Improved Lossless Depth Image Compression

Roland Fischer, Philipp Dittmann, Christoph Schröder, Gabriel Zachmann
University of Bremen, Germany
rfischer@cs.uni-bremen.de

WSCG, 19-21 May 2020
Motivation

- Small, affordable, RGB-D cameras getting popular
- Resolution increases

Microsoft’s Azure Kinect RGB-D camera

Color image

Depth image
Motivation

- Small, affordable, RGB-D cameras getting popular
  - Resolution increases
- Many applications:
  - Robotics
  - Computer vision
  - Telepresence
  - VR/AR
Motivation

• Small, affordable, RGB-D cameras getting popular
  • Resolution increases

• Many applications:
  • Robotics
  • Computer vision
  • Telepresence
  • VR/AR

Remote robot with RGB-D camera [Nenci14]

Mapped environment [Labbé14]
Motivation

- Small, affordable, RGB-D cameras getting popular
  - Resolution increases
- Many applications:
  - Robotics
  - **Computer vision**
  - Telepresence
  - VR/AR

Autonomous lamps [Teuber17]
Motivation

- Small, affordable, RGB-D cameras getting popular
  - Resolution increases
- Many applications:
  - Robotics
  - Computer vision
  - Telepresence
  - VR/AR

Virtual conference room [Wilson17]
Motivation

- Small, affordable, RGB-D cameras getting popular
  - Resolution increases
- Many applications:
  - Robotics
  - Computer vision
  - Telepresence
  - VR/AR

Virtual operation room [VIVATOP]
Motivation

- Efficient compression for real-time transmission
  - Limited bandwidth (1 Gbit/s ethernet)
  - One Kinect V2 RGB-D frame: 6.6 MB (1.6 Gbit/s @30 Hz)
Motivation

- Efficient compression for real-time transmission
  - Limited bandwidth (1 Gbit/s ethernet)
  - One Kinect V2 RGB-D frame: 6.6 MB (1.6 Gbit/s @30 Hz)
- Standard image/video compression algorithms for color
Motivation

- Efficient compression for real-time transmission
  - Limited bandwidth (1 Gbit/s ethernet)
  - One Kinect V2 RGB-D frame: 6.6 MB (1.6 Gbit/s @30 Hz)
- Standard image/video compression algorithms for color
- Depth has unique characteristics → custom algorithms
  - Homogeneous regions with abrupt depth-discontinuities
  - Distributed regions of invalid (zero) pixels
Related Work

- Point cloud based [Thanou16, Mekuria17] and mesh based methods [Bannò12, Mekuria13] not real-time capable
Related Work

- Point cloud based [Thanou16, Mekuria17] and mesh based methods [Bannò12, Mekuria13] not real-time capable

- Methods based on adapted image and video codecs mostly lossy [Pece11, Liu15, Zhang15, Hamout19]
Related Work

- Point cloud based [Thanou16, Mekuria17] and mesh based methods [Bannò12, Mekuria13] not real-time capable

- Methods based on adapted image and video codecs mostly lossy [Pece11, Liu15, Zhang15, Hamout19]

- Few real-time lossless solutions, e.g. [Mehrotra11]
  - The RVL algorithm [Wilson17] is the most promising one
RVL Recap

- Fast, efficient, lossless depth-image compression
- Accounts for unique depth image characteristics
  - Run-length coding of zero pixels
  - Variable-bit-length coding of non-zero pixels
  - Depth-adapted intra-image prediction
- Only moderately high compression ratio
Our Contributions

- Novel real-time lossless depth-image compression algorithm
  - Inspired by RVL, aimed at stronger compression
  - Inter-frame delta computation
  - Span-based adaptive prediction
  - Bit reduction
  - Multi-threading
Our Contributions

• Novel real-time lossless depth-image compression algorithm
  • Inspired by RVL, aimed at stronger compression
  • Inter-frame delta computation
  • Span-based adaptive prediction
  • Bit reduction
  • Multi-threading

• Empirical evaluation:
  • Several lossless compression algorithms
  • Multiple static and dynamic scenes with different cameras
Compression Pipeline

Depth Image → Frame Delta → Span-Based Adaptive Prediction → Run-length and Variable-length Coding → Zstandard → Compressed Image
Compression Pipeline

- Pipeline is lossless
- Individual steps are multi-threaded
- Analogous decompression
Inter-Frame Delta

- Pixel-wise differences of consecutive images
- Uses temporal coherence
Inter-Frame Delta

- Pixel-wise differences of consecutive images
  - Uses temporal coherence

- Optional: temporal filtering
  - Skips update of pixels if continually $\Delta < \epsilon$
  - Counters noisy depth readings
  - Not lossless anymore

\[ \Delta < \epsilon \]
Adaptive Prediction

- Adaptively switch between multiple predictors
- Use predictor with lowest residual $r$ for pixel $p$ at position $x$

Grey indicates invalid (zero) pixels
Adaptive Prediction

- Adaptively switch between multiple predictors
  - Use predictor with lowest residual $r$
    for pixel $p$ at position $x$
- We use 4 simple but effective predictors:
  - Previous valid: $\text{Pred}_0(p) = p_X - p_A$

Grey indicates invalid (zero) pixels

Grey indicates depth
Adaptive Prediction

- Adaptively switch between multiple predictors
- Use predictor with lowest residual $r$ for pixel $p$ at position $x$
- We use 4 simple but effective predictors:
  - Previous valid: $\text{Pred}_0(p) = p_X - p_A$

Grey indicates invalid (zero) pixels
Grey indicates depth
Adaptive Prediction

- Adaptively switch between multiple predictors
  - Use predictor with lowest residual $r$ for pixel $p$ at position $x$
- We use 4 simple but effective predictors:
  - *Previous valid:* $\text{Pred}_0(p) = p_X - p_A$
  - *Up:* $\text{Pred}_1(p) = p_X - p_B$
Adaptively switch between multiple predictors

- Use predictor with lowest residual $r$ for pixel $p$ at position $x$

We use 4 simple but effective predictors:

- Previous valid: $\text{Pred}_0(p) = p_X - p_A$
- Up: $\text{Pred}_1(p) = p_X - p_B$
- Average: $\text{Pred}_2(p) = p_X - \frac{p_A + p_B}{2}$
Adaptively switch between multiple predictors

- Use predictor with lowest residual $r$
  for pixel $p$ at position $x$

- We use 4 simple but effective predictors:
  - *Previous valid:* $\text{Pred}_0(p) = p_X - p_A$
  - *Up:* $\text{Pred}_1(p) = p_X - p_B$
  - *Average:* $\text{Pred}_2(p) = p_X - \frac{p_A + p_B}{2}$
  - *MED-like:* $\text{Pred}_3(p) = p_X - (p_A + p_B - p_C)$

- Pixel-wise switching leads to high bit-overhead
Span-Based Adaptive Prediction

- Dynamically segment image into spans (1D blocks) of $n$ valid pixels
- Best suited regarding current memory layout

Grey indicates invalid (zero) pixels
Span-Based Adaptive Prediction

- Dynamically segment image into spans (1D blocks) of \( n \) valid pixels
  - Best suited regarding current memory layout
- Adaptively switch predictor per span
  - Evaluate all predictors for each pixel in span
  - Choose and encode best predictor \( k \) per span \( S \), based on minimal accumulated absolute error

\[
k = \arg\min_{i \in [0, 3]} \left\{ \sum_{p \in \text{valid}(S)} |\text{Pred}_i(p)| \right\}
\]
Span-Based Adaptive Prediction

- Dynamically segment image into spans (1D blocks) of $n$ valid pixels
  - Best suited regarding current memory layout
- Adaptively switch predictor per span
  - Evaluate all predictors for each pixel in span
  - Choose and encode best predictor $k$ per span $S$, based on minimal accumulated absolute error
  - Encode final residuals $r_p$ using $k$
  - Results in 2 bits for predictor ID per span

\[
    k = \arg\min_{i \in [0,3]} \left\{ \sum_{p \in \text{valid}(S)} |\text{Pred}_i(p)| \right\}
\]

\[
    r_p = \text{Pred}_k(p)
\]
Bit Reduction

- RVL has lower limit of 4 bits per valid pixel
Bit Reduction

- RVL has lower limit of 4 bits per valid pixel
- We additionally use Zstandard for further compression
  - Zstandard combines dynamic dictionary-based and ANS-based entropy compression
Parallelization

- Partition image in equal blocks
- Simultaneous processing by threads $t_i$
Parallelization

- Partition image in equal blocks
- Simultaneous processing by threads $t_i$
- Concatenation of results $r_j$ and relevant variables
Parallelization

- Partition image in equal blocks
- Simultaneous processing by threads $t_i$
- Concatenation of results $r_j$ and relevant variables

- Applied on prediction, (cut-down) RVL, and Zstandard
Result: Compression Ratio vs. Speed

The graph shows the compression ratio on the x-axis and the compression + decompression speed in MB/s on the y-axis. Different algorithms such as ANS, LZ4, RVL, Pred*, Zstd, PNG, and PredZ* are plotted on the graph. LZ4 and RVL have the highest speed with considerable compression ratio compared to others.
Result: Compression Ratio vs. Speed

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Usage in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24.4</td>
</tr>
<tr>
<td>2</td>
<td>26.6</td>
</tr>
<tr>
<td>3</td>
<td>21.2</td>
</tr>
<tr>
<td>4</td>
<td>27.7</td>
</tr>
</tbody>
</table>
Our PredZ:
- Has best compression
- Still reasonably fast
Result: Frame Delta and Filtering

![Diagram showing compression ratio for different algorithms with configuration options: None, Frame Delta, Temporal Filter, Both.](image-url)
Result: Frame Delta and Filtering

Configuration:
- None
- Frame Delta
- Temporal Filter
- Both

Algorithm:
- Pred*
- PredZ*
- RVL
- Zstd

Compression Ratio
Result: Speed Breakdown

![Speed Breakdown Diagram](image)

- **Algorithm**: Zstd, RVL, Pred*, PredZ*
- **Speed (MB/s)**: 0, 500, 1000, 1500

- **Compression**
- **Decompression**
Conclusion

• Novel real-time lossless depth-image compression algorithm
  • Effective temporal delta computation
  • Adaptive span-based prediction
  • Bit reduction
  • Multi-threaded implementation
Conclusion

- Novel real-time lossless depth-image compression algorithm
  - Effective temporal delta computation
  - Adaptive span-based prediction
  - Bit reduction
  - Multi-threaded implementation
- Significantly higher compression ratio than existing algorithms
  - Factor 1.73x higher than original RVL, 1.3x higher than Zstandard
- Real-time capable
Future Work

- General performance optimization
- SIMD
- Zigzag encoding
- 2D block prediction
- Last image’s neighbor values for intra-image prediction
Thank you!

rfischer@cs.uni-bremen.de