Misi

Binary Commands in lossless Real-time compression

Using Binary Commands and Pixel Manipulation as Basis for lossless Real-time compression

By Ela Shapira and Tamir Fridman - Managing Directors

www.misi-tech.com

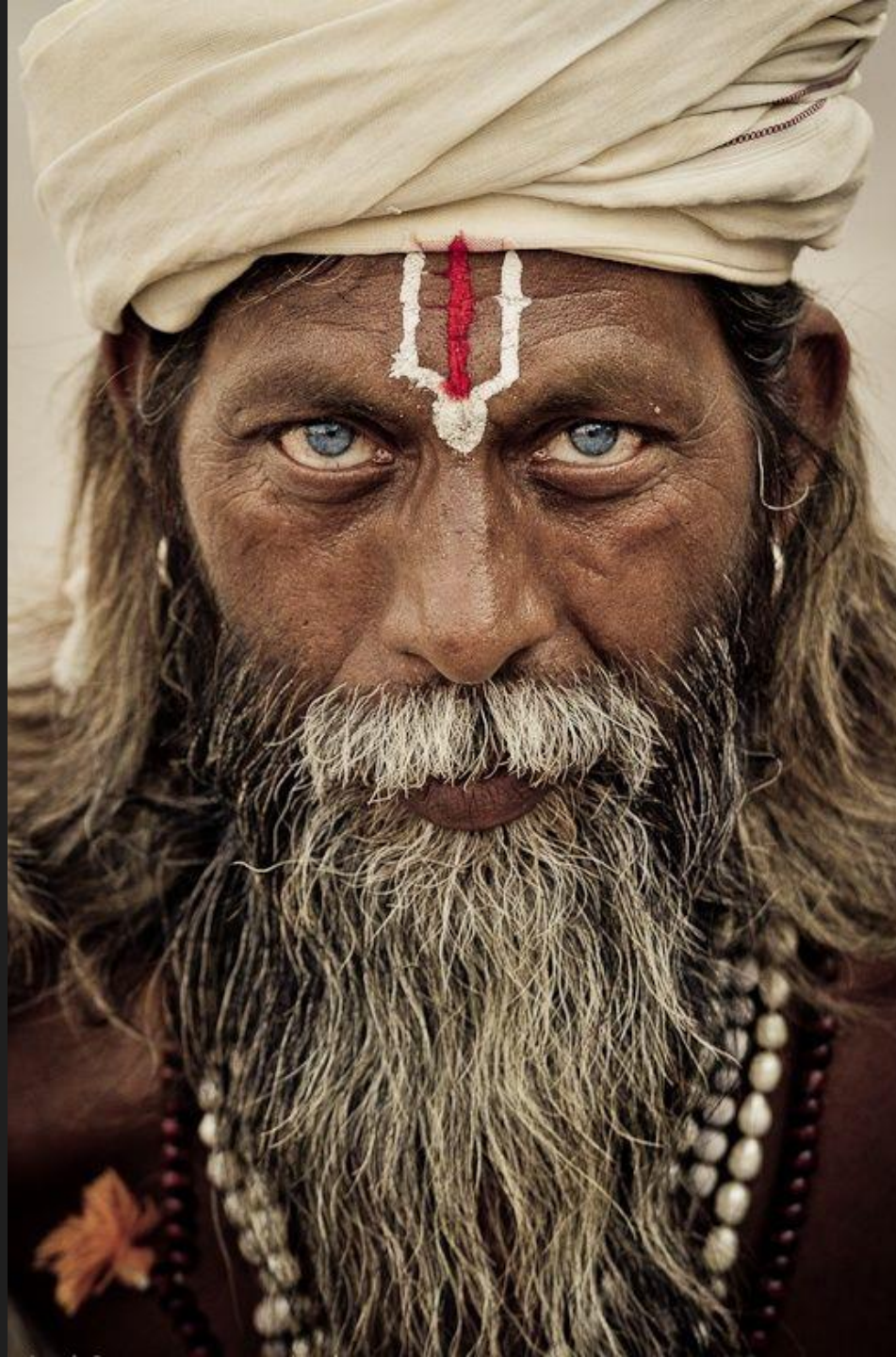

Table of content

Introduction

Purpose

The purpose of this paper is to describe new ideas in lossless compression as a key element for High-resolution streaming. These ideas are based upon review of the physics of the problem and an analysis of applicable technological approaches.

Understanding compression in real-time

Streaming of high-resolution video in real-time requires suitable compression of data to preserve appropriate frame rate during streaming. Large amount of visual data should be compressed and decompressed in real time without any loss of quality. None of the available compression algorithm provides a robust solution for per-frame loss-less compression in real time.

Challenges of high-resolution compression

Wide availability of High Resolution video cameras in handheld devices poses a challenge while trying to apply real-time streaming technology through cellular network. Limited network traffic creates the need to reduce the amount of data sent as much as possible, while preserving the high resolution. Limited CPU capabilities, compared to a media server, require innovative approach to compressing and decompressing the video stream.

Scope of binary commands in image compression

What is loss-less compression?

Lossless compression is an algorithm which rewrites the original data in a more compact manner, while ensuring that decompression results in the original data.

Using sequence analysis for compression

Image compression, being a compression of a very specific data structure, allows for sequence analysis as base for data compression. Color distribution in an image has strict observable rules, which can be described using ranges of functions instead of actual data.

Scope of binary commands in image compression

Classification of numeric sequences

Classifying repeating data sequences as binary commands enables replacing large sets of data with small sets of instructions with parameters, effectively resulting in lossless compression.

Class	Sequence type	Binary CMD
Solid color	Equal pixels	Repeat [pixel]
Gradient	Ascending / Descending	Asc / Desc [base] [delta]
Outline	Single sparse	Sparse [pixel] [step]

Indexing data as binary commands

When all sequence types are mapped in advance, instructions index is defined such that each instruction with its parameters takes at most four bytes while replacing at least six. Rigid set of predefined instructions enables quick dictionary use for compression and decompression on hand-held devices.

Using binary index in real-time

Compressing high-resolution image

Using sequence-based compression algorithm in real-time requires image processing to take about 10ms for compression and 10ms for decompression, to keep steady rate of 30 frames per second, while leaving 10ms for rendering. This limit allows for one pass only on the frame matrix, which means that no preliminary calculations can be made.

Generate new data-buffer

Generation of new binary stream out of incoming frames requires building of new data-buffer with mixed binary commands and data to happen in real-time, in one pass over the frame data. Sequences are defined as relations between consecutive pixels, in a way which reflects natural distribution of light.

Streaming data as binary commands

Stream is an ordered binary data; streaming of commands instead of original data raises additional challenges, as packet loss might result in indecipherable data. This forces meticulous separation of sets of instructions into data packets, such that lost data can be inferred from existing data.

Using binary index in real-time

Extraction of data-units

Extraction of original frame out of incoming binary stream requires rebuild of original frame pixels from binary data buffer in real-time, as the packets arrive. Each instruction is unfolded into a data stripe with a specific location in the image matrix, so when all instructions are processed the image is complete and ready to be rendered on screen.

Using binary index in real-time

Code Example

```
//pseudo code

For each pixel in image matrix
{
    If matches existing sequence
        Update sequence and continue
    Else
        Start new sequence
}
```


Summary

This paper presented a summary of the needs for new advanced real-time lossless compression algorithm when approaching high-resolution streaming technology to be done in real-time from hand-held devices with virtually no latency and without any loss of data and quality.

Thank you

Misi Tech Inc.

MisiCam video-camera and gaming console, download:

Download on the App Store

ANDROID APP ON Google play

[Watch our demo on YouTube]

Misi Tech INC.
340 S LEMON AVE #2396N , WALNUT, CA, 91789
www.misi-tech.com info@misi-tech.com

Misi