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 $M \times N$ for greyscale
- Matrix $M \times N \times 3$ for color images
- Formulas shown for greyscale images
Image as a function

- In formulas often a real function of two variables $\mathbb{R}^2 \rightarrow \mathbb{R}^+$, 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: $f \sim 50$ mm
- Telephoto lens: ($f > 100$ mm)
What happens if camera moves?

- Sharp image – movement less than $\frac{1}{2}$ pixel
- Influence of focal length, shutter speed, sensor resolution (pixel density)
- Velocity field, PSF $\sim$ blur kernel
3D camera motion

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

\[ T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad \Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \]
Roll, Yaw, Pitch movements

\[ \Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \]

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

\[ \mathbf{v}(x, y) = \frac{1}{d(x, y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \\ xy & -xy & -x \end{bmatrix} \begin{bmatrix} -1 - x^2 \\ y \\ -x \end{bmatrix} \]

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

\[
v(x, y) = \frac{1}{d(x, y)} \left[ \begin{array}{ccc} -1 & 0 & x \\ 0 & -1 & y \end{array} \right] T + \\
\left[ \begin{array}{ccc} xy & -1 - x^2 & y \\ 1 + y^2 & -xy & -x \end{array} \right] \Omega
\]
Translation

\[ \mathbf{v}(x, y) = \frac{1}{d(x, y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T + \ldots \]
Translation along optical axis

\[ \mathbf{v}(x, y) = \frac{1}{d(x, y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T + \ldots \]
Point-spread function - PSF

Integration of velocity field → PSF

\[ h(s,t; x_2, y_2) \]

\[ h(s,t; x_1, y_1) \]
Mathematical model of blurring

\[ u \ast_v h \ [x, y] = \int_{\Omega} u(x - s, y - t) h(s, t; x, y) \, ds \, dt \]

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

$h(s,t; x_1, y_1)$

$h(s,t; x_2, y_2)$

$h(s,t; x_3, y_3)$
Blur description – summary (I)

- What we have learned
  - What happens when a camera is moving
  - 4 motion components
  - Velocity field
  - How PSF describes the blur and its relation with velocity field
Blur description – summary (II)

<table>
<thead>
<tr>
<th>Motion component</th>
<th>Dependence on distance</th>
<th>Space-variant blur</th>
</tr>
</thead>
<tbody>
<tr>
<td>YAW, PITCH (x,y-axis rotation)</td>
<td>NO</td>
<td>YES (a bit)</td>
</tr>
<tr>
<td>ROLL (z-axis rotation)</td>
<td>NO</td>
<td>YES (a lot)</td>
</tr>
<tr>
<td>X,Y-axis translation</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Z-axis translation</td>
<td>YES</td>
<td>YES (a lot)</td>
</tr>
</tbody>
</table>
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 $\Rightarrow$ ISO 200
  $\sim f$-number/2, $2^t$ (1 EV or 1 stop)
- ISO 100 $\Rightarrow$ ISO 3200 $\sim 32^t$ (5 stops)

Photon noise (Poisson) $\quad$ SNR $\sim$ SNR$_0^* t$

$\text{SNR}_{1600} = \text{SNR}_{100} / 16$ $(-12 \text{ dB})$
$\text{SNR}_{3200} = \text{SNR}_{100} / 32$ $(-15 \text{ dB})$
SNR

30 dB

20 dB

15 dB

5 dB
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 ~ 500 pixels
http://www.cloudcaptech.com
Helicopter – external demo
Gimbal stabilization - demo
Stabilizer precision/resolution

\[\text{prec} = 0.05^\circ\]

- \(60^\circ \sim 60/0.05 = 1200\) pixels
- \(30^\circ \sim 30/0.05 = 600\) 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

Success rate with/without image stabilization

- Rule of 1/f
- Success rate

- 3-4 stops $\rightarrow$ 8-16 times longer exposure and size of convolution kernel $\sim$ 4-8 pixels
## Hardware-based stabilization summary

<table>
<thead>
<tr>
<th></th>
<th>+</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosting ISO</td>
<td>Cheap, almost no additional hardware</td>
<td>Noisy image</td>
</tr>
<tr>
<td>Gyro-stabilized gimbals</td>
<td>Universal, can stabilize large motions</td>
<td>Heavy, expensive</td>
</tr>
<tr>
<td>OIS systems (Optical image stabilization)</td>
<td>3-4 stops improvement</td>
<td>High energy consumption, no „roll“ stabilization, in all lenses – expensive</td>
</tr>
<tr>
<td>Moving sensor stabilization</td>
<td>Roll stabilization, one device for all lenses</td>
<td></td>
</tr>
</tbody>
</table>
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 $\rightarrow$ SNR $\sim$ SNR$_0 \times t$
- increasing contrast amplifies noise
Multiple noisy images

- 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
- \( \frac{1}{4} \times \frac{1}{60} \text{s} \) (15 times, \( \sim 4 \) stops)
  - 15 images \( \rightarrow \) 15*(1/3) = 5s
- Faster chips in near future allow averaging 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

\[ z = u \ast h + n \]
Solution of deconvolution problem

- Model

\[ z = u \ast h + n \]

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

- Image model \[ z = u \ast h + n \]
- Minimize
  \[ E(u) = \frac{1}{2\sigma^2} \| u \ast h - z \|^2 + \lambda Q(u) \]
  \[ Q(u) = \int |\nabla u|^2 \quad Q(u) = \int |\nabla u| \]
- Regularization constant \( \square \) - no one correct value
Role of regularization parameter

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

Mean least squares error /pixel

![Graph showing the role of regularization with SNR values 15 dB, 20 dB, and 30 dB, and corresponding error minima.]
Matrix notation

\[ E(u) = \frac{1}{2\sigma^2} \| u \ast h - z \|^2 + \lambda \| c \ast u \|^2 \]

Tikhonov reg. \( c = [1\, -1] \)

\( u, z \) ... vectors

\( H \) ... matrix of 2D convolution

\( C \) ... regularization matrix

\[ E(u) = \frac{1}{2\sigma^2} \| Hu - z \|^2 + \lambda \| Cu \|^2 \]
Solution in Fourier domain

\[ E(u) = \frac{1}{2\sigma^2} \| u \ast h - z \|^2 + \lambda \| c \ast u \|^2 \]

Tikhonov reg.  \( c = [1\ -1] \)

Parseval’s theorem

Convolution theorem

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

Wiener filter
Bayesian view – MAP estimate

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

\[ p(u|z, h) \propto p(z|u, h)p(u) \]

- Minimize

\[ -\ln p(u|z, h) = -\ln p(z|u, h) - \ln p(u) \]
Deconvolution as MAP estimate

- Minimize

\[-\ln p(u \mid z, h) = -\ln p(z \mid u, h) - \ln p(u)\]

\[-\ln p(z \mid u, h) = -\ln \prod_{i} e^{-\frac{(z_i - [u \ast h]_i)^2}{2\sigma^2}} = \frac{1}{2\sigma^2} \|z - u \ast h\|^2\]

\[z = u \ast h + n\]
Image prior (first order statistics)

\[- \ln p(u) = - \ln \prod_{i} p(\nabla u_i) = \sum_{i} - \ln p(\nabla u_i)\]

Intensity histogram

Gradient log-histogram
Equivalence of the two views

Tikhonov regularization

\[ E(u) = \frac{1}{2\sigma^2} \left\| u \ast h - z \right\|^2 + \lambda \int |\nabla u|^2 \]

\[ - \ln p(u|z, h) = - \ln p(z|u, h) - \ln p(u) \]

where \( p(u) \propto \prod_i e^{\Phi(\nabla u_i)} \) and \( \Phi(\nabla u_i) = |\nabla u|^2 \)
Image priors

\[ Q(u) = \int |\nabla u|^2 \]

Tikhonov regularization

\[ Q(u) = \int |\nabla u| \]

TV regularization
Space-variant deblurring

Minimization of

\[ E(u) = \frac{1}{2} \| u \ast_v h - z \|^2 + \lambda Q(u) \]

\[ u \ast_v h [x, y] = \int_{\Omega} u(x - s, y - t) h(s, t; x - s, y - t) \, ds \, dt \]
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

\[
\max_{u,h} p(u, h|z) \propto p(z|u, h) p(u) p(h)
\]

\[
\max_h p(h|z) = \int p(u, h|z) du
\]

- \( \ln p(h|z) \) difficult to compute \( \rightarrow \) approximation
Image prior

\[ p(u) \propto \prod e^{\Phi(\nabla u_i)} \]

\[-\ln p(u) = \sum_{i} -\Phi(\nabla u_i)\]

Gradient log-histogram

(approximation of \( \ln p(\nabla u_i) \))
Image priors

\[ Q(u) = \int |\nabla u|^2 \]

Tikhonov regularization

\[ Q(u) = \int |\nabla u| \]

TV regularization
Approximation by Gaussian mix
PSF prior

\[ p(h) \propto \prod_i \sum_k \beta_k e^{-\tau_k h_i} \]

\[ -\ln p(h) \propto \sum_i -\ln \sum_k \beta_k e^{-\tau_k h_i} \]
Rob Fergus (Example I)
Rob Fergus (Example II)
MAP approach at SIGGRAPH 08

\[
p(u, h|z) \propto p(z|u, h)p(u)p(h)
\]

\[- \ln p(u, h|z) = - \ln p(z|u, h) - \ln p(u) - \ln p(h)\]

\[
\frac{1}{\sigma^2} \| z - u \ast h \|^2 + \sum_i \Phi(\partial u_i) + \tau \| h \|_1 + \cdots
\]
Single image deblurring - summary

- Difficult, underdetermined problem
- Needs strong priors on both image and convolution kernel
- First really successful algorithm (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

\[ [u \ast h_k](x, y) + n_k(x, y) = z_k(x, y) \]
Multi-image blind deconvolution

System of integral equations (ill-posed, underdetermined)

\[ z_k(x) = (h_k * u)(x) + n_k(x) \]

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

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

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

\[ R(\{h_i\}) = \frac{1}{2} \sum_{1 \leq i, j \leq K} \|z_i * h_j - z_j * h_i\|^2 \]
PSF regularization

\[ R(\{h_i\}) = \frac{1}{2} \sum_{1 \leq i, j \leq K} \|z_i \ast h_j - z_j \ast h_i\|^2 \]

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

\[ z_1 = u \ast h_1 \quad z_2 = u \ast h_2 \]

\[ z_1 \ast h_2 = u \ast h_1 \ast h_2 = u \ast h_2 \ast h_1 = z_2 \ast h_1 \]
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 $E(u, \{g_i\})$ 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

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

\[ Q(u) = \int_{\Omega} |\nabla u| \]

\[ R(\{h_i\}) = \frac{1}{2} \sum_{1 \leq i, j \leq K} \| z_i * h_j - z_j * h_i \|^2 \]
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 transform found by RANSAC
- Lens distortion can be neglected
- Less important parts of scene can move
Blurred + underexposed results

\[ h_{i,j} = \arg \min_k \|u_{i,j} * k - z_{i,j}^T\|^2 + \alpha \| \nabla k \|^2, \quad k(s,t) \geq 0, \]
Blur kernel adjustment

- Regions lacking texture
- Regions of pixel saturation
Restoration

• Minimization of functional

\[ E(u) = \frac{1}{2} \left\| u \ast_v h - z \right\|^2 + \lambda \int |\nabla u| \]

\[ u \ast_v h [x, y] = \int_{\Omega} u(x - s, y - t) h(s, t; x - s, y - t) \, ds \, dt \]

• 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

- 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)

PSF consists of scaled versions of camera trajectory.
Rough depth map estimation (step II)

\[ \text{err}(d) = [z_1 \ast h_2(d) - z_2 \ast h_1(d)]^2 \]
Functional minization (step III)

- Input images $z_1, z_2, \ldots$
- Minimization initialized by depth map $d_0$
- Goal – sharp image and depth map computed as $\text{argmin}_{u,d} E(u,d)$

$$E(u, d) = \frac{1}{2} \sum_{p=1}^{P} \| u * h_p(d) - z_p \|^2 + \lambda_u Q(u) + \lambda_d R(\frac{1}{d})$$
Functional minimization (step III)
Motion blur + limited depth of focus

F/4
Out-of-focus blur

$Z_1$  
(F/5.0)

$Z_2$  
(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 blurry - 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