Chair of Spacecraft Systems
School of Engineering and Design
Technical University of Munich

Event Camera Lossless Compression for Satellite Applications

Event Data Compression on a Nvidia Jetson Orin Nano for CubeSats

Ottobrunn, 20.10.25

Antonio Junco de Haas– ge27cuy@mytum.de

Space present unique opportunities and challenges. CubeSats allow universities to build their own space missions.

Opportunities

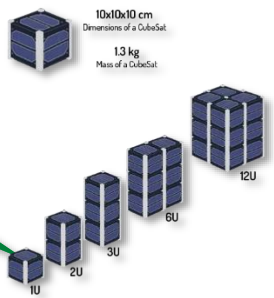
Satellites can observe space without atmospheric interference.

Cube Satellites have brought down costs, making it possible for universities to develop satellite missions.

- They use standardized 10x10x10cm cube Units (U).
- Popular for universities
- Mostly Low Earth Orbit Satellites

Satellite launch is possible thanks to rocket ride sharing.

Satellites are above the atmosphere



Each satellite Unit is 10x10x10 cm

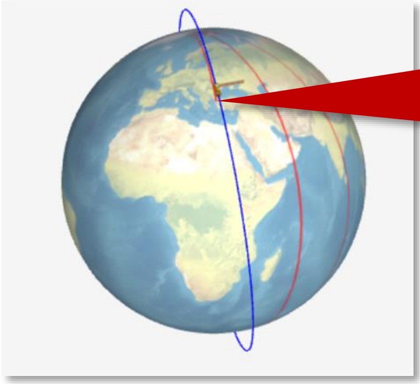
No need to pay for a whole rocket, just hitch a ride!

Satellite offer opportunities due to their unique location **in space**. Cubesats reduce costs and simplify development, allowing universities to build missions.

Space also presents communication challenges that must be addressed for mission to succeed.

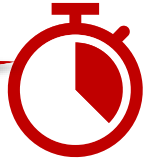
Challenges[1]

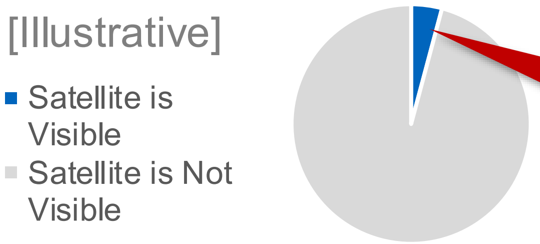
- **Limited communication windows with the satellite.**
- Satellite moves quickly through the sky.
- Data transmission down from the satellite (downlink) is a difficult task.
  - Satellites need to be visible to antennas on Earth to send and receive messages.



Communication is only possible when the satellite "passes over" a ground station

Maximum overpass estimated to be 9 minutes





Satellite is only visible for a fraction of the time

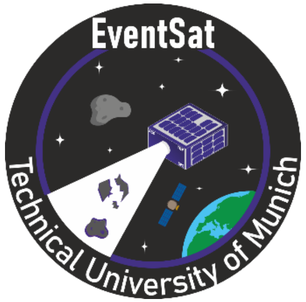
Satellite offer opportunities due to their unique location **in space**. **However, space introduces challenges**, as such **data must be clear, concise and comrpressed.**

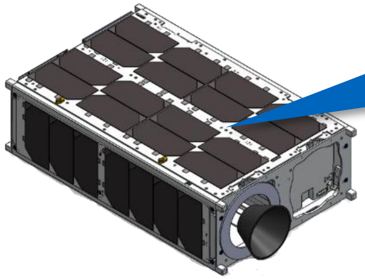
1. For a Low Earth Orbit Satellites

EventSat is a TUM CubeSat mission that will perform space observation in a Low Earth Orbit

Mission Parameters

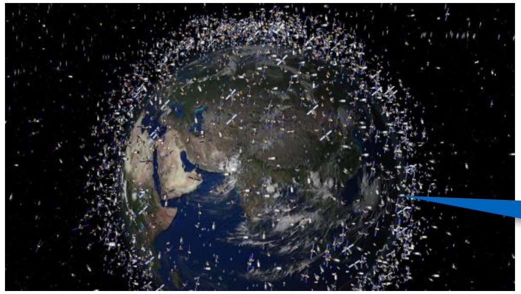
- University satellite mission developed by the **Chair of Spacecraft Systems.**
- The size of the cube satellite is **6-units (6U).**
- **Launch into Low Earth Orbit.**
- **The satellite will carry an Event Camera.**
- EventSat will observe stars, planets, inter-planetary objects, other satellites and **space debris.**
  - Space debris is of particular interest as it poses risks to future satellite missions





Eventsat badge and render

The event camera will be the main observation instrument



Space debris is becoming a risk

The onboard Event Camera seeks to observe and document **space observations.**

EventSat will be focused on observing the surrounding space environment.

Mission Objective 1

Perform real-time detection to identify objects in the event camera field of view using computer vision.

Mission Objective 2

Develop a database of space-based event camera observations.

This objective is relevant for the thesis, as a lot of data will need to be transmitted

The onboard Event Camera seeks to observe and document **space observations.**

The satellite will generate huge amounts of data. Which must be downlinked to the ground station.

Regular Image

An HD picture from a regular black and white camera is 1 MB. Less than 1% of an event camera observation.

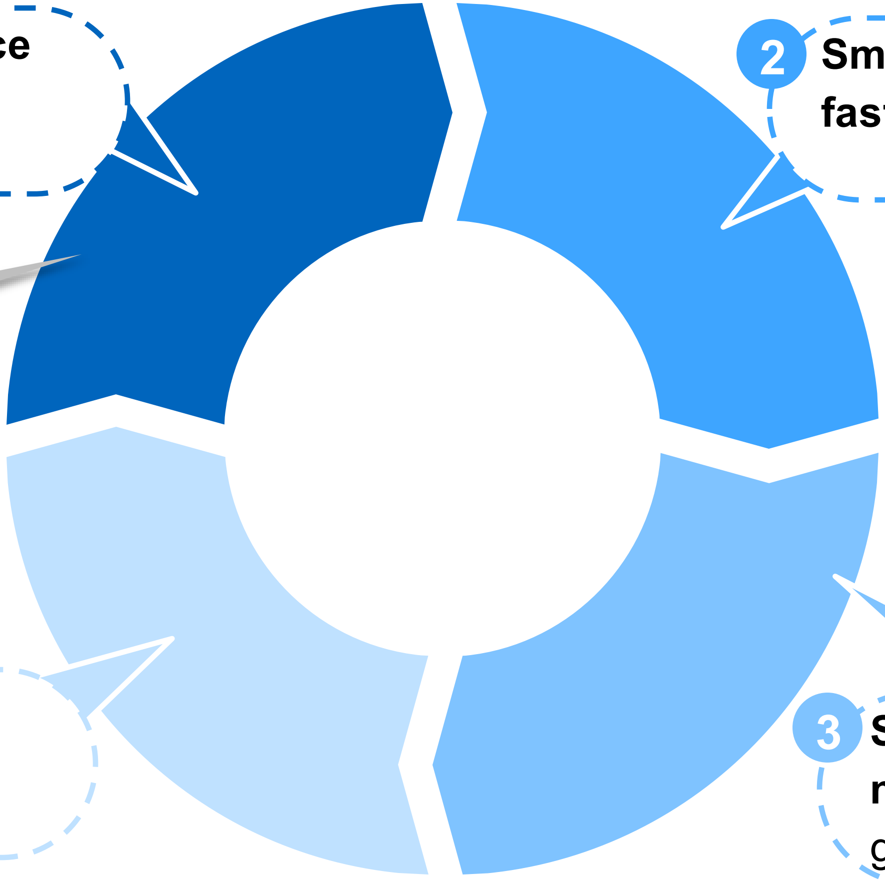
Satellite Observation

130 MB

A 44 second event camera observation can be upwards of 130 MB (after lossy compression)

Event cameras can generate huge files, as such data transmission will be a significant bottleneck for the mission.

Lossless compression will yields smaller files. Satellite can perform other operations, such as more observations.



1 **Lossless compression can reduce data file** size without deleting any information

Lossy compression exists, but is not the focus of this thesis

2 **Smaller files can be transmitted faster** to the ground station

3 **Smaller transmission times means more data can be transmitted** to the ground station

4 **More data can be transmitted**, therefore the satellite can **perform more observations.**

By utilizing lossless compression, the satellite can **reduce information size** without losing any data, **so satellite may then prioritize other operations** that can contribute to the mission objective.

An investigation for event data compression algorithms was performed, the conclusion was to create one in-house.

Algorithm	Result
Proprietary compression algorithms developed **by** the event camera **manufacturers**	**The only relevant event data compression algorithms** found were developed by the event camera manufacturers. Leaving **little demand for additional algorithm development**
 Generic zip file	**Not optimized for event data**, as such, algorithm does not understand the data structure to the same degree as the proprietary compression algorithms

Because of the **lack of third party algorithm development**, and the lack of data understanding, it was concluded that an **in-house compression lossless compression algorithm would be beneficial** to EventSat.

These thesis will seek to fullfill the following Resarch Contributions to deliver a successful solution

1. **A novel event data compression algorithm that uses a hybrid dictionary and variable-length encoding strategies.** The dictionary encoding approach categorizes the data by size and compresses it in a separate section from the rest of the file. The variable-length encoding approach integratescompression instructions, indicating size right next to the data.

2. **A comprehensive trade-off histogram analysis to determine the optimal compression thresholds** for dictionary and variable length. The algorithm calculates file sizes for each threshold until the minimum size is found.

The Research contributions revolve around compressing the file to the smalles size possible by finding optimal data thresholds

Example Event Camera Video of the Berlin night sky.

Event Data Explanation

- Camera is panning, creating the illusion of movement
- White pixels are where brightness increased
- Blue pixels are where brightness decreased
- Stars move directly upwards
- Other space objects are moving diagonally

An Event Camera is a specialized sensor that only records brigtness changes as Events, optimal for space applications.

Example Event Camera Observation

What is an Event Camera

- They **mimic human eyes** (neuromorphic)
- **They record Events, brightness intensity changes** , instead of taking pictures
- They can record with **ultra-low latency.**
- They have the potential to **surpass** traditional frame-based **"picture" cameras by detecting fast-moving objects**
- They are **considered for space applications** such as Space Situational Awareness (understanding the space environment around a spacecraft)

Event cameras have certain advantages for detecting objects. Potential applications is space are being considered

Events only record the pixel, polarity (brightness increased or decreased) and the timestamp where an event occurred

Raw Data

Number of pixels

```
%geometry:1280,720
189,513,0,13958339
640,467,1,13958368
887,102,0,13958570
243,693,0,13958593
674,220,0,13958653
1075,510,0,13958664
494,676,1,13958757
304,606,0,13958795
699,411,0,13959103
...
```

Processed Data

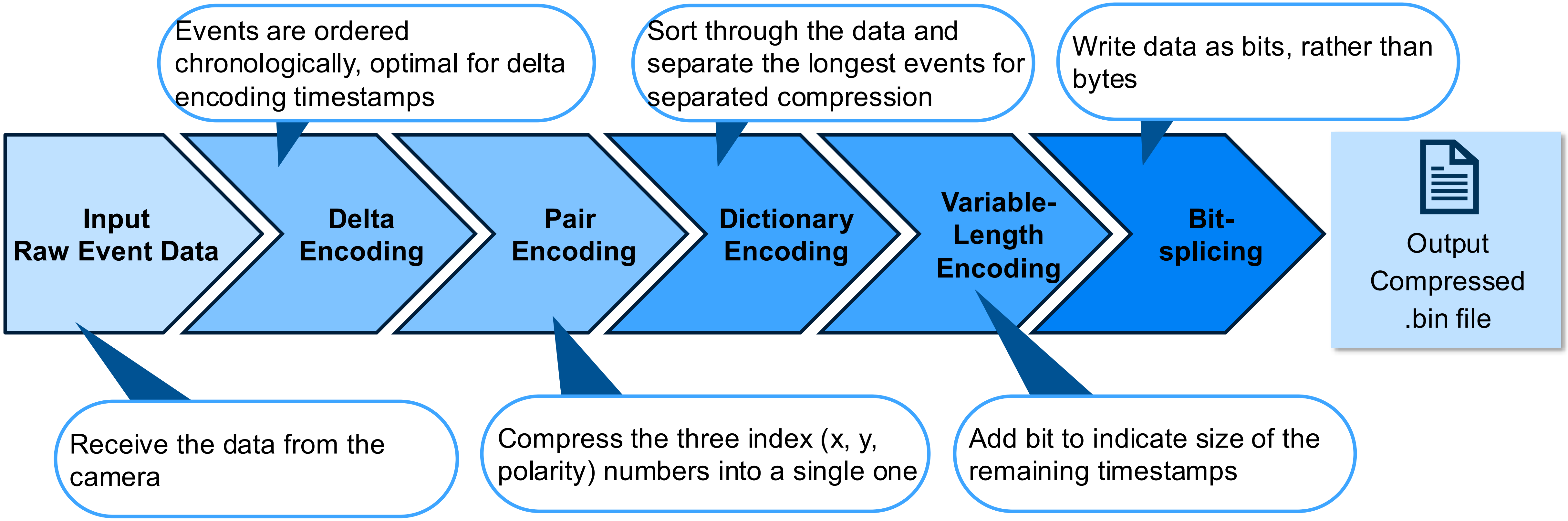
Id	X Coordinate	Y Coordinate	Polarity	Timestamp
1	189	513	0	13958339
2	640	467	1	13958368
3	887	102	0	13958570
4	243	693	0	13958593
5	674	220	0	13958653
	…	…	…	…

Index: Values always within a range x: [0-1279], y[0-719], p[0-1]

Timestamp: Always incremental

The geometry indicates number of pixels, **the index values** are can be any number **within the range** indicated by the geometry, **the timestamp is always incremental.**

Algorithm Overview



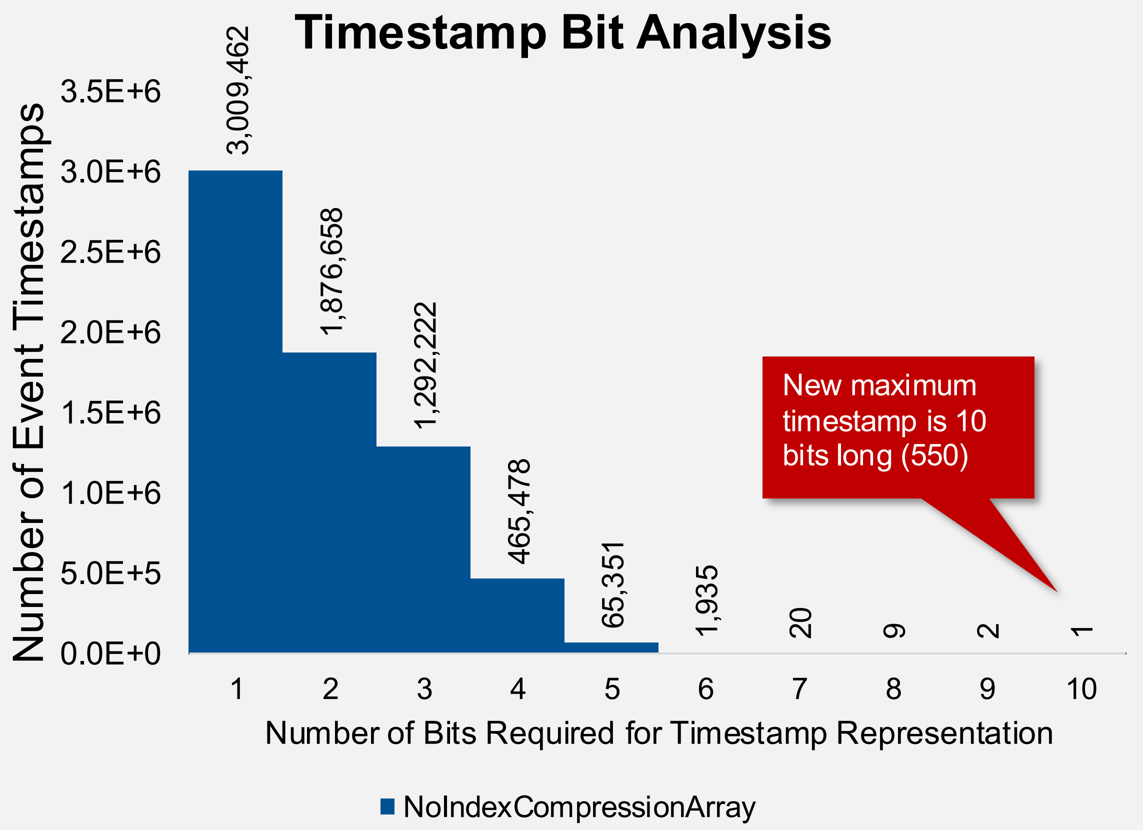
Delta encoding will reduce timestamp size by only recording the differences in values.

File values are changed

x,y,p	*t*
189,513,0	**13958339**
640,467,1	**13958368**
887,102,0	**13958570**
243,693,0	**13958593**
674,220,0	**13958653**
1075,510,0	**13958664**
494,676,1	**13958757**
304,606,0	**13958795**
699,411,0	**13959103**
…	**…**

→

x,y,p	*t*
189,513,0	**0**
640,467,1	**29**
887,102,0	**202**
243,693,0	**23**
674,220,0	**60**
1075,510,0	**11**
494,676,1	**93**
304,606,0	**38**
699,411,0	**308**
…	**…**



To delta encode optimally it is necessary to delve into **how the data works**. First **data needs to be in chronological order** (no negative values). **Only the difference in values is recorded,** if events are moved, they need a pointer to indicate location.

Pair encoding combines multiple natural numbers into a single one. It is an efficient way to compres index values.

File values are changed

x,y,p	t
189,513,0	0
640,467,1	29
887,102,0	202
243,693,0	23
674,220,0	60
1075,510,0	11
494,676,1	93
304,606,0	38
699,411,0	308
…	…

→

Index i	t
273186	0
922535	29
1277484	202
351306	23
971000	60
1549020	11
712713	93
438972	38
1007382	308
…	…

Values are multiplied by ranges to generate a unique index.

Index		
x coordinate	y coordinate	polarity
189	513	0
640	467	1
…	…	…

Multiply index by the maximum range value + 1

$(x_i)(y_{range}+1)(pol_{range}+1)$ $(y_i)(pol_{range}+1)$ (pol_i)

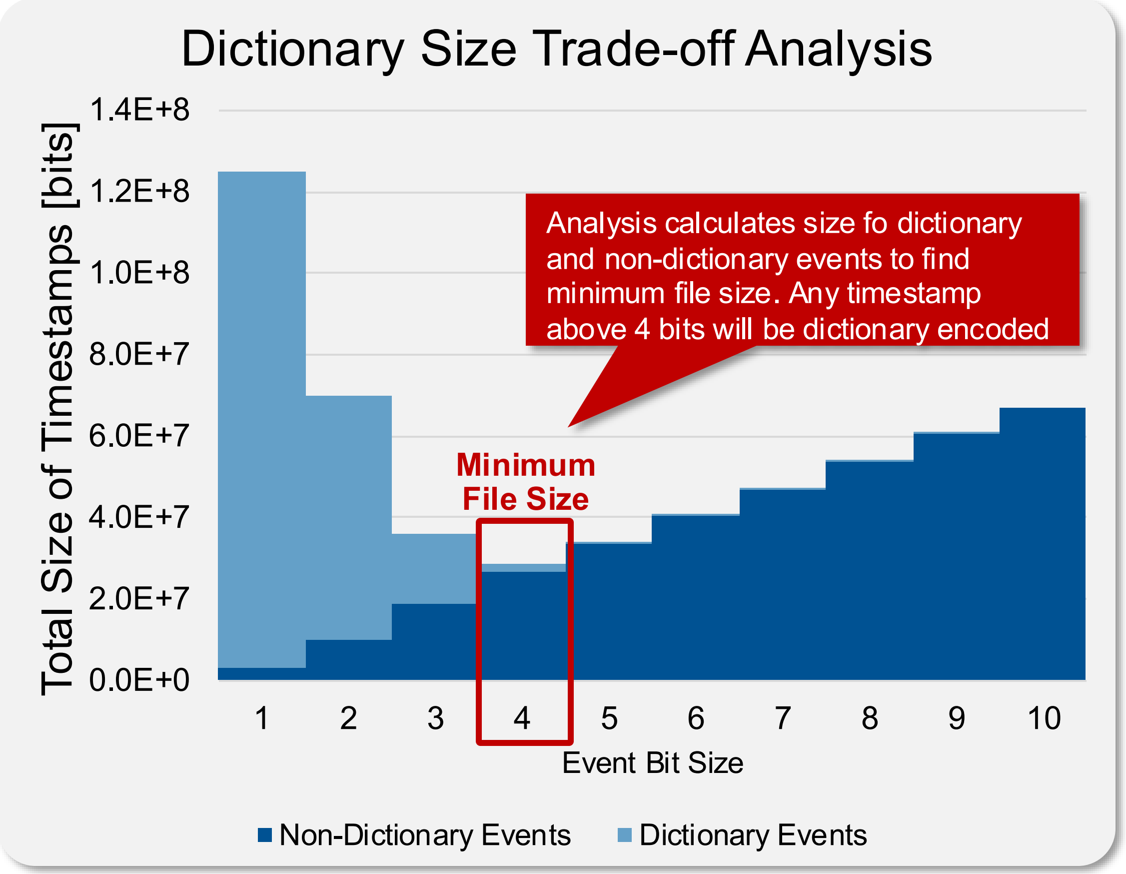
$(189)(720)(2)$ = 272160	$(513)(2)$ = 1026	0 = 0
$(640)(720)(2)$ = 921600	$(467)(2)$ = 934	1 = 1
…	…	…

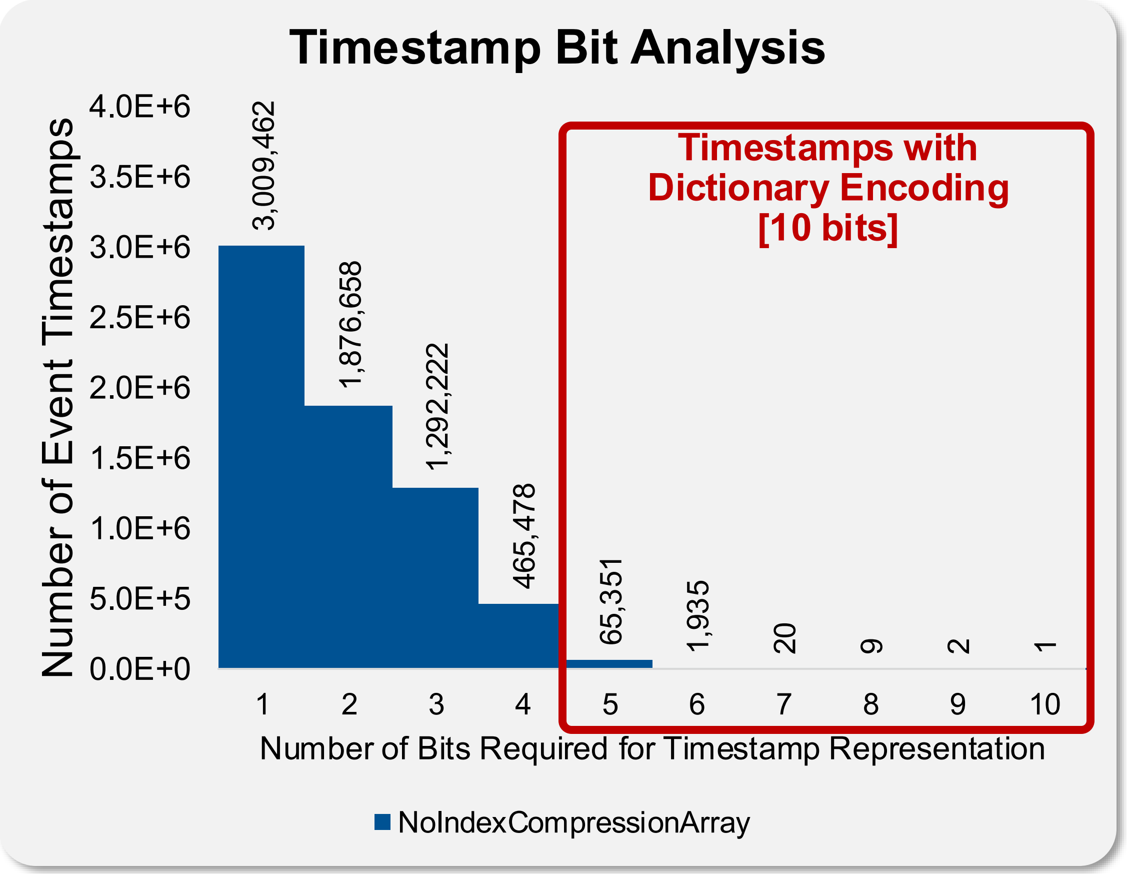
Sum the values together into a single Index number

Index number
272160 + 1026 + 0 = 273186
921600 + 934 + 1 = 922535
…

Pair encoding enables data to compress **3 integers into a 1 integer**.

Dictionary encoding separates long timestamp events. Requires a threshold to indicate events to compress.

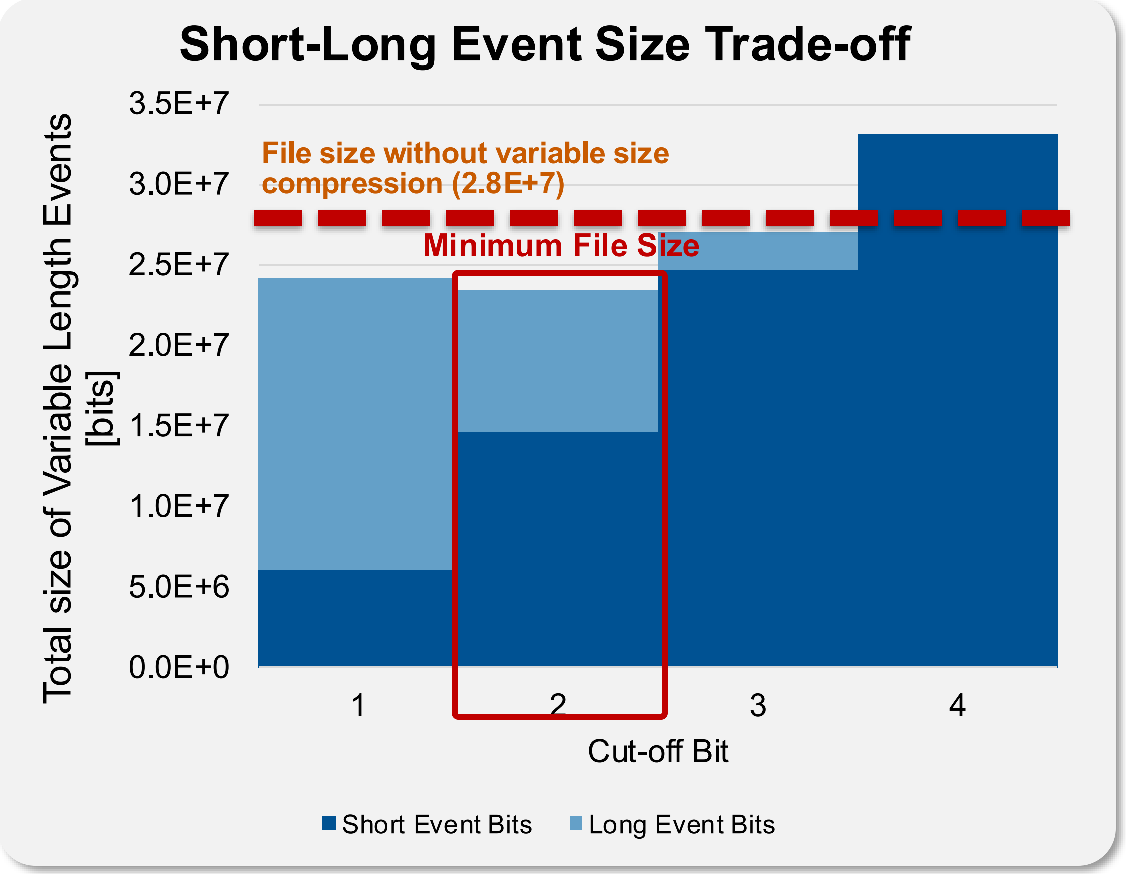


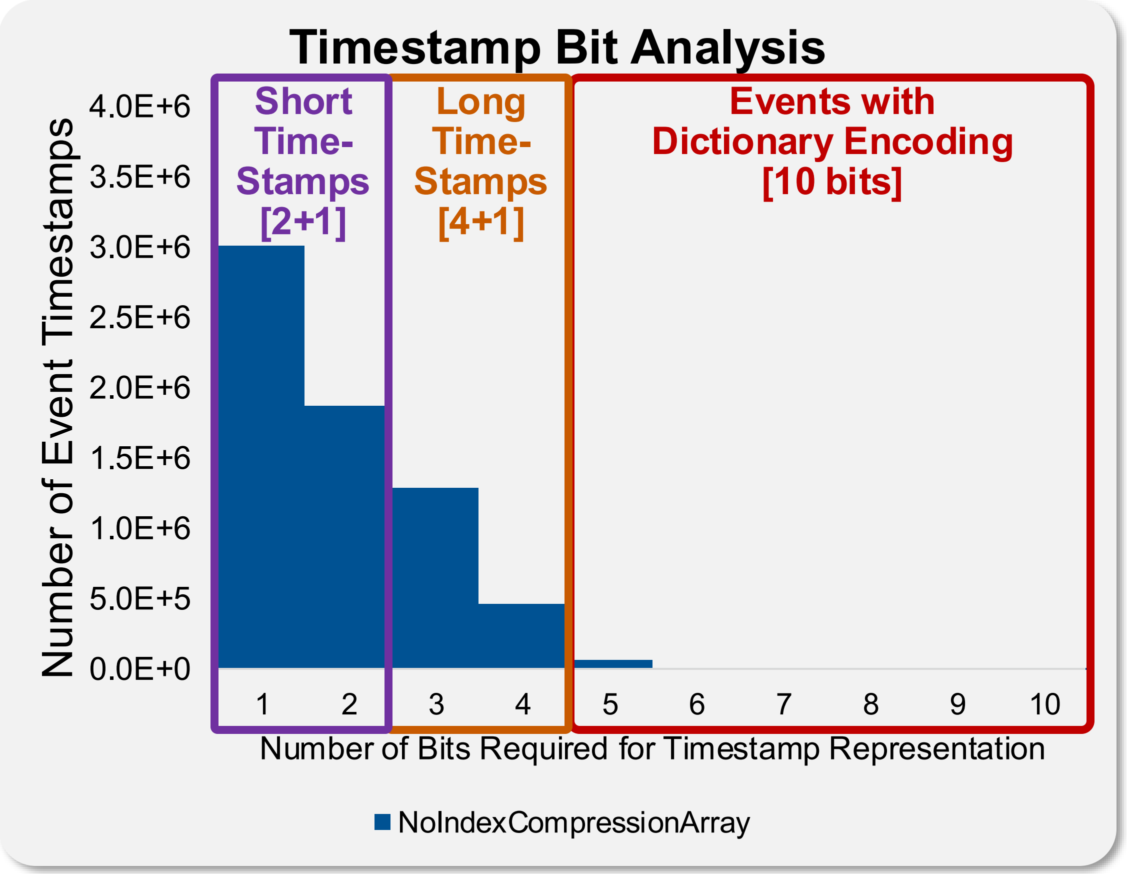


To produce the dictionary trade-off histogram, and extensive mathematical analysis was developed. File size was simulated for each bit threshold, and the optimal minimum file size is selected.

Dictionary events require an id. As such they will be slightly larger than before, however the rest of the events will need less memory allocation

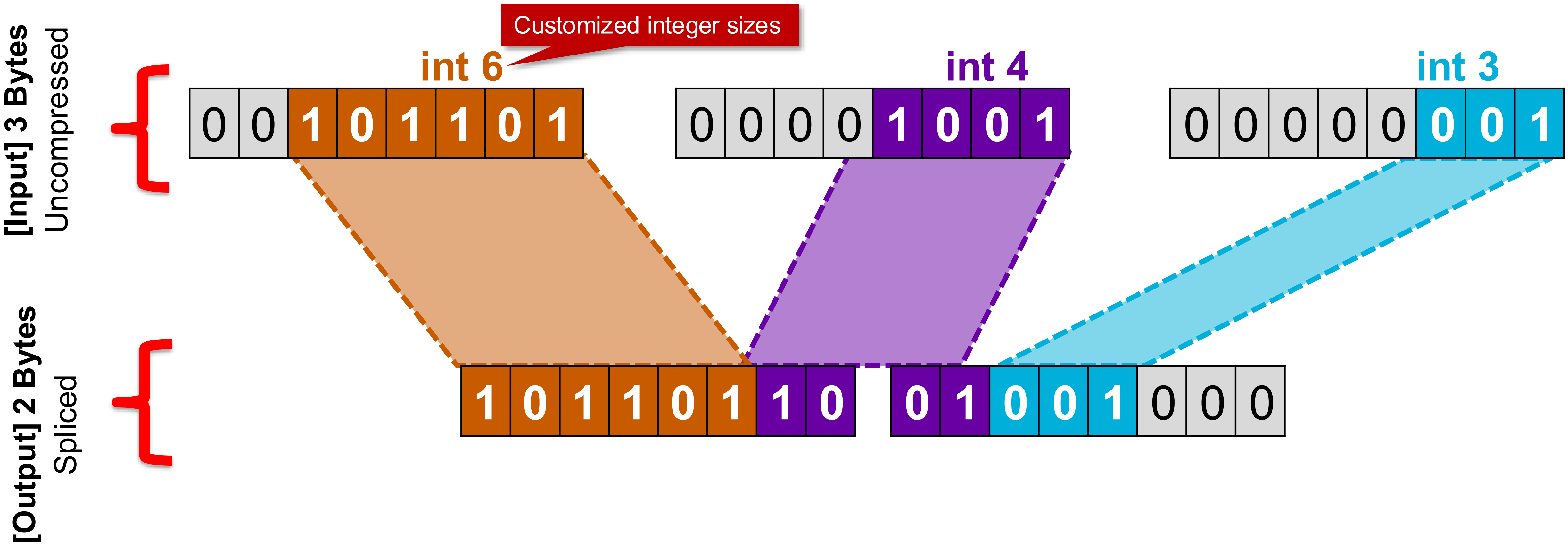
Variable length encoding adds one bit to indicate timestamp size, optimal threshold calculated.





An extensive histogram analysis was also developed for the variable length, similar to the dictionary histogram but with the additional consideration that results may be larger than the original size.

Values in computers are always written in bytes (8 bits) but with bit splicing, it is possible to write multiple numbers in a byte



Bit splicing permits computer to write sub-byte (1 byte = 8 bits) values. This improves algorithm compression rate, especially with variable length, where short and long timestamps are both less than 8 bits.

Berlin Dataset

London Dataset (low visible movement)

Madrid Dataset

The in-house algorithm [.bin] achieved the best compression ratio out of all the tested compression algorithms.

Algorithm	Compression Ratio [%]			
	Berlin Dataset	**Spin Dataset**	**Madrid Dataset**	**London Dataset**
[.raw] (Camera Manufacturers)	76.29	75.60	4.96	4.96
[.zip] (Windows OS)	68.03	67.02	64.14	64.18
[.bin] (EventSat Team)	**84.22**	**83.78**	**81.45**	**81.75**

The EventSat compression algorithm outperformed the competition. A higher compression ratio means better compression.

The in-house algorithm [.bin] achieved the smallest compression file out of all the tested compression algorithms

Algorithm	Compression Ratio [%]			
	Berlin Dataset	Spin Dataset	Madrid Dataset	London Dataset
Human Readable [.csv]	133,177,219	173,610,480	8,250,704	5,121,357
PROPHESEE METAVISION FOR MACHINES [.raw]	31,577,649	42,355,357	7,841,209	4,418,273
[.zip]	42,575,962	57,257,994	2,958,836	1,834,221
EventSat Technical University of Munich [.bin]	**21,019,730**	**28,167,826**	**1,530,245**	**934,837**

The EventSat compression algorithm outperformed the competition. A smaller file size means less data needs to be transmitted

To measure algorithm's performance it is important to analyze each encoding step to visualize the change in file size.

Ablation study breaks down each individual step of the algoithm

Data Format	Individual Event Sizes	Event Distribution[1] [%]	File Size[1] [MB]
Raw String	**18, 19, 20 bytes**	100	109
Event as Integers	**16 bytes**	100	91
Dictionary Encoding	Dictionary: **11 bytes** Non-Dictionary: **7 bytes**	1 99	39.7
Variable Encoding	Dictionary: **11 bytes** Long Events: **7 bytes** Short Events: **7 bytes**	1 73 26	39.7
UnsignedIntegers/ Custom Integer Length/ Bit Splicing	Dictionary: **54 bits** Long Events: **27 bits** Short Events: **25 bits**	1 73 26	19.13

The combination of all the strategies in files that are approximately 20% of the original file size. This result is due to the hybrid approach between the dictionary and variable-length encoding. As well as calculating optimal thresholds.

In summary, the compression algorithm was a success with up to 84% compression rate.

Summary

- The algorithm achieved 84% compression rate**. Better than the state of the art.**
- The reason this result was achieved was by understanding data at a deeper level.
- I would like to detail more about the histogram analyses, as they required considerable mathematical calculations for thresholds. More details in the thesis
- Other compression strategies were tested, but were abandoned due to lack of file integrity
- Files are documented in the EventSat Gitlab

Future Work

- Dictionary compression can be improved by adding encoding instructions in timestamp, rather than utilizing an id
- Huffman encoding could reduce indexes

A lossless event compression algorithm was delivered that outperformed the competition. Resulting in files up to 84% of compression ratio

Back Up Slides

https://observablehq.com/@jake-low/satellite-ground-track-visualizer

https://www.racecar-engineering.com/articles/f1/how-to-create-a-2017-f1-car/

…

Indexing Experiment

...

Zip (what it is using)

...

Backup Arithmetic

...

Chair of Spacecraft Systems
School of Engineering and Design
Technical University of Munich

Hello

Justification and relevance for the EventSat event file compression algorithm

Question	Answer/Justification
What is this presentation about?	**Lossless compression strategies** for event data file reduction.
Why is this relevant to discuss?	Due to several mission parameters, **data transmission will be the main bottleneck**. **Compression can provide support**.
What is the benefits of using lossless and lossy data compression?	**Lossless maintains integrity of the observations, this is the main topic of thesis.** Lossy is best to filter out spurious data.
Is there an evaluation benchmark?	**Camera** has it's a **compression algorithm** that yields a .hdf5 file. **The algorithm proposed shall compress smaller files**.

Compression will help mission observe longer by reducing data size for easier transmission

General information (1/2)
Data bytes and the difference between signed and unsigned

Data Types and Files

- Typical numerical representation is handled in integers, long long, or short
- **All variables and files** in c++ (including binary files) **must be written in full bytes**, files are traditionally in chars

integer (4 bytes):	0000 0000 0000 0000 0000 0000 0000 0000
long long (8 bytes):	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
short (2 bytes):	0000 0000 0000 0000
char (1 byte):	0000 0000

Signed vs Unsigned

- The **first bit of an int** indicates if **value is negative**, **unsigned integers** therefore allow for **double values**

max signed int: **0**111 1111 … 1111 1111 = 2,147,483,648 = 2^{31}

max unsigned int: **1**111 1111 … 1111 1111 = 4,294,967,296 = 2^{32}

Key considerations: Files and variables are written in bytes; Utilizing **unsigned** variables **reduces** each measurement by **one bit**
Expected measurements number in from hundreds of thousands to **millions**, therefore **reducing bits per measurement is necessary**

General information (2/2)
Bit Splicing and Encoding strategies

Custom int length and Bit Splicing

- If the recorded value of a measurement is considerably lower than integer one could apply a "**customized integers length**"
- **Bit splicing** is a technique to help write multiple values even if the data type isn't measured in bytes

Maximum recorded value: **286**
int_9(using short): **1000 1111 0000 0000 [black is empty data]**
Values to be recorded: **286**, **160**, **32**
Bit spliced **1000 1111 0010 1000 0000 0100 0000**

Delta Encoding

- When **values are stored in ascending or descending order**, delta encoding helps reduce maximum data values by storing the difference
- **Works** especially **well when data is dense** (small changes in between measurements compared to data size)

Normal Values: 13465485, 13465958, 13465974, 13465978, 13466166
Delta Encoded Values: 13465485, 473, 16, 4, 188
Delta divided by Normal Values: 1, 0.0000351, 0.0000011, 0.0000003, 0.0000140

The length of the custom int is known beforehand, and recorded somewhere when decompression is required. **Delta encoding** can only be used in **sequential order**, **if** values are **resorted**, some tracking **id must be implemented**

Below is the Event File Data Structure, unlike regular “RGB” cameras, events only consist of coordinates polarity and time

- An uncompressed event file is usually represented as a .csv or a .txt
- Data is human readable, however data filesize is large, as each written character is 1 byte (including commas and enters [¶] after each event)

	Values written in the file				
# of Event	X Coordinate	Y Coordinate	Polarity	Timestamp	Total Event Size
Event 1	272,	212,	0,	13465485¶	19 bytes = 152 bits
Event 2	1069,	703,	0,	13465958¶	20 bytes = 160 bits
Event 3	271,	455,	1,	13465974¶	19 bytes = 152 bits
Event 4	150,	76,	0,	13465978¶	18 bytes = 144 bits
Event 5	1040,	57,	0,	13466166¶	19 bytes = 152 bits
	…	…	…	…	…

Simply transforming values to the appropriate c++ data types would be inefficient. As an integer has a total of 4 bytes (32 bits), and Boolean is 1 byte (8 bits). An event would then be 3 ints and 1 bool for a total of 13 bytes (104 bits). **Data reduction is necessary.**

Strategy 1: Timestamp Delta Encoding

- Each timestamp has the full time value written, these are also written in **sequential order**
- Due to this, values could be reduced by simply reading the difference

Pixel and Polarity	**Timestamp**
272,212,0,	**13465485**
1069,703,0,	**13465958**
271,455,1,	**13465974**
150,76,0,	**13465978**
1040,57,0,	**13466166**
368,506,1,	**13466205**
688,123,0,	**13466217**
989,199,0,	**13466219**
478,665,0,	**13466338**
462,630,0,	**13466467**
80,711,0,	**13466496**

Pixel and Polarity	**Timestamp**
272,212,0,	**13465485**
1069,703,0,	**473**
271,455,1,	**16**
150,76,0,	**4**
1040,57,0,	**188**
368,506,1,	**39**
688,123,0,	**12**
989,199,0,	**2**
478,665,0,	**119**
462,630,0,	**129**
80,711,0,	**29**

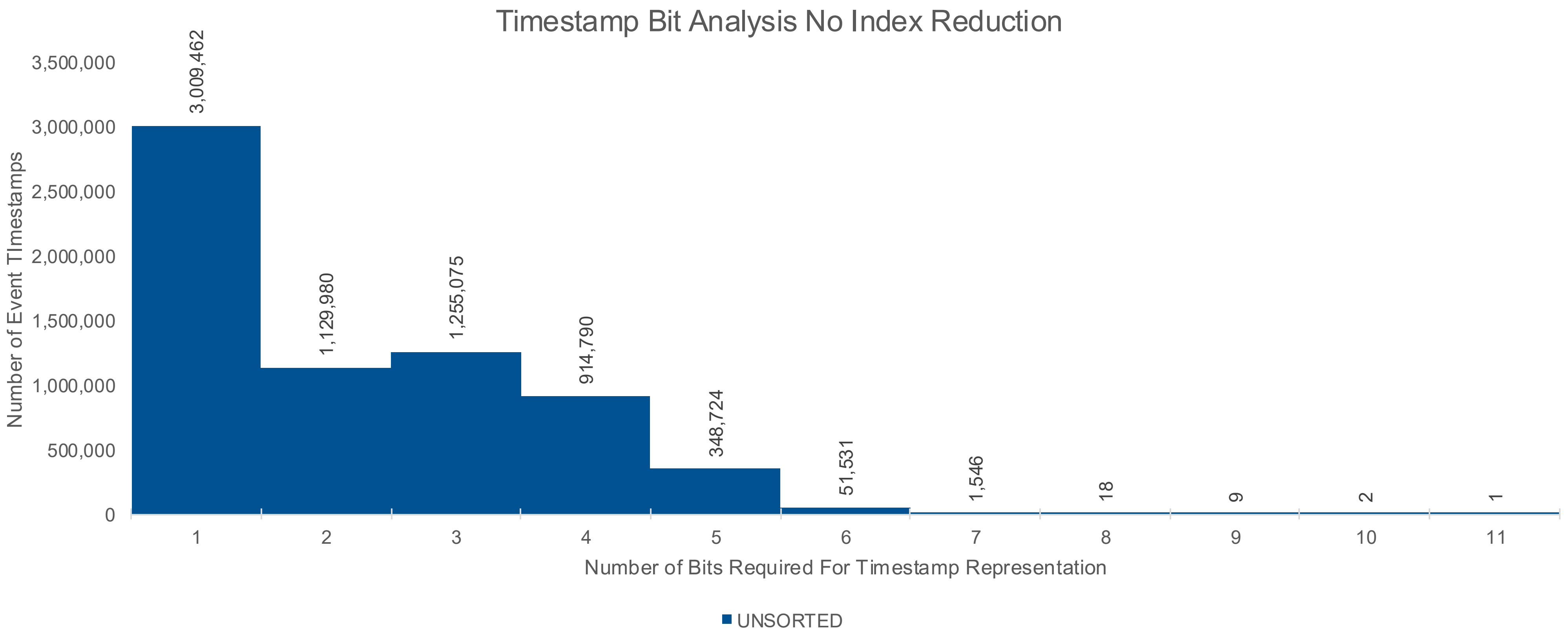
Timestamp Delta Encoding

- As a consequence, Timestamp is greatly reduced, in our test the **largest value** (without counting the initial) **was of 1440**, which can be **represented with 11 bits** rather than the **largest uncompressed timestamp** of **28237230**, which needs **25 bits**.

Pixel and Polarity	**Timestamp**
272,212,0,	**13465485**
1069,703,0,	**13465958**
271,455,1,	**13465974**
150,76,0,	**13465978**
1040,57,0,	**13466166**
368,506,1,	**13466205**
688,123,0,	**13466217**
989,199,0,	**13466219**
478,665,0,	**13466338**
462,630,0,	**13466467**
80,711,0,	**13466496**

Pixel and Polarity	**Timestamp**
272,212,0,	**13465485**
1069,703,0,	**473**
271,455,1,	**16**
150,76,0,	**4**
1040,57,0,	**188**
368,506,1,	**39**
688,123,0,	**12**
989,199,0,	**2**
478,665,0,	**119**
462,630,0,	**129**
80,711,0,	**29**

After Delta encoding a file, the number of bits required to represent an event will be reduced



Key Takeaway: After delta encoding, the maximum number of bits required is only 11

Visual Representation of Pair Enoding

Quick example, imagine we have 3 variables x, y, z that each can have a three values: ***0, 1, 2***

As such one possible representation of x, y, z would be a **6 bit number:** 100110_2 in binary

But with **arithmetic encoding** it can be reduced to a **5 bit number: 10111_2** in binary

To decompress simply use the modulo function and floor division

Multiply original values by maximums + 1

$x = 2$
$y = 1$
$z = 2$

$(2)(y_{max}+1)(z_{max}+1) = (2)(3)(3) = 18$
$(1)(z_{max}+1) = (1)(3) = 3$
$(2) = 2$

Sum the values together

Decimal	**Binary**
18	10010_2
+ 3	+ 00011_2
2	00010_2
23	10111_2

$\boxed{1\,\boxed{0\,1\,\boxed{1\,1}}}_2$

Decompress by using modulo (%) and floor division (⌊/⌋)

$z = (23 \% (z_{max}+1)) = (23 \% 3) = 2 \mid \lfloor 23/3 \rfloor = 7$
$y = (7 \% (y_{max}+1)) = (7 \% 3) = 1 \mid \lfloor 7/3 \rfloor = 2$
$x = 2 = 2$

Visual Representation of Pair Enoding

Index		
x coordinate	y coordinate	polarity
189	513	0
640	467	1
…	…	…

Multiply index by the maximum range value + 1

$(x_i)(y_{range}+1)(pol_{range}+1)$	$(y_i)(pol_{range}+1)$	(pol_i)
(189)(720)(2) = 272160	(513)(2) = 1026	0 = 0
(640)(720)(2) = 921600	(467)(2) = 934	1 = 1
…	…	…

Sum the values together into a single Index number

Index number
272160 + 1026 + 0 = 273186
921600 + 934 + 1 = 922535
…

Arithmetic Encoding for Index Pixel and Polarity compression

- **Pixels** and **polarity** can be represented as a **single number** (X × 720x2) + (Y × 2) + (POL × 1).
- This is a “Customized variable base” where Polarity is [0,1], Y [0,719], X [0,1279].
- Therefore **pixels** can be **represented by** a **unique number** from [0,1843200] in **21 bits**

Pixel and Polarity	Timestamp
272,212,0,	13465485
1069,703,0,	473
271,455,1,	16
150,76,0,	4
1040,57,0,	188
368,506,1,	39
688,123,0,	12
989,199,0,	2
478,665,0,	119
462,630,0,	129
80,711,0,	29

Pixel and Polarity	Timestamp
392104	13465485
1540766	473
391151	16
216152	4
1497714	188
530933	39
990966	12
1424558	2
689650	119
666540	129
116622	29

Run Length Encoding for Polarity

- It is possible to further reduce the Index by utilizing Run Lenth Encoding. First separate the values [0,1] and record how many events are on each. Similar to how one orders words alphabetically

Sort the events by the Index Polarity

Pixel and Polarity	Timestamp
272,212,**0**,	13465485
1069,703,**0**,	13465958
271,455,**1**,	13465974
150,76,**0**,	13465978
1040,57,**0**,	13466166
368,506,**1**,	13466205
688,123,**0**,	13466217
989,199,**0**,	13466219

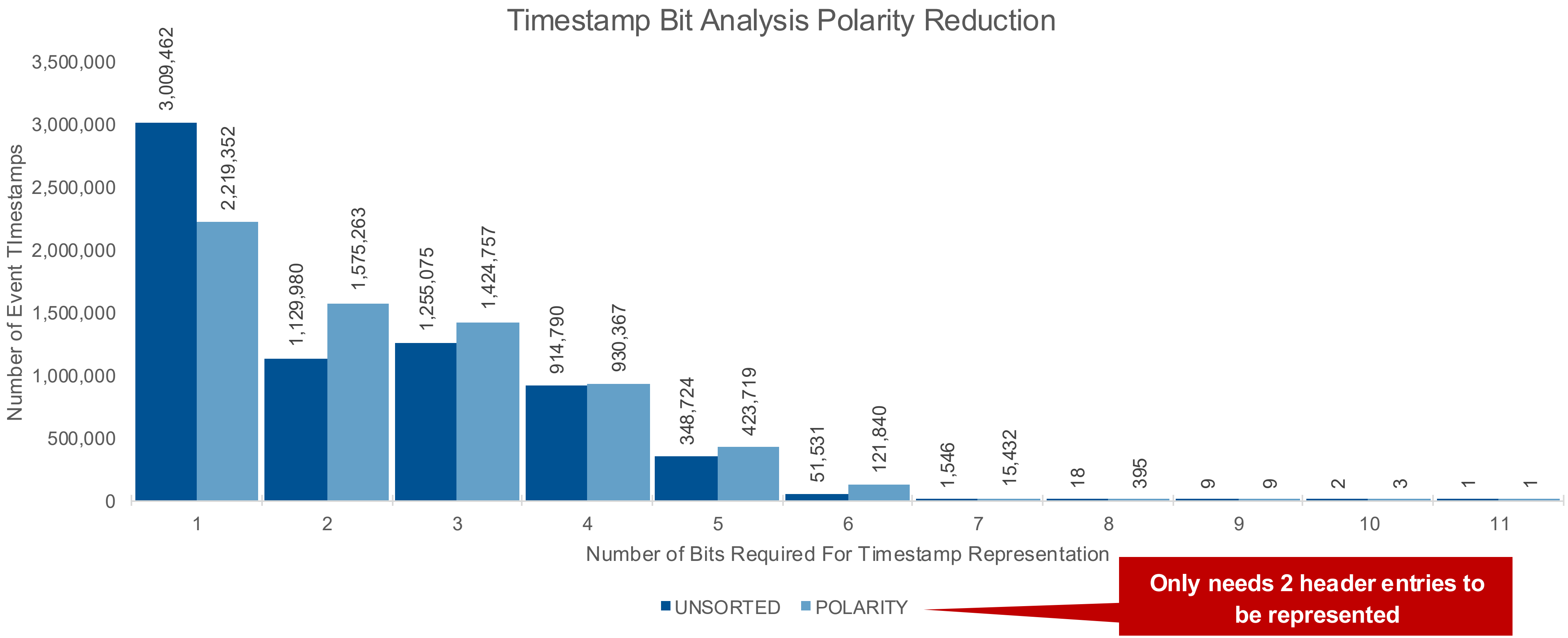
POLARITY 0, 3584555
POLARITY 1

	P	
Header for Polarity 0: 3,355,579	272,212,**0**,	13465485
	1069,703,**0**,	13465958
	150,76,**0**,	13465978
	1040,57,**0**,	13466166
	…………………………..	………………
Header for Polarity 1: 3,355,559	271,455,**1**,	13465974
	368,506,**1**,	13466205
	41,65,**1**,	13467692
	1019,256,**1**,	13467785

After sorting by polarity, timestamp is no longer sequential

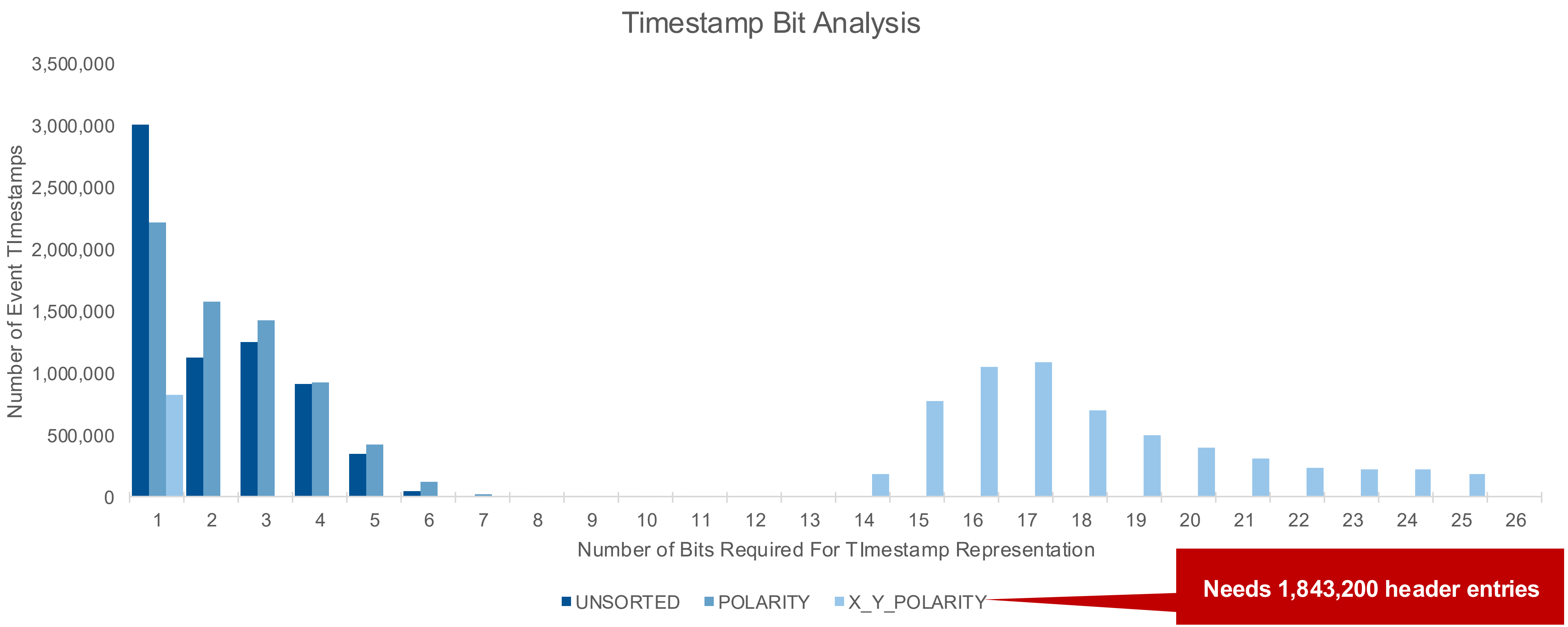
This will reduce each event entry by 1 bit as long as there is a header that indicates how many polarity 0 an 1 events exist

Because Run-Length Encoding and Delta encoding both require sorting, they reduce the effectivity of each other



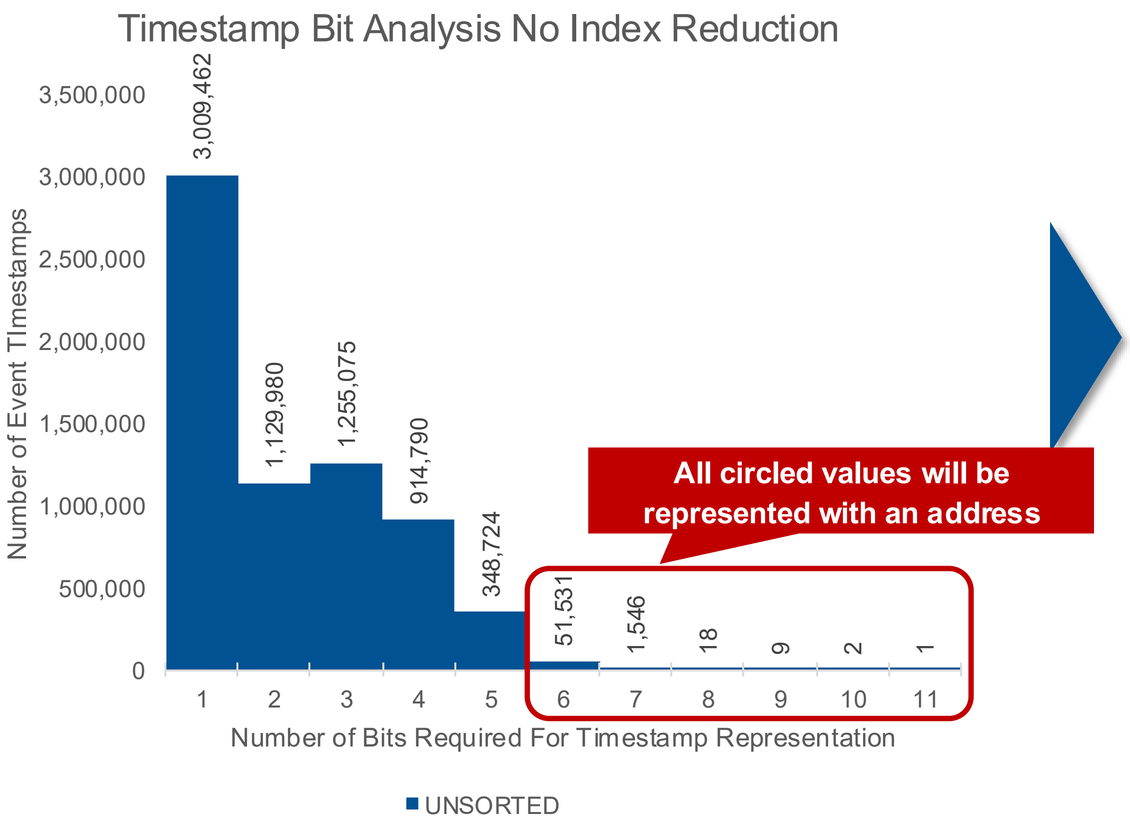
Unsorted values require less bits for timestamp representation, while polarity sorted events require more. **However, in this particular example** they **both need a maximum of 11 bits to represent the timestamp**

To show the incompatibility, a Run-Length encoding for XYPol will fully eliminate the Index, but larger Timestamps remain



If the algorithm were to **fully remove the index, timestamps** would **require 26 bits** to be represented **plus an additional header** to record the number of events for each pixel and polarity must be implemented, **a balance between index, timestamp, and header must be calculated**

Dictionary Encoding can be utilized to reduce the outlier values, as they can be represented with another number



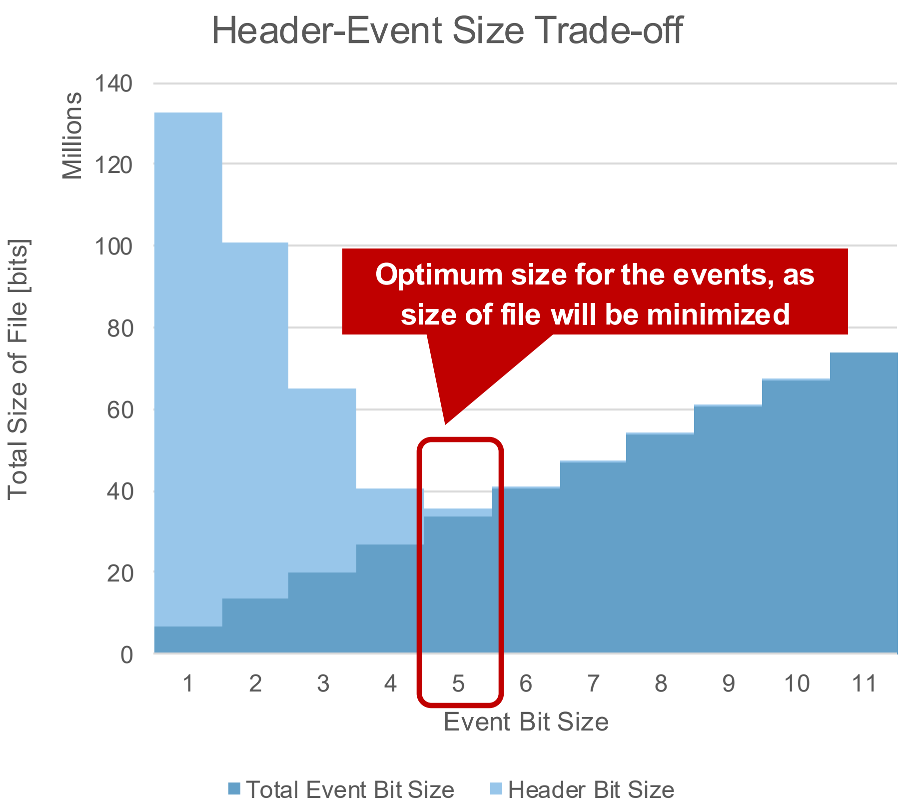
- The last 4 bit buckets in the graph are sparse, and as such many values are completely empty.
- Dictionary Encoding can allow us to represent these large values with a smaller "pointer" addresses if we utilize an id system

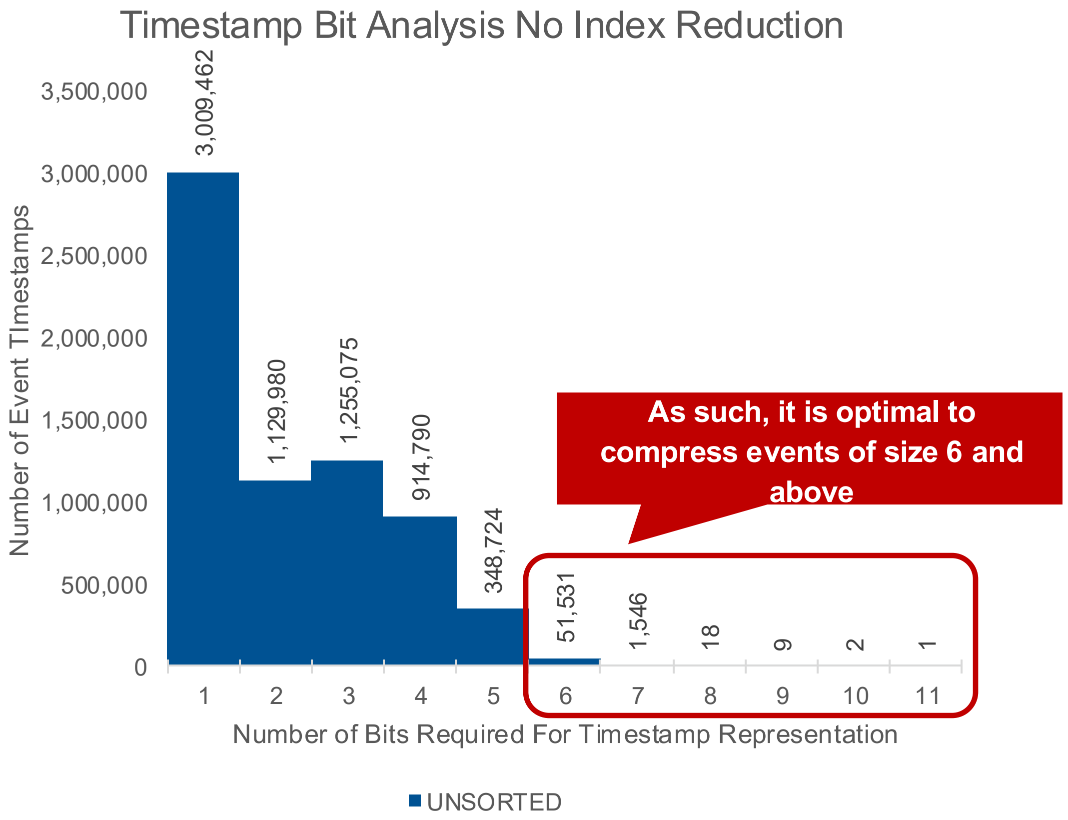
Event Id	Timestamp
400,578	290
1,752,005	514
2,278,656	3977
…	…

- For this encoding to work data must be sparse, as each pointer will need to represent a large number for the Event Id (size will be max number of events)
- There are some cases where it could be encoded into 7 bits, if enough zero values are available, but a further analysis is required to show the frequency of each number
- Will be impacted by Run-Length encoding

Afterwards the maximum bit necessary for timestamps will be 8 instead of 11

To find the optimal event size a histogram analysis is calculated with the sum of the header and event entries





Afterwards the maximum bit necessary for timestamps will be 8 instead of 11

Chair of Spacecraft Systems
School of Engineering and Design
Technical University of Munich

Recalculate numbers

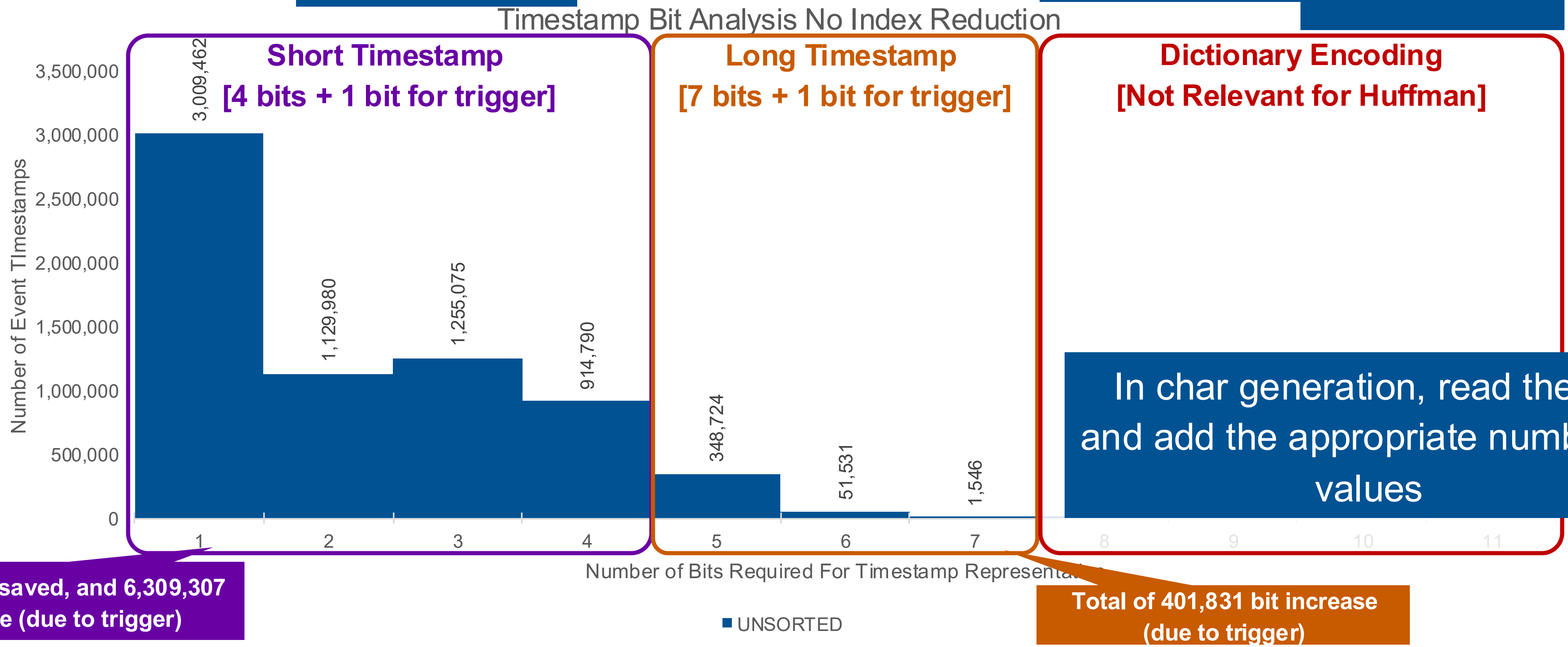
Huffman Encoding is a way to have variable length data, therefore events … shor…

Add id = 0

Add id = 1

Add id = 2

%%%



In char generation, read the id, and add the appropriate number of values

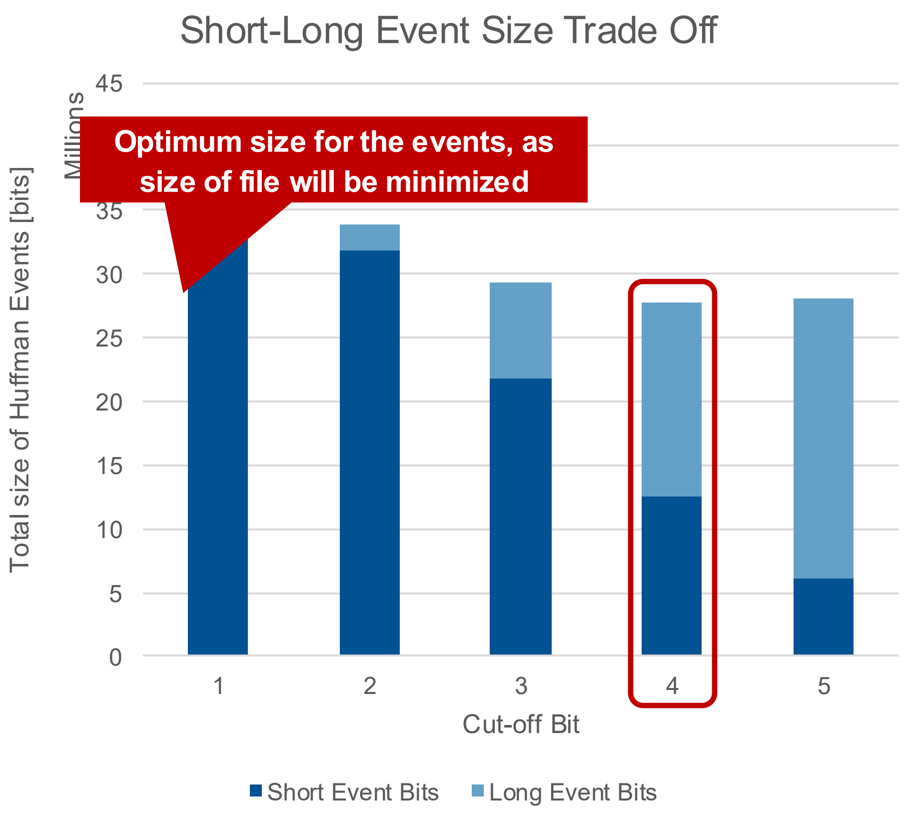
44,165,149 bit saved, and 6,309,307 bit increase (due to trigger)

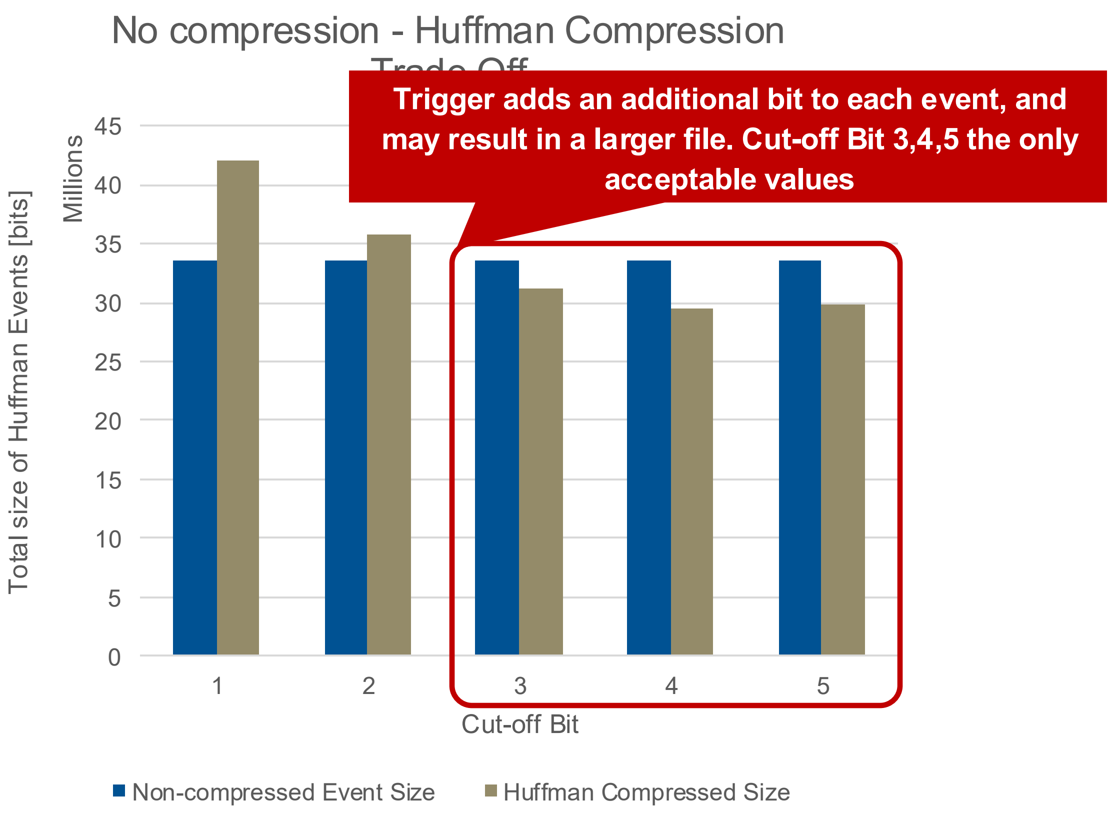
Total of 401,831 bit increase (due to trigger)

The Huffman encoding will add a trigger [0,1] that will indicate if an event has a **long or short timestamp**. The length of the short/long timestamp will be **file specific**. The choice of the trigger locations will be influenced by the Dictionary Encoding

Chair of Spacecraft Systems
School of Engineering and Design
Technical University of Munich

To find the optimal short-long event size a histogram analysis must be made, as well as a non-compression comparison





Afterwards the maximum bit necessary for timestamps will be 8 instead of 11

These encoding strategies will be utilized with different parameters to find an acceptable filesize for downlink

Huffman, and Dictionary are still a work in progress

Should it be a bit every single entry? Or should it be a trigger flag that executes after a certain amount of entries?

Future strategies to be implemented (1/2)
Data reduction by adding indexes

Dictionary based encoding

- Dictionaries work by having an **index** that **represents values**.
- This can **compress long** and **complex** that are **repeated** often **into smaller** ones.
- **Not all** values **need** to have an **index**

The most common word in English is **The**, a dictionary encoding would assign it a number **[1]** to compress a sentence.
When **1** stars shine over **1** lake, **1** stargazer went to observe **1** celestial objects, and record **1** movements in **1** journal.

Variable Length Data

- Most measurements don't need the full memory allotted to them, having some kind of indicator for a reduced memory entry to minimize empty space

0 index means only assign 4 bits per entry measurement
1 index means assign max 8 bit per entry measurment
Entry 1: 0 1101 0001 1000 | Total 13 bits
Entry 2: 1 0010 1000 0001 0011 0001 1111 | Total 25 bits
Entry 3: 0 1000 0011 1010 | Total 13 bits

These strategies are **heavily dependent on the specific details of the data**, and will **yield different results** even if the instrument and measurement time doesn't vary

Future strategies to be implemented (2/2)
Iterating to find the best compression strategy for file

Iterative File Specific Encoding

- Encode a file with different strategies multiple times to see which ones yield a smaller result file

Some strategies such as variable length data benefit from being run multiple times (iterative) to find optimal result.

TBD

- TBD

TBD

Running the compression algorithm multiple times will **utilize more energy**, **however** this is not the current bottleneck projected for the mission, as such **data downlink budget takes priority**

Agenda - Content

- Esda
- Sadas
- Sadasdf
- Sa
- As
- sa

Status Bus

Summary of subsystem (EPS..working, first powered, not purchased)

Status Payload

- We worked a lot
- What we learned with results:
  - What we can observe
  - Filtering is not enough → Star tracking
  - Lara does the slides

Status Payload

Challenge - Mission Objective

- FIRST: We are not aligned in the objective mission
- Vincenzo presents

Challenge - Time

THEN: Time challenge

Jaspar does the slide

Vincenzo presents

Solution Propose

- Voluntary students: Reward for them?
- Thesis students: Can we do thesis based on EventSat that do not have scientific output (not Master‘s Thesis) Semester Thesis needs to have a scientific output, practical research course/teamwork don‘t need scientific output?
- Jaspar does the slide