Videosignalverarbeitung für zukünftige Bildkommunikationssysteme

Prof. Dr.-Ing. André Kaup

Lehrstuhl für Multimediakommunikation und Signalverarbeitung
Friedrich-Alexander-Universität Erlangen-Nürnberg
Typical Video Data Rate

**Data rate** of uncompressed HD video:

1920 columns x 1080 lines x 24 Bit/pixel x 30 frames/second =

1.49 GBit/Second or 187 MByte/Second

- One CD every 4 seconds
- One DVD every 50 seconds
- One Blu-ray disc every 267 seconds

**Consequences:** Video signals must be compressed before transport or storage
Intraframe Prediction

- **Assumption:**
  images have a high spatial correlation

- **Intraframe prediction**
  uses already decoded image blocks in causal neighborhood

- **Best prediction mode**
  according to optimization constraint is transmitted as side information
Interframe Prediction

- **Assumption:** video has a high temporal correlation

- **Block is predicted** using motion compensated reference

- **Multiple** reference images for long term prediction

- **Motion vector** is transmitted as side information

---

Kaup: Video Signal Processing for Future Communication Systems
Chair of Multimedia Communications and Signal Processing

10/22/2012
Block Diagram of Hybrid Video Coder

Image block

Transform

Quantization

Entropy Coding

Inverse Transform

Memory

Intra Prediction

Inter Prediction

Irrelevancy Reduction

Reduction of Statistical Redundancy

Removal of Spatial Redundancy

Removal of Temporal Redundancy

Kaup: Video Signal Processing for Future Communication Systems
Chair of Multimedia Communications and Signal Processing
Page 7
Overview

- Introduction into video coding
- **Spatiotemporal prediction**
- In-loop video denoising
- Compensation of rotational motion
- Rate distortion optimization
- Multi-view video using super-resolution
- Conclusions and outlook
Spatiotemporal Prediction

- **State-of-the-art:**
  switching between interframe and intraframe prediction modes

- **Decision** taken using rate-distortion optimization

- **Extended approach:**
  joint spatiotemporal prediction as additional mode
Hybrid Video Coder Using Spatiotemporal Prediction

Image block → + → Transform format → Quanti-zation → Entropy Coding

Inverse Transform

Intra-Prediction

Inter-Prediction

Non-Local Means Filtering

Memory
Simplified approach for prediction both in time and space

Basic idea:
- Motion compensated prediction gives initial estimation based on previous image of the video sequence
- Integration ("superposition") of local neighborhood information from current image

Practical realization:
- Cogging of two predictors with non-local means denoising algorithm
Non-Local Means Refined Prediction

**Processing area** \( \mathcal{L} = \mathcal{B} \cup \mathcal{R} \) is regarded for prediction

- Motion compensated block \( \mathcal{B} \)
- Reconstructed neighboring blocks \( \mathcal{R} \)
Non-Local Means Refined Prediction

**Task:** Recover original signal $s[m, n]$ from given signal samples $\tilde{s}[m, n]$ in area $\mathcal{L} = \mathcal{B} \cup \mathcal{R}$

**Non-local means:** Estimate refined samples $\hat{s}[m, n]$ in $\mathcal{B}$ using weighted non-local average filter

\[
\hat{s}[m, n] = \frac{\sum_{(k, l) \in \mathcal{L}} \tilde{s}[k, l] w_{(m, n)}[k, l]}{\sum_{(k, l) \in \mathcal{L}} w_{(m, n)}[k, l]}
\]
Non-Local Means Refined Prediction

Filter weights at pixel position \((m,n)\)

\[
w_{(m,n)}[k, l] = e^{-d_{(m,n)}[k, l]/h^2}
\]

- depend on similarity between neighborhood around sample \(\tilde{s}[m, n]\) and \(\tilde{s}[k, l]\)

Similarity is measured by sum of squared differences:

\[
d_{(m,n)}[k, l] = \sum_{\mu, \nu = -d_m, \ldots, d_m} \left( \tilde{s}[m + \mu, n + \nu] - \tilde{s}[k + \mu, l + \nu] \right)^2
\]

Parameter \(h\) controls strength of average filtering
Non-Local Means Refined Prediction

Example for weight calculation:

- Samples with similar neighborhood get a large weight
- Samples with dissimilar neighborhood get a small weight
Test Sequences

Crew

Foreman

Vimto
Simulation Results

H.264/AVC JM10.2
Baseline Profile, Level 2.0
CIF sequences
IPPP..., 100 frames
Search range: 16 sample
1 bit/block for signaling the new mode

NLM-RP parameters:
\( h = 25 \)
\( d_m = 3 \)

[Seiler, Richter, Kaup, PCS 2010]
## Simulation Results

Mean rate reduction and PSNR gain according to Bjøntegaard metric:

<table>
<thead>
<tr>
<th></th>
<th>“Crew”</th>
<th>“Foreman”</th>
<th>“Vimto”</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ratenreduktion</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSA [1]</td>
<td>7.32 %</td>
<td>3.20 %</td>
<td>13.42 %</td>
<td>7.98 %</td>
</tr>
<tr>
<td>MSA [2]</td>
<td>7.69 %</td>
<td>2.26 %</td>
<td>14.98 %</td>
<td>8.31 %</td>
</tr>
<tr>
<td>NLM-RP</td>
<td>10.44 %</td>
<td>4.77 %</td>
<td>13.68 %</td>
<td>9.63 %</td>
</tr>
<tr>
<td><strong>PSNR gain</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSA [1]</td>
<td>0.37 dB</td>
<td>0.13 dB</td>
<td>0.66 dB</td>
<td>0.39 dB</td>
</tr>
<tr>
<td>MSA [2]</td>
<td>0.39 dB</td>
<td>0.09 dB</td>
<td>0.74 dB</td>
<td>0.41 dB</td>
</tr>
<tr>
<td>NLM-RP</td>
<td>0.54 dB</td>
<td>0.19 dB</td>
<td>0.67 dB</td>
<td>0.47 dB</td>
</tr>
</tbody>
</table>


Simulation Results

Motion compensated prediction

Non-local means refined prediction

QP34: 33.54 dB @ 447 kbit/s

QP34: 34.25 dB @ 434 kbit/s
## Simulation Results

<table>
<thead>
<tr>
<th>Method</th>
<th>SSIM</th>
<th>Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion compensated prediction</td>
<td>33.54 dB</td>
<td>447 kbit/s</td>
</tr>
<tr>
<td>Non-local means refined prediction</td>
<td>34.25 dB</td>
<td>434 kbit/s</td>
</tr>
</tbody>
</table>
Overview

- Introduction into video coding
- Spatiotemporal prediction
- **In-loop video denoising**
- Compensation of rotational motion
- Rate distortion optimization
- Multi-view video using super-resolution
- Conclusions and outlook
Prediction Error Signal

**Problem**
- Prediction error signal has more noise than the current frame itself

\[
\text{current frame } g[i] = s_g[i] + n_g[i]
\]

\[
\text{motion compensated reference frame } \hat{f}[i] = s_{\hat{f}}[i] + n_{\hat{f}}[i]
\]

\[
\text{prediction error } e[i] = s_g[i] - s_{\hat{f}}[i] + n_g[i] - n_{\hat{f}}[i]
\]

**Solution**
- Remove noise from the predictor
Inter-Frame Encoder with In-Loop Denoising

- Simplified block diagram of an inter-frame encoder with in-loop denoising

\[ g[i] \xrightarrow{+} e[i] \xrightarrow{\hat{f}[i]} T \xrightarrow{Q} EC \xrightarrow{coded bitstream} \]

\[ \text{ME/MC} \xrightarrow{z^{-1}} p[i] \xrightarrow{\hat{f}[i]} e_q[i] \xrightarrow{T^{-1}}\]

\[ g_q[i] \xrightarrow{+} \hat{f}[i] \]
Inter-Frame Decoder with In-Loop Denoising

- Simplified block diagram of an inter-frame decoder with in-loop denoising

![Diagram of inter-frame decoder]

- Denoising is performed after displaying the decoded frames
Quantization of Noise

- Noise filtering works on **transformed** and **quantized** reference signal

- Analytical model for Gaussian noise and perfect prediction

- **Observation**: Noise depends on $\Delta$

\[
\sigma_{n_q}^2(\Delta) = 2 \cdot \Delta^2 \sum_{\lambda=1}^{\infty} \lambda^2 w(\lambda)
\]

\[
w(\lambda) = \text{erf} \left( \frac{\lambda \Delta + \Delta}{\sigma_n \sqrt{2}} \right) - \text{erf} \left( \frac{\lambda \Delta - \Delta}{\sigma_n \sqrt{2}} \right)
\]
Quantization of Noise in H.264/AVC

- **Generalization** of analytical model for quantization offset $0 < f < \Delta$

\[
\alpha_q[\mu] = \left\lfloor \left( \frac{|\alpha[\mu]| + f}{\Delta} \right) \right\rfloor \text{sign}(\alpha[\mu]) \Delta
\]

- **Gaussian noise** yields

\[
wf(\lambda) = \text{erf} \left( \frac{(\lambda + 1)\Delta - f}{\sigma_n \sqrt{2}} \right)
- \text{erf} \left( \frac{\lambda \Delta - f}{\sigma_n \sqrt{2}} \right)
\]

- Noise is still present for a wide range of quantization parameters

[Wige, Kaup, PCS 2010]
Fast Noise Estimation

- Filtering operation
  \[ \sigma_{n_q} = \frac{1}{6|\Omega|} \sqrt{\frac{\pi}{2}} \sum_{j \in \Omega} |g_q[j] * h| \]

- Filtering mask
  \[
  h = \begin{bmatrix}
  1 & -2 & 1 \\
  -2 & 4 & -2 \\
  1 & -2 & 1 \\
  \end{bmatrix}
  \]

[Immerkaer, Computer Vision and Image Understanding 1996]
Adaptive Wiener (Averaging) Filter

- Filtering operation
  \[ p_{AWF}[i] = \mu_f[i] + \frac{\sigma_f^2[i]}{\sigma_f^2[i] + \xi \sigma^n_q} \cdot (g_q[i] - \mu_f[i]) \]

- Local mean
  \[ \hat{\mu}_f[i] = \frac{1}{|N_i|} \sum_{j \in N_i} g_q[j] \]

- Local variance
  \[ \hat{\sigma}_f^2[i] = \begin{cases} \sigma_{gq}[i] - \sigma^n_q, & \text{if } \sigma_{gq}[i] > \sigma^n_q \\ 0, & \text{else} \end{cases} \]
  \[ \sigma_{gq}^2[i] = \frac{1}{|N_i|} \sum_{j \in N_i} (g_q[j] - \hat{\mu}_f[i])^2 \]

[Wige, Kaup, ICIP 2010]
Simulation Conditions

- HEVC reference software HM-2.2
  - Coding of 100 frames
  - QP 2 \{12 \ldots 37\}
  - Coding configurations: ldlc_P, ldhe_P, ldlc, ldhe

- SVT test sequences from ftp://vqeg.its.bldrdoc.gov/
  - Resolution of 3840x2160 pixels with 50 frames per second
  - Using a centrically cropped version of 2560x2160 pixels

- Denoising parameters
  - AWF: window of 3x3 pixels, »=20
  - \Delta_L=1, \Delta_H=3
Simulation Results for ParkJoy (ldlc)

- Estimated noise of the input sequence $\sigma_n \approx 1.8$
Simulation Results ldlc

- Average bitrate savings for low delay low complexity (B.frames)
  - HQ for QP 2 {12 … 27}
  - MQ for QP 2 {22 … 37}
  - NA = not adaptive, AS = adaptive selection, bound = perfect estimate
  - PJ=ParkJoy, CR=CrowdRun, ITT=InToTree, OTC=OldTownCross, DTO=DucksTakeOff

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Δ bitrate in % for ldlc B-frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AWF-NA</td>
</tr>
<tr>
<td></td>
<td>HQ</td>
</tr>
<tr>
<td><em>PJ</em></td>
<td>-0.32</td>
</tr>
<tr>
<td><em>CR</em></td>
<td>-1.52</td>
</tr>
<tr>
<td><em>ITT</em></td>
<td>+2.85</td>
</tr>
<tr>
<td><em>OTC</em></td>
<td>-1.10</td>
</tr>
<tr>
<td><em>DTO</em></td>
<td>-2.03</td>
</tr>
</tbody>
</table>
Overview

- Introduction into video coding
- Spatiotemporal prediction
- In-loop video denoising
- **Compensation of rotational motion**
- Rate distortion optimization
- Multi-view video using super-resolution
- Conclusions and outlook
Automated quality monitoring in car infotainment systems

Grabbed video content has significantly different characteristics:
- static menus
- conventional television
- navigation video sequences
Common display style: navigation map is aligned to the orientation of the user (i.e. car driver)
→ Map rotates, user icon remains aligned in upward fashion

Complex overlays: status bars, text, POIs
**Observation:** block-based MC cannot handle rotation properly  
- map rotation is approximated by small MB partitioning

**Solution:** Global motion compensation using rotation compensation  
- Computation of a suitable rotated reference  
- Use rate-distortion optimization to handle static areas
**Challenges** in rotation estimation:

- Identify and exclude static areas from parameter estimation
- Keep complexity low

**Proposed solution:**

- Feature based matching instead of optic flow
- Use rotation invariant fast ORB feature
- Removal of ambiguous matches using geometry constraints
Compensation of Rotational Motion

Extended reference list:
- rotationally compensated frame is added to list of references frames

Static areas are also rotated but efficiently handled by RD optimization
- no explicit detection and segmentation necessary

Navigation encoder (proposed)  Traditional H.264/AVC encoding

Colors: intra-MBs red, p-predictive MBs blue, skipped MBs green
RD Optimization: Chosen MB Partitions and Modes

Colors: intra-MBs red, p-predictive MBs blue, skipped MBs green
Compression Efficiency

- Comparison to traditional H.264/AVC High Profile encoding on sequences Ingolstadt1, Denver1, and Denver2 (resolution 800x480)
- Using Bjontegaard metric, mean bitrate savings of 19.52% (Denver1), 18.77% (Denver2), and 11.1% (Ingolstadt1) are achieved
Overview

- Introduction into video coding
- Spatiotemporal prediction
- In-loop video denoising
- Compensation of rotational motion
- **Rate distortion optimization**
- Multi-view video using super-resolution
- Conclusions and outlook
In the Sense of Rate-Distortion

\[ \min \{ D \} \]

subject to \( R \leq R_c \)

Mode Selection?

Intra Prediction
or
Inter Prediction

Dynamic Programming
- High computational complexity

Lagrangian Rate-Distortion Optimization
- Simple and efficient
Lagrangian Rate-Distortion Optimization

- By Lagrange Multiplier Method
  \[
  \min\{D\} \quad \text{subject to } R \leq R_c
  \]
  \[
  \min\{J\} \quad \text{where } J = D + \lambda \cdot R
  \]

  Theoretically, with a proper \( \lambda \), mode decision and motion estimation process can be optimized.

- How to determine \( \lambda \)?
  If \( R \) and \( D \) are differentiable everywhere, minimum \( J \) is given by setting its derivative to zero, i.e.

  \[
  \frac{dJ}{dR} = \frac{dD}{dR} + \lambda = 0
  \]

  leading to

  \[
  \lambda = -\frac{dD}{dR}
  \]

  \( \lambda \) corresponds to the negative slope of R-D curve
Well Known Algorithm – High Rate Assumption

- Under “High Rate” assumption, rate $R$ can be derived according to approximation curve for entropy constrained scalar quantization

$$R(D) = a \log_2 \left( \frac{b}{D} \right)$$

- Uniform distribution of coefficients within each quantization interval in “High Rate” environment

$$D = \frac{(2Q)^2}{12} = \frac{Q^2}{3}$$

- **Final $\lambda$**

$$\lambda = -\frac{dD}{dR} = c \cdot Q^2$$

- **Advantages**
  - Simple and efficient and was adopted into the reference software (JM) of H.264/AVC

- **Drawbacks**
  - “High Rate” assumption is not realistic all the time
  - Not adaptive: no properties of input sequences are considered
Proposed Entropy and Distortion Models

- More general assumption -- **Laplace** distribution of transformed residuals

\[
f_{Lap}(x) = \frac{\Lambda}{2} e^{-\Lambda |x|}
\]
\[
\Lambda = \frac{\sqrt{2}}{\sigma}
\]

- Related Entropy

\[
P_0 = \int_{-Q-\gamma Q}^{Q-\gamma Q} f_{Lap}(x) \, dx
\]
\[
P_n = \int_{nQ-\gamma Q}^{(n+1)Q-\gamma Q} f_{Lap}(x) \, dx
\]

\[
H = -P_0 \cdot \log_2 P_0 - 2 \sum_{n=1}^{\infty} P_n \cdot \log_2 P_n
\]

- Related Distortion

\[
D = \int_{-Q-\gamma Q}^{Q-\gamma Q} x^2 f_{Lap}(x) \, dx
\]
\[
+ 2 \sum_{n=1}^{\infty} \int_{nQ-\gamma Q}^{(n+1)Q-\gamma Q} (x - nQ)^2 f_{Lap}(x) \, dx
\]

[Li, Kaup, IEEE T-CSVT 2009]
On average, 0.20 dB gain over HR-Λ in JM

Big gains for slow sequences
- Up to 0.74 dB for hall monitor
- Up to 0.65 dB for salesman
- Up to 0.44 dB for waterfall

Almost no gains for bus, coastguard and stefan

Big gains for B frames
- On average, 0.85 dB for hall monitor
- On average, 0.75 dB for salesman
- On average, 0.76 dB for waterfall
High Efficiency Video Coding

New ITU/ISO standard: High Efficiency Video Coding (HEVC)
- To be finalized in January 2013
- Same system concept as H.264/AVC (see T-CSVT Dec. 2012)

Major differences to H.264/AVC:
- Variable block sizes (coding units) up to 64x64
- Better intraframe and motion prediction

Coding gain vs. H.264/AVC:
- Up to 30 % rate savings for all intra
- Around 40 % for inter, up to 50% subjective

Complexity: decoder around factor 2, encoder possibly much higher
Overview

- Introduction into video coding
- Spatiotemporal prediction
- In-loop video denoising
- Compensation of rotational motion
- Rate distortion optimization
- Multi-view video using super-resolution
- Conclusions and outlook
Super-Resolution

- **Super-Resolution** (SR) is a key issue in image and video processing domain

- **Goal**: create reasonable high-frequency content for a low-resolution image or video sequence

\[
\begin{align*}
\text{Original Image} & \quad + \quad \text{High-Frequency Content} & \quad = \quad \text{Super-Resolved Image}
\end{align*}
\]
Motivation

- **Mixed-resolution** multi-view video plus depth format (MR-MVD)

- **Goal**: Usage of neighboring high-frequency content to refine low-resolution destination view
Super-Resolution Based on High-Frequency Synthesis

State of the art:

\[ l(u,v) \xrightarrow{\uparrow} l_i(u,v) \xrightarrow{+} \hat{l}(u,v) \]

left view

right view

\[ r(u,v) \xrightarrow{\uparrow} r_i(u,v) \xrightarrow{d_r(u,v)} r_h(u,v) \]

warping
Super-Resolution Based on High-Frequency Synthesis

- Approach requires accurate depth information
- Incorrect depth entries typically occur at depth boundaries
- Various approaches to remove ghosting artifacts already exist
Potential additional **depth errors** and inaccuracies:

- Inaccurate registration of depth camera
- Limitation of depth range
Impact of additional depth inaccuracies on visual SR quality:

- Different depth distortion scenarios have different impact on SR quality

- **Goal:** Create an algorithm that is robust to each of those distortions
Displacement-Compensated Super-Resolution

\[ l(u,v) \rightarrow \mathbf{w} \rightarrow l_f(u,v) \rightarrow + \rightarrow \hat{l}_{dc}(u,v) \]

\[ \hat{l}_l(u,v) \rightarrow \text{Displacement estimation} \rightarrow \hat{l}_h(u,v) \]

Displacement estimation

Displacement compensation

\[ r(u,v) \rightarrow \mathbf{w} \rightarrow r_f(u,v) \rightarrow d_r(u,v) \rightarrow r_h(u,v) \]

left view

right view

[Richter, Kaup, MMSP 2012]
Displacement-Compensated Super-Resolution

- **Displacement estimation** (DE):
  
  - Estimate a proper displacement between the warped low-frequency information \( \hat{l}_i(u,v) \) and the upsampled destination view \( l_i(u,v) \)
  
  - Depth errors may vary both across the two image dimensions, as well as across the depth range

\[\rightarrow\] DE is done depth-dependently and blockwise
Displacement-Compensated Super-Resolution

- Depth-dependent subdivision:
  - Every block $B^{(i)}$ consists of pixels with similar entries in the initial depth map $d_r(u,v)$.
Displacement-Compensated Super-Resolution

- Depth-dependent subdivision not sufficient to assume a constant displacement

Further subdivision of every block $\hat{B}^{(i)}$ into $\hat{B}^{(i_1)}$ and $\hat{B}^{(i_2)}$ by distinguishing whether a pixel was originally located near a depth edge in $d_r(u,v)$.
Displacement-Compensated Super-Resolution

- Displacement in a block $B\cap(i_k)$ can be assumed to be constant.

- Calculation of a displacement vector for every block $B\cap(i_k)$ by searching for the most similar block in $l_i(u,v)$.

$$\left(u_d^{(i_k)}, v_d^{(i_k)}\right) = \min_{(u,v) \in S} SAD(B\cap(i_k), B_{(u,v)})$$

- Refinement vectors $\left(u_d^{(i_k)}, v_d^{(i_k)}\right)$ are used afterwards to compensate the displacement between $l_h(u,v)$ and $l_i(u,v)$.
Simulation Results

- Simulation Setup:
  - Test sequences: *ballet, breakdancers, cones, teddy*
  - Downsampling factor for low-resolution view: 2, 4
  - Blocksize: 32
  - Depth stepsize: 10
  - Search range: 100
  - Evaluation for original, translated, scaled and zoomed depth maps
Simulation Results

- **Translation:**
  - Shifting all depth entries 5 pixel positions to the top right.

- **Scaling:**
  - Limiting the 8 bit depth entries [0; 255] to [0; 127].

- **Zoom:**
  - Dropping 10% of rows and columns and resizing the cropped depth map via nearest neighbor interpolation.
## Simulation Results

PSNR evaluation, $\downarrow 2$

<table>
<thead>
<tr>
<th></th>
<th>Ballet</th>
<th>Breakdancers</th>
<th>Cones</th>
<th>Teddy</th>
<th>Avg. gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_i(u,v)$</td>
<td>36.97</td>
<td><strong>38.82</strong></td>
<td>33.10</td>
<td>33.71</td>
<td></td>
</tr>
<tr>
<td><strong>Original</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{l}(u,v)$</td>
<td>36.68</td>
<td>37.38</td>
<td>34.63</td>
<td>35.09</td>
<td></td>
</tr>
<tr>
<td>$\hat{l}_{dc}(u,v)$</td>
<td><strong>38.01</strong></td>
<td>37.95</td>
<td><strong>34.71</strong></td>
<td>35.43</td>
<td><strong>0.58</strong></td>
</tr>
<tr>
<td><strong>Translated</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{l}(u,v)$</td>
<td>35.98</td>
<td>37.47</td>
<td>34.18</td>
<td>34.82</td>
<td></td>
</tr>
<tr>
<td>$\hat{l}_{dc}(u,v)$</td>
<td><strong>38.11</strong></td>
<td>38.04</td>
<td><strong>34.61</strong></td>
<td>35.48</td>
<td><strong>0.95</strong></td>
</tr>
<tr>
<td><strong>Scaled</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{l}(u,v)$</td>
<td>34.77</td>
<td>36.11</td>
<td>30.35</td>
<td>31.04</td>
<td></td>
</tr>
<tr>
<td>$\hat{l}_{dc}(u,v)$</td>
<td><strong>37.83</strong></td>
<td>38.01</td>
<td><strong>34.24</strong></td>
<td>35.06</td>
<td><strong>3.22</strong></td>
</tr>
<tr>
<td><strong>Zoomed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{l}(u,v)$</td>
<td>34.16</td>
<td>36.86</td>
<td>32.23</td>
<td>33.16</td>
<td></td>
</tr>
<tr>
<td>$\hat{l}_{dc}(u,v)$</td>
<td><strong>37.80</strong></td>
<td>37.86</td>
<td><strong>34.09</strong></td>
<td>35.07</td>
<td><strong>2.10</strong></td>
</tr>
</tbody>
</table>
## Simulation Results

PSNR evaluation, \(\downarrow 4\)

<table>
<thead>
<tr>
<th></th>
<th>Ballet</th>
<th>Breakdancers</th>
<th>Cones</th>
<th>Teddy</th>
<th>Avg. gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>(l_1(u,v))</td>
<td>31.32</td>
<td>33.82</td>
<td>27.02</td>
<td>27.87</td>
<td></td>
</tr>
<tr>
<td>Original depth</td>
<td>(\hat{l}(u,v))</td>
<td>32.70</td>
<td>34.45</td>
<td>29.50</td>
<td>30.30</td>
</tr>
<tr>
<td></td>
<td>(\hat{l}_{dc}(u,v))</td>
<td>32.83</td>
<td>34.89</td>
<td>29.26</td>
<td>30.48</td>
</tr>
<tr>
<td>Translated</td>
<td>(\hat{l}(u,v))</td>
<td>31.64</td>
<td>34.47</td>
<td>28.99</td>
<td>30.09</td>
</tr>
<tr>
<td>depth</td>
<td>(\hat{l}_{dc}(u,v))</td>
<td>32.78</td>
<td>34.94</td>
<td>29.18</td>
<td>30.62</td>
</tr>
<tr>
<td>Scaled depth</td>
<td>(\hat{l}(u,v))</td>
<td>29.41</td>
<td>31.15</td>
<td>24.29</td>
<td>25.23</td>
</tr>
<tr>
<td></td>
<td>(\hat{l}_{dc}(u,v))</td>
<td>32.87</td>
<td>34.96</td>
<td>28.76</td>
<td>30.01</td>
</tr>
<tr>
<td>Zoomed depth</td>
<td>(\hat{l}(u,v))</td>
<td>28.17</td>
<td>33.36</td>
<td>26.86</td>
<td>28.10</td>
</tr>
<tr>
<td></td>
<td>(\hat{l}_{dc}(u,v))</td>
<td>32.47</td>
<td>34.70</td>
<td>28.61</td>
<td>30.22</td>
</tr>
</tbody>
</table>
Visual comparison: ballet

Simulation Results

\[ l_1(u, v) \]

\[ \hat{l}(u, v) \]

\[ l(u, v) \]

\[ \hat{l}_{dc}(u, v) \]

Kaup: Video Signal Processing for Future Communication Systems
Chair of Multimedia Communications and Signal Processing
Simulation Results

Visual comparison: teddy

\( I(u,v) \)

\( \hat{I}(u,v) \)

\( l(u,v) \)

\( \hat{l}(u,v) \)

\( l_{dc}(u,v) \)

\( \hat{l}_{dc}(u,v) \)
Summary and Conclusions

Video compression is essential for future communication systems.

Prior knowledge should be used whenever possible:
- Video is a cube: Spatiotemporal prediction
- Noise might be significant: in-loop denoising filter
- Special data adaptation: rotation compensated prediction

Rate-distortion optimization crucial for performance:
- HEVC will set new standard

Multi-view video systems still under research

Single solution to multiple problems might not be optimal:
- More application specific solutions will appear in the future