MDPI

Article

Real-Time Compression for Tactile Internet Data Streams [†]

Patrick Seeling [1,*], Martin Reisslein [2] and Frank H. P. Fitzek [3]

1 Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48859, USA
2 School of Electrical, Computer, and Energy Engineering, Arizona State University,
   Tempe, AZ 85287-5706, USA; reisslein@asu.edu
3 Centre for Tactile Internet with Human-in-the-Loop, Technische Universität Dresden,
   01062 Dresden, Germany; frank.fitzek@tu-dresden.de
* Correspondence: patrick.seeling@cmich.edu
† This paper is an extended version of our paper published in Proceedings of the 2020 IEEE Global
   Communications (Globecom) Conference, Taipei, Taiwan, 7–11 December 2020.

Abstract: The Tactile Internet will require ultra-low latencies for combining machines and humans in systems where humans are in the control loop. Real-time and perceptual coding in these systems commonly require content-specific approaches. We present a generic approach based on deliberately reduced number accuracy and evaluate the trade-off between savings achieved and errors introduced with real-world data for kinesthetic movement and tele-surgery. Our combination of bitplane-level accuracy adaptability with perceptual threshold-based limits allows for great flexibility in broad application scenarios. Combining the attainable savings with the relatively small introduced errors enables the optimal selection of a working point for the method in actual implementations.

Keywords: tactile internet; real-time compression

1. Introduction

Mobile communication network bandwidth has become abundant in recent years, with new use cases emerging predominantly in the Internet of Everything realm. Although initially focused on the Internet of Things, which was dominantly driven by low frequency, low bandwidth sensing and monitoring, use cases have changed dramatically over the past decade. The increasing use of sensing, processing, and actuating control loops across sectors from agriculture over industry to leisure application scenarios has become ever increasing [1–6]. More recently, this has resulted in the consideration of these control loops with human operators in the loop, coined as the Tactile Internet (TI) [7,8]. As a result, future communication networks will experience significant challenges to cope with ever-increasing latency demands in general, while the current focus is on industrial scenarios [9–13].

In general, the human–machine cooperation that is at the nexus of several disciplines is gathering an increasing interest. Haptic considerations in the ultra-low-latency context were described in, e.g., [14]. In [15], the authors focus on compression and encoding for tactile (i.e., haptic as well as kinesthetic) information. Some of the original works were described in [16], where the authors provide first forays into perceptually optimized encoding of tactile information. Commonly, perceptual coding approaches are based on the concerns of the Quality of Experience (QoE) or Quality of Interaction (QoI) by exploiting human sensory processing particularities. One such approach is the reliance on *just noticeable differences (JND)*, coined in [17]. The JND principle has been popularly employed in mapping of objectively measured metrics from the Quality of Service (QoS) to derive the human perception-focused QoE [18]. The relationship between QoS and QoE is typically described as a logarithmic one, such as in [19]. The Weber–Fechner law has evolved to a metric allowing a comparison of the impacts of QoS on different types of media, see, e.g., [20].

As human operators are increasingly becoming part of control loops across application scenarios, an increase in bi-directional communications will likely follow [21], which constitutes the Tactile Internet with Human-in-the-Loop (HITL). With different perceptual coding parameters, existing studies in [22,23] exhibit a broader range of packet inter-arrival times than estimated [24]. These could result in significant concerns for these types of application scenarios, which are tightly delay bound around the 1 ms boundary.

The need to deliver round-trip times of 1 ms for sensing, processing, and resultant actuating (inclusive of all networking delays) will put significant strain on existing networks [25]. New approaches are required to reign in latencies, e.g., through processing in the network near the sensing and actuating locations [26–28] in conjunction with accelerated processing techniques [29–33]. Current projections for Tactile Internet scenarios [33] allocate the 1 ms delay budget as follows for a generic sensor-to-actuator control loop: 0.1 ms for the embedded sensing processing, 0.1 ms for the wireless and wired transmission and reception processing to the edge computing location, with an additional 0.125 ms budget for propagation delay over a distance up to 25 km. Then, 0.35 ms for the computing processing at the edge computing site, followed by 0.125 ms for propagation, 0.1 ms for the wireless and wired transmission and reception processing, and 0.1 ms for the embedded actuator processing. Thus, the typical delay budget for sensor processing is 0.1 ms, requiring low-complexity sensor data processing—a main design imperative for BINBLISS. Also, specific strategies for long-distance robot control loops are needed, for example, the model-mediated virtual reality remote robot control systems, e.g., the so-called digital twin technology [34–38], can overcome the latency challenges for sensor-to-actuator loops that span long physical distances for automated or non-automated control loops. Although initially, concerns for single data streams can be alleviated by focusing on the relative infinite bandwidth (several bytes of sensed data vs. several gigabytes of network bandwidth), the capacity of networks, especially in dense deployments could become a future bottleneck.

Data compression is generally desirable in sensor networks to alleviate such a bottleneck [39]. The density can stem from dense population or machine deployments, such as in metropolitan areas or in factories, but also be the result of an increase of the sensors/actuators that are placed, e.g., on individual operators (humans or robots) [40]. A common example is, e.g., a tactile glove [41] or similar sensing garments [42,43]. Although the actual underlying sensing/actuating might not be readily modifiable to result in a reduction of latencies, the sending of multiple of those readings could yield compression benefits, if it were possible to perform compression in real time to avoid significant negative impacts on the overall system latency.

This article describes our BINBLISS algorithm that can be executed at endpoints almost without delays by compressing the actually transmitted numbers. We note that a preliminary abridged version of the BINBLISS algorithm has been presented in the conference paper [44]. This journal article substantially extends the prior conference paper presentation with (i) an expanded description of the BINBLISS algorithm in Section 2 with a novel algorithm for incorporating perceptual change into BINBLISS (see Algorithm 1 in Section 2.2), (ii) an expanded compression evaluation with TU Munich kinesthetic traces and JIGSAW traces of robotic surgery in Section 3 (whereas [44] gave only a few compression results for the TUM traces), and (iii) a novel combined performance metric definition with extensive numerical evaluation in Section 4.

We note that this present study focuses on the compression of the tactile Internet data stream; and may be combined with compression of other data stream components, e.g., video streams [45–47], and compression of the Internet protocol headers [48]. We furthermore note that in this article, we focus on the theoretical impacts based on the previously described data sets, leaving human subject experimentation for future studies. This article provides the full conceptual description of our approach, including the pseudo-code algorithm, to facilitate these future studies.

The remainder of this manuscript is structured as follows. Section 2.1 describes the overall compression algorithm while Section 2.2 describes the overall evaluation configuration and methods in greater detail, followed by data set descriptions in Sections 2.3 and 2.4 for kinesthetic codec development and robotic surgery training, respectively. Section 3 provides the performance evaluation results for compression savings and losses before we describe a combined metric in Section 4. We conclude with a discussion and outlook on future works in Section 5.

2. Materials and Methods

We commence this section with a brief description of the Binary Indicated Numbers with Bit-Level Integrated Scalability Support (BINBLISS) real-time compression algorithm. We follow with details of the overall configuration and kinesthetic and surgery data sets we employ throughout.

2.1. Bit-Level Integrated Scalability Support (BINBLISS)

Consider an initially lossless compression approach that employs delta-coding on a per-value basis. A fixed number of values (such as originating from, e.g., a frequently repeated sensor reading) needs to be communicated in well-determined messages. Individual values are always in the same position within a message, which enables a fundamental indicator to signal whether a changed value is contained in a specific message. This binary indicator is prepended to the original message and allows the skipping of those values that are not contained in the message as they are unchanged with respect to the prior message. This approach is lossless, but introduces a small overhead for the new binary header. A single start delay is incurred in form of the very first message composition (in form of, e.g., one sensor reading), after which the delays are only dependent on the processing speed. We provide a visual representation of the packaging process for multiple floating-point values into BINBLISS messages in Figure 1. Other values, such as communicated binary flags, can readily be incorporated as well through simple reordering of a message's content.

For example, four floating-point values are sent with several binary flags as data in Figure 1. They are re-ordered, and changes evaluated. Here, it is assumed that only the first and third values have changed and are indicated in the new binary header that is followed by the re-allocated binary values.
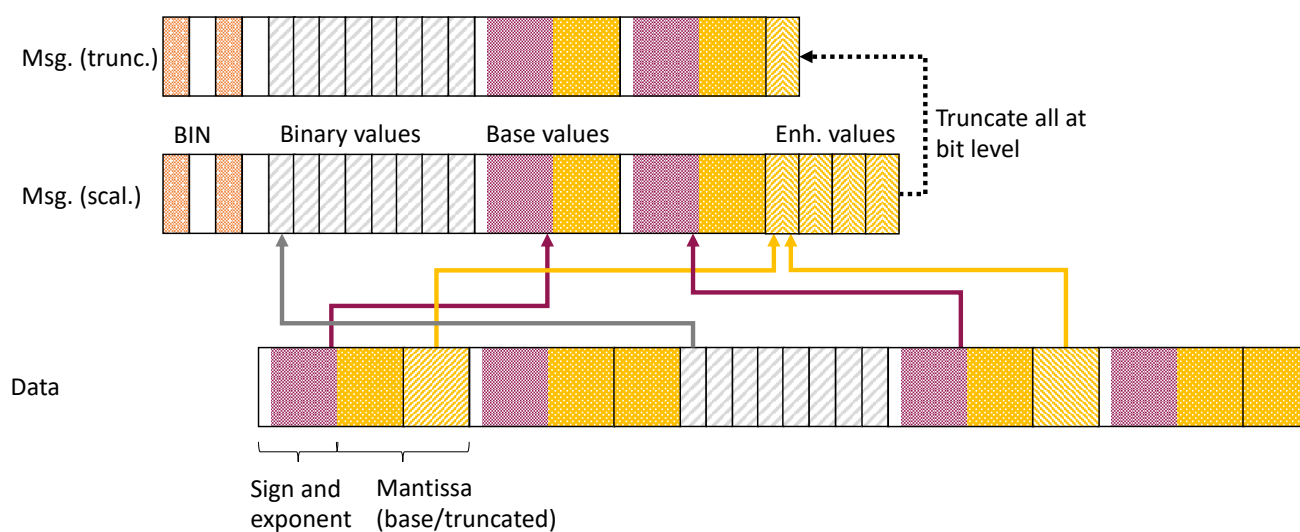

Figure 1. BINBLISS low-latency compression concept: Floating-point values are truncated in granularity, delta-coded with a threshold, and subsequently repackaged by adding binary indicators for the non-changed and modified values. Truncated bits can be added at the message end to enable straight-forward bit-level scalability.

Additionally, the floating-point values themselves can be modified as well, leading to additional savings through lossy, but speedy compression. BINBLISS enables arbitrary truncation of the number of bits used for the significand or digit part of floating-point values, as in the common IEEE 754 standard [49]. This bit reduction reduces accuracy, but yields additional compression data savings. The removed bits can be appended to the message and be dynamically used or dropped by intermediate or receiving nodes, resulting in a bitplane-level scalability. Figure 1 illustrates this principle by moving the base values (exponent part and truncated significand part) into the main message and the removed bits are appended after all contained values in round-robin fashion, enabling the bitplane-level scalability, which subsequently can be truncated as needed.

Extending the initial consideration for changed values to include perceptual coding approaches, we additionally enable the option of including only values that have changed from the last communicated value by a certain threshold limit. This approach initially performs any value operations, such as truncation of bits to a certain number of significant bits, before comparing the resulting value to the last communicated one. The result is an effective perceptual coding approach, similar to those discussed in, e.g., [15,16,23].

2.2. Configuration

We now describe the overall configuration and approach before we explain individual employed data sets separately in greater detail in the following subsections. Commonly, TI data streams exhibit high frequency readings of individually sensed values. One example could be the torque determined in a specific direction at a collaborative robot's axis to enable safe collaboration. Let the i-th of such general values be denoted as x_i. More specifically, i denotes the place of x_i in a time-dependent series of values as they are captured. Following the outline in Section 2.1, we retain the general view of $i = 0, \ldots, I - 1$, i.e., we generically consider I ordered data points (noting that the initial pre-sample would be x_{-1}). Employing this ordered list of values, we determine the average and the standard deviation, respectively denoted as $\mu(x)$ and $\sigma(x)$.

An evaluation of modified values with respect to their original counterparts is performed between the original x_i and the modified version $\hat{x}_i$. One of these modifications is the described removal of bits from x_i to the internal value $\dot{x}_i$. The second change represents the skip of values, i.e., we maintain the previous changed value $\dot{x}_j$ if the change since the last modified value is below the pre-determined (perceptual) threshold level l as in Algorithm 1. (We note that in our evaluations, we always use the first value encountered, i.e., x_0, effectively assuming that l was exceeded.)

Algorithm 1: Perceptual change incorporation into BINBLISS.

$\hat{x}_0 = \dot{x}_0$;
j = 0;
for $i = 1; i < I; i{+}{+}$ **do**
  if $\frac{\dot{x}_i - \dot{x}_j}{\dot{x}_j} > l$ **then**
   j = i;
  end
  $\hat{x}_i = \dot{x}_j$;
end

Subsequently, we determine the Mean Squared Error (MSE) as

$$\text{MSE}(x, \hat{x}) = \frac{1}{I} \sum_{i=0}^{I-1} (x_i - \hat{x}_i)^2. \tag{1}$$

Moreover, especially for small values, it might be equally interesting to derive an overview absolute deviation metric. Here, we consider the Median Absolute Error (MedAE), which is less prone to outliers as an additional benefit, as

$$\text{MedAE}(x, \hat{x}) = \text{median}(\mid x_0 - \hat{x}_0 \mid, \ldots, \mid x_{I-1} - \hat{x}_{I-1} \mid). \tag{2}$$

Oftentimes it is also desired to evaluate relative metrics that can be observed to compare different application scenarios. Let ρ denote the ratio between values as $\rho = \hat{x}/x$. We subsequently consider the relative average and standard deviation as $\mu(\rho)$ and $\sigma(\rho)$, respectively. Other metrics can readily be obtained in a similar fashion.

2.3. TU Munich Kinesthetic Traces

We employ the reference traces for codec development from the IEEE P1918.1 Tactile Internet Working Group's Haptic Codec Task Group. Specifically, these traces were generated as described in greater detail in [50] using a well-defined reference setup. The resultant data set provided by the Technische Universität München (TU Munich, TUM) contains three static and three dynamic interactions, each with 1 ms spaced values that are for force, velocity, and position in each of the three dimensions. We consider the motion vector contents as inputs and the resulting positions as outputs to determine the overall system performance. Each data set covers 25 s of interaction. In our evaluation, we consider the motion vector trace contents x, y, z individually (i.e., as x_i inputs as in Section 2.2). Furthermore, we consider the combined origin vector length as input, i.e., we employ $v = \sqrt{x^2 + y^2 + z^2}$. The individual raw values in the data set are interpreted in our performance evaluation as 32-bit float values. In our performance evaluation, we provide the aggregated outcomes from the individual traces combined.

2.4. JIGSAWS Traces

The JIGSAWS data traces are based on data captures from the DaVinci robotic surgery application, described in greater detail in [51]. Specifically, the data set is derived from various training tasks performed and captured at a fixed frequency, based on the manufacturers tool chain. The data set's individual human subject driven evaluations each consist of 76 kinematic floating-point variables for each data point. These values represent the states and current kinematic properties of the two human operator input interfaces and the surgery robot's arms as well as the tools attached to them. Each tooltip or effector is characterized further via 3-d coordinates, velocities, and angles. The DaVinci tele-surgery robot features Maxon motors, noting that the highest resolution reported for the company's encoders is 32,768 counts per turn. (However, it is debatable whether this precision can be achieved jointly with the motor actuation under load, even with micro-stepping.) Raw values in the data set are again interpreted in our performance evaluation of BINBLISS as 32-bit float values.

Incorporating the data set to represent the changes that are caused by BINBLISS between human operator and tooltip/effector requires some additional modifications as follows. Let $x_i^{M1}, y_i^{M1}, z_i^{M1}$ denote the coordinates of the first human operator manual gripper in the JIGSAWS traces, respectively. Furthermore, let $x_i^{P1}(x_i^{M1}), y_i^{P1}(y_i^{M1}), z_i^{P1}(z_i^{M1})$ denote the patient-side robot tool tip positions that correspond to the first human operator gripper's actuation. (We note that we evaluate only the first tool set's data throughout, as the second set would have similar characteristics.) Dropping the index i for readability, we now determine the impact of our BINBLISS approach as follows. We initially determine the factors between the two individual x coordinate pairs as

$$x^{M1,P1} = \frac{x^{M1}}{x^{P1}(x_i^{M1})}. \tag{3}$$

This captures the dynamic fluctuations between manual gripper and effector which are present due to internal DaVinci processing. Let $\hat{x}^{M1}$ denote the modified (compressed)

value of x^{M1} as in Section 2.2. We subsequently determine the result of compressing the operator's input value for the robot arm's output as

$$\hat{x}^{P1} = \frac{\hat{x}^{M1}}{x^{M1,P1}} = \frac{\hat{x}^{M1} \cdot x^{P1}(x_i^{M1})}{x^{M1}}. \tag{4}$$

Other dimensions as well as the origin vector's length are determined in a similar fashion. Subsequently, all performance evaluations are conducted on the patient side only, as we focus on the resulting impacts of BINBLISS, i.e., we use $x = x^{P1}(x^{M1})$ and $\hat{x} = \hat{x}^{P1}$. We again note that we provide aggregate statistics for the performance evaluation, but group these by activities, namely *Knot* tying, *Needle* passing, and *Suture* creation.

3. Compression Results

In this section, we discuss the compression results with a focus on the BINBLISS outcomes for introduced errors with different perceptual and non-perceptual coding configurations as well as for the attainable savings.

3.1. TU Munich Kinesthetic Traces

We initially investigate the basic effects of the binary compression scheme alone, i.e., we employ a forced repeat on every individual value, independent of content changes. Figure 2 illustrates the resulting medians and standard deviations of outcomes for the TUM data set for absolute deltas, relative deltas, and the MSE, respectively.
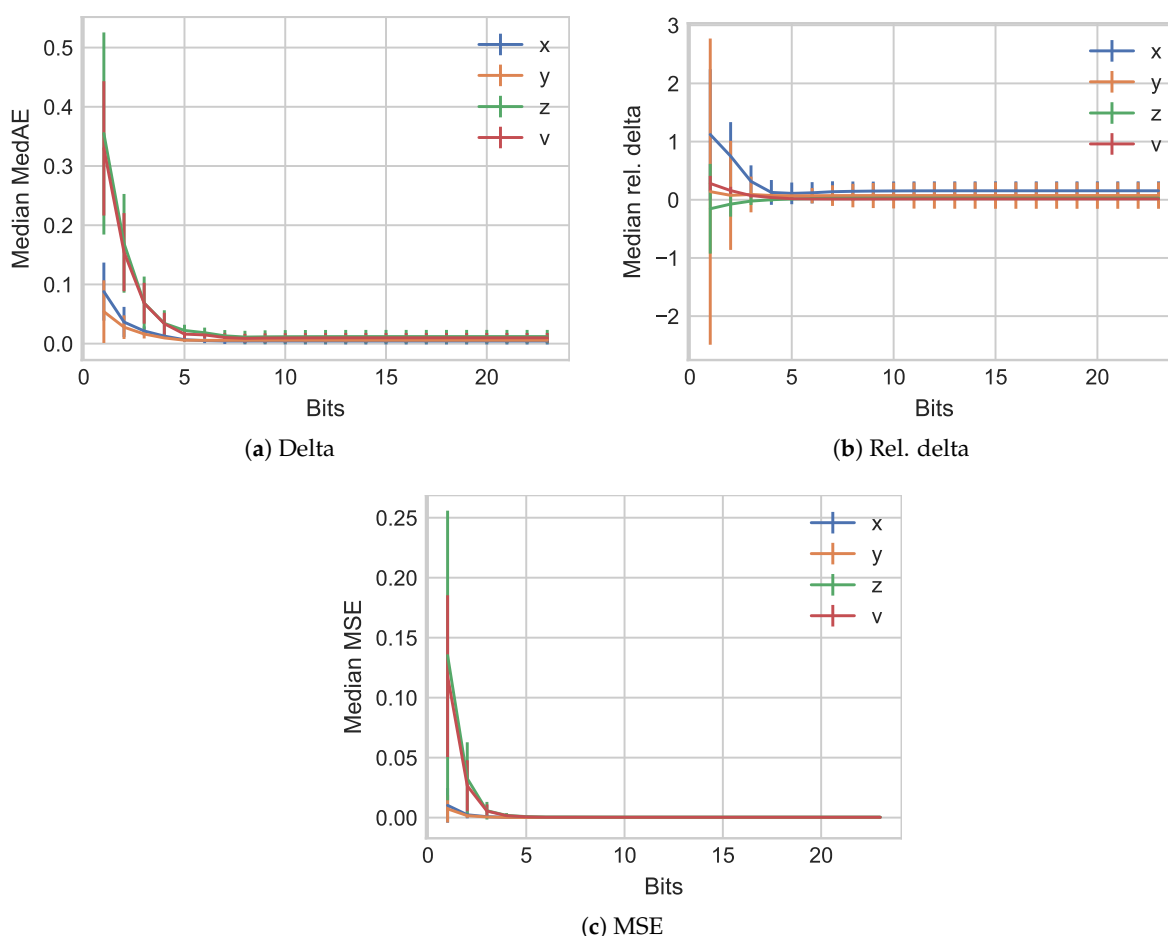

(**a**) Delta

(**b**) Rel. delta

(**c**) MSE

Figure 2. Median results for the *TUM Kinesthetic* data set, evaluated with forced repeats every measurement and no limit (effect of bit reduction only).

We observe a significant drop in the delta as the number of mantissa/significand bits increases, with more than 9 bits not yielding any additional improvements. In turn, we can derive that the overall granularity of measurements could likely be well-captured with 16-bit floats. The exponential decrease of errors, further separated into the three dimensions x, y, z as well as the combined origin vector length v. We note that the dominant deviation is for the z-axis, which is almost the same as the resulting v. Once the overall threshold of about 4–5 bits is passed, the changes become minor. However, one must put these changes into perspective of the underlying base values. The relative delta showcases that initially, very high deviations occur, which rapidly fade. The relative displacement of v as result of the compression indicates that the combined effect of the compression might be less noticeable than the absolute values indicate, especially considering for more than 3 bits. A view of the MSE results additionally showcases the impacts we noticed when considering the absolute delta values, though more pronounced. Again, the median MSE drops sharply after 3 bits.

Next, we shift our evaluation to enable a foundational change-based skipping of values. Should either x, y, or z value not be changed, it will be dropped and indicated as described in Section 2.1. The median results for different relative limits l are illustrated in Figure 3.

(**a**) Rel. delta (limit 0.0)

(**b**) Rel. delta (limit 0.05)

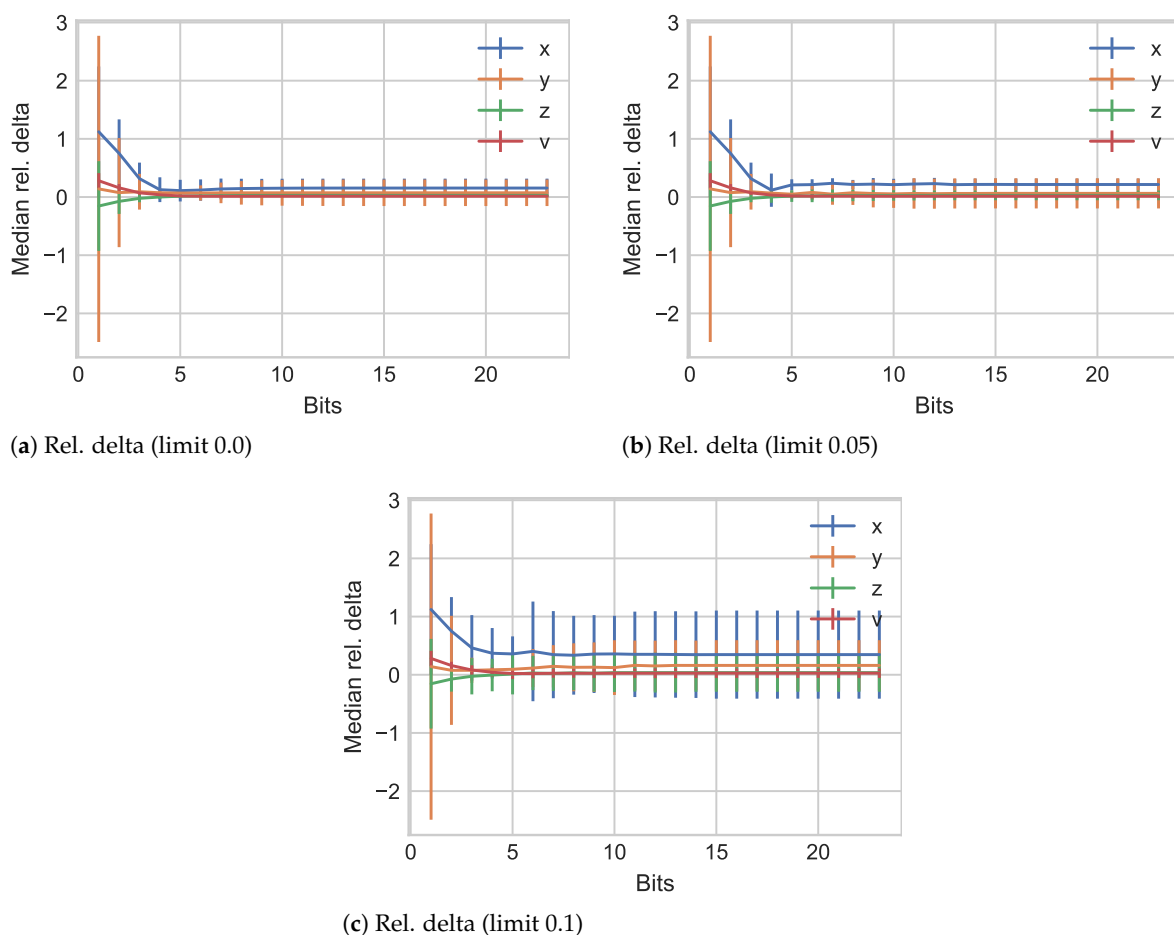(**c**) Rel. delta (limit 0.1)

Figure 3. Median relative delta results for the *TUM Kinesthetic* data set with different limits as change thresholds for perceptual coding.

We note that the relative limit of $l = 0.0$ represents the sending of any changed value, while the remaining two examples can be regarded as perceptual coding approaches that limit required changes to 5% and 10%, respectively. For the regular delta-coding paired with the bit-level compression of BINBLISS in Figure 3a, we observe a similar behavior than for the forced updates, with rapidly decreasing impacts. However, some small changes

remain due to employed number transformations for statistics overall (we note that we add an infinitesimally small number to avoid divisions by zero). This behavior is to be expected, as small changes are captured in the same fashion that all changes are captured when sending every value independent of changes. Increasing the threshold to 5%, we observe that predominantly the x dimension values result is shifted and exhibits a higher relative error level. A similar behavior is visible when considering a further shift of l to 10%. There is, however, little impact on the median combined metric v for both cases. Only in the last case does the y value exhibit a visual increase in the median relative error.

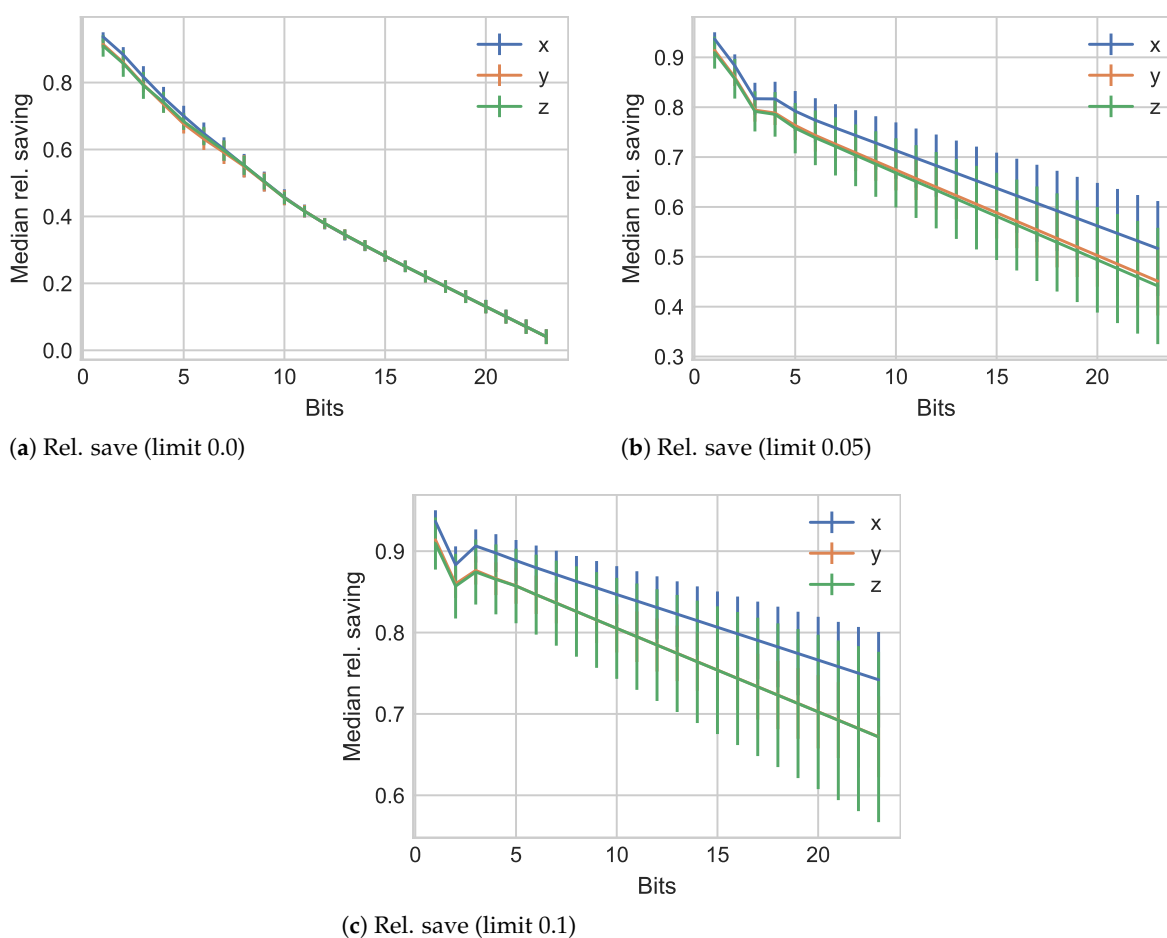We now shift the view from the introduced errors to the attainable savings for these three cases in Figure 4.

(**a**) Rel. save (limit 0.0)

(**b**) Rel. save (limit 0.05)

(**c**) Rel. save (limit 0.1)

Figure 4. Relative savings results for the *TUM Kinesthetic* dataset with different limits as change thresholds for perceptual coding.

We initially observe significant changes for all three values around 85% as maximum when using only one bit for the mantissa. Although there is a minor variation in between the three axes' values, overall, they remain close. Increasing the number of bits almost linearly decreases the attained savings with values beyond 9 bits identical for all three dimensions. As the change limit is increased to 5%, we notice a significant separation of the x and y, z value results. Specifically, we observe that the relative savings for y, z values decrease faster as more bits are added. A slight plateau forms between 3 and 4 mantissa bits. We also note that the highest overall relative savings level for x values is now close to 95%, while y, z values trail by a few percent. One of the reasons here could be the actual source of the data, with less dynamic data in y, z directions and more on the x axis. Increasing the change limit threshold for perceptual coding considerations further, we notice an increase of the prior observations. Noteworthy here is the slight dip at 2 mantissa bits, followed by a rise and subsequent linear trend as before. The savings shifts

and plateaus can be explained by the different captures of difference levels in the available bit levels.

3.2. JIGSAW

The JIGSAWS data set evaluation considers the three separate user interactions of *Knot* tying, *Needle* passing, and *Suture* creation. Figure 5 illustrates the results for these three different application scenarios. We again begin with a view on the compression without delta-coding first.
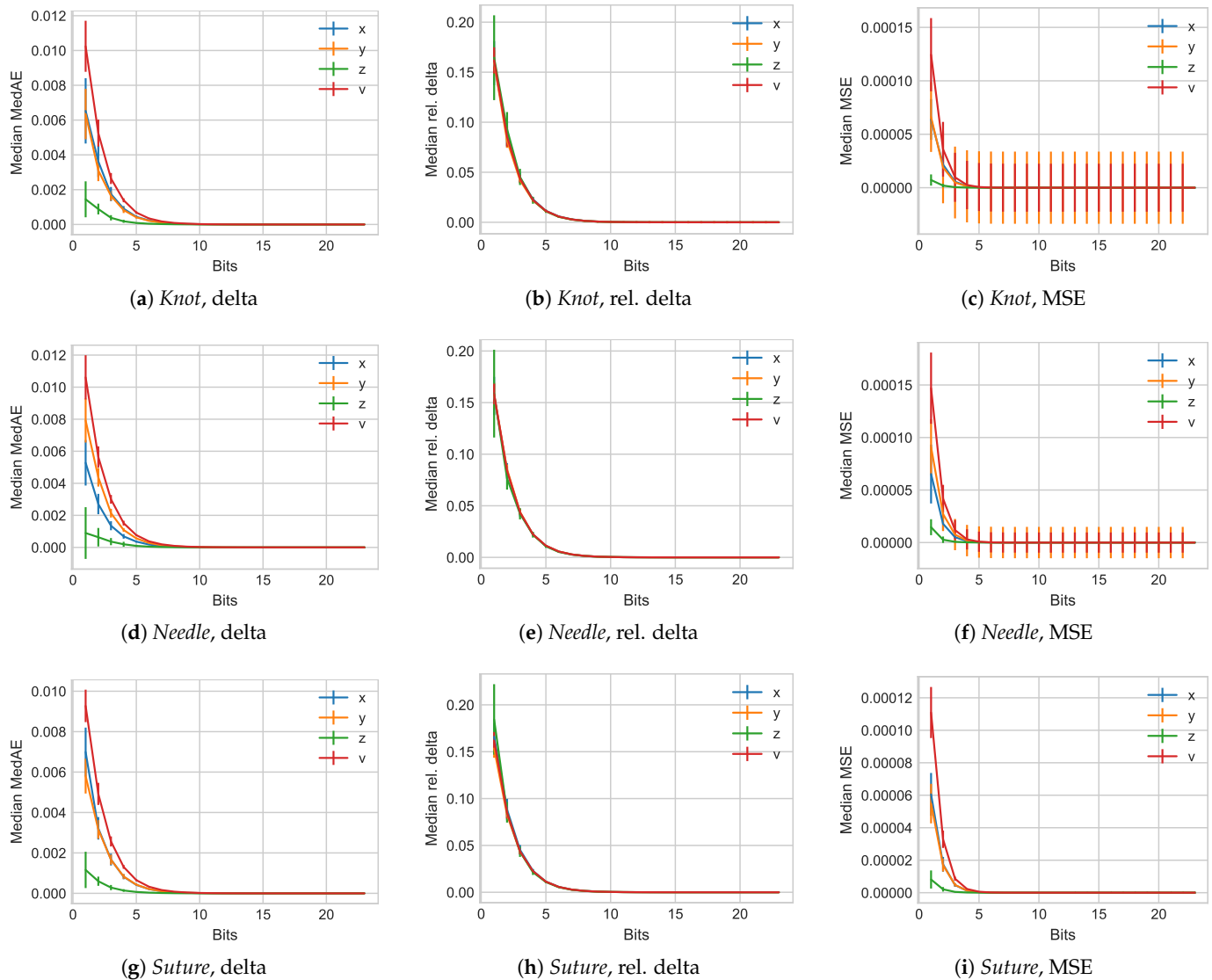
Figure 5. Median error results for all subject experiments in the *Knot*, *Needle*, and *Suture* groups from the *JIGSAW* data set.

We initially observe that all three exhibit trends similar to those we observe for the *TUM* data set as well, with slight variations. Here, for all three data sets, the median vector displacement length is highest, with commonly the *z* axis value being lowest. One of the reasons here might well be that the overall movements are higher in their dynamics, but the actual displacement error on the robot's patient side is lower. Moving on to the relative delta, we notice an upper limit of about 20% for all, which exponential drops similar to prior evaluations. For the *JIGSAWS* data set, we notice a significantly tighter range of all 4 evaluated values than we observe for the *TUM* data set. As before, the MSE reflects the overall observations from the median actual delta.

Next, we include the delta-coding approach of the BINBLISS method, again with 0%, 5%, and 10% relative change between values as threshold limits. Figure 6 illustrates the median result outcomes for the relative errors.
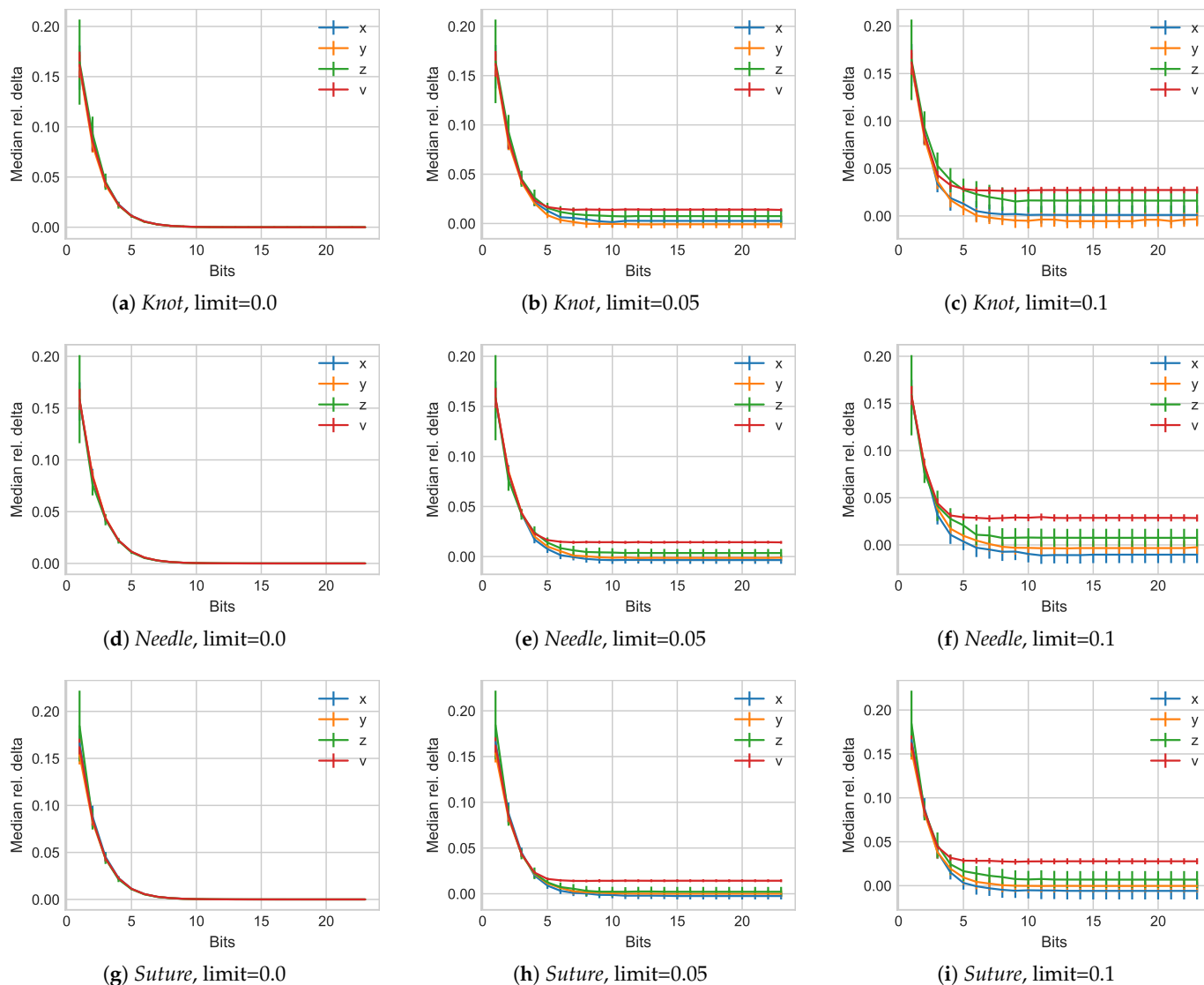

(**a**) *Knot*, limit=0.0

(**b**) *Knot*, limit=0.05

(**c**) *Knot*, limit=0.1

(**d**) *Needle*, limit=0.0

(**e**) *Needle*, limit=0.05

(**f**) *Needle*, limit=0.1

(**g**) *Suture*, limit=0.0

(**h**) *Suture*, limit=0.05

(**i**) *Suture*, limit=0.1

Figure 6. Relative delta results for all subject experiments in the *Knot*, *Needle*, and *Suture* groups from the *JIGSAW* data set.

We initially observe the same behavior as before with a limit of $l = 0$ resulting in almost the same outcomes that one would obtain if delta-coding were disabled. As a limit of 5% changes enables perceptual coding, we observe a separation of the four different values, with v exhibiting the highest relative delta overall. Although for very low numbers of bits employed all exhibit median relative deltas around 17.5%, this quickly diminishes to about 2.5% when 5 bits are employed. Overall, this also seems to provide an overall threshold for the displacement vector to remain stable while the remaining x, y, z components continue to drop slightly lower until reaching 9 bits. As before, increasing the perceptual limits further results in additional error increases, albeit at a lower level.

We now contrast these introduced errors with the attained savings in Figure 7.
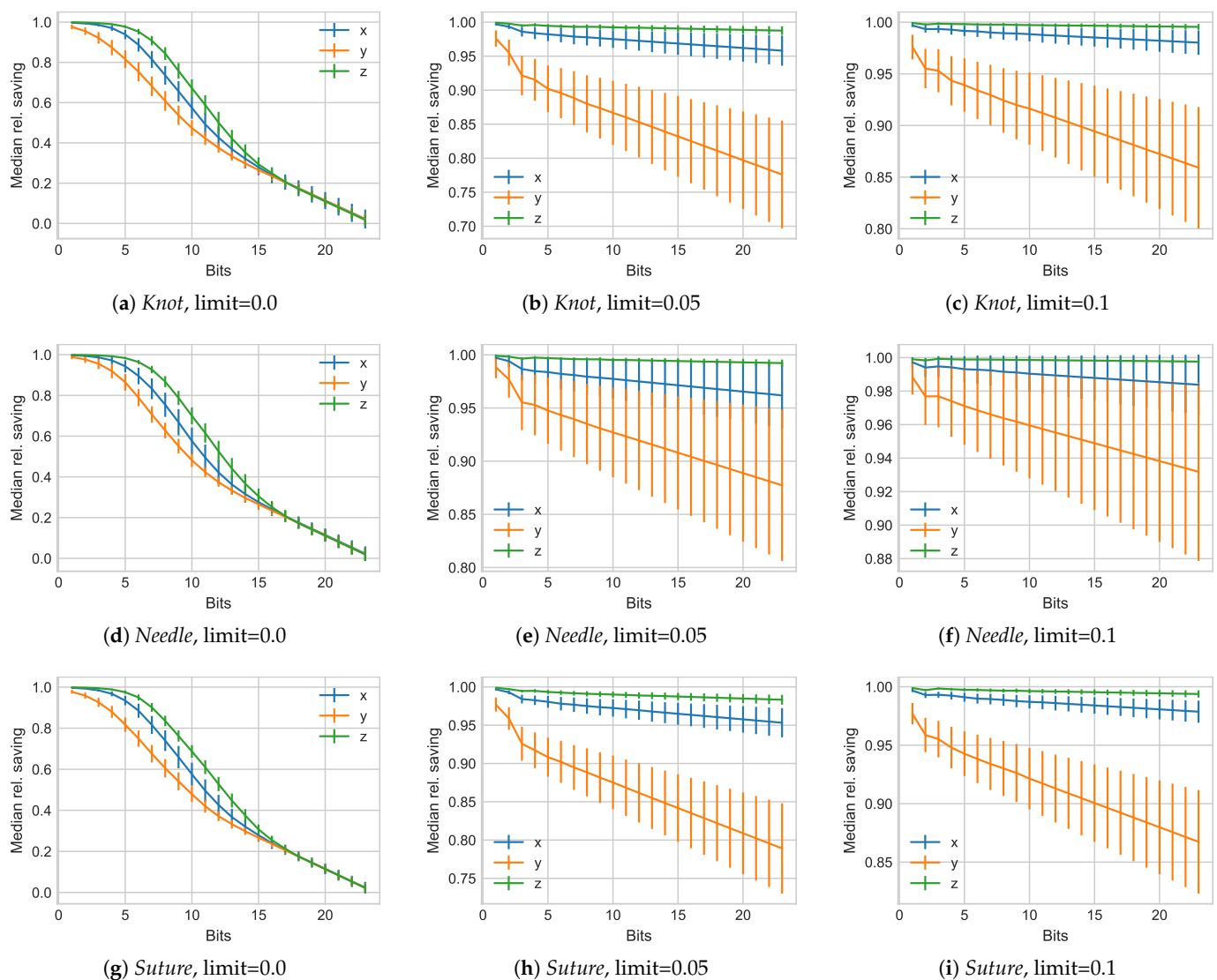
Figure 7. Relative saving results for all subject experiments in the *Knot*, *Needle*, and *Suture* groups from the *JIGSAW* data set.

We initially observe that a limit of $l = 0.0$ results in a hysteresis resembling spread between the x, y, z values, with x providing the highest savings, followed by z and y, respectively. A removal of all but one bit results in almost 100% savings, as the granularity of changes is no longer captured at all. As the numbers of bits increase, the savings slowly diminish, but even at 9 bits, removing subsequent identical values results in more than 50% savings across application scenarios. A particular reason for this can be seen in the large number of values per data set entry, which include longer periods of inactivity. Once perceptual coding is enabled by allowing a limit threshold of 5%, variability, as captured by the standard deviation across the various human subject traces, greatly increases. The overall trend shapes now more closely resemble those we obtained for the *TUM* data set. We additionally observe significant lower savings for the y component than for x, z, which can be attributed to the nature of the source data exhibiting significance lower changes in the x, z dimensions.

4. Combined Performance Metric

In this section, we provide an initial performance metric that combines the relative savings as well as the relative errors introduced to determine an ideal operational point. We provide an overview of the median relative combined errors, savings, and a combined metric as result of the interplay of bits and perceptual change threshold in Figure 8 for

the TUM traces. We provide this overview for the combined error $v_e rr$ and the individual combined savings $x_{save}, y_{save}, z_{save}$. We finally combine the two in a simple metric as

$$V_c = -\alpha * v_{err} + (1 - \alpha) * (x_{save} + y_{save} + z_{save}), \tag{5}$$

which allows the weighing of errors and savings differently. In Figure 8, we employ $\alpha = 0.5$ for a balanced view.

(**a**) Rel. combined errors

(**b**) Rel. combined savings

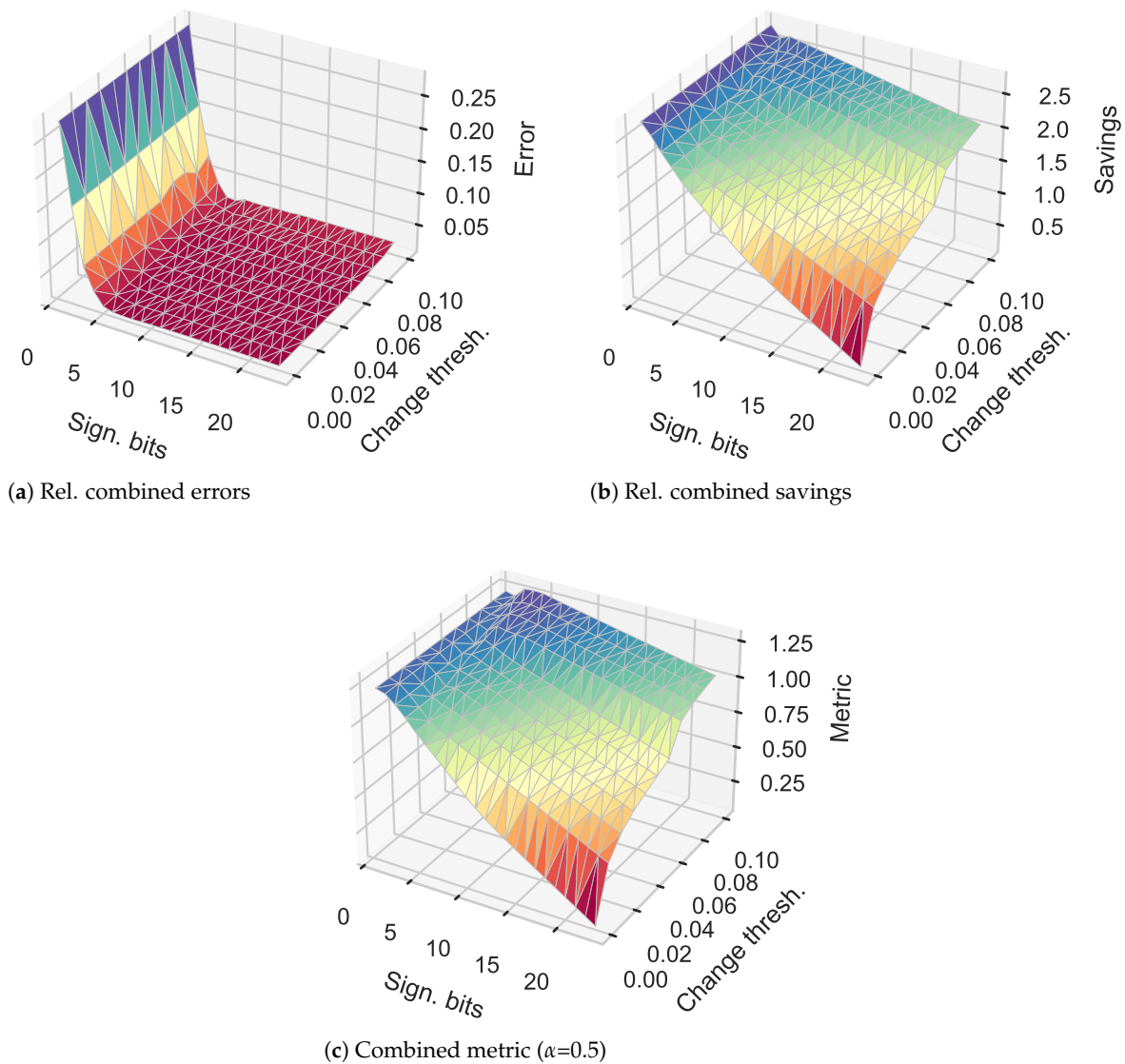(**c**) Combined metric (α=0.5)

Figure 8. Relative errors, savings, and combined metric results for the *TUM Kinesthetic* dataset with different bit limits and change thresholds for perceptual coding.

We initially note that the relative overall error is primarily dominated by the number of bits employed, and only to a smaller degree by the coding threshold. This is in line with the prior evaluations in Section 3.1. Furthermore, we notice a waterfall-type drop-off for the combined savings. Here, the impact of the perceptual coding threshold is not negligible and contributes significantly to the savings when paired with a reduction in bits. Ultimately, this illustrates the combination of savings in bits per message paired with message content reductions. We notice that the combined result resembles more closely the savings rather than the combined errors, with a maximum V_c at 4 bits and a perceptual delta limit of 0.1 (which represents the upper limit employed).

We perform the evaluation for the *JIGSAWS* data set as well, with results illustrated in Figure 9.
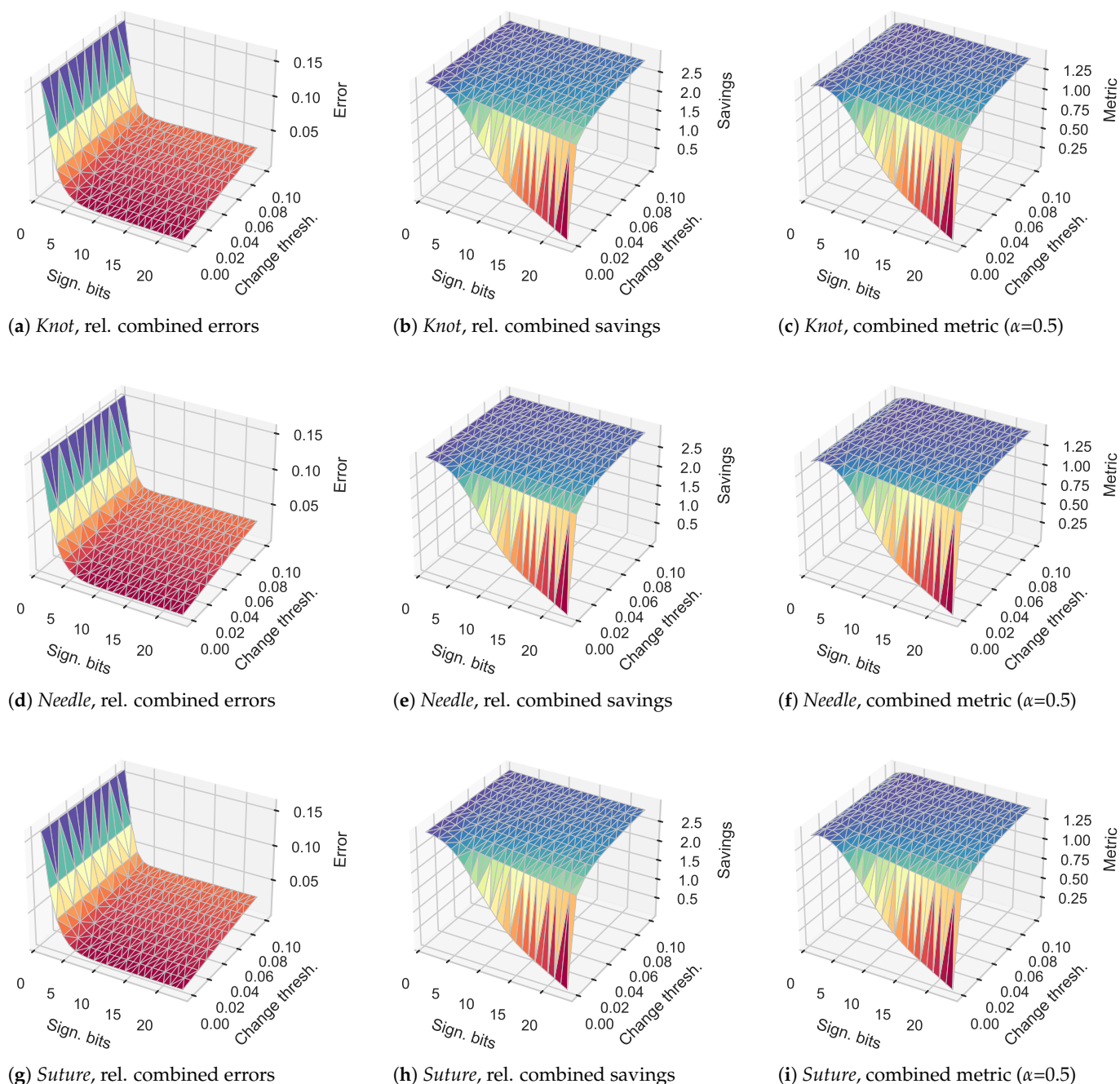

(**a**) *Knot*, rel. combined errors   (**b**) *Knot*, rel. combined savings   (**c**) *Knot*, combined metric (α=0.5)

(**d**) *Needle*, rel. combined errors   (**e**) *Needle*, rel. combined savings   (**f**) *Needle*, combined metric (α=0.5)

(**g**) *Suture*, rel. combined errors   (**h**) *Suture*, rel. combined savings   (**i**) *Suture*, combined metric (α=0.5)

Figure 9. Relative errors, savings, and combined metric results for the three different *JIGSAWS* datasets with different bit limits and change thresholds for perceptual coding.

We observe that overall, the results mimic those we observed for the *TUM* data set, but with a more pronounced drop in the combined savings. Each of these scenarios exhibit their maximum at 4 bits and highest evaluated perceptual delta-coding limit of 10%.

The presented results are based on our generic BINBLISS approach and its evaluation without impacts on task performance or human perception. Determining these impacts requires an initial evaluation with human subjects and Institutional Review Board oversight, which is out of the scope of the present study. The BINBLISS approach specified in this article is highly flexible to facilitate a wide range of future research studies and

implementation experiments. In particular, the presented BINBLISS design allows implementations to readily adjust the combination of different compression factors and impacts on granularity in specific evaluation scenarios with human subjects.

5. Conclusions

This article examined the trade-off between time-sensitive compression based on foundational floating-point number accuracy reduction and the resulting errors for real-life scenarios common to the Tactile Internet: kinesthetic movement and tele-surgery. A significant reduction in data can be obtained through a reduction in the number of bits, with only minor impacts on the resulting error, a benefit for this real-time approach. Similarly, by an introduced delta threshold, further reductions can be achieved that take perceptual considerations into accounts and result in only minor increases in the introduced errors. Combining the two individual outcomes, we derive a main view on the saving/error trade-off, which can be adjusted as needed to determine favorable operational points.

The present study can serve as basis for several important future research directions. One direction is to refine the presented approach through detailed timing measurements in testbed implementations to achieve further data reductions under real-time constraints. A first prototypical (non-optimized) implementation employing Python on an i5 CPU incurred around 0.3 ms of added sending delay on general-purpose computing hardware. We believe that implementation optimizations can substantially reduce this delay. Additional optimizations, including dynamic configuration options for sender and receiver, provide interesting avenues for future research. Based on the BINBLISS approach specified in this article, future research and development efforts should create testbeds to enable more fine-grained evaluation measurements. Such testbeds should also enable human subject experimentation to determine the impacts of our approach on actual task completion performance as well as perceptual impacts, e.g., through experience sampling [52].

Another direction is to explore combinations of the source compression of the tactile Internet data stream with channel coding to make the tactile data stream transmissions robust against network impairments, e.g., wireless network errors. One potential approach is to explore flexible low-delay network coding that may interleave the tactile data stream with related data streams [53–56]. A broad future research direction is to explore how tactile data stream compression can cooperate with distributed computing, e.g., through multi-access edge computing or fog computing [57–60], of the tactile signals close to the sensing and actuation sites to achieve ultra-short round-trip computing for tactile applications.

References

1. Forge, S.; Vu, K. Forming a 5G Strategy for Developing Countries: A Note for Policy Makers. *Telecommun. Policy* **2020**, *44*, 101975.1–101975.24. [CrossRef]
2. Hoeschele, T.; Dietzel, C.; Kopp, D.; Fitzek, F.; Reisslein, M. Importance of Internet Exchange Point (IXP) Infrastructure for 5G: Estimating the Impact of 5G Use Cases. *Telecommun. Policy* **2021**, *45*, 102091.1–102091.18. [CrossRef]
3. Maternia, M.; Ayoubi, S.E.E.; Fallgren, M.; Spapis, P.; Qi, Y.; Martin-Sacristan, D.; Óscar, C.; Fresia, M.; Payaro, M.; Schubert, M.; et al. 5G PPP use cases and performance evaluation models. 2016. Available online: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1.0.pdf (accessed on 8 March 2021).
4. Navarro-Ortiz, J.; Romero-Diaz, P.; Sendra, S.; Ameigeiras, P.; Ramos-Munoz, J.J.; Lopez-Soler, J.M. A survey on 5G usage scenarios and traffic models. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 905–929. [CrossRef]
5. Rendon Schneir, J.; Ajibulu, A.; Konstantinou, K.; Bradford, J.; Zimmermann, G.; Droste, H.; Canto, R. A Business Case for 5G Mobile Broadband in a Dense Urban Area. *Telecommun. Policy* **2019**, *43*, 101813.1–101813.19. [CrossRef]
6. Verde, S.; Marcon, M.; Milani, S. Advanced Assistive Maintenance Based on Augmented Reality and 5G Networking. *Sensors* **2020**, *20*, 7157. [CrossRef]
7. Fettweis, G.P. The Tactile Internet: Applications and Challenges. *IEEE Veh. Technol. Mag.* **2014**, *9*, 64–70. [CrossRef]
8. Fitzek, F.; Li, S.C.; Speidel, S.; Strufe, T.; Simsek, M.; Reisslein, M. *Tactile Internet with Human-in-the-Loop*; Academic Press: Cambridge, MA, USA, 2021.
9. Bello, L.L.; Steiner, W. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proc. IEEE* **2019**, *107*, 1094–1120. [CrossRef]
10. Finn, N. Introduction to time-sensitive networking. *IEEE Commun. Stand. Mag.* **2018**, *2*, 22–28. [CrossRef]
11. Gavriluţ, V.; Pop, P. Traffic-type Assignment for TSN-based Mixed-criticality Cyber-physical Systems. *ACM Trans. Cyber Phys. Syst.* **2020**, *4*, 1–27. [CrossRef]
12. Nasrallah, A.; Thyagaturu, A.S.; Alharbi, Z.; Wang, C.; Shao, X.; Reisslein, M.; ElBakoury, H. Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 88–145. [CrossRef]
13. Varga, P.; Peto, J.; Franko, A.; Balla, D.; Haja, D.; Janky, F.; Soos, G.; Ficzere, D.; Maliosz, M.; Toka, L. 5G Support for Industrial IoT Applications—Challenges, Solutions, and Research gaps. *Sensors* **2020**, *20*, 828. [CrossRef]
14. Antonakoglou, K.; Xu, X.; Steinbach, E.; Mahmoodi, T.; Dohler, M. Towards haptic communications over the 5G Tactile Internet. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3034–3059. [CrossRef]
15. Steinbach, E.; Strese, M.; Eid, M.; Liu, X.; Bhardwaj, A.; Liu, Q.; Al-Ja'afreh, M.; Mahmoodi, T.; Hassen, R.; El Saddik, A.; et al. Haptic codecs for the Tactile Internet. *Proc. IEEE* **2018**, *107*, 447–470. [CrossRef]
16. Hinterseer, P.; Hirche, S.; Chaudhuri, S.; Steinbach, E. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Trans. Signal Process.* **2008**, *56*, 588–597. [CrossRef]
17. Weber, E.H. *Die Lehre vom Tastsinne und Gemeingefühle auf Versuche gegründet*; Friedrich Vieweg und Sohn: Braunschweig, Germany, 1851.
18. Mohammadi Amin, F.; Rezayati, M.; van de Venn, H.; Karimpour, H.A. A Mixed-Perception Approach for Safe Human–Robot Collaboration in Industrial Automation. *Sensors* **2020**, *20*, 6347. [CrossRef] [PubMed]
19. Fiedler, M.; Hossfeld, T.; Tran-Gia, P. A generic quantitative relationship between quality of experience and quality of service. *IEEE Netw.* **2010**, *24*, 36–41. [CrossRef]
20. Reichl, P.; Tuffin, B.; Schatz, R. Logarithmic laws in service quality perception: Where microeconomics meets psychophysics and quality of experience. *Telecommun. Syst.* **2013**, *52*, 587–600. [CrossRef]
21. Maier, M.; Ebrahimzadeh, A. Towards Immersive Tactile Internet Experiences: Low-Latency FiWi Enhanced Mobile Networks with Edge Intelligence. *J. Opt. Commun. Netw.* **2019**, *11*, B10–B25. [CrossRef]
22. Meli, L.; Pacchierotti, C.; Prattichizzo, D. Experimental evaluation of magnified haptic feedback for robot-assisted needle insertion and palpation. *Int. J. Med Robot. Comput. Surg.* **2017**, *13*, e1809. [CrossRef]
23. Xu, X.; Schuwerk, C.; Cizmeci, B.; Steinbach, E. Energy prediction for teleoperation systems that combine the time domain passivity approach with perceptual deadband-based haptic data reduction. *IEEE Trans. Haptics* **2016**, *9*, 560–573. [CrossRef]
24. Wong, E.; Dias, M.P.I.; Ruan, L. Predictive resource allocation for Tactile Internet capable passive optical LANs. *IEEE/OSA J. Light. Technol.* **2017**, *35*, 2629–2641. [CrossRef]
25. Rico, D.; Merino, P. A Survey of End-to-End Solutions for Reliable Low-Latency Communications in 5G Networks. *IEEE Access* **2020**, *8*, 192808–192834. [CrossRef]
26. Ferrer, A.J.; Marquès, J.M.; Jorba, J. Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing. *ACM Comput. Surv. (CSUR)* **2019**, *51*, 111.1–111.36. [CrossRef]
27. Fitzek, F.; Granelli, F.; Seeling, P. *Computing in Communication Networks*; Academic Press Books: Cambridge, MA, USA, 2020.
28. Mehrabi, M.; You, D.; Latzko, V.; Salah, H.; Reisslein, M.; Fitzek, F.H. Device-Enhanced MEC: Multi-Access Edge Computing (MEC) Aided by End Device Computation and Caching: A Survey. *IEEE Access* **2019**, *7*, 166079–166108. [CrossRef]
29. Junior, J.C.; Torquato, M.F.; Mahmoodi, T.; Dohler, M.; Fernandes, M.A. Reconfigurable Computing Applied to Latency Reduction for the Tactile Internet. *arXiv* **2020**, arXiv:2003.12463.
30. Linguaglossa, L.; Lange, S.; Pontarelli, S.; Rétvári, G.; Rossi, D.; Zinner, T.; Bifulco, R.; Jarschel, M.; Bianchi, G. Survey of performance acceleration techniques for network function virtualization. *Proc. IEEE* **2019**, *107*, 746–764. [CrossRef]

31. Niemiec, G.S.; Batista, L.M.; Schaeffer-Filho, A.E.; Nazar, G.L. A Survey on FPGA Support for the Feasible Execution of Virtualized Network Functions. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 504–525. [CrossRef]

32. Shantharama, P.; Thyagaturu, A.S.; Reisslein, M. Hardware-Accelerated Platforms and Infrastructures for Network Functions: A Survey of Enabling Technologies and Research Studies. *IEEE Access* **2020**, *8*, 132021–132085. [CrossRef]

33. Xiang, Z.; Gabriel, F.; Urbano, E.; Nguyen, G.T.; Reisslein, M.; Fitzek, F.H. Reducing latency in virtual machines: Enabling tactile Internet for human-machine co-working. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1098–1116. [CrossRef]

34. Freund, E.; Rossmann, J. Projective virtual reality: Bridging the gap between virtual reality and robotics. *IEEE Trans. Robot. Autom.* **1999**, *15*, 411–422. [CrossRef]

35. Guivant, J.; Cossell, S.; Whitty, M.; Katupitiya, J. Internet-based operation of autonomous robots: The role of data replication, compression, bandwidth allocation and visualization. *J. Field Robot.* **2012**, *29*, 793–818. [CrossRef]

36. Kelly, A.; Chan, N.; Herman, H.; Huber, D.; Meyers, R.; Rander, P.; Warner, R.; Ziglar, J.; Capstick, E. Real-time photorealistic virtualized reality interface for remote mobile robot control. *Int. J. Robot. Res.* **2011**, *30*, 384–404. [CrossRef]

37. Tsokalo, I.A.; Kuss, D.; Kharabet, I.; Fitzek, F.H.P.; Reisslein, M. Remote Robot Control with Human-in-the-Loop over Long Distances Using Digital Twins. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]

38. Xu, X.; Cizmeci, B.; Schuwerk, C.; Steinbach, E. Model-mediated teleoperation: Toward stable and transparent teleoperation systems. *IEEE Access* **2016**, *4*, 425–449. [CrossRef]

39. Qie, Y.; Hao, C.; Song, P. Wireless Transmission Method for Large Data Based on Hierarchical Compressed Sensing and Sparse Decomposition. *Sensors* **2020**, *20*, 7146. [CrossRef]

40. Pirozzi, S. Tactile Sensors for Robotic Applications. *Sensors* **2020**, *20*, 7009. [CrossRef] [PubMed]

41. Junior, J.C.V.S.; Torquato, M.F.; Noronha, D.H.; Silva, S.N.; Fernandes, M.A.C. Proposal of the Tactile Glove Device. *Sensors* **2019**, *19*, 5029. [CrossRef]

42. Sayem, M.; Sadat, A.; Hon Teay, S.; Shahariar, H.; Fink, P.L.; Albarbar, A. Review on Smart Electro-Clothing Systems (SeCSs). *Sensors* **2020**, *20*, 587.

43. Wilson, S.; Laing, R. Fabrics and Garments as Sensors: A Research Update. *Sensors* **2019**, *19*, 3570. [CrossRef]

44. Seeling, P.; Fitzek, F.H.P. Binary Indicated Numbers with Bit-Level Integrated Scalability Support (BINBLISS). In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020.

45. Seeling, P.; Reisslein, M. Video Traffic Characteristics of Modern Encoding Standards: H.264/AVC with SVC and MVC Extensions and H.265/HEVC. *Sci. World J.* **2014**, *2014*. [CrossRef] [PubMed]

46. Xu, X.; Liu, S. Recent Advances in Video Coding Beyond the HEVC Standard. *APSIPA Trans. Signal Inf. Process.* **2019**, *8*, e18-1–e18-10. [CrossRef]

47. Zhang, T.; Mao, S. An Overview of Emerging Video Coding Standards. *ACM GetMobile: Mob. Comput. Commun.* **2019**, *22*, 13–20. [CrossRef]

48. Sanchez-Gomez, J.; Gallego-Madrid, J.; Sanchez-Iborra, R.; Santa, J.; Skarmeta, A. Impact of SCHC Compression and Fragmentation in LPWAN: A Case Study with LoRaWAN. *Sensors* **2020**, *20*, 280. [CrossRef] [PubMed]

49. Hough, D.G. The IEEE Standard 754: One for the History Books. *Computer* **2019**, *52*, 109–112. [CrossRef]

50. Kirsch, J.; Noll, A.; Strese, M.; Liu, Q.; Steinbach, E. A Low-Cost Acquisition, Display, and Evaluation Setup for Tactile Codec Development. In Proceedings of the 2018 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), Dalian, China, 20–21 September 2018; pp. 1–6. [CrossRef]

51. Gao, Y.; Vedula, S.S.; Reiley, C.E.; Ahmidi, N.; Varadarajan, B.; Lin, H.C.; Tao, L.; Zappella, L.; Bejar, B.; Yuh, D.D.; et al. JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling. In Proceedings of the Fifth Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI), BOSTON, MA, USA, 14–18 September 2014; Volume 3, pp. 1–10.

52. Hektner, J.M.; Schmidt, J.A.; Csikszentmihalyi, M. *Experience Sampling Method: Measuring the Quality of Everyday Life*; SAGE: Thousand Oaks, CA, USA, 2007.

53. Migabo, E.; Djouani, K.; Kurien, A. An Energy-Efficient and Adaptive Channel Coding Approach for Narrowband Internet of Things (NB-IoT) Systems. *Sensors* **2020**, *20*, 3465. [CrossRef]

54. Nguyen, V.; Tasdemir, E.; Nguyen, G.T.; Lucani, D.E.; Fitzek, F.H.; Reisslein, M. DSEP Fulcrum: Dynamic Sparsity and Expansion Packets for Fulcrum Network Coding. *IEEE Access* **2020**, *8*, 78293–78314. [CrossRef]

55. Peralta, G.; Cid-Fuentes, R.G.; Bilbao, J.; Crespo, P.M. Homomorphic Encryption and Network Coding in IoT Architectures: Advantages and Future Challenges. *Electronics* **2019**, *8*, 827. [CrossRef]

56. Tanjung, D.; Byeon, S.; Huh, J.; Kim, D.; Kim, J. Opportunistic and On-Demand Network Coding-Based Solutions for LPWAN Forwarding. *Sensors* **2020**, *20*, 5792. [CrossRef]

57. Chukhno, O.; Chukhno, N.; Araniti, G.; Campolo, C.; Iera, A. Optimal Placement of Social Digital Twins in Edge IoT Networks. *Sensors* **2020**, *20*, 6181. [CrossRef]

58. Gu, X.; Ji, C.; Zhang, G. Energy-Optimal Latency-Constrained Application Offloading in Mobile-Edge Computing. *Sensors* **2020**, *20*, 3064. [CrossRef]

59. Hamdan, S.; Ayyash, M.; Almajali, S. Edge-Computing Architectures for Internet of Things Applications: A Survey. *Sensors* **2020**, *20*, 6441. [CrossRef] [PubMed]
60. Zhao, L. Privacy-Preserving Distributed Analytics in Fog-Enabled IoT Systems. *Sensors* **2020**, *20*, 6153. [CrossRef] [PubMed]