
181	174	170	141	113	111	115	112	113	105	119	130	132	134	144	153	156	148	152	151
180	172	168	140	114	114	118	113	112	107	119	128	130	134	146	157	162	153	153	148
186	176	171	142	114	114	116	110	108	104	116	125	128	134	148	161	165	159	157	149
185	178	171	138	109	110	114	110	109	97	110	121	127	136	150	160	163	158	156	150

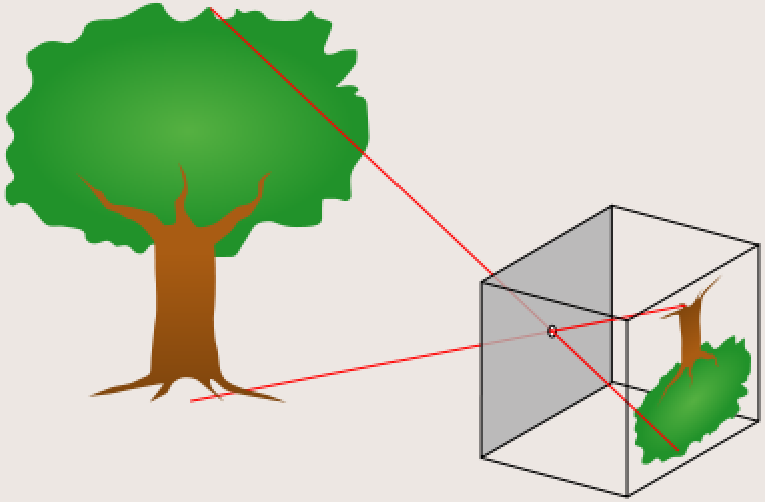
- Matrix **M × N ×** 3 for color images
- Formulas shown for greyscale images

Image as a function

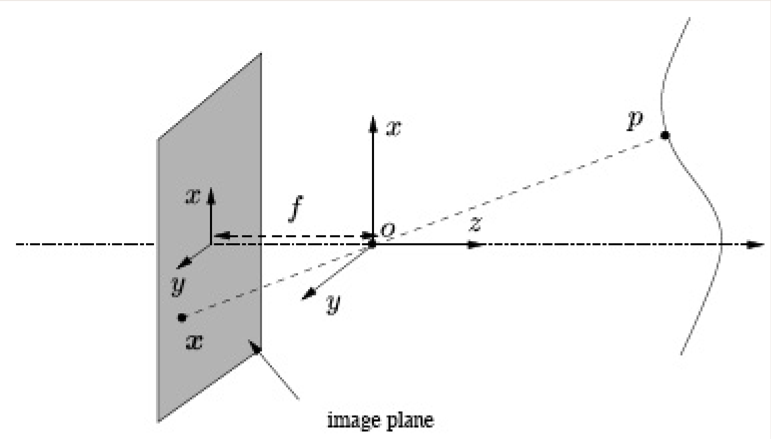
- In formulas often a real function of two variables $R^2 \rightarrow R^+$, mostly 0..1



Pinhole camera model



Pinhole camera
(Camera obscura)



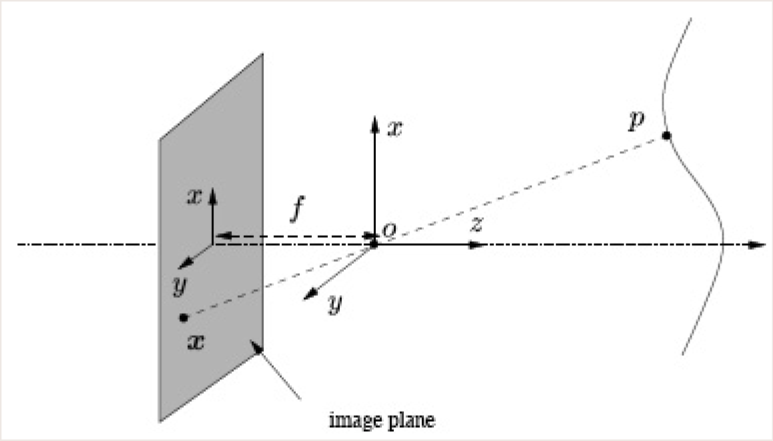
Pinhole camera
model

Focal length and sensor size

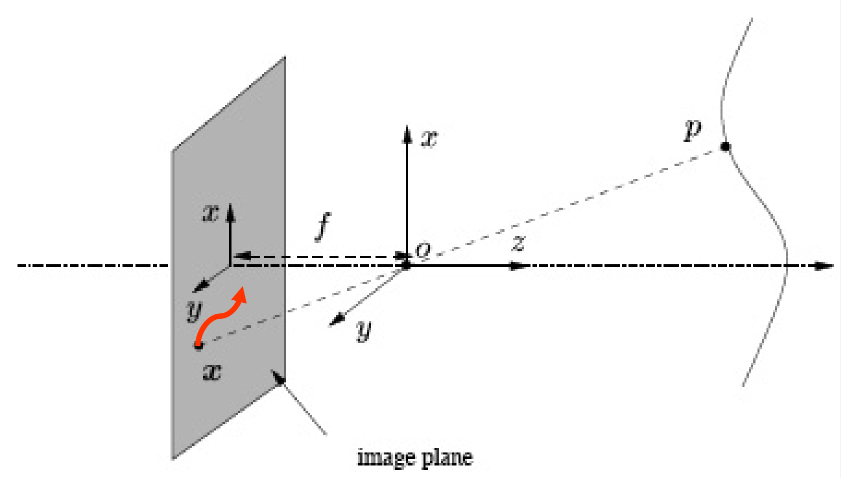
fish-eye lens
f down to 5mm

normal lens
f ~ 50 mm

telephoto lens
(**f** > 100 mm)



What happens if camera moves?



- Sharp image – movement less than ½ pixel
- Influence of focal length, shutter speed, sensor resolution (pixel density)
- Velocity field, PSF ~ blur kernel

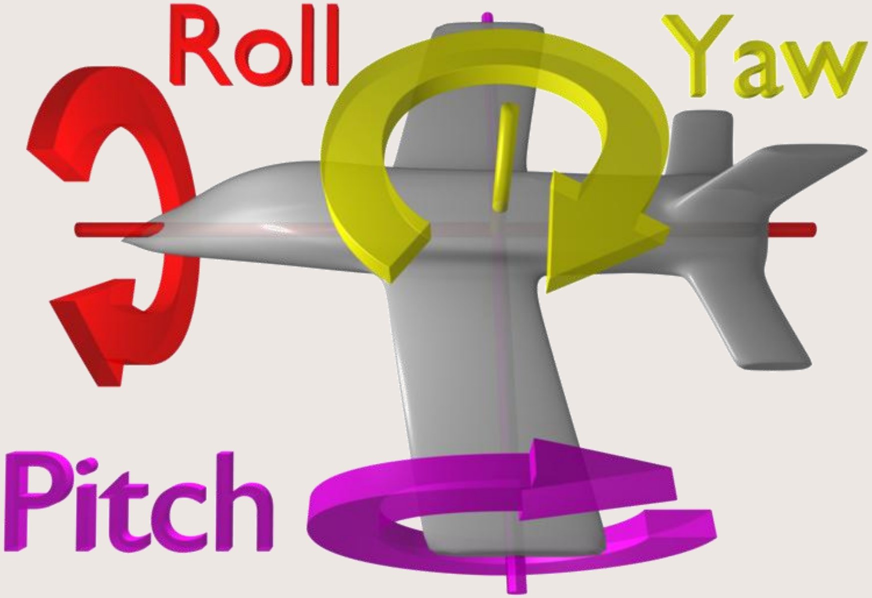
3D camera motion

- Rigid body – 6 degrees of freedom
- Natural coordinate system
- 2 vectors of camera velocity:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \qquad \Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$$

Roll, Yaw, Pitch movements

$$\Omega = \begin{bmatrix} \boxed{\Omega_x} \\ \boxed{\Omega_y} \\ \boxed{\Omega_z} \end{bmatrix}$$

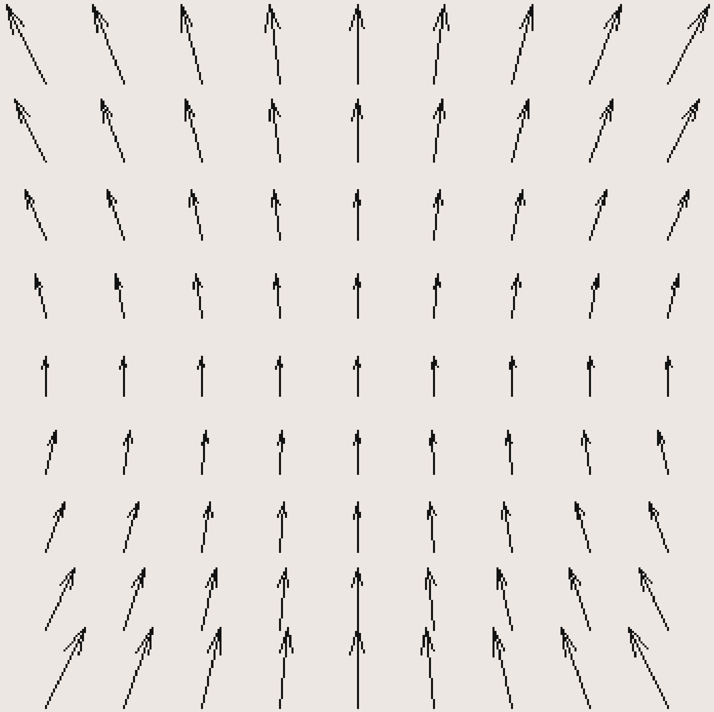


Pan ... follow an object by a camera (often refers to horizontal motion)

Rotation down - demonstration

Camera rotates downwards ↓ (pitch motion)

Velocity field



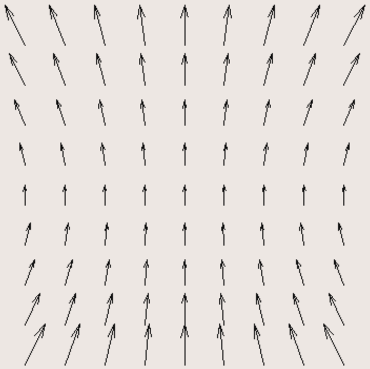
$$\Omega = \begin{bmatrix} \Omega_x \\ 0 \\ 0 \end{bmatrix}$$

Velocity field

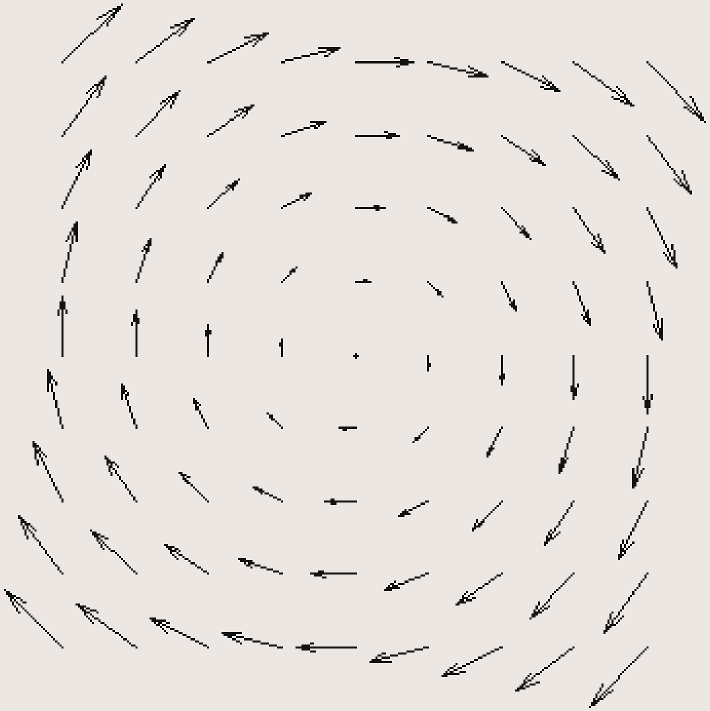
$$\mathbf{v}(x,y) = \frac{1}{\mathbf{d}(x,y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T +$$

$$\begin{bmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \end{bmatrix} \Omega$$

d - depth map

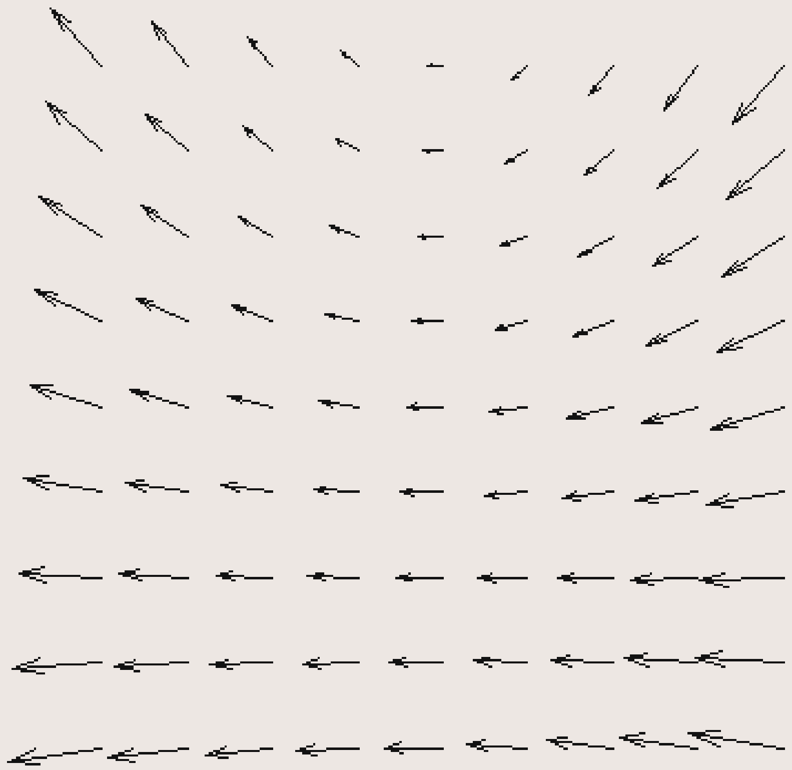


Rotation about optical axis (roll)



$$\Omega = \begin{bmatrix} 0 \\ 0 \\ \Omega_z \end{bmatrix}$$

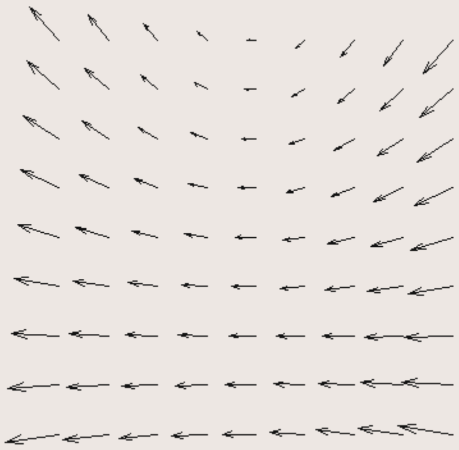
General 3D rotation



Stabilizer of 3D camera rotation

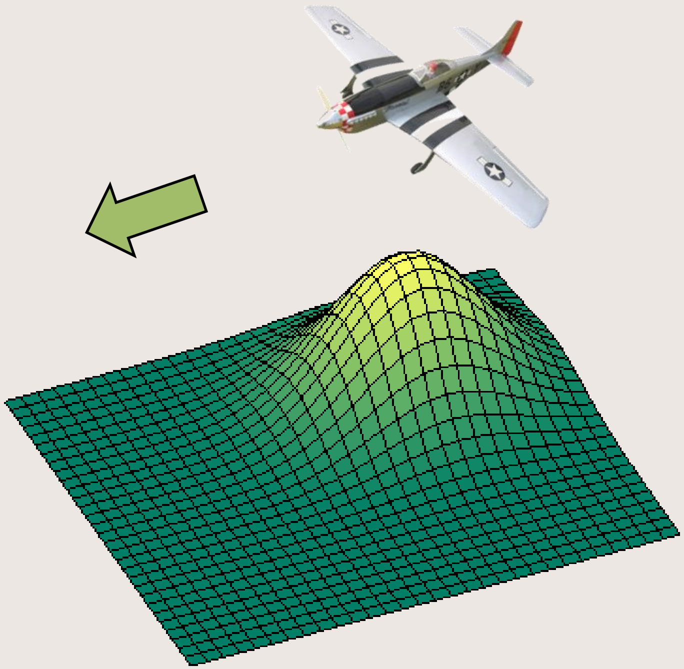
- For hand shake, camera rotation is mostly dominant
- Blur is independent of scene depth (that is why optical image stabilizers can work) and changes gradually

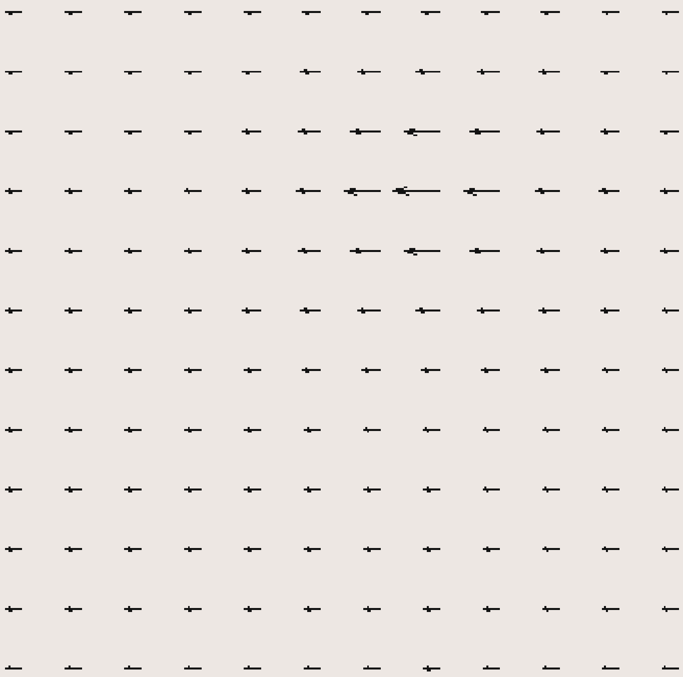
$$\mathbf{v}(x,y) = \frac{1}{\mathbf{d}(x,y)}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T +$$
$$\begin{bmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \end{bmatrix} \Omega$$



Translation

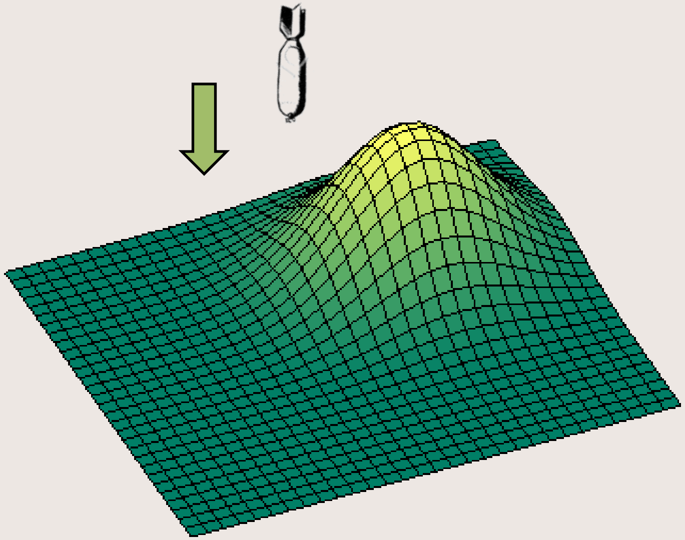
$$\mathbf{v}(x,y) = \frac{1}{\mathbf{d}(x,y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T + \dots$$

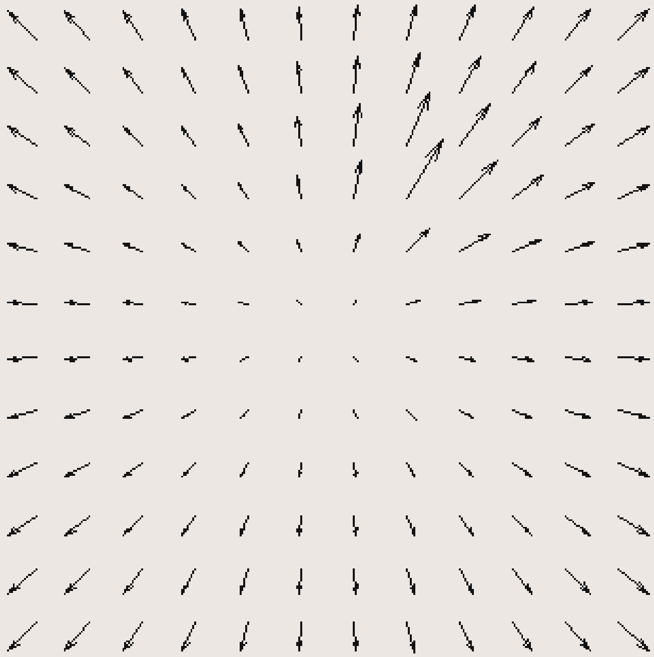




Translation along optical axis

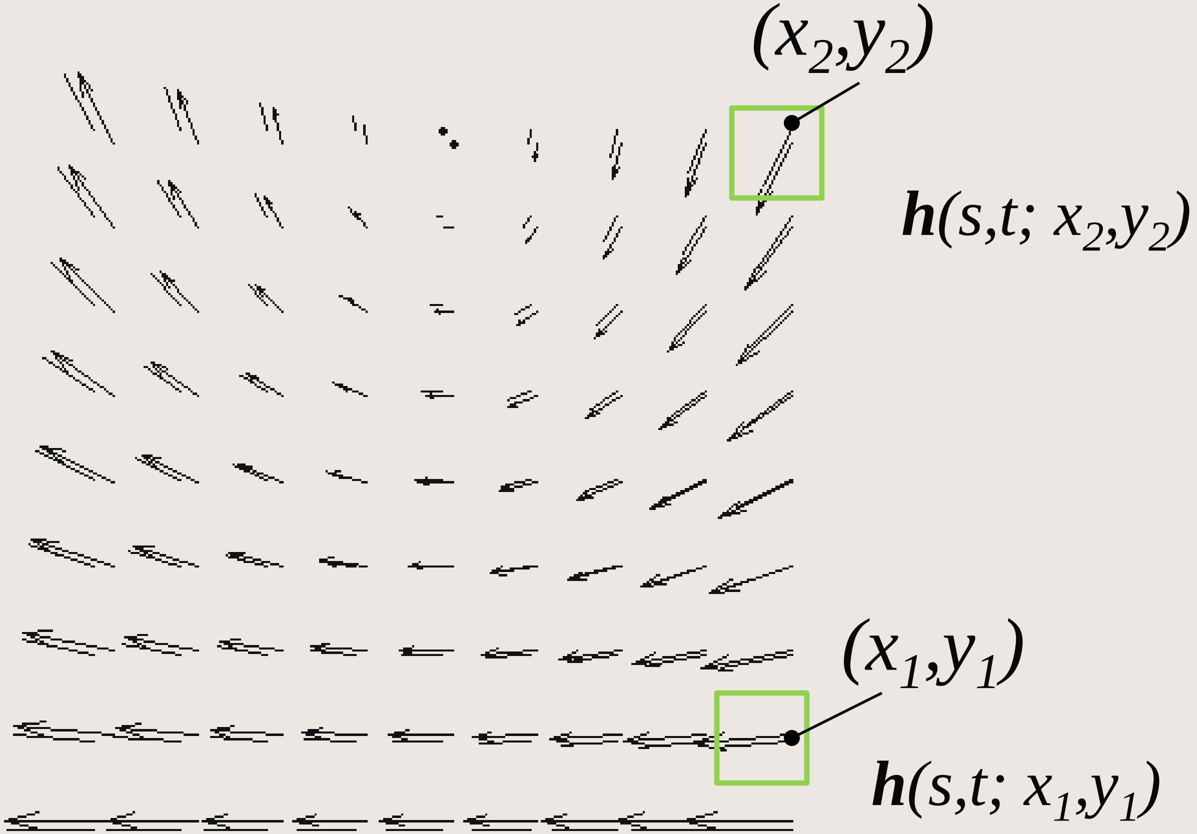
$$\mathbf{v}(x,y) = \frac{1}{\mathbf{d}(x,y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T + \ldots$$





Point-spread function - PSF

- Integration of velocity field → **PSF**



(x_2,y_2)

$\boldsymbol{h}(s,t;\, x_2,y_2)$

(x_1,y_1)

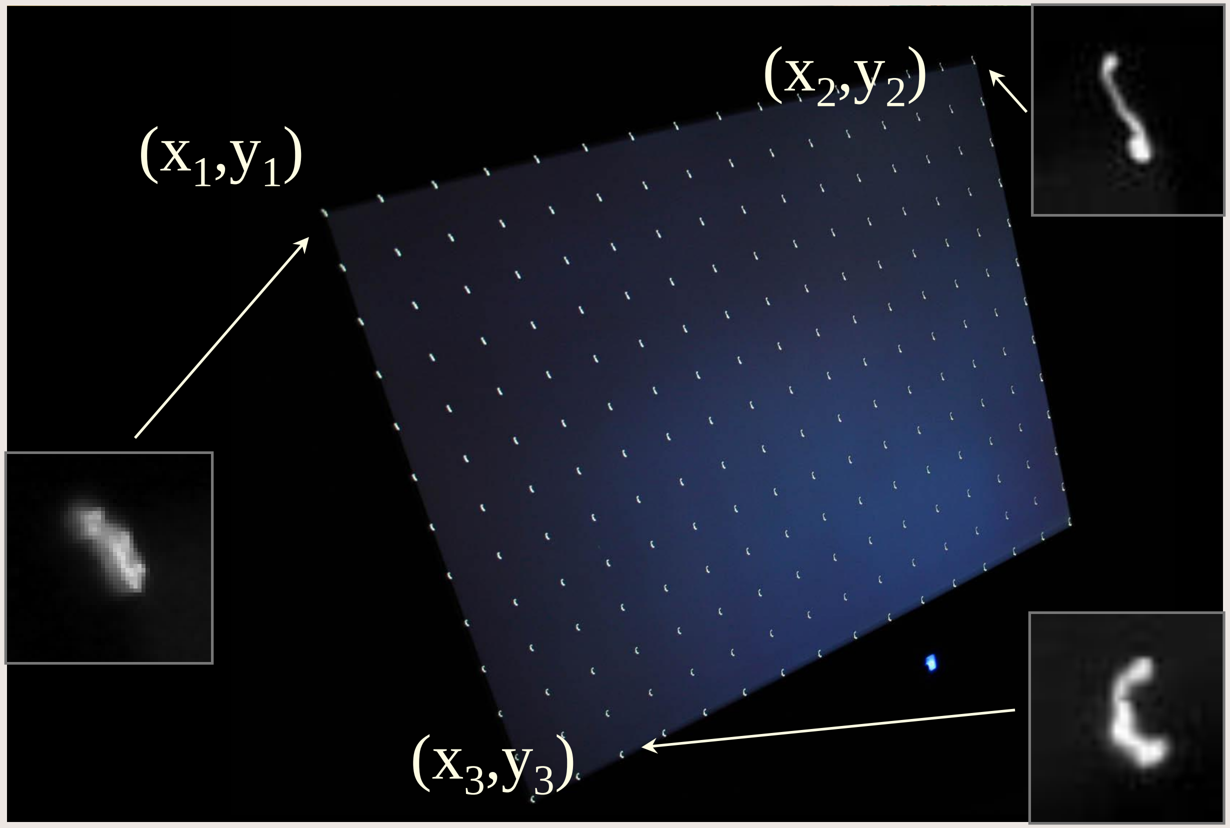
$\boldsymbol{h}(s,t;\, x_1,y_1)$

Mathematical model of blurring

$$\mathbf{u} *_v \mathbf{h}\,[x,y] = \int_\Omega \mathbf{u}(x-s, y-t)\mathbf{h}(s,t;\xcancel{x-s, y-t})\,dsdt$$

- PSF **h** ... depends on position (x,y)
- Generalized convolution
- Convolution case – **h** is called convolution kernel or convolution mask

PSF for camera shake



$\boldsymbol{h}(s,t;\, x_2,y_2)$

$\boldsymbol{h}(s,t;\, x_1,y_1)$

$\boldsymbol{h}(s,t;\, x_3,y_3)$

Blur description – summary (I)

- What we have learned
  - What happens when a camera is moving
  - 4 motion compoments
  - Velocity field
  - How PSF describes the blur and its relation with velocity field

Blur description – summary (II)

Motion component	Dependence on distance	Space-variant blur
YAW, PITCH (x,y-axis rotation)	NO	YES (a bit)
ROLL (z-axis rotation)	NO	YES (a lot)
X,Y-axis translation	YES	NO
Z-axis translation	YES	YES (a lot)

Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

Hardware approaches to suppress blur

- Boosting ISO (100, 200, 400, 800, 1600, 3200)
- External stabilization/gyro-stabilized gimbals (two principles)
- Optical image stabilization (OIS) systems

High ISO is not a solution

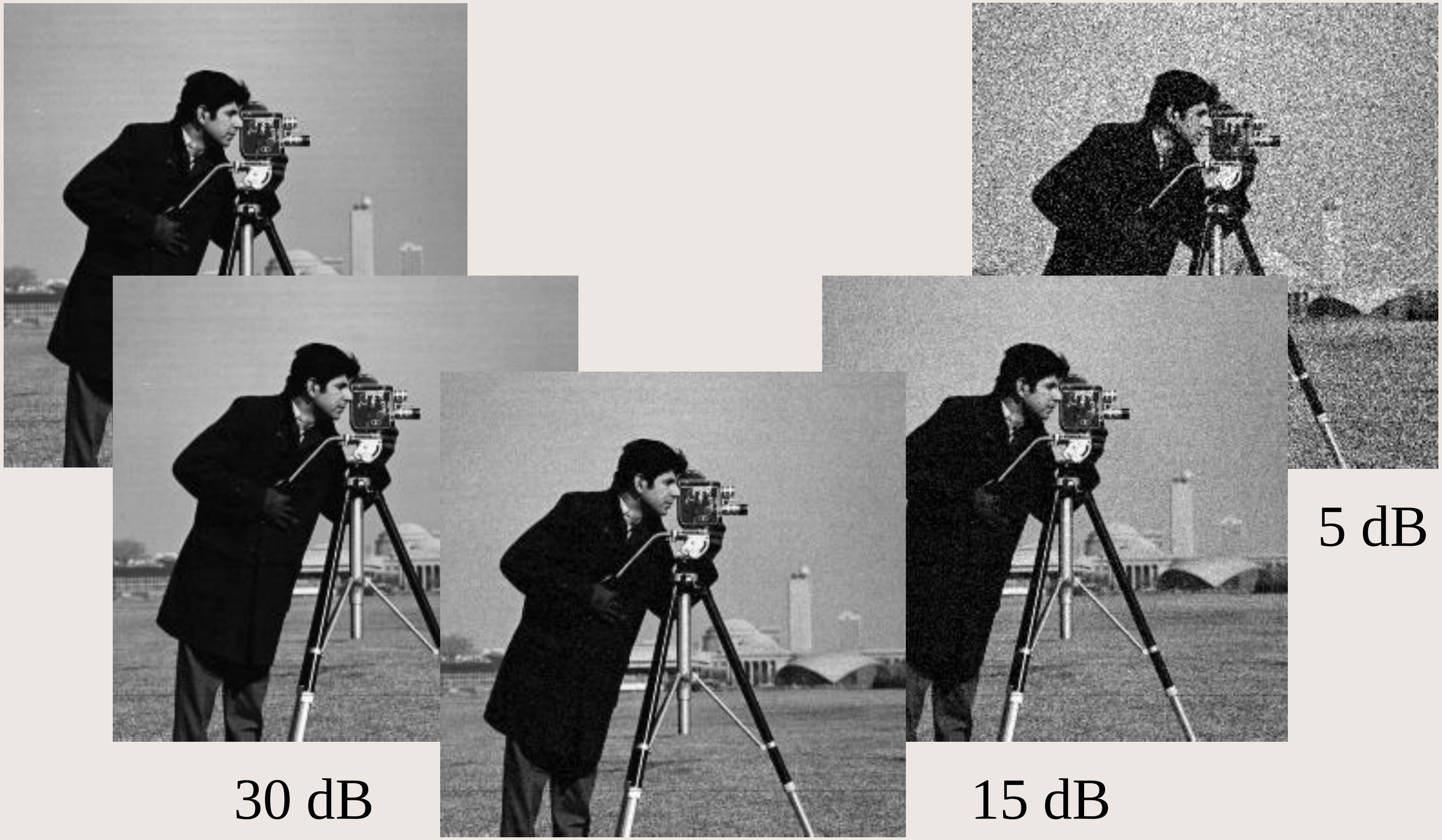
- ISO - 100, 200, 400, 800, 1600, 3200
- ISO 100 ⇨ ISO 200
  ~ f-number/2, 2*t (1 EV or 1 stop)
- ISO 100 ⇨ ISO 3200 ~ 32*t (5 stops)

Photon noise (Poisson) SNR ~ SNR_0* t

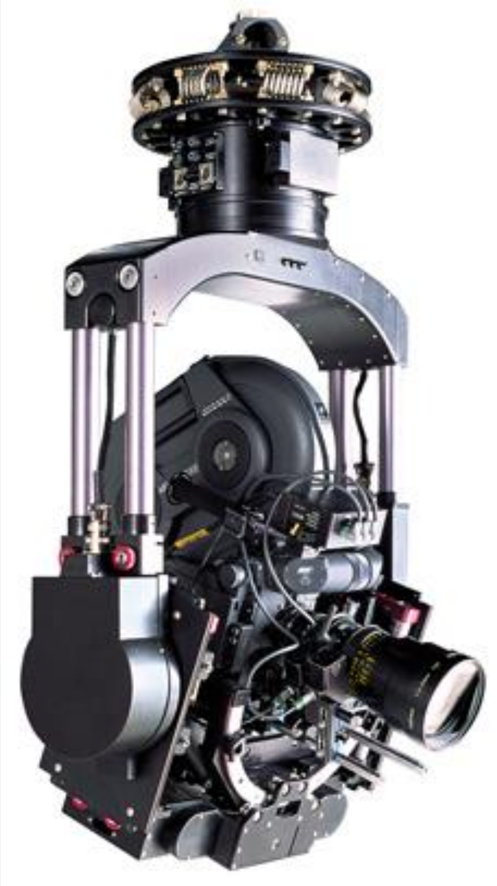
$\mathbf{SNR}_{\mathbf{1600}} = \mathbf{SNR}_{\mathbf{100}} / \mathbf{16}$ (-12 dB)

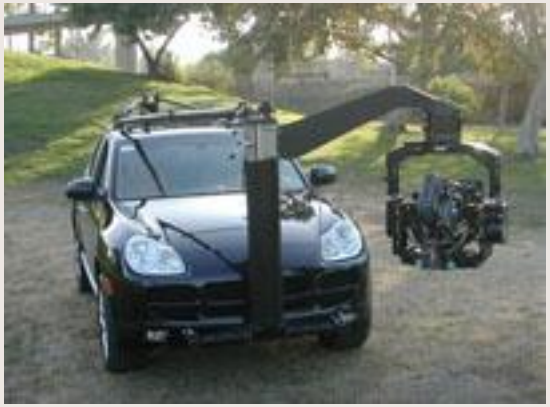
$\mathbf{SNR}_{\mathbf{3200}} = \mathbf{SNR}_{\mathbf{100}} / \mathbf{32}$ (-15 dB)

SNR



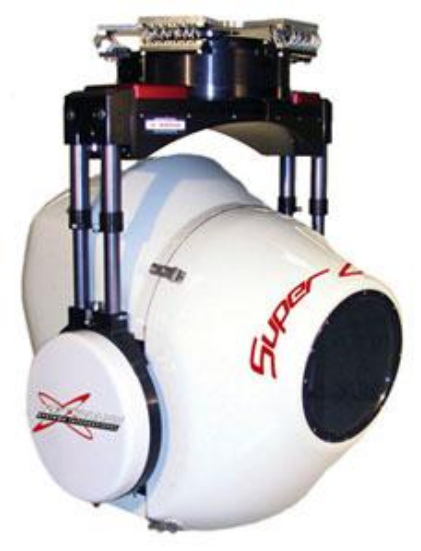
Gyro-stabilized gimbals

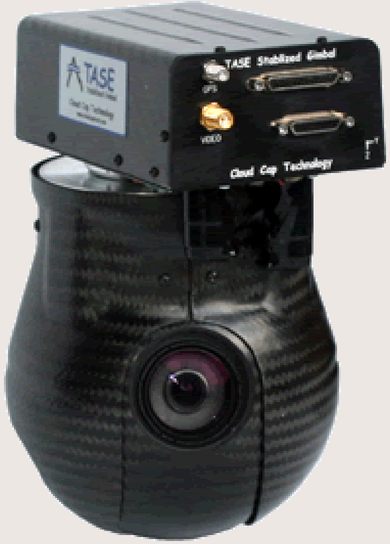




Gyron FS
(Nettmann systems international)
http://www.camerasystems.com/gyronfs.htm

Gyro-stabilized gimbals (airborn)





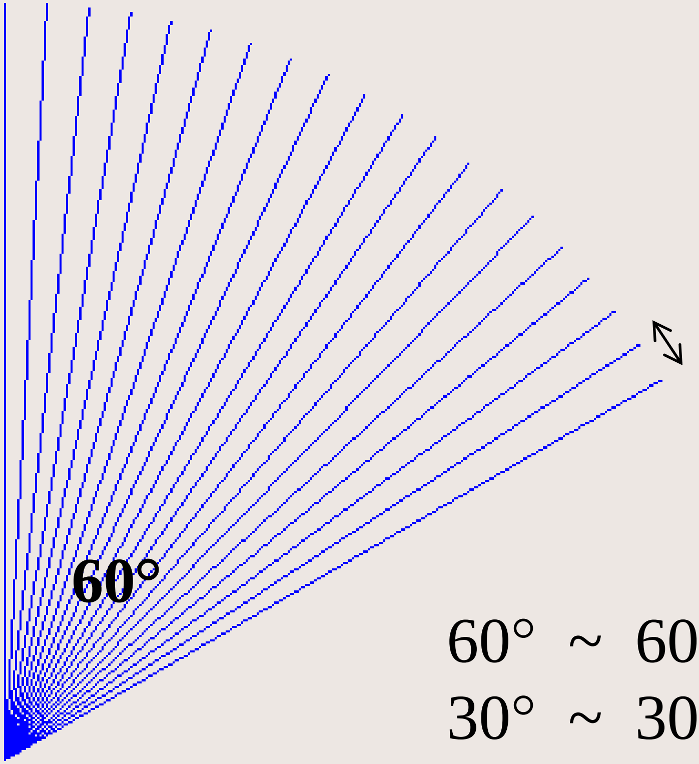
SUPER G (Nettman)
Panavision, IMAX cameras
5-axis Aerial Camera System
91 kg
up to 220 km/h

TASE (Cloud cap tech. - for UAVs), 13x17x11 cm
0.9 kg
0.05° pointing resolution
f=32mm ~ 500pixels
http://www.cloudcaptech.com

Helicopter – external demo

Gimbal stabilization - demo

Stabilizer precision/resolution



prec = 0.05°

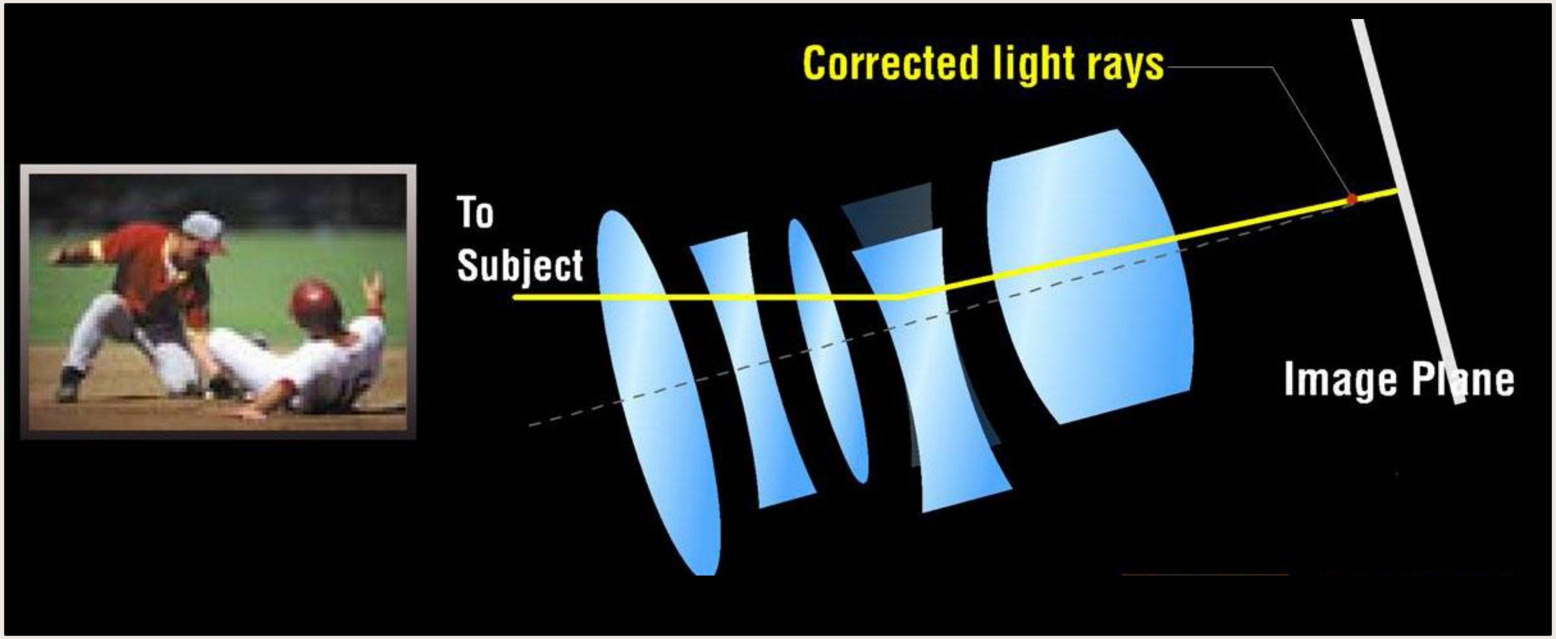
60° ~ 60/0.05 = 1200 pixels

30° ~ 30/0.05 = 600 pixels

Hardware-based image stabilization

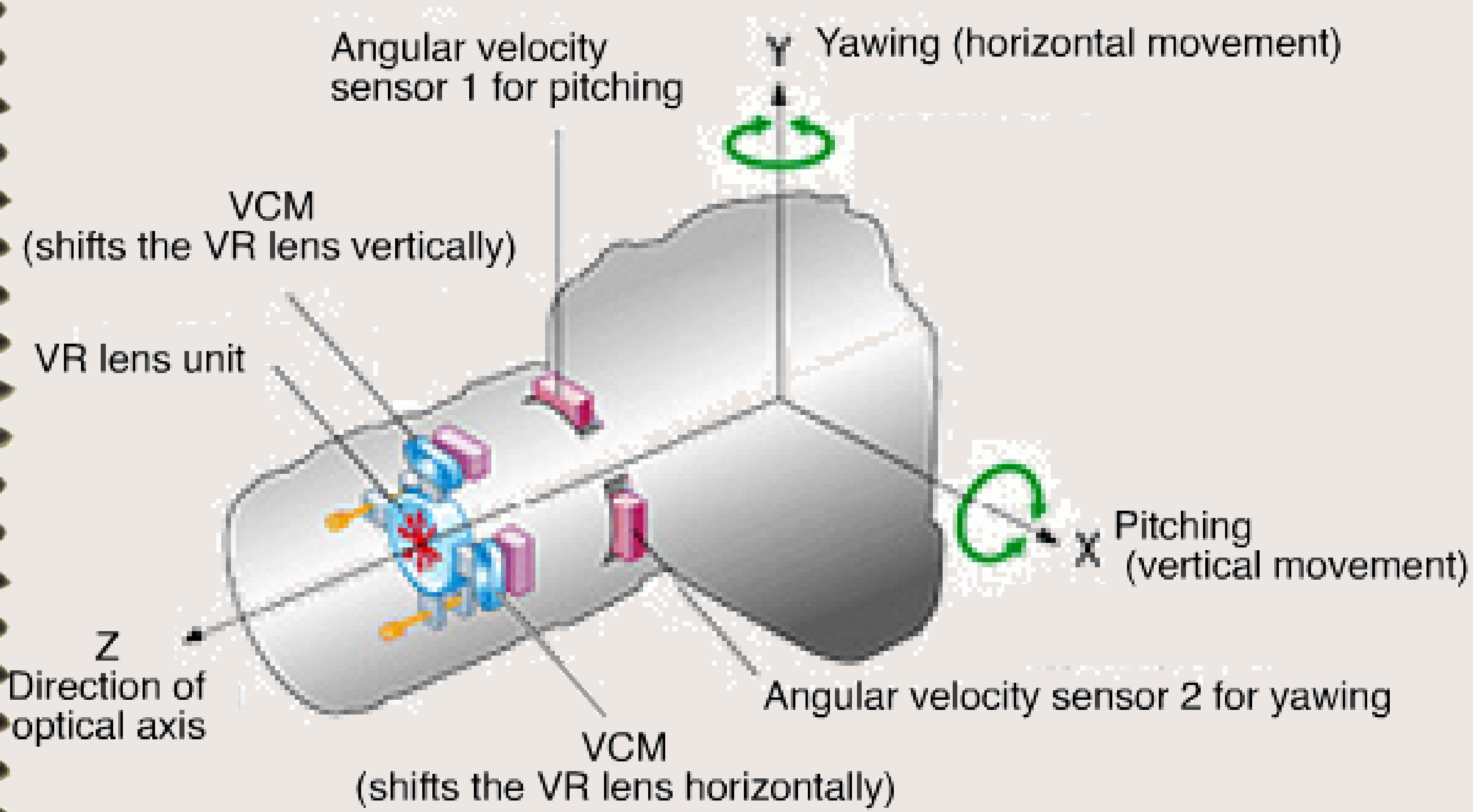
- Optical image stabilization
  - – Canon (IS - Image stabilization)
  - – Nikon (VR – Vibration Reduction)
  - – Panasonic, Leica, Sony, Sigma, Tamron, Pentax ....
- Moving sensor
  - – Konika-Minolta (Sony □–line)
  - – Olympus

Image stabilization



www.canon.com

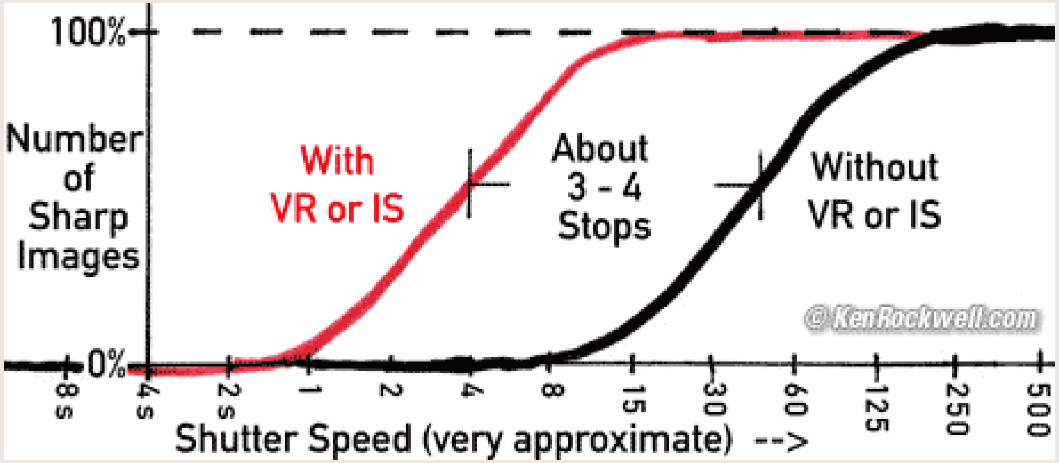
Nikon VR


Angular velocity sensor 1 for pitching
Y Yawing (horizontal movement)
VCM (shifts the VR lens vertically)
VR lens unit
Pitching (vertical movement)
X
Z Direction of optical axis
Angular velocity sensor 2 for yawing
VCM (shifts the VR lens horizontally)

http://imaging.nikon.com/products/imaging/technology/vr/index.htm

Success rate with/without image stabilization

- Rule of 1/f
- Success rate



- 3-4 stops ⇨ 8-16 times longer exposure and size of convolution kernel ~ 4-8 pixels

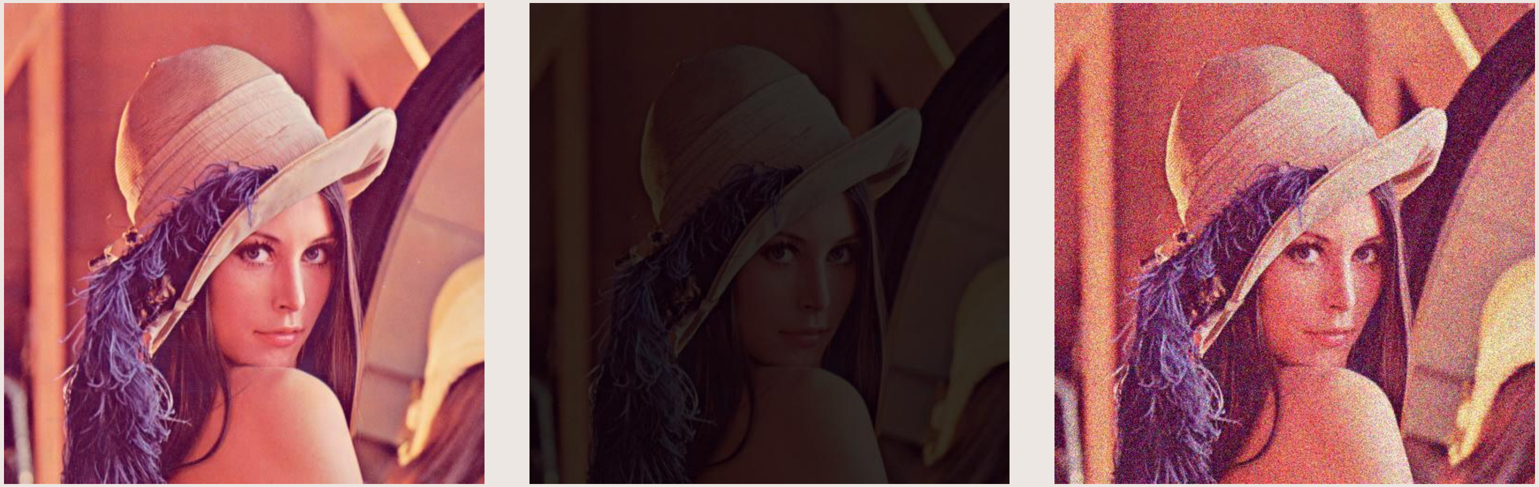
Hardware-based stabilization summary

	+	-
Boosting ISO	Cheap, almost no additional hardware	Noisy image
Gyro-stabilized gimbals	Universal, can stabilize large motions	Heavy, expensive
OIS systems (Optical image stabilization)	3-4 stops improvement	High energy consumption, no „roll“ stabilization, in all lenses – expensive
Moving sensor stabilization	Roll stabilization, one device for all lenses	

Talk outline

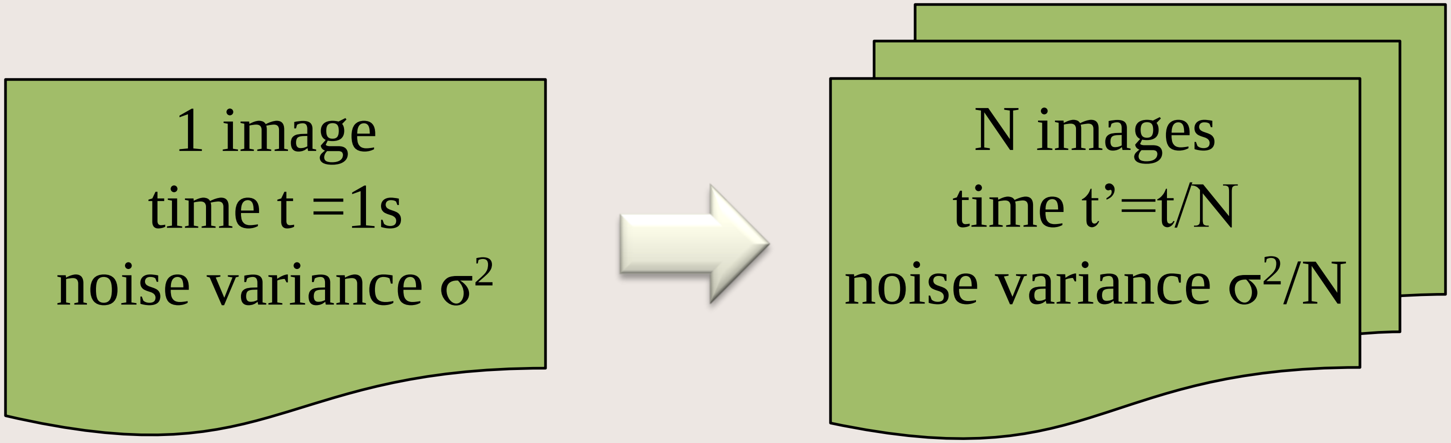
- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

underexposed = noisy



- Photon noise → $\mathrm{SNR} \sim \mathrm{SNR}_0 * t$
- increasing contrast amplifies noise

Multiple noisy images



1 image
time t =1s
noise variance σ^2

N images
time t’=t/N
noise variance σ^2/N

- Noise variance (and SNR) of the sum of N images is the same as of the original image
- The difficult part is registration

Multiple noisy images

- Main problem slow read-out
- ¼ × 1/60s (15 times, ~4 stops)
  15 images → 15*(1/3) = 5s
- Faster chips in near future allow avering of 4 - 8 images.

Talk outline

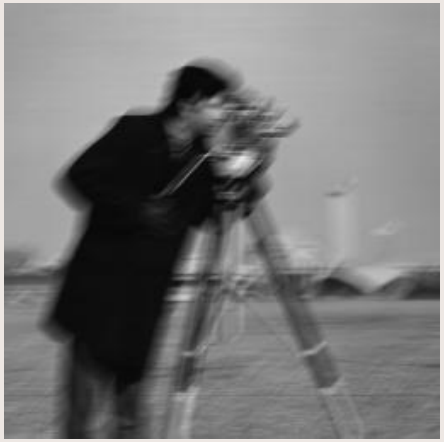
- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

Restoration using known PSF

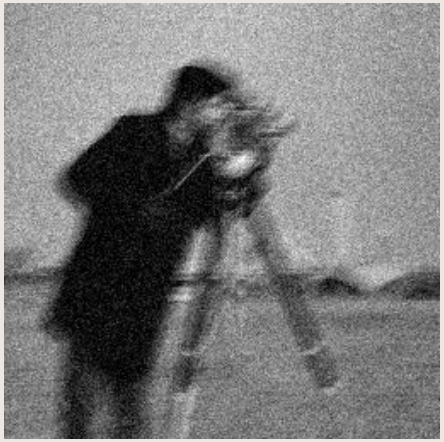
- Degradation model – for homogenous blur

u

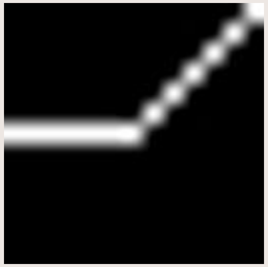
u □ h



z



$$\mathbf{z} = \mathbf{u} * \mathbf{h} + \mathbf{n}$$



h

Solution of deconvolution problem

- Model

$$\mathbf{z} = \mathbf{u} * \mathbf{h} + \mathbf{n}$$

- 2 viewes
  - Minimization of the model least squares error (least squares fitting)
  - Bayesian MAP estimation

Minimization of LS error

- Image model $\mathbf{z} = \mathbf{u} * \mathbf{h} + \mathbf{n}$
- Minimize

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{u} * \mathbf{h} - \mathbf{z}\|^2 + \lambda Q(\mathbf{u})$$

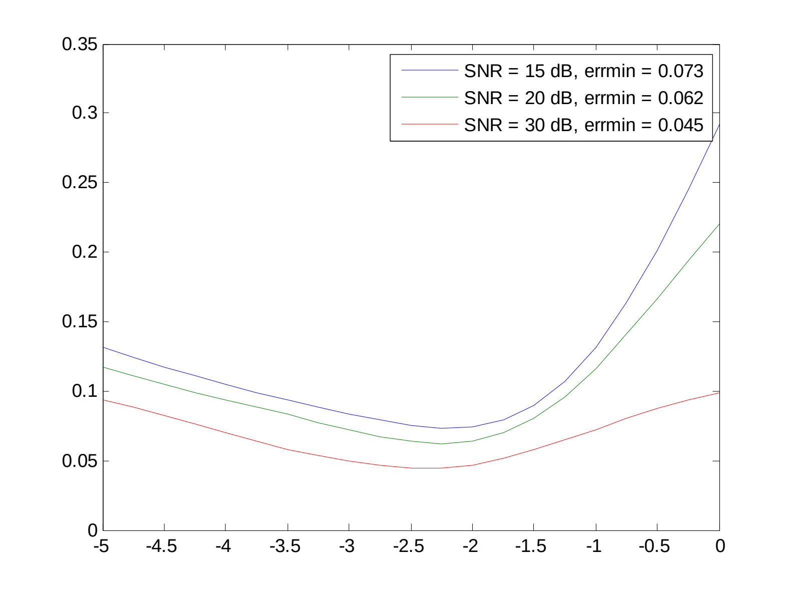
$$Q(\mathbf{u}) = \int |\nabla \mathbf{u}|^2 \quad Q(\mathbf{u}) = \int |\nabla \mathbf{u}|$$

- Regularization constant □ - no one correct value

Role of regularization parameter

$$\min_{\mathbf{u}} E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{u} * \mathbf{h} - \mathbf{z}\|^2 + \lambda \int |\nabla \mathbf{u}|^2$$

Mean least squares error /pixel



- log □

Matrix notation

Tikhonov reg. c = [1 -1]

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{u} * \mathbf{h} - \mathbf{z}\|^2 + \lambda\|\mathbf{c} * \mathbf{u}\|^2$$

u, **z** ... vectors

H ... matrix of 2D convolution

C ... regularization matrix

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{H}\mathbf{u} - \mathbf{z}\|^2 + \lambda\|\mathbf{C}\mathbf{u}\|^2$$

Solution in Fourier domain

Tikhonov reg. c = [1 -1]

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{u} * \mathbf{h} - \mathbf{z}\|^2 + \lambda\|\mathbf{c} * \mathbf{u}\|^2$$

Parseval’s theorem

Convolution theorem

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\hat{\mathbf{u}}\hat{\mathbf{h}} - \hat{\mathbf{z}}\|^2 + \lambda\|\hat{\mathbf{c}}\hat{\mathbf{u}}\|^2$$

Wiener filter

Bayesian view – MAP estimate

- MAP – Maximum a posteriori probability
- Maximize (using Bayes formula)

$$p(\mathbf{u}|\mathbf{z},\mathbf{h}) \propto p(\mathbf{z}|\mathbf{u},\mathbf{h})p(\mathbf{u})$$

$$\Updownarrow$$

- Minimize

$$-\ln p(\mathbf{u}|\mathbf{z},\mathbf{h}) = -\ln p(\mathbf{z}|\mathbf{u},\mathbf{h}) - \ln p(\mathbf{u})$$

Deconvolution as MAP estimate

- Minimize

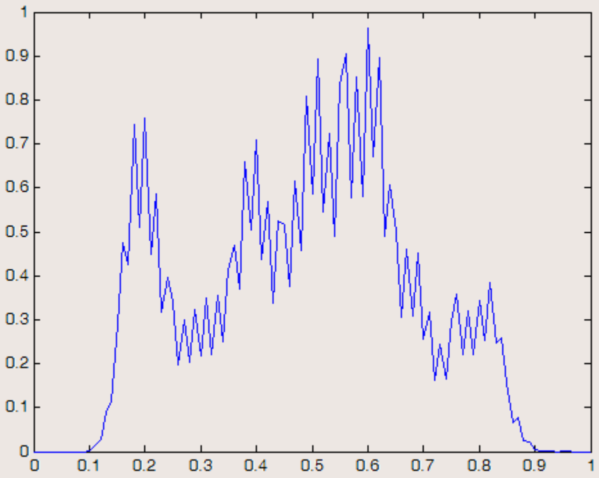
$$-\ln p(\mathbf{u}|\mathbf{z},\mathbf{h}) = \boxed{-\ln p(\mathbf{z}|\mathbf{u},\mathbf{h})} - \ln p(\mathbf{u})$$

$$-\ln p(\mathbf{z}|\mathbf{u},\mathbf{h}) = -\ln \prod_i e^{-\frac{(\mathbf{z}_i - [\mathbf{u}*\mathbf{h}]_i)^2}{2\sigma^2}} = \frac{1}{2\sigma^2}\|\mathbf{z}-\mathbf{u}*\mathbf{h}\|^2$$

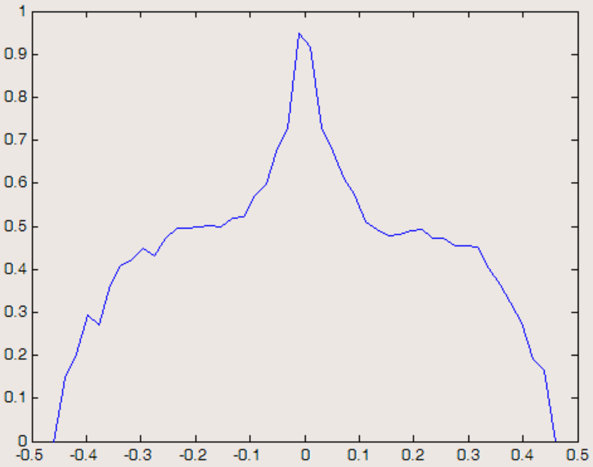
$$\mathbf{z} = \mathbf{u}*\mathbf{h}+\mathbf{n}$$

Image prior (first order statistics)

$$-\ln p(\mathbf{u}) = -\ln \prod_i p(\nabla \mathbf{u}_i) = \sum_i -\ln p(\nabla \mathbf{u}_i)$$



Intensity histogram



Gradient log-histogram

Equivalence of the two views

Tikhonov regularization

$$E(\mathbf{u}) = \frac{1}{2\sigma^2}\|\mathbf{u} * \mathbf{h} - \mathbf{z}\|^2 + \lambda \int |\nabla \mathbf{u}|^2$$

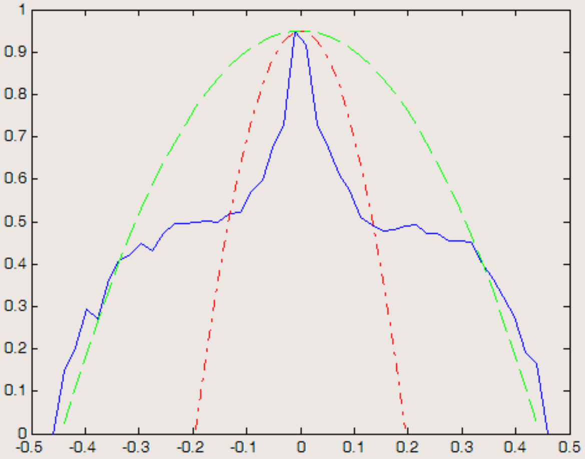
$$-\ln p(\mathbf{u}|\mathbf{z}, \mathbf{h}) = -\ln p(\mathbf{z}|\mathbf{u}, \mathbf{h}) - \ln p(\mathbf{u})$$

where $p(\mathbf{u}) \propto \prod_i e^{\Phi(\nabla \mathbf{u}_i)}$ and $\Phi(\nabla \mathbf{u}_i) = |\nabla \mathbf{u}|^2$

Image priors

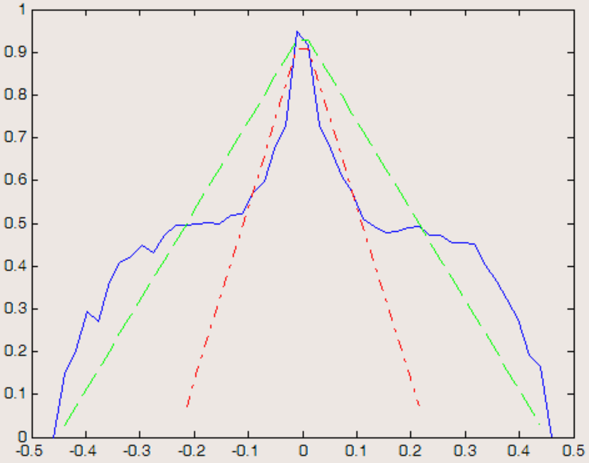
$$Q(\mathbf{u}) = \int |\nabla \mathbf{u}|^2$$

Tikhonov regularization



$$Q(\mathbf{u}) = \int |\nabla \mathbf{u}|$$

TV regularization



Space-variant deblurring

Minimization of

$$E(\mathbf{u}) = \frac{1}{2}\|\mathbf{u} *_v \mathbf{h} - \mathbf{z}\|^2 + \lambda Q(\mathbf{u})$$

$$\mathbf{u} *_v \mathbf{h}\,[x,y] = \int_\Omega \mathbf{u}(x-s, y-t)\mathbf{h}(s,t;x-s,y-t)\,dsdt$$

Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

Single image deblurring - history

- Rob Fergus (2006) building on the work of James Miskin
- Bayesian approach
- Approximation – conditional distributions of PSF and image are considered independent
- Priors on image gradients and blur kernels as a mixture of Gaussians and exponential functions

Marginalization

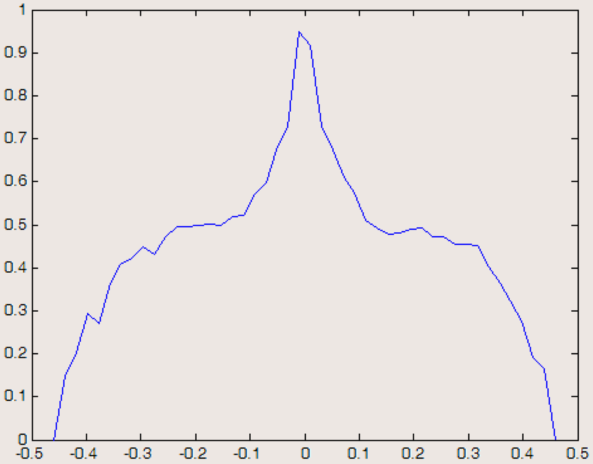
$$\max_{\mathbf{u},\mathbf{h}} \; p(\mathbf{u},\mathbf{h}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{u},\mathbf{h})p(\mathbf{u})p(\mathbf{h})$$

⇓

$$\max_{\mathbf{h}} \; p(\mathbf{h}|\mathbf{z}) = \int p(\mathbf{u},\mathbf{h}|\mathbf{z})du$$

- ln p(**h**|**z**) difficult to compute → approximation

Image prior



Gradient log-histogram
(approximation of ln p(∇$\mathbf{u}_i$))

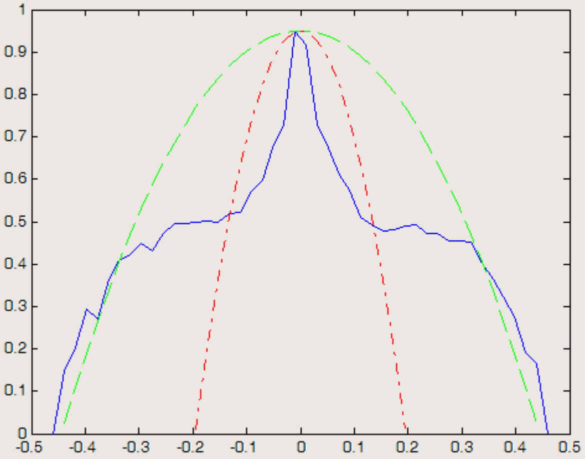
$$p(\mathbf{u}) \propto \prod_i e^{\Phi(\nabla \mathbf{u}_i)}$$

$$-\ln p(\mathbf{u}) = \sum_i -\Phi(\nabla \mathbf{u}_i)$$

Image priors

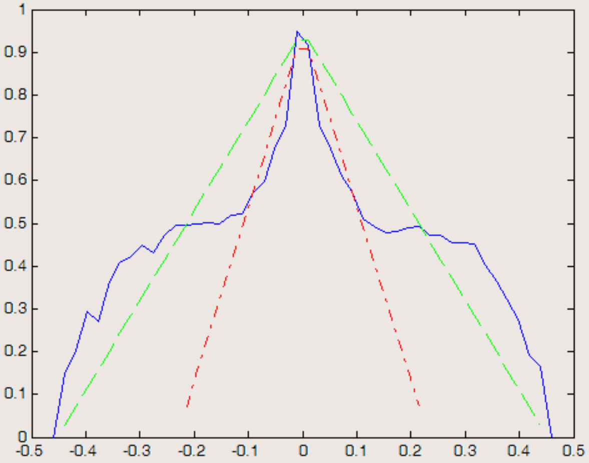
$$Q(\mathbf{u}) = \int |\nabla \mathbf{u}|^2$$

Tikhonov regularization



$$Q(\mathbf{u}) = \int |\nabla \mathbf{u}|$$

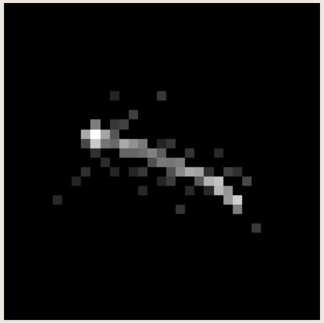
TV regularization



Approximation by Gaussian mix

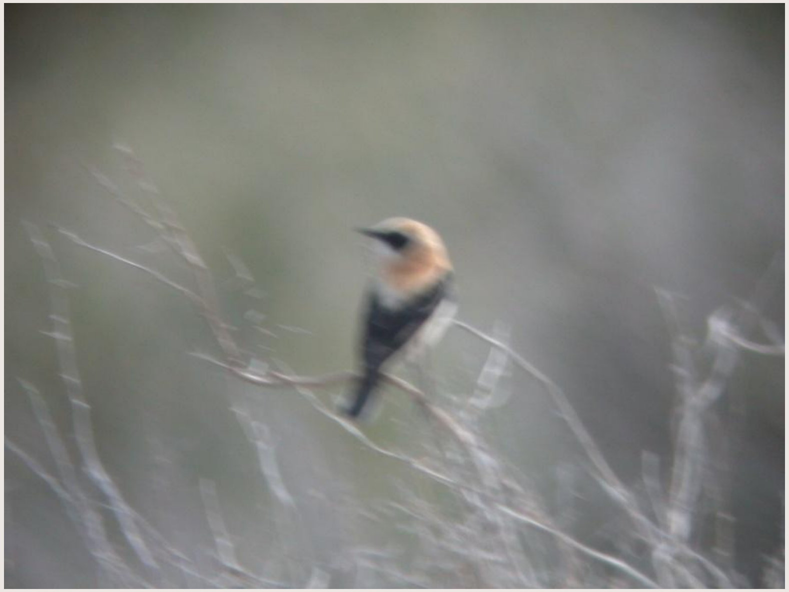
PSF prior

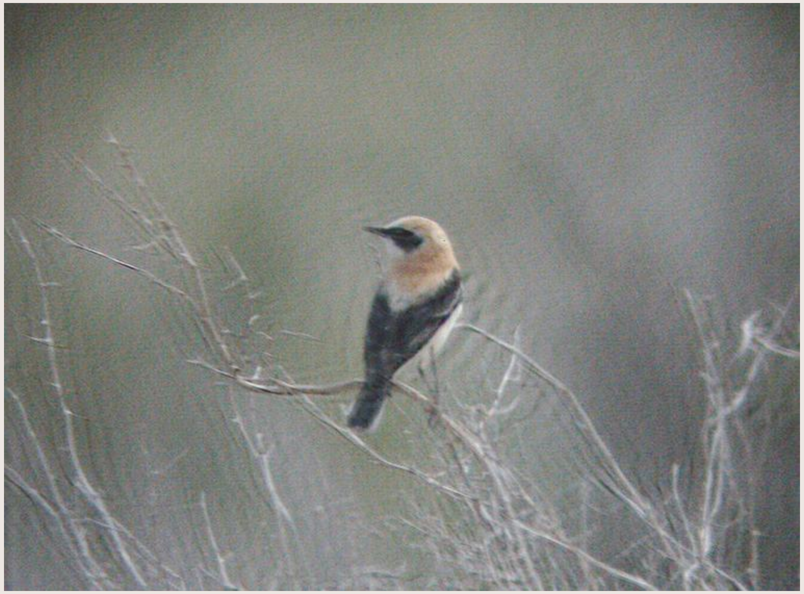
$$p(\mathbf{h}) \propto \prod_i \sum_k \beta_k e^{-\tau_k \mathbf{h}_i}$$

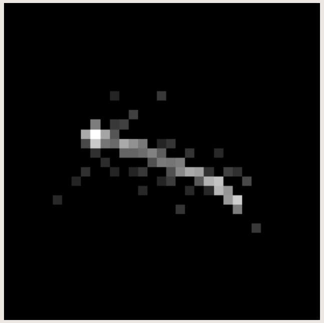


$$-\ln p(\mathbf{h}) \propto \sum_i -\ln \sum_k \beta_k e^{-\tau_k \mathbf{h}_i}$$

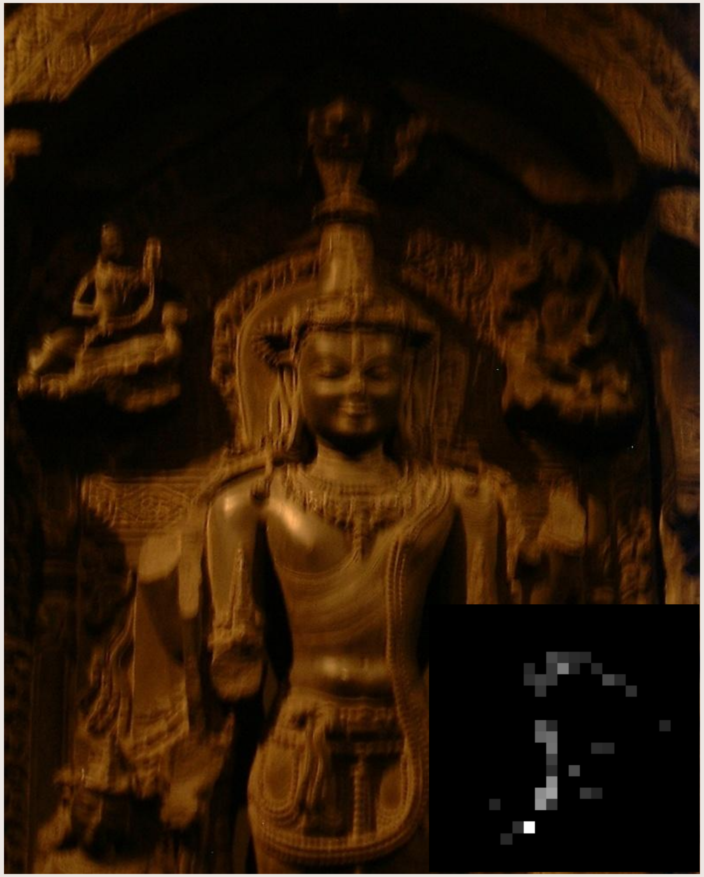
Rob Fergus (Example I)

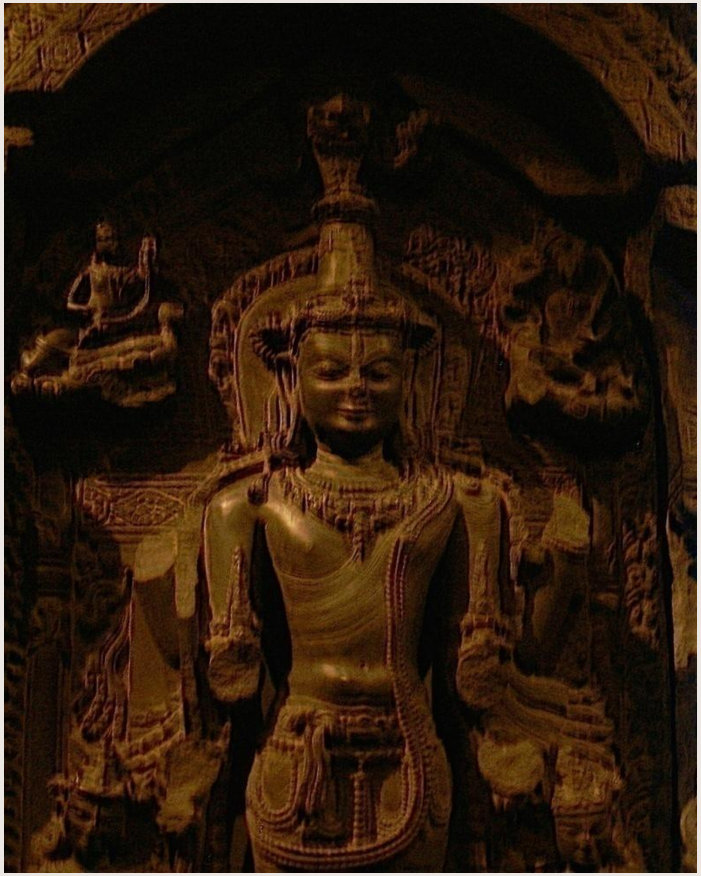






Rob Fergus (Example II)





MAP approach at SIGGRAPH 08

$$p(\mathbf{u},\mathbf{h}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{u},\mathbf{h})p(\mathbf{u})p(\mathbf{h})$$

$$-\ln p(\mathbf{u},\mathbf{h}|\mathbf{z}) = -\ln p(\mathbf{z}|\mathbf{u},\mathbf{h}) - \ln p(\mathbf{u}) - \ln p(\mathbf{h})$$

$$\frac{1}{\sigma^2}\|\mathbf{z}-\mathbf{u}*\mathbf{h}\|^2 + \sum_i \Phi(\partial \mathbf{u}_i) + \tau\|\mathbf{h}\|_1 + \cdots$$

Single image deblurring - summary

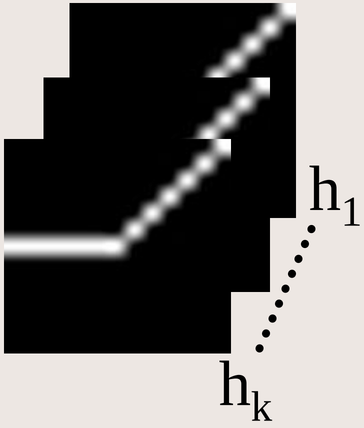
- Difficult, underdetermined problem
- Needs strong priors on both image and convolution kernel
- First really successful algoritm (Fergus 2006) uses Bayesian variational approach, priors are learned from example images
- MAP approaches less stable
- Hardly extensible to space-variant case

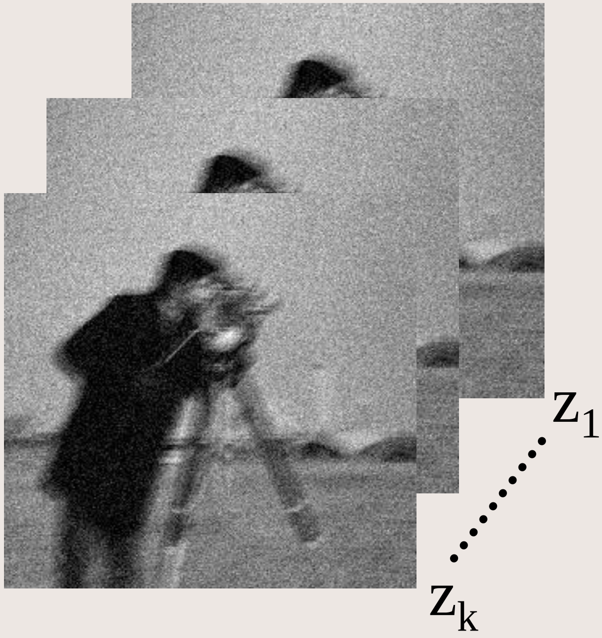
Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

Multiple blurred images

original image





$$[u \quad * \quad h_k](x, y) \; + \quad n_k(x, y) = \; z_k(x, y)$$

Multi-image blind deconvolution

System of integral equations (ill-posed, underdetermined)

$$z_k(x) = (h_k * u)(x) + n_k(x)$$

↓

Energy minimization problem (well-posed)

$$E(u, \{h_i\}) = \frac{1}{2}\sum_{i=1}^{K} \|h_i * u - z_i\|^2 + \lambda Q(u) + \gamma R(\{h_i\})$$

Regularization terms

$$E(u,\{h_i\}) = \frac{1}{2}\sum_{i=1}^{K}\|h_i * u - z_i\|^2 + \lambda Q(u) + \gamma R(\{h_i\})$$

$$Q(u) = \int_{\Omega}\phi(|\nabla u|)$$

$$R(\{h_i\}) = \frac{1}{2}\sum_{1\le i,j\le K}\|z_i * h_j - z_j * h_i\|^2$$

PSF regularization

$$R(\{h_i\}) = \frac{1}{2}\sum_{1\le i,j\le K} \|z_i * h_j - z_j * h_i\|^2$$

with one additional constraint $0 \le h_i(x) \le 1, \quad \forall x, i$

$z_1 = u * h_1$ $z_2 = u * h_2$

$z_1 * h_2 = u * h_1 * h_2 - u * h_2 * h_1 = z_2 * h_1$

Incorporating a between-image shift

$$[u * h_k](\tau_k(x,y)) + n_k(x,y) = z_k(x,y)$$

$$[u * g_k](x,y) + n_k(x,y) = z_k(x,y)$$

Alternating minimization (AM)

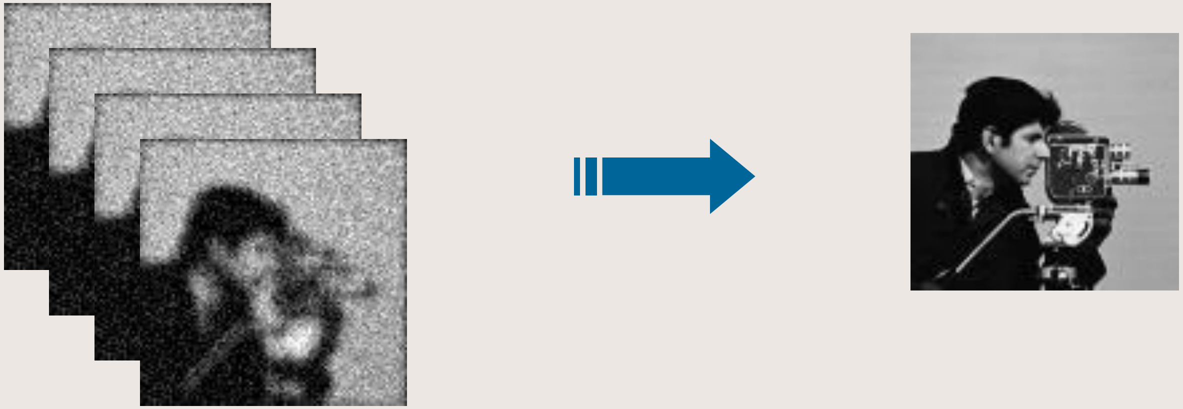
AM of $E(u,\{g_\mathrm{i}\})$ over u and g_i

Input: - blurred images

- estimation of the PSF size

Output: - reconstructed image

- the PSF’s



Multiple blurred images

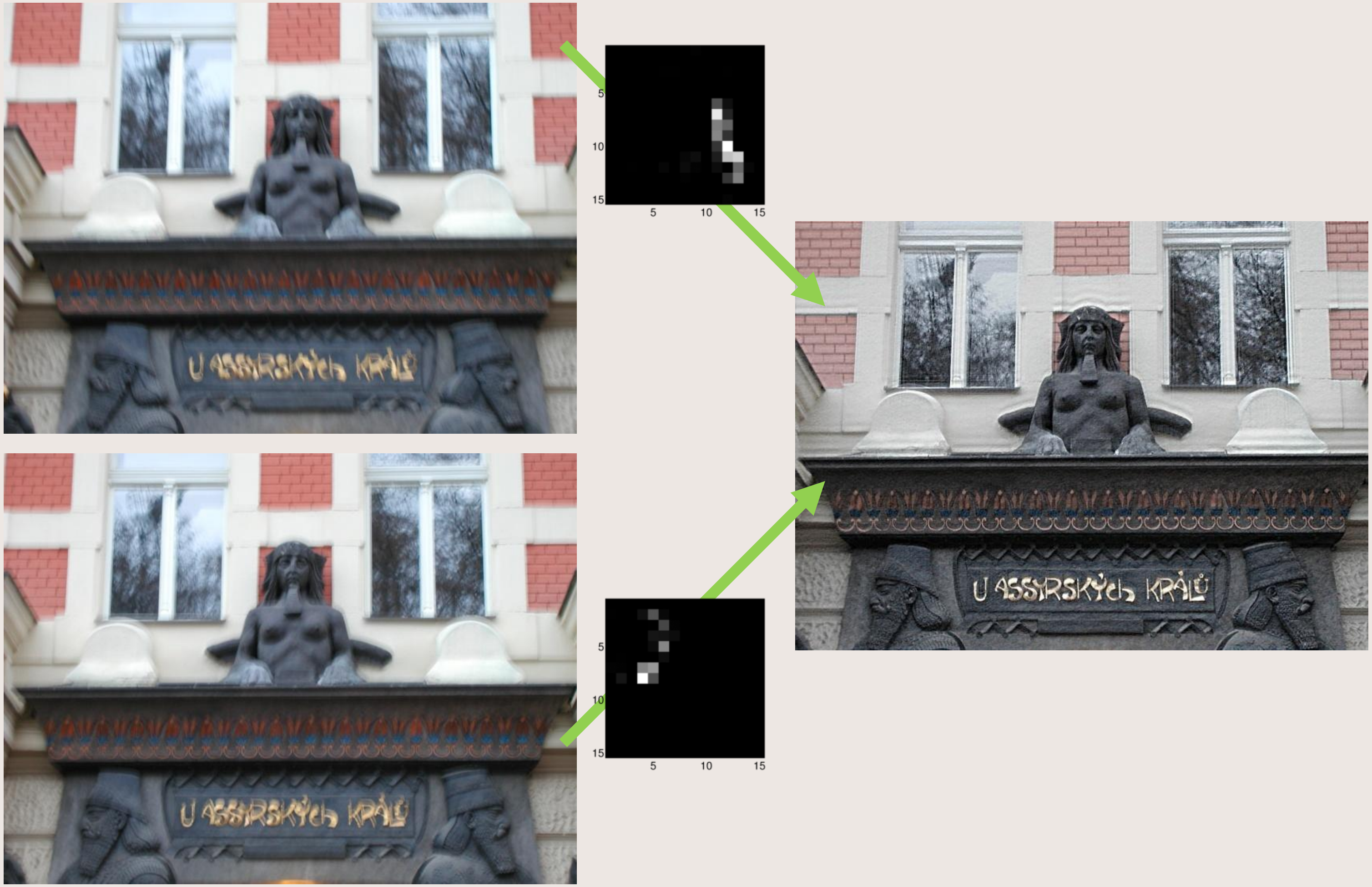
- Multichannel blind deconvolution
- Convolution model of blurring
- Solved by minimization of

$$E(u,\{h_i\}) = \frac{1}{2}\sum_{i=1}^{K}\|h_i * u - z_i\|^2 + \lambda Q(u) + \gamma R(\{h_i\}),$$

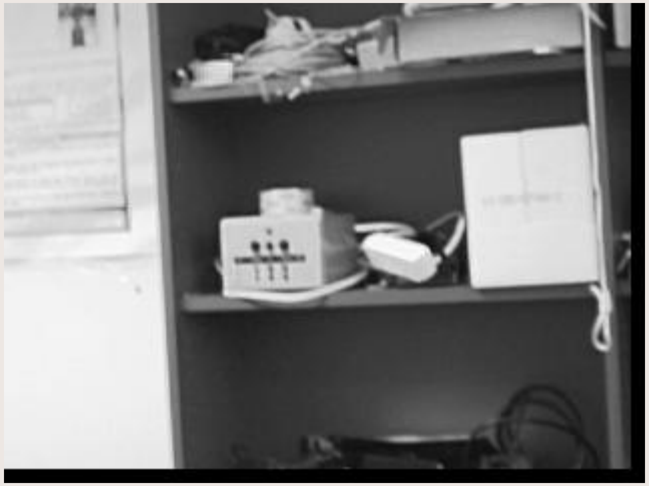
$$Q(\mathbf{u}) = \int_\Omega |\nabla \mathbf{u}|$$

$$R(\{h_i\}) = \frac{1}{2}\sum_{1\le i,j\le K}\|z_i * h_j - z_j * h_i\|^2$$

Multiple blurred images



3-image deblurring (video)



Multi-image deblurring - summary

- Similar to methods used for single-image deconvolution
- Much more data than in single-image case

  → we need less strong priors
- Can be applied to video
- In theory could be applied to space-variant case, but slow

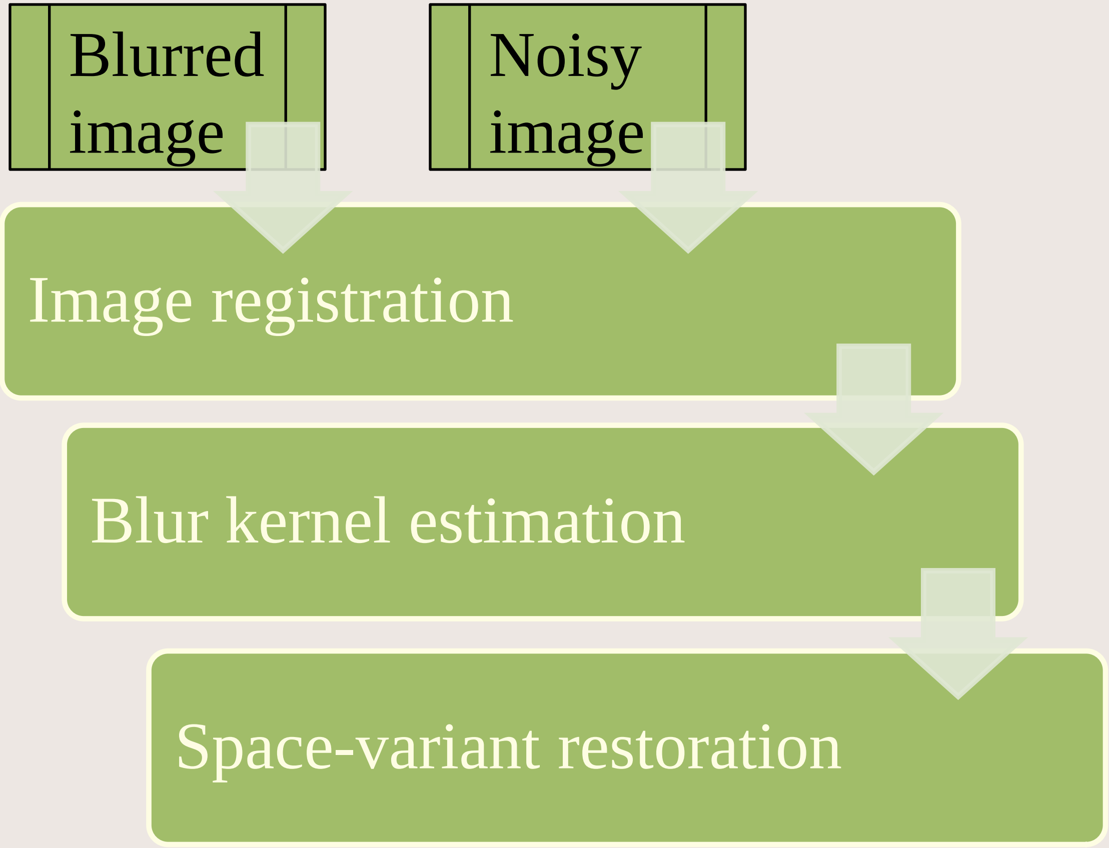
Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

Blurred/underexposed - history

- 2006
  - patented in US
  - since 2006 - several papers assuming convolution model
  - simpler approach only match histograms, no deconvolution
  - Samsung introduced ASR (Advanced shake reduction)

Deblurring algorithm



Blurred image

Noisy image

Image registration

Blur kernel estimation

Space-variant restoration

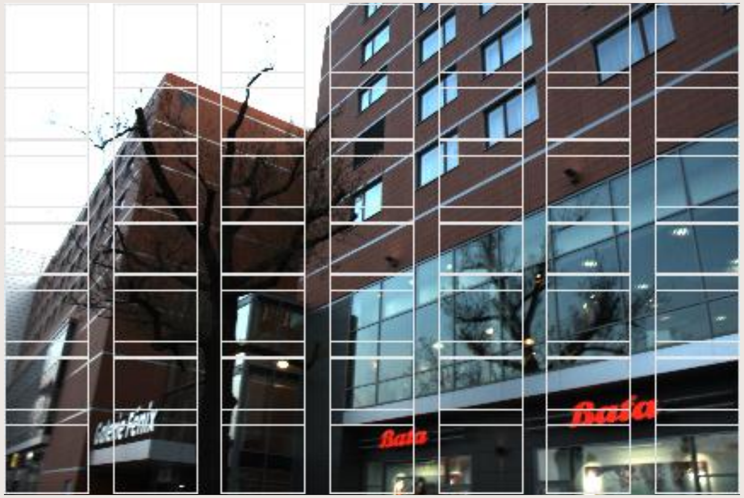
Image registration

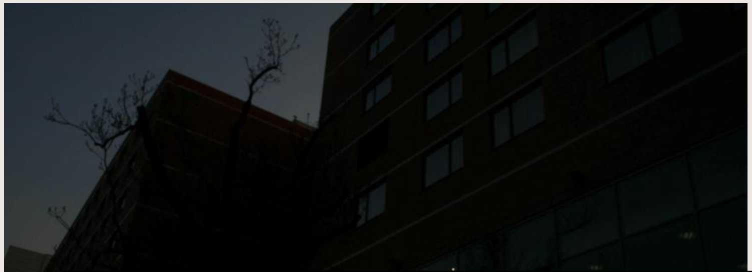
- Small change of camera position – small stereo base

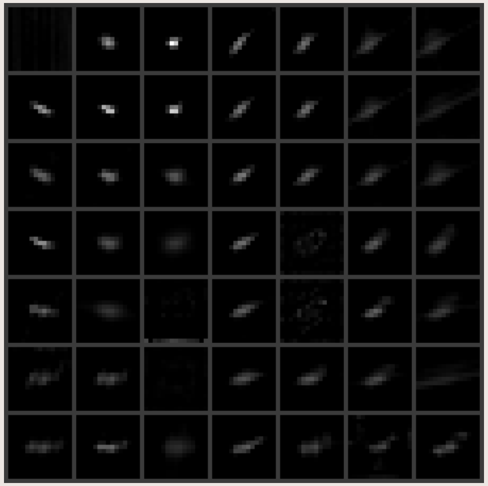
↓

- Static parts of the scene can be modelled by projective tranform found by RANSAC
- Lens distortion can be neglected
- Less important parts of scene can move

Blurred + underexposed results

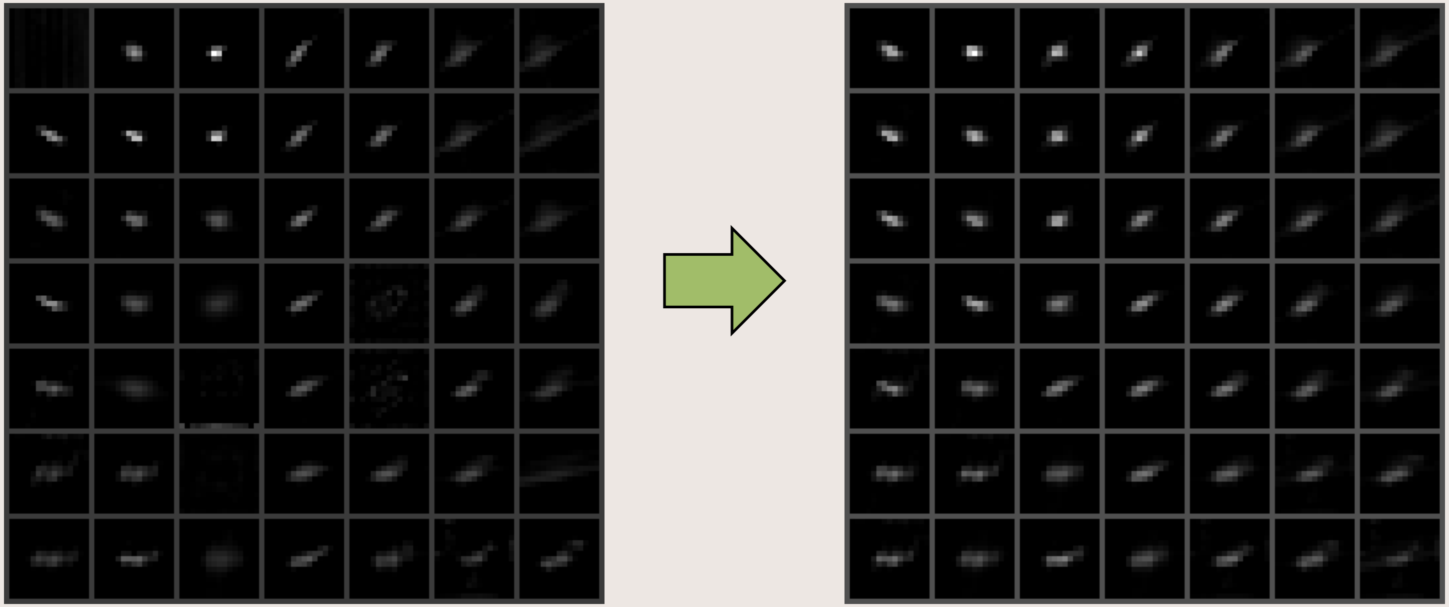






$$\mathbf{h}_{i,j} = \arg\min_{\mathbf{k}} \|\mathbf{u}_{i,j} * \mathbf{k} - \mathbf{z}_{i,j}^T\|^2 + \alpha\|\nabla\mathbf{k}\|^2, \qquad \mathbf{k}(s,t) \geq 0,$$

Blur kernel adjustment



- Regions lacking texture
- Regions of pixel saturation

Restoration

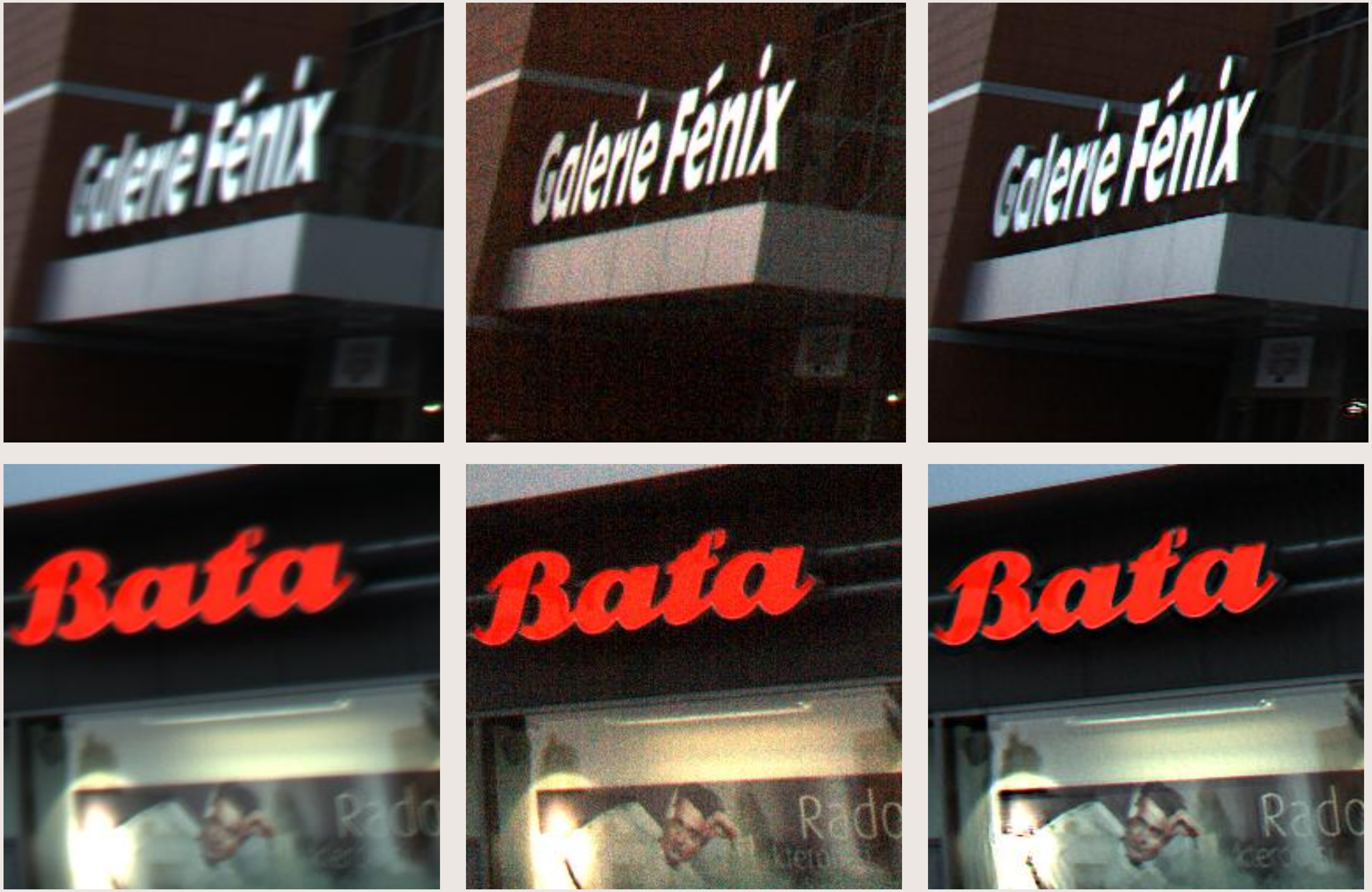
- Minimization of functional

$$E(\mathbf{u}) = \frac{1}{2}\|\mathbf{u} *_v h - \mathbf{z}\|^2 + \lambda \int |\nabla \mathbf{u}|$$

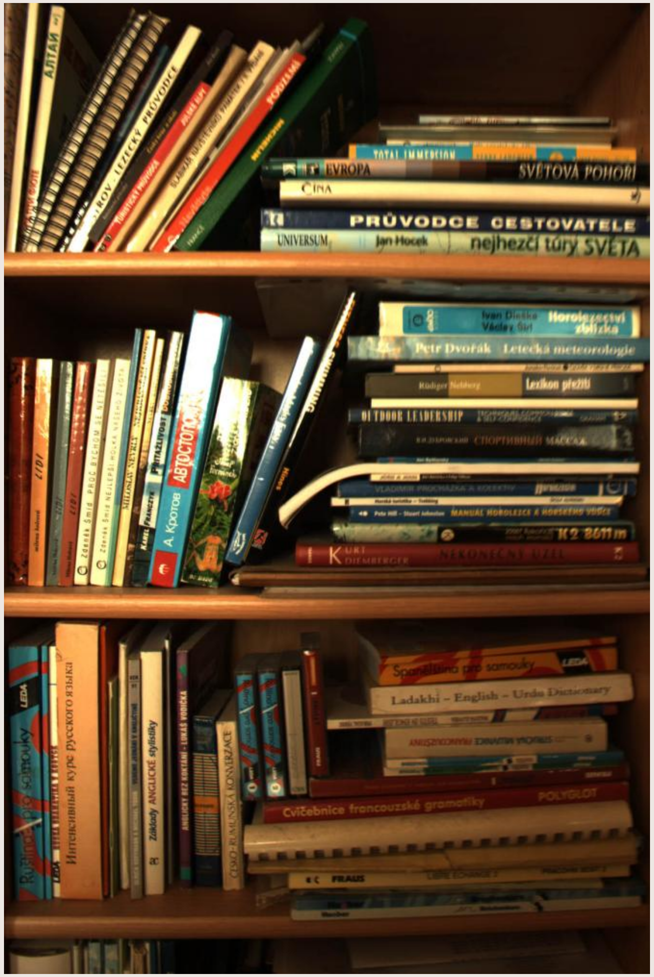
$$\mathbf{u} *_v \mathbf{h}\,[x, y] = \int_\Omega \mathbf{u}(x - s, y - t)\mathbf{h}(s, t; x - s, y - t)\, dsdt$$

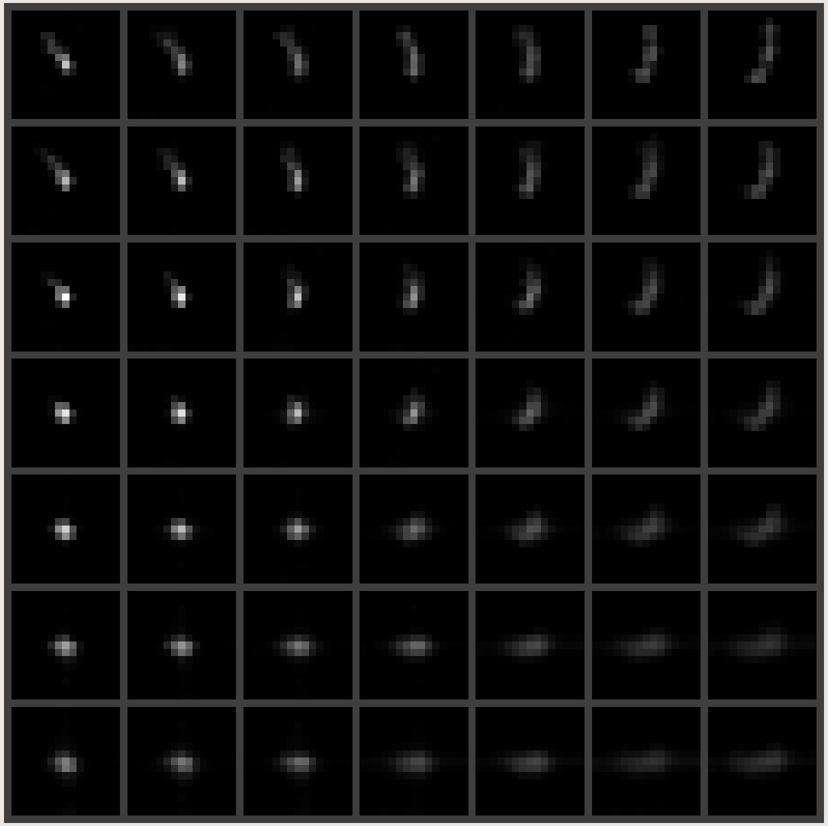
- PSF h interpolated from estimated convolution kernels

Shopping center (details)



Bookcase example





Bookcase (details)

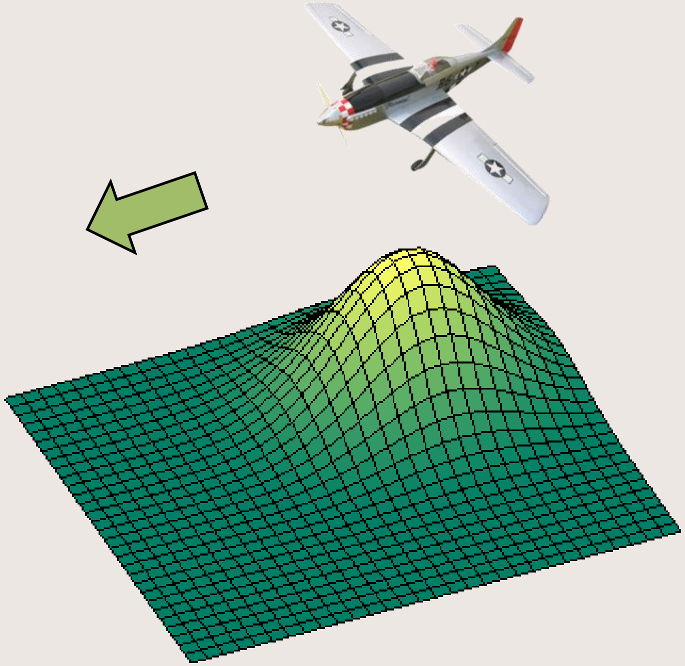
Shot-long exposure - summary

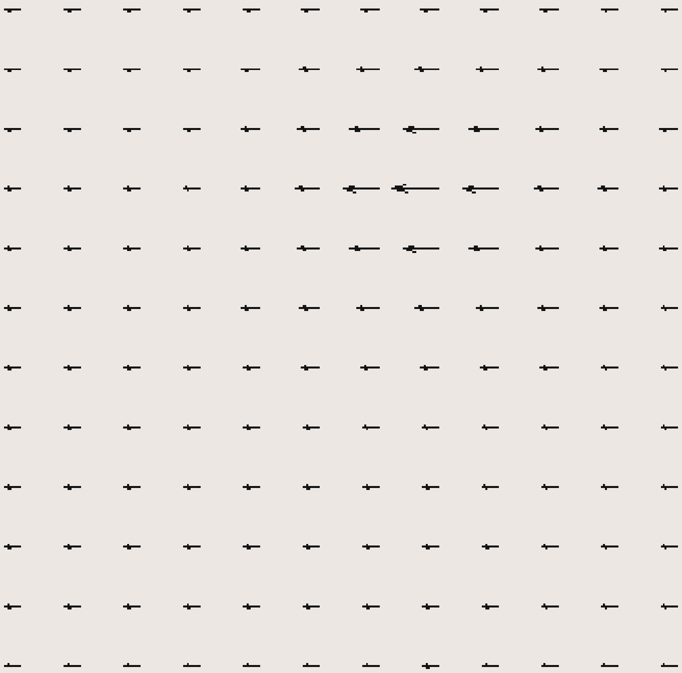
- fast and reliable
- works for space-variant blur
- potential for segmentation of moving objects
- could be also extended to more images

Talk outline

- How to describe the blur? (convolution, velocity field, PSF)
- Hardware-based stabilization
- Software deblurring
  - Multiple underexposed/noisy images
  - Non-blind restoration
  - Single blurred image (deconvolution)
  - Multiple blurred images (deconvolution)
  - One blurred and one underexposed image
  - Multiple images blurred by sideways vibrations

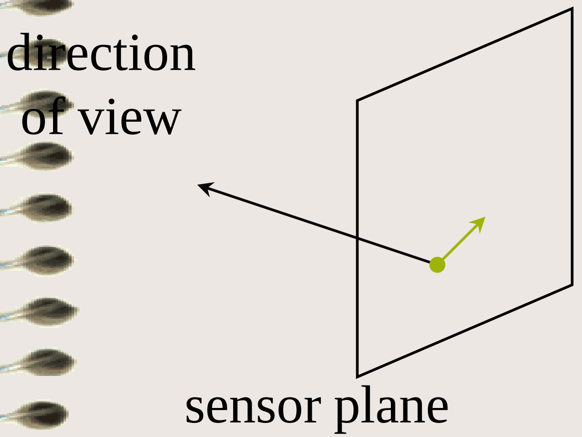
In-plane translation





How we compute camera trajectory

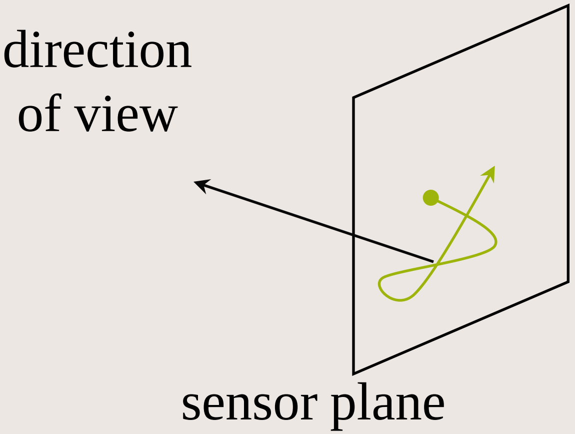
Existing methods



direction of view

sensor plane

Our method



direction of view

sensor plane

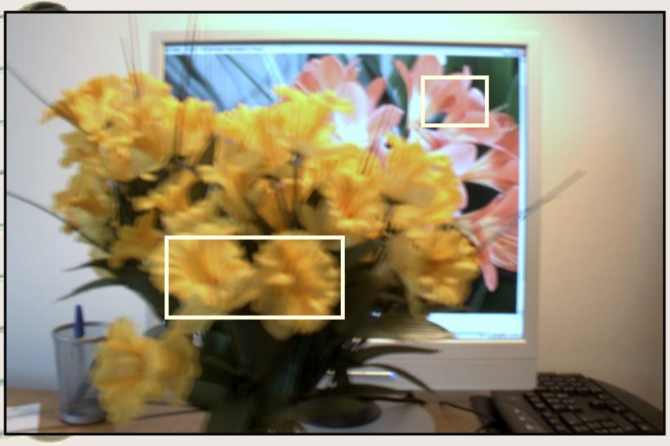
- Point traces (PSF) are scaled versions of camera trajectory
- Estimation of camera motion from the blurred images is possible

Algorithm removing motion blur

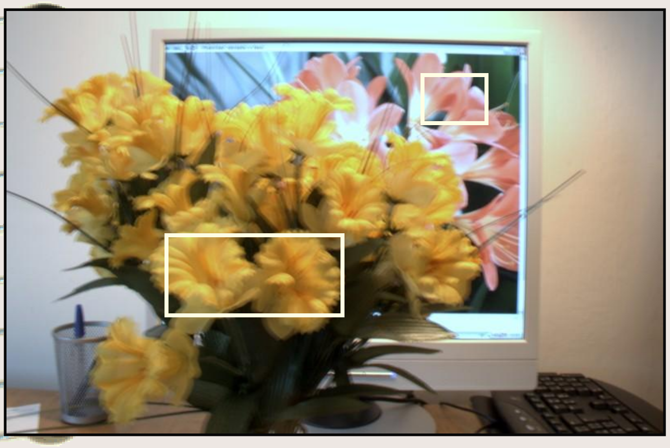
- 3 steps
- Explained on example images
- Algorithm for out-of-focus blur based on similar principle but does not need step 1

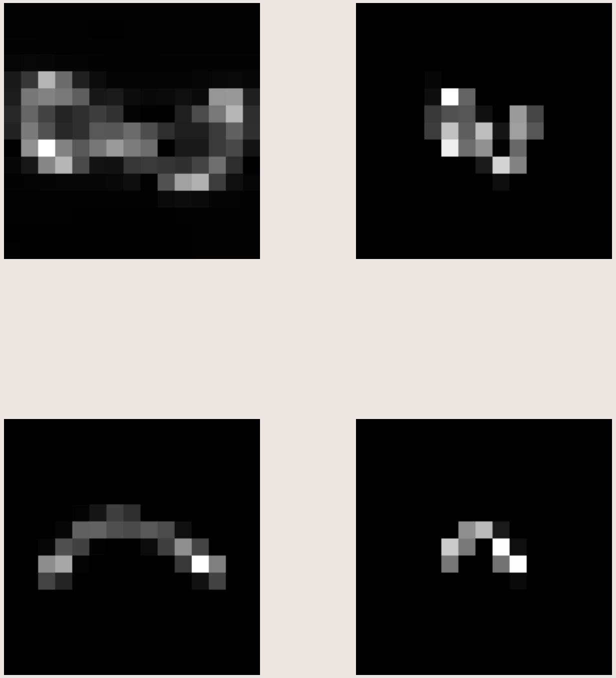
Estimation of camera motion (step I)

$\mathbf{z}_1$



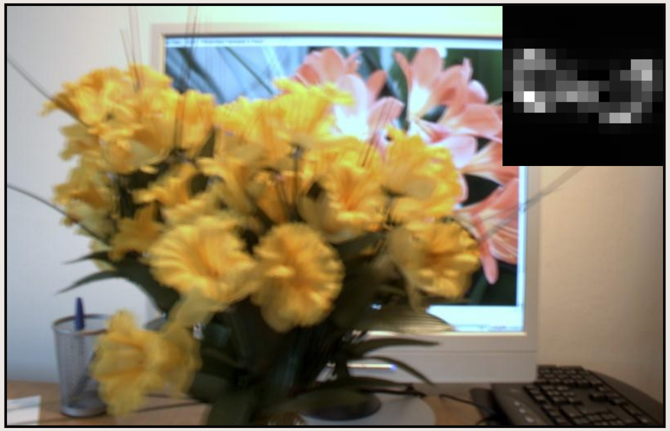
$\mathbf{z}_2$



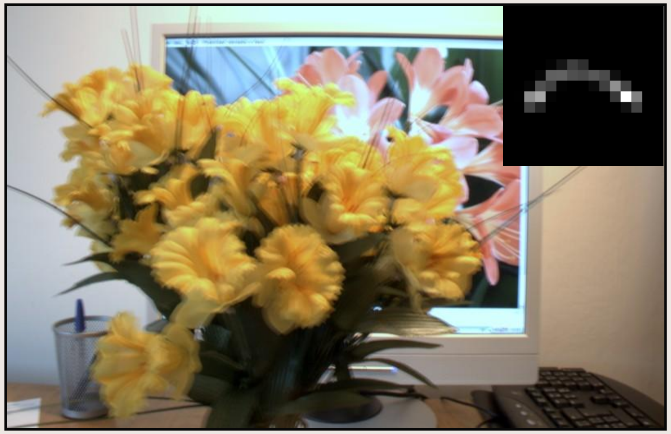


PSF consists of scaled versions of camera trajectory

Rough depth map estimation (step II)

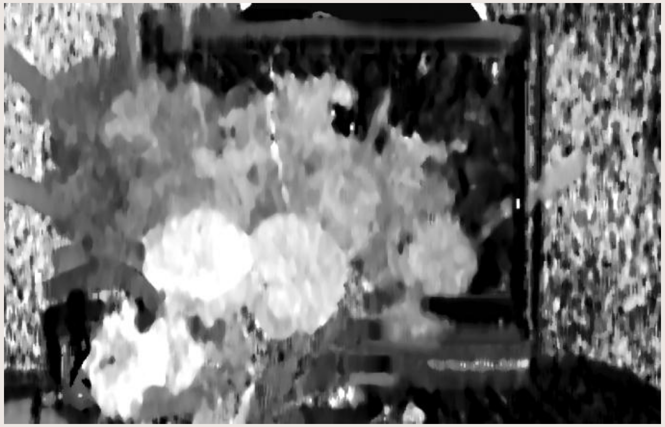


$\mathbf{z}_1$



$\mathbf{z}_2$

$$\mathrm{err}(d) = [\mathbf{z}_1 * h_2(d) - \mathbf{z}_2 * h_1(d)]^2$$



d_0

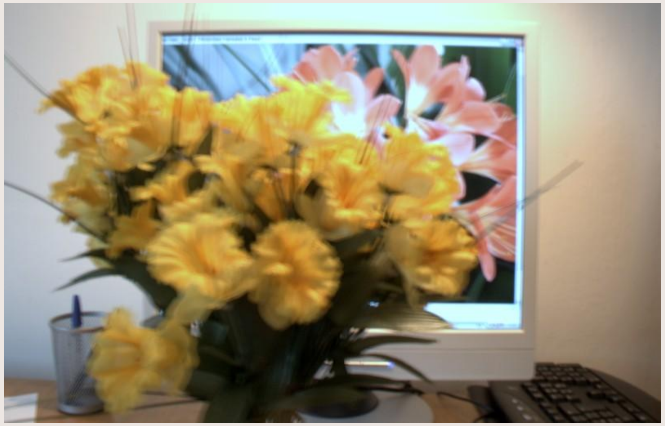
Functional minization (step III)

- Input images $\mathbf{z}_1$, $\mathbf{z}_2$, …
- Minimization initialized by depth map $\mathbf{d}_0$
- Goal – sharp image and depth map computed as $\text{argmin}_{u,d}$ E(u,d)

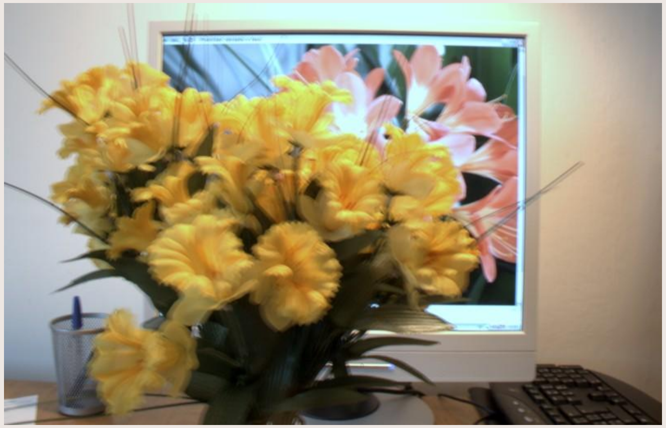
$$E(\mathbf{u},\mathbf{d}) = \frac{1}{2}\sum_{p=1}^{P} \|\mathbf{u} *_v h_p(\mathbf{d}) - \mathbf{z}_p\|^2 + \lambda_u Q(\mathbf{u}) + \lambda_d R(\frac{1}{\mathbf{d}})$$

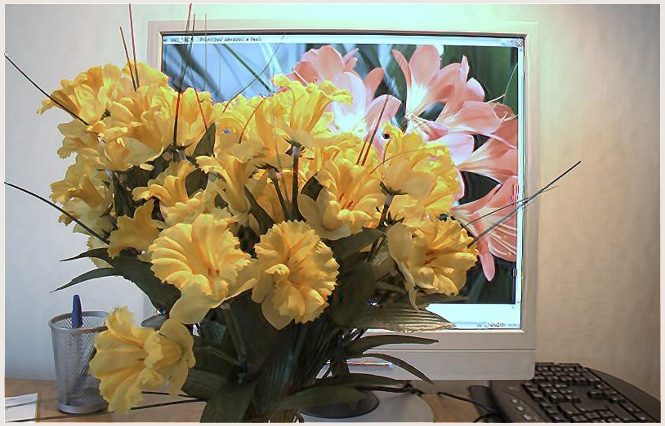
Functional minimization (step III)

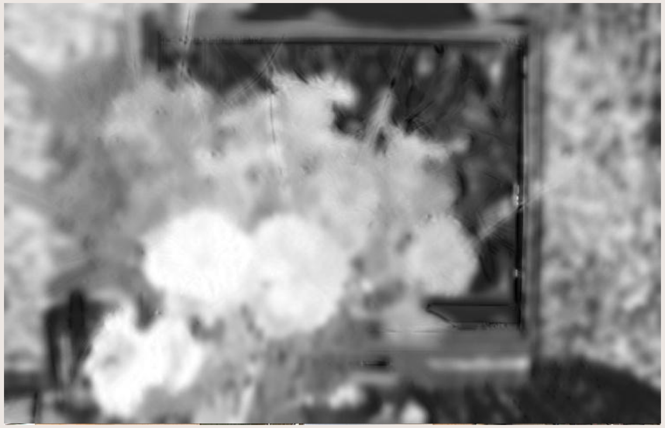
$\mathbf{z}_1$



$\mathbf{z}_2$

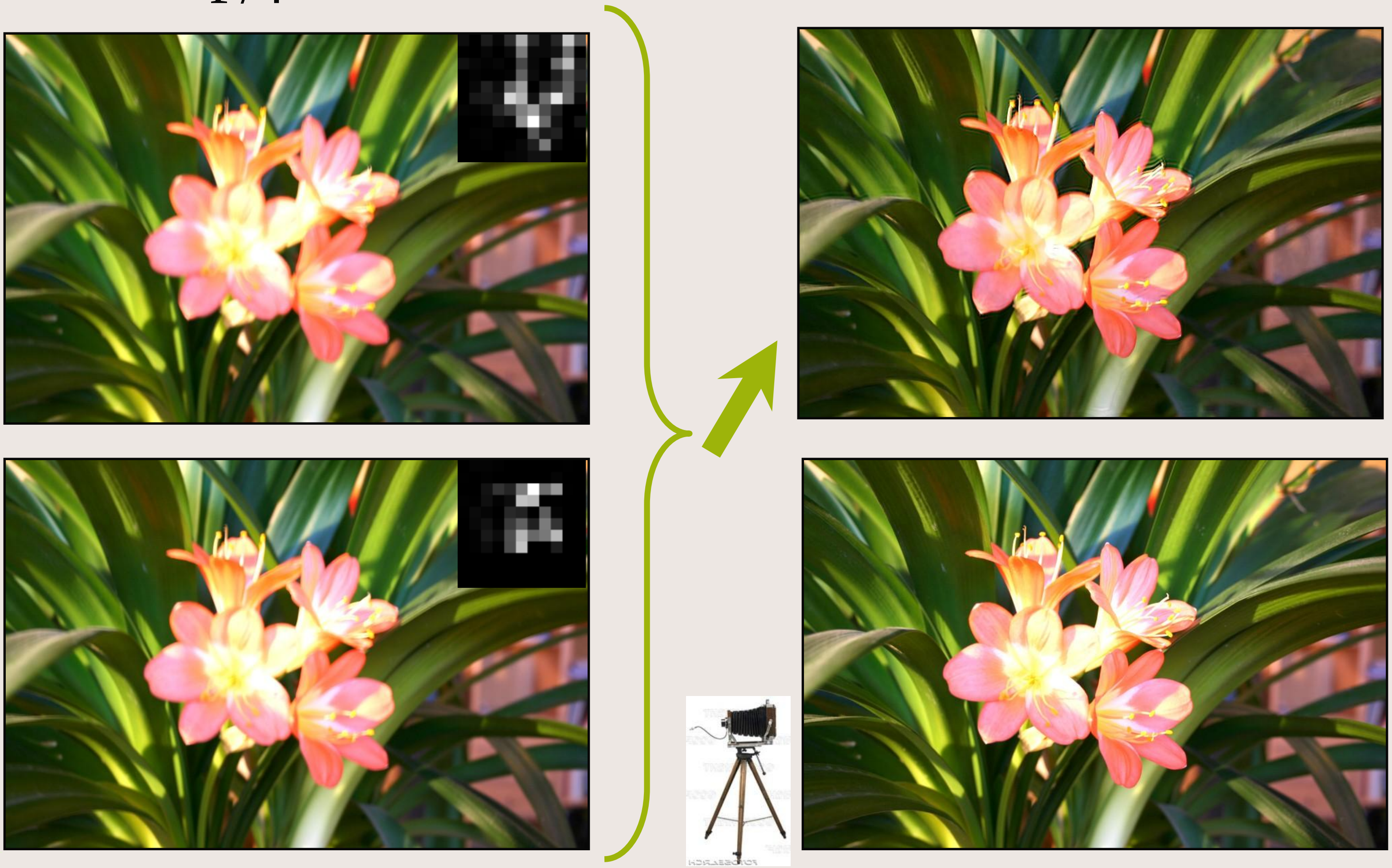






Motion blur + limited depth of focus

F/4



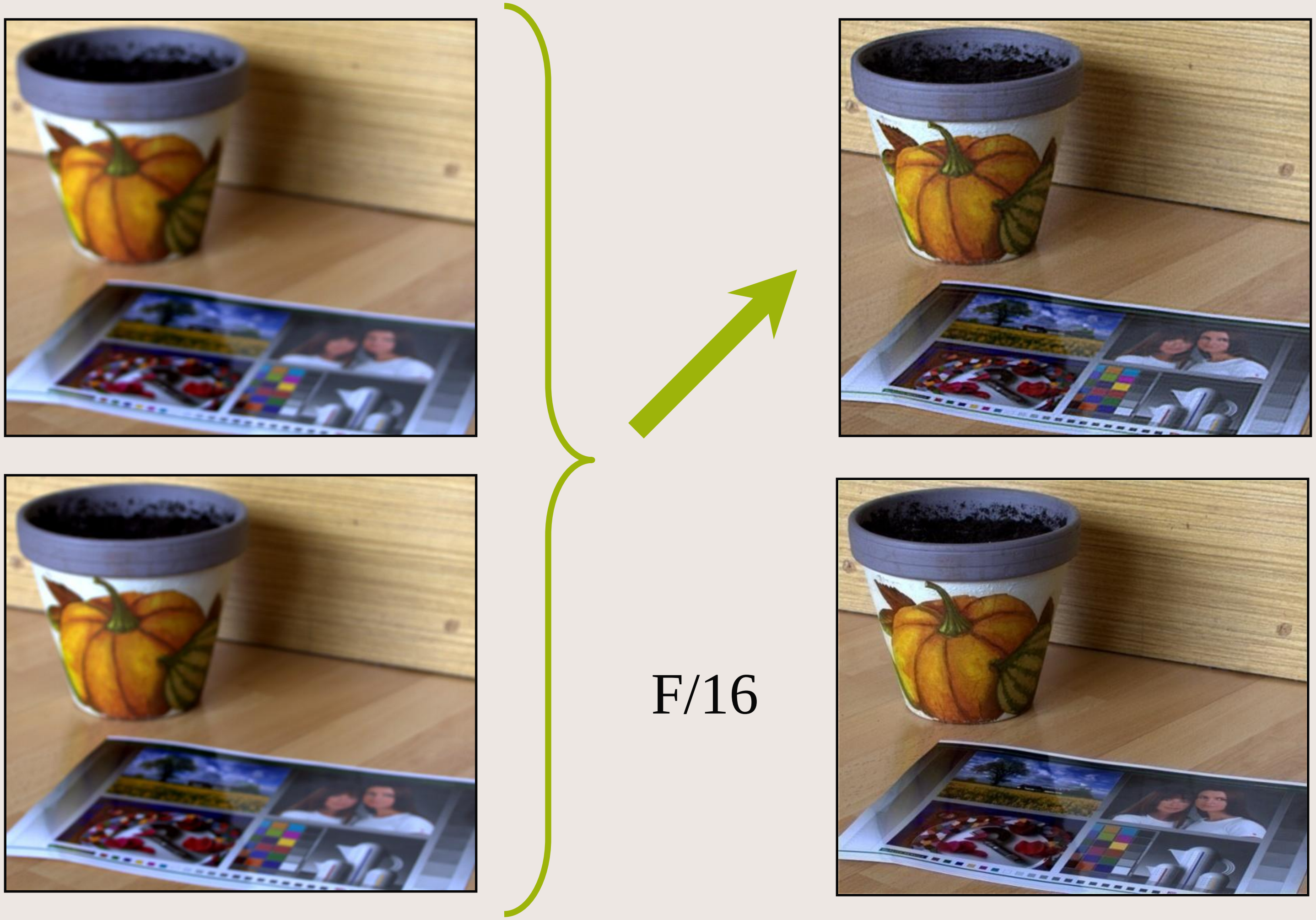
Out-of-focus blur

$\mathbf{z}_1$

(F/5.0)

$\mathbf{z}_2$

(F/6.3)



F/16

Software deblurring in present-day cameras

- Usually no deblurring
- Samsung ASR system
  - may use two images, one underexposed and one blury - only simple algorithm, no „deconvolution“
- Sony DSC-HX1 superimposes six photos (update)
- Reason: speed and energy consumption

Summary/Perspectives

- Denoising – readout speed problems – only way for now, limited EV improvement
- Single image approach – takes time, imprecise PSF, unable to distinguish intentional depth of focus, limited to convolution model
- Multiple blurred images – computationally expensive, fewer artifacts
- Blurred + underexposed image – relatively fast, but (so far) not enough to be used with real deblurring inside a camera

Comparison with OIS

- Can remove roll motion (z-axis rotation) blur
- Handle larger range of EV (exposure values) but with growing number of artifacts
- Ideal solution – both hardware and software image stabilization

Discussion, questions...

Michal Šorel
Academy of Sciences of the Czech Republic
sorel@utia.cas.cz
www.zoi.utia.cas.cz