Foveated Video Coding for Real Time Streaming Applications

Masterarbeit

vorgelegt von

Oliver Marcel Wiedemann

an der

Universität Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Fachbereich Informatik und Informationswissenschaft

1. Gutachter: Prof. Dr. Dietmar Saupe

2. Gutachter: Prof. Dr. Bastian Goldlücke

Konstanz, 2020

Abstract

Video streaming under real-time constraints is an increasingly widespread application. Many recent video encoders are unsuitable for this scenario due to theoretical limitations or run time requirements. In this thesis, we present a framework for the perceptual evaluation of foveated video coding schemes. Foveation describes the process of adapting a visual stimulus according to the acuity of the human eye.

In contrast to traditional region-of-interest coding, where certain areas are statically encoded at a higher quality, we utilize feedback from an eye-tracker to spatially steer the bit allocation scheme in real-time. We evaluate the performance of an H.264 based foveated coding scheme in a lab environment by comparing the bitrates at the point of just noticeable distortion (JND). In our trials, we achieve an average bitrate savings of 62.76% in comparison to the unfoveated baseline.

Contents

0. Preamble

Video streaming is ubiquitous and imposes ever-growing demands on content and network providers. Increasing resolutions coupled with bandwidth limitations motivate ongoing research on sophisticated video codecs and compression algorithms. This thesis is concerned with the emerging subclass of real-time video streaming. Besides transmitting and rendering a video sufficiently fast, these applications require to encode it under strict latency constraints.

Video codecs rely on the analysis and exploitation of correlations between pixel color values to achieve high visual quality at low bitrates. Recent approaches combine a multitude of incremental and often marginal improvements to coding techniques. On the one hand, these methods have practical limitations, which can be solved often by fast, hardware-supported implementations of certain mathematical operations.

On the other hand, there are theoretical requirements and limitations that constraint methodology choices. Bi-directional inter-prediction schemes, for example, rely on future keyframes and are thus not applicable in real-time scenarios, if the required temporal buffering violates latency constraints.

Region of interest coding aims to represent the parts of a frame that are more relevant to the viewer at a higher visual quality. However, conventional video applications allow only limited assumptions about the spatial relevance of the content within a given frame. This constraints potential improvements through ROI coding. The limitations stem primarily from not knowing what region an observer will be interested in. Content-based algorithmic predictions of ROIs are speculative and often inaccurate.

We investigate possible improvements for video coding by incorporating gaze information in the encoding process. We expect this approach to be more accessible in the near future due to the prevalence of large high-resolution screens in combination with the advent of eye-tracking devices in consumer hardware. Directly measuring an observers gaze allows making strong assumptions about the active region of interest in each frame. It enables us to devise a spatial coding scheme that gradually decreases the quality as a function of the distance to the current fixation point.

The approach of adapting a visual medium according to the acuity of the observer's eyes is called foveation. Possible application scenarios include not only traditional video telephony and live streaming, but predominantly and novel technologies that could be bolstered by our approach, e.g., human assistance in steering semi-autonomous vehicles, medical or industrial robots, drones and streaming of cloud rendered video games.

We present a modular software to assess of foveated video coding schemes, including a reference implementation based on `x264`. We evaluate our approach's performance in a lab study and quantify the bitrate gains at the point of just noticeable distortion. Our experiments show that our foveated codec achieves an average of 62.76% bitrate savings compared to the unfoveated baseline.

0.1. Structure of this Thesis

The range of appreciable and related topics and contributions is rather broad. This thesis is compartmentalized into the following Sections with the intent of logically guiding the reader through the increasingly specific subjects.

- Section 1 provides a review of fundamental aspects of the human visual system, which forms the basis for any discussion of perception and concepts of perceptual video quality.
- In Section 2, we demonstrate the discrepancies between traditional signal fidelity measures and human perception. We then elaborate on methodologies to measure and quantify perceived visual quality.
- A formal connection between media quality and coding is established in Section 3 by means of Shannon's fundamental contributions in information theory and rate-distortion theory.
- Section 4 introduces modern approaches to standard video coding. We showcase their performance and hint at their limitations for our specific purpose.
- The contributions discussed in Section 5 are closely related works to our implementation. We categorize them to differentiate between static and dynamic region of interest coding.
- We discuss implementation details of our video coding framework in Section 6.
- Section describes the execution and results of an empirical lab study that we conducted to assess our proposed method's performance.

Personal contributions are indicated at the end of the thesis.

1. Introduction to Vision and Perception

The first section of this thesis is devoted to the introduction of topics from biology and psychology. As the observer and critic of multimedia systems is commonly a human, the concepts presented here will be fruitful in their evaluation and optimization.

1.1. Structure of the Eye and Receptor Density

The eye is the sensoriy component of the human visual system. This organ's schematic is given in Figure 1, depicting its spherical shape with the cornea, iris, and lens on top, and the optic nerve at the bottom. The interior of approximately the lower half is lined by the retina, a layer of photoreceptive cells that trigger neural responses upon incident light [8]. One distinguishes two major types of these cells. In each eye, there are approximately 100 million *rods*, which are sensitive to minute brightness intensities. The human eye additionally possesses around 5 million *cones*, that can be subdivided into short, medium and long wavelength receptors, which enable color perception [81].

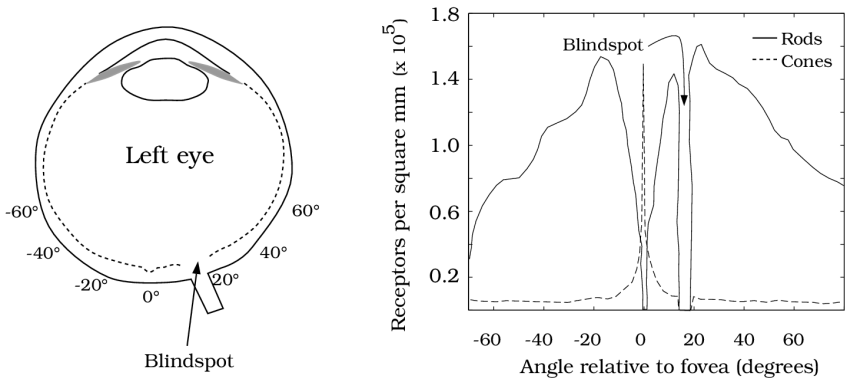

Figure 1: Structure of the eye and receptor density. [1]

Our understanding of vision on a higher, neurological level is linked to our understanding of the human brain, which is far from complete. However, we can make a simple, almost mechanical observation about the eye's capabilities based on the receptor density on the retina. The point of highest resolution, the fovea centralis, lies right on the eye's visual axis. Cone density decreases rapidly with increasing angular distance from its center, as depicted on the right in Figure 1.

This leads to the conclusion that color stimuli outside of a relatively narrow region of about 2.5° around the current fixation point can only be perceived at a significantly lower resolution. Conversely, there are no rods present in the fovea centralis, but their density peaks roughly at 20°. This effect is noticeable when observing dim light sources, such as distant stars on the night sky. Peripherally, these objects are visible, but they seem to vanish once one tries to fix one's gaze on them [81].

[1] Source: Figure 3.1, Chapter 3: The Photoreceptor Mosaic in *Foundations of Vision* [81]

1.2. Trichromatic Vision and Cone Sensitivity

The insinuated fact that the color vision of the human eye is trichromatic deserves further emphasis. Every perceivable hue is created as a mixture of the responses of the three cone types. Figure 2 presents the normalized cone sensitivity as a function of the light's wavelength. In this measurement, the light source was placed in front of the cornea; thus it includes possible aberrations induced by the eye's optically active constituents in the responses [66, 81].

Each response curve in the plot is normalized such that the maximum response is equal to one, though this equality does not necessarily hold in terms of actual neural response across the three cone types.

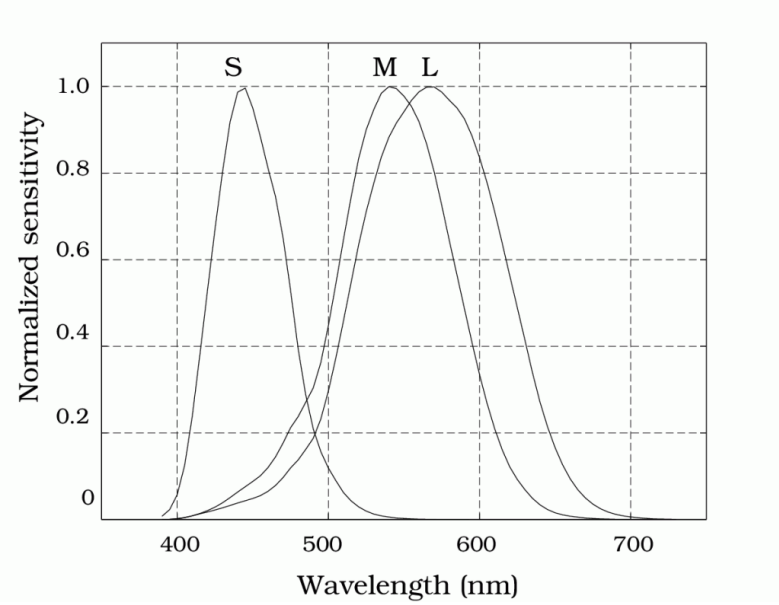

Figure 2: Normalized cone sensitivities. [2]

1.3. Higher Neural Levels

Processing of the initial photosensitive response such as detecting simple patterns arguably starts already at early neural layers following the receptors. The signal then follows a path through the optic chiasm and the lateral geniculate nuclei towards the visual cortex at the back of the head [81].

This suffices as an excursus into biology and neurology, whose sole purpose was to justify certain choices regarding video coding later in the thesis. However, there exists a whole branch of research on measuring the perception of visual quality directly through brain-related characteristics, such as through electroencephalography [2, 62]. This topic out of scope for this thesis, but might be of interest to the gentle reader.

[2]Source: Figure 3.3, Chapter 3 in *Foundations of Vision* [81]. Data originally based on [66].

2. Perceptual Quality in Multimedia

2.1. A Technical Perspective

Developing multimedia systems is commonly approached from a technical perspective, to enable novel applications using limited computational resources. Consequently, ideas from engineering, computer science, and mathematics are predominant in the field. This also holds for *quality*-related problems and research questions. A prime example of this phenomenon is the mean squared error,

$$MSE(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2 \quad \text{for } x, y \in \mathbb{R}^n$$

which serves as a fidelity metric in audio, image, and video processing [85]. The peak signal-to-noise ratio relates the MSE to the maximal signal value $I_{\max}$ on a logarithmic scale and is still one of the most popular metrics in the compression domain. Both have appealing properties, e.g. the MSE's close relationship to the energy of a physical signal or their sheer simplicity.

$$PSNR(x, y) = 10 \log_{10} \left(\frac{I_{\max}^2}{MSE(x, y)} \right)$$

Optimization solely based on formal signal characteristics is misguided, as the recipient of a multimedia system is commonly a human. This aspect is already criticised in the literature [22, 29]. An illustration of the discrepancy between the $PSNR$ and perceived quality of images is given in Figure 3 and in [85, 88].

Figure 3: Gaussian noise, Gaussian blur and salt-and-pepper noise with an equal MSE of 275. The original is displayed in the upper left quadrant.[3]

[3] Derived from `20522527.jpg`, KonIQ-10k dataset [25].

2.2. Subjective Quality Assessment Methodologies

The colloquial term "quality" has a context-dependent meaning, which is affected by domain- and user-specific expectations and biases: A photographer might appreciate *bokeh* in a portrait, while it is likely to favor a sharp depiction in, e.g., medical imagery.

This is unsatisfactory from an engineering perspective, as it entails the need to optimize systems for specific purposes based on assumptions of user preference. As this aspect of optimality is use-case dependent, there is hardly a general solution to the problem.

However, the difficulties of quality as a notion of user preference *can* be approached based on consensus. This section discusses quality assessment methodologies, scoring procedures, limitations, and practicability aspects.

2.2.1. Assessment Formalization

Out of the hypothetical set $\bar{\mathcal{I}}$ of all media items of a specific *type*, let $\mathcal{I}$ be a finite subset of *permissible* items to be assessed. We are interested in constructing a mapping

$$\psi : \mathcal{I} \to Q = [q_{\min}, q_{\max}] \subseteq \mathbb{R}$$

that assigns a *quality score* to a given media item. The terms *type* and *permissible* are used to formalize restrictions on $\mathcal{I}$ that are easily expressed in natural language:

We generally disallow the joint assessment of media items of different types and confine ourselves to comparing, e.g., only videos on one common scale. The psychological implications of cross-type comparisons are certainly interesting[4]. This is, however, not pursued further, as there is no foreseeable utility regarding the topic of this thesis.

Within a particular type of media, it is often desirable to narrow down what samples are to be considered *permissible* in the construction of ψ. It is customary to agree upon certain technical aspects in for study- and database design, such as, e.g., video duration and resolution. Media content also has to be considered, as there arise questions that extend comparisons among, e.g., depictions of different scenes or motifs are meaning- and useful [99] in quality assessment.

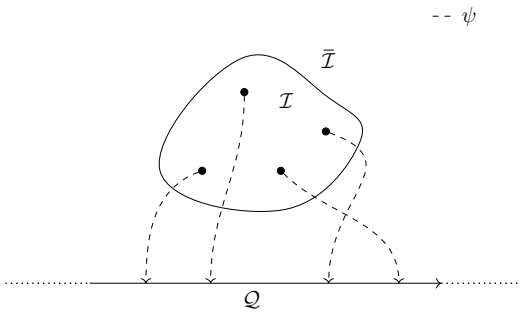

Figure 4: Mapping media items to quality scores.

[4]E.g. comparisons of emotional reaction strength for both images *and* audio sequences.

2.2.2. Single Stimulus Experiments

The most prevalent strategy for quality assessment is to define ψ in terms of mean opinion scores acquired through single stimulus experiments. The ITU standardized an absolute category rating (ACR) [35, 36] scheme with a labeled five-point scale for this purpose. Participants are queried for an absolute judgement on one media item at a time.

Figure 5: Single stimulus absolute category rating.[5]

The labels are mapped to values in $\{1, \ldots, 5\}$, and the mean over all n_i available observations o_k^i, where i is the item, k the study participant, defines the quality score:

$$\psi(i) := \frac{1}{n_i} \sum_{k=1}^{n_i} o_k^i$$

This simplicity can be advantageous and problematic, as we will elaborate:

Granularity, Anchoring and Reference Items The corarse quantization into only few selecatble quality levels combined with the restricted number of observations per item can jeopardize the whole approach, e.g. if all votes for items of clearly distinct quality fall into the same bin. Some authors allow participants to choose continuous scores instead, thereby deviating from the standard [87].

This issue is related to the problem of specifying the quality range that participants shall expect and consider with regard to the items presented in an experiment. Many older datasets include severely distorted images and videos [82], but technology advances, and so do user expectations and research questions: As the "baseline" quality for media increases, the margins for practically relevant distortions diminish. Even careful, quasi-standard instructions [54] might not suffice enable participants to differentiate properly between items with minute quality differences. An assessment task is arguably easier with a reference to compare a given item to, as will be discussed in the following section.

[5] Derived from `2313142.jpg` KonIQ-10k dataset [25].

2.2.3. Pairwise Comparison

An alternative approach to quality assessment is pairwise comparison. The central idea is to "define the [psychological] scale in terms of [vote] frequencies" [73]. This requires only relative judgments between two stimuli at a time. There is no need for participants to construct a full mental image of visual quality, in which all stimuli have to be integrated. For a given pair $(i, j) \in \mathcal{I}^2$, the task is merely to indicate which of the items is preferred.

Stimulus Presentation Double stimulus experiments involve additional choices regarding the media presentation as compared to their single stimulus counterparts.

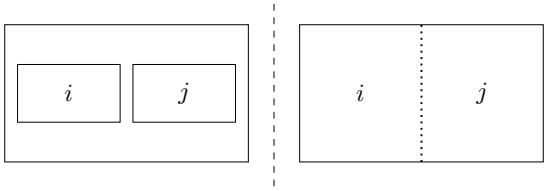

Figure 6: Configurations for simultaneous stimulus presentation.

A fundamental decision is whether the items shall be displayed simultaneously or consecutively. Figure 6 shows two possible configurations of the former. On the left, two stimuli are depicted on a common screen, with matching aspect ratios and additional margins around the items. The right side shows a vertical center crop, in which only half of the items are displayed at a time.

Both approaches entail spatial limitations and reduce the available screen area available to each stimulus. A second screen may be added to circumvent this problem in some lab scenarios, however, this is neither suitable for experiments in conjunction with most eye-tracking hardware, nor is it scalable to crowdsourcing platforms. Within these limitations, the parallel stimulus presentation is well suited for static images, whereas videos, especially with rapid content changes, may be difficult to assess.

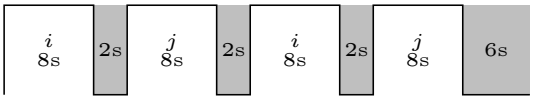

Figure 7: VQEG alternating stimulus presentation[6].

The alternative approach is consecutive stimulus presentation, where the whole screen surface is used to depict one stimulus at a time. The ITU proposes the scheme depicted in Figure 7 [54], in which the actual rating takes place after two cycles of alternating stimulus presentations, intermitted by short separator intervals of monochrome gray.

[6] 2.4 "Presentation structure of test material", VQEG subjective test plan, p. 14 in [54]

The results of an entire double stimulus experiment with binary choice can be aggregated in the form of a *count matrix*:

$$C_{i,j} = \begin{cases} \#\text{of votes preferring } i \text{ over } j \text{ for } i \neq j \\ 0 \text{ else} \end{cases}$$

Multiple approaches for the reconstruction of ψ, in this context also called *scaling*, have been proposed in the literature. A simplistic but widely used strategy is taking the sum over the columns,

$$\psi(i) := \alpha \sum_j C_{i,j}$$

which is often normalized by choosing α e.g., with regard to the numbers of comparisons per stimulus. All pairs have to be assessed by the same number of participants. This requires $\mathcal{O}(|\mathcal{I}|^2)$ comparisons, in contrast to the judgments in single stimulus experiments, which grow with $\mathcal{O}(|\mathcal{I}|)$.

Furthermore, one can neither assume consistency among participants nor in between trial repetitions. This is especially true for items that are sufficiently *close* on the perceptual scale: At the point of being indistinguishable, a random outcome is expected.

Thurstonian Scale Reconstruction Defining ψ through a probabilistic model based on Thurstone's *law of comparative judgment* [73] allows us to formally handle participant behavior and obviates the need for equally many judgements on each pair. The quality of stimuli $i, j \in \mathcal{I}$ is represented by Gaussian random variables:

$$X_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \qquad\qquad X_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

Let φ denote the standard normal probability density function and Φ the cummulative distribution function:

$$\varphi(t) = \tfrac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} \qquad\qquad \Phi(t) = \int_{-\infty}^{t} \varphi(x)dx$$

The probabilities for realizations x_i, x_j drawn from X_i, X_j are given as

$$P(X_i = x_i) = \frac{1}{\sigma_i}\varphi\left(\frac{x_i - \mu_i}{\sigma_i}\right) \qquad\qquad P(X_j = x_j) = \frac{1}{\sigma_j}\varphi\left(\frac{x_j - \mu_j}{\sigma_j}\right)$$

Note that x_i, x_j correspond to absolute quality scores in $\mathcal{Q}$. This approach models deviations of a participant's perception from the *assumed true quality scores* μ_i, μ_j. Thurstone argues that such realizations are drawn and compared in a pairwise comparison, which defines the outcome of the preference decision. Following the approach in [75], the probability of preferring item i over item j can be expressed as a difference of Gaussians:

$$P(X_i > X_j) = P(X_i - X_j > 0)$$

This quantity is illustrated as the green area in Figure 8.

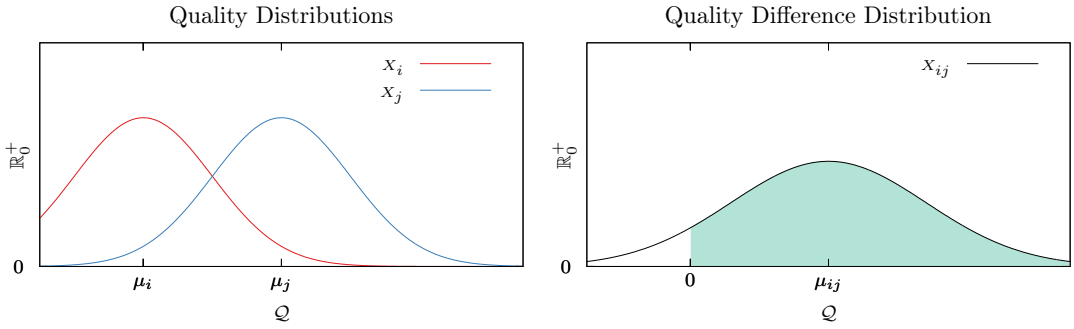

Figure 8: Thurstonian reconstruction: Quality scores modeled by Gaussian distributions.

This difference is again a Gaussian random variable with the following properties:

$$\begin{aligned} X_{ij} &= X_i - X_j \sim \mathcal{N}(\mu_{ij}, \mu_{ij}) \\ \mu_{ij} &= \mu_i - \mu_j \\ \sigma_{ij}^2 &= \sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j \end{aligned}$$

As detailed in [75], $P(X_{ij} > 0)$ can be expressed in terms of Φ as follows

$$\begin{aligned} P\left(X_{ij} > 0\right) &= \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(\frac{-\left(x-\mu_{ij}\right)^2}{2\sigma_{ij}^2}\right) dx \\ &= \int_{-\infty}^{\mu_{ij}} \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(\frac{-x^2}{2\sigma_{ij}^2}\right) dx \\ &= \int_{-\infty}^{\frac{\mu_{ij}}{\sigma_{ij}}} \varphi\left(t\right) dt \\ &= \Phi\left(\frac{\mu_{ij}}{\sigma_{ij}}\right) \end{aligned}$$

Applying the inverse CDF and solving for the quality difference yields:

$$\mu_{ij} = \sigma_{ij}\Phi^{-1}\left(P(X_{ij} > 0)\right)$$

The proportion of the preferences is then used as an empirical estimate for $P(X_{ij} > 0)$:

$$\hat{\mu}_{ij} = \sigma_{ij}\Phi^{-1}\left(\frac{C_{i,j}}{C_{i,j} + C_{j,i}}\right)$$

Case V Simplifications Thurstone proposed five versions of his model with increasingly strict assumptions on the score distributions of the items under comparison [73]. These reduce the degree of freedom, allowing to compute the estimate $\hat{\mu}_{ij}$ from the data given in C. The Case V model requires X_i, X_j to be uncorrelated with equal variances:

$$\sigma_i^2 = \sigma_j^2 \qquad\qquad \rho_{ij} = 0$$

W.l.o.g set $\sigma_i^2 = \sigma_j^2 = \frac{1}{2}$ to obtain unit variance for X_{ij}, simplifying the estimate to:

$$\hat{\mu}_{ij} = \Phi^{-1}\left(\frac{C_{i,j}}{C_{i,j} + C_{j,i}}\right)$$

Multiple approaches exist to align the pairwise differences on a common scale [75]. To apply a maximum likelihood strategy, we phrase the likelihood of $\mu_{ij} = \mu_i - \mu_j$ as

$$\begin{aligned} L(\mu_{ij}) &= P(C_{i,j}, C_{j,i} \,|\, \mu_{ij}) \\ &= \frac{1}{\gamma} P(X_i > X_j)^{C_{i,j}} P(X_j > X_i)^{C_{j,i}} \\ &= \frac{1}{\gamma} \Phi(\mu_{ij})^{C_{i,j}} (1 - \Phi(\mu_{ij}))^{C_{j,i}} \\ &= \frac{1}{\gamma} \Phi(\mu_{ij})^{C_{i,j}} \Phi(-\mu_{ij})^{C_{j,i}} \end{aligned}$$

where γ is a scale factor to adjust for sample size. The maximum log-likelihood of μ_{ij} is

$$\hat{\mu}_{ij} = \arg\max_{\mu_{ij}} \; C_{i,j} \log(\Phi(\mu_{ij})) + C_{j,i} \log(\Phi(-\mu_{ij}))$$

and the log-likelihood for all quality scores $\mu = [\mu_1, \mu_2, ...] \in \mathcal{Q}$ given C is

$$\mathcal{L}(\mu|C) = \sum_{i,j} C_{i,j} \log\left(\Phi(\mu_i - \mu_j)\right)$$

The quality scores μ are obtained by solving the following maximization problem:

$$\arg\max_{\mu} \; \mathcal{L}(\mu|C) \quad \text{subject to} \quad \sum_i \mu_i = 0$$

whereby the constraint ensures a unique solution.

Pairwise Comparison with Multiple Options Thurstone's model can be extended to permit more than two options to choose from when assessing a pair of items. This is useful to facilitate *undecided* responses and multiple degrees of preference. The formalism is to partition the quality difference scale into k intervals representing the options, with boundaries symmetric around zero. An example is given in Figure 9.

For odd choices of k, define boundary variables $\delta_1 < \ldots < \delta_{\frac{k-1}{2}}$ and arrange them into an auxiliary tuple for simplified indexing:

$$\beta = (-\infty, -\delta_{\frac{k-1}{2}} \ldots, -\delta_1, \delta_1, \ldots, \delta_{\frac{k-1}{2}}, \infty)$$

If k is even, define $\frac{k}{2} - 1$ such variables, and include a zero in the tuple as follows:

$$\beta = (-\infty, -\delta_{\frac{k}{2}-1} \ldots, -\delta_1, 0, \delta_1, \ldots, \delta_{\frac{k}{2}-1}, \infty)$$

Extending the previous definition of C, define a tuple of count matrices:

$$\widetilde{C} = (c^1, \ldots, c^k) \qquad \text{with} \qquad c^l_{ij} = \text{frequency of option } l \text{ on tuple } (i, j)$$

The likelihood of a quality difference μ_{ij} with option boundaries β is then given as:

$$\mathcal{L}(\mu_{ij}, \beta) = P(\widetilde{C} \,|\, \mu_{ij}, \beta) = \frac{1}{\gamma} \prod_l P(\beta_l < X_{ij} \leq \beta_{l+1})^{c^l_{ij}}$$

Finally, quality scores μ and option boundaries β are then again recovered by solving

$$\underset{\mu,\beta}{\arg\max} \sum_{i,j,l} c^l_{ij} \log \left(\int_{\beta_l}^{\beta_{l+1}} \varphi(t - \mu_{ij}) \, \mathrm{d}t \right)$$

subject to $\sum_i \mu_i = 0$ and the aforementioned ordering condition on the δ's.

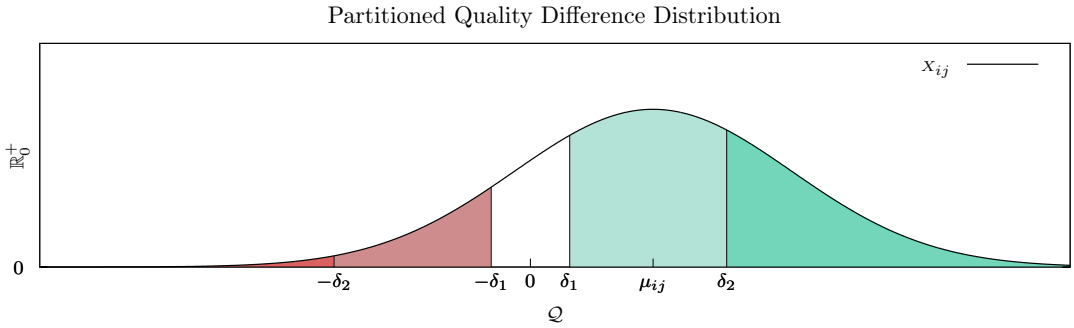

Figure 9: Model for pairwise comparison with five options.

Instead of optimizing for β, one can also specify fixed boundary variables and thereby reduce the problem to an optimization for μ. This enforces a certain notion of distances between the different choices.

2.2.4. The Just Noticeable Difference

The concept of a *just noticeable difference* is immanent to comparative assessment paradigms and closely related to stimulus indistinguishability and perceptual equality. It describes the smallest perceivable change in a stimulus for a given comparison task.

Weber-Fechner Laws The perceptual impact of physical stimuli on human observers is studied in *psychophysics*, a branch of psychology initiated by Gustav Fechner. In his publication [18], he presented two rules, which became well-known as Weber's law, named after his professor and colleague Ernst Weber, and Fechner's law, which is an extension of the former. These ideas are generally applicable to the human perception of physical stimuli. Typical examples include the comparison of weights or the volume of sounds.

Weber's law states that the just noticeable difference ΔS relative to a reference stimulus S in terms of a physically measurable intensity or magnitue is constant:

$$\frac{\Delta S}{S} = c$$

Fechner's law is a result of integrating the former while assuming c is independent of S. The formula below essentially states that the perceived stimulus intensity p grows only linearly when the measured intensity S grows exponentially. The constants c_0 and S_0 depend the stimulus type and the participant.

$$p = c_0 \cdot \ln \frac{S}{S_0}$$

JNDs in Quality Assessment To utilize the concept of a just noticeable difference for the purpose of quality assessment, we require an additional function to be defined on the itemset $\mathcal{I}$. It models a systematic distortion that is present on all of its elements:

$$d : \mathcal{I} \to \mathbb{R},\ i \mapsto d(i)$$

This is especially suited for artificially distorted data, where $\mathcal{I}$ contains multiple *versions* of an item derived from a single source. Each version is affected by the same distortion with varying severity. When studying e.g., compression, d can be defined for each item according to its compression rate. A fine granularity in terms of perceived distortions is favorable in the dataset, but often predefined through e.g., the compression levels that the algorithm under inspection offers. When ordering all items according to an index by their distortion level, it is desirable that consecutive pairs of items i_k and i_{k+1} are perceptually indistinguishable, though this is again often impossible in practice.

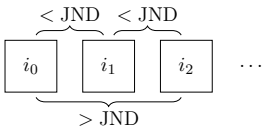

Figure 10: Comparing i_0 with i_2 is above the JND, adjacent pairs are indistinguishable.

The task for a study participant is to find the item with the smallest noticeable distortion:

$$\arg\min_{i \in \mathcal{I}} d(i) \quad \text{such that a difference is visible.}$$

Item presentation styles for conventional pairwise comparison, such as those presented in Section 2.2.3, are generally applicable to JND experiments. However, the presentation sequence with regard to the distortion magnitude is of additional concern, as salient regions of interest and distortion visibility affect each other.

Unimpaired original. JPEG compression ($qp = 10$).

Figure 11: Compression artifacts in smooth background regions.[7]

Binary search would be an obvious choice to identify the item at the point of just noticeable difference. Bisection search methods will present items above the JND at some point. This can incur a visual *priming* effect on regions most affected by the distortion, shifting the observer's attention to these areas.

An example of this effect is given on the right in Figure 11, where the distortions are most visible in the presumably non-salient, blurry background. This is an aspect to consider in experimental design: Is a notion of JND preferred where such hints help the observer to spot distortions of smaller magnitude, or is it a higher tolerance level preferable, at which the observer has to notice the difference without pointing it out? An alternative would be linear search, starting from the lowest distortion, through which this effect can be circumvented. This comes at the cost of a theoretically higher number of required comparisons in the worst-case, though this can be avoided in practice through sensible database design.

Just Noticeable Distortion The last method in this enumeration is merely an alteration of the just noticeable difference; it can even be hard to distinguish between these two properly, depending on the presentation type. By a *just noticeable distortion* experiment, we denote a just noticeable difference scenario without a reference. The *difference* is implicit and relative to the initial state of the presented medium.

This is carried out through single stimulus experiments, in which the distortion severity is gradually increased over time until the observer reports to notice a difference.

[7] Derived from `3074454678.jpg`, KonIQ-10k dataset [25].

Beyond Participant and Source Specificity The notion of a just noticeable difference in the Weber/Fechtner sense is both *participant* and *source*-specific. This is inadequate for many problems in quality research. Often, more generally valid results are sought after, independent of the specific content or a particular observer's distortion sensitivity.

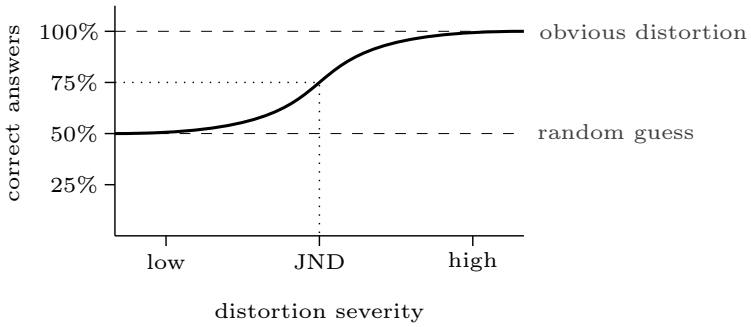

Figure 12: The JND in pairwise comparison experiments with forced choice[8].

The point of just noticeable difference can be defined alternatively as the distortion level, at which a certain fraction of observers is able to tell the difference. For pairwise comparison experiments with a forced choice on two possible answers, a sensible definition is the point at which 75% of the answers are correctly indicating the distorted version [24, 41, 47]. An equal statement would be that 50% of the participants are able to see a difference, and therefore answer correctly, while the other half has to choose at random.

The convention is to *indicate the percentage of observers that can tell the difference*, the corresponding distortion level is then said to be at the x% JND. A lower percentage leads to stricter, less distortion-forgiving quality requirements. For our experiments in the subsequent chapters, we will even report results at the 10% mark. The assumption that cross-participant JNDs are reasonable is somewhat akin to Thurstone's simplifications [73] for pairwise comparisons, although taking percentiles isn't as elaborate as the probabilistic modeling in Section 2.2.3.

2.3. Quality Databases

Annotated quality databases are closely tied to assessment methodologies. They are required to evaluate quality prediction algorithms, and for the training of machine learning models [88]. One distinguishes artificially [46] from authentically distorted [27] databases, depending on whether pristine media items were deliberately impaired or whether they expose the mixtures of degradations found *in the wild*. Another concern is whether the data was gathered in a controlled lab environment, or through a web-based crowdsourcing experiment, which is often the only possibility to gather sufficient amounts of ratings. A comprehensive list of quality-related databases is given in [92][9], however the state of the art does not factor in foveation, which increases database design complexity.

[8] Adapted from [24], Figure 1.

[9] The author curates an up to date list at `https://stefan.winkler.site/resources.html`

3. Communication Systems and Optimality Criteria

A formal link between the previous and the subsequent chapters is established through rate-distortion theory [55, 64], which connects a codec's reconstruction quality and its efficiency within the framework of information theory. It gives rise to theoretical optimality considerations and corroborates practical codec performance measures.

3.1. General Communication Systems

Shannon's seminal work [64] introduced a model of a general communication system. It compartmentalizes the involved operations into the parts depicted below, thereby enabling or simplifying their analysis and optimization:

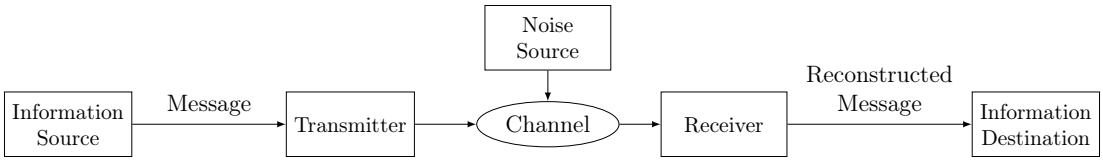

Figure 13: Schematic of a general communication system.[10]

The *message*, which gets emitted by the *information source*, is of a certain type, which is classified through a function describing its content: An analog signal can be modeled as a function of time, a grayscale image might be defined on a two-dimensional domain, and the color value of a video pixel would depend on e.g., spatial location, time and the color channel.

The *transmitter* prepares the message in a suitable way for transmission over the *channel*, which in this model is an analog medium. We utilize a subset of the model's descriptive power and confine ourselves purely to encoding and reconstruction within a digital system, leaving the transport channel peculiarities unappreciated. The ability to describe the continuous aspects of analog communication are nowadays more of a topic in electrical engineering rather than computer science. The transmitter's operation depends on the message- and channel characteristics, examples from [64] include e.g. analog telephony and communication through pulse coded modulation.

A *noise source* influences the signal during transmission over the channel. This is commonly described stochastically, e.g., through a Gaussian distribution that models the noise signal characteristics. An aspect of interest is a communication system's robustness against noise, and the cost induced for (statistical) guarantees on transmission success.

The *receiver* accepts the noisy signal and reconstructs the original message by inverting the operations applied by the transmitter. It is then forwarded to the *destination*, which can be either "person (or thing)" in Shannon's definition [64].

Though this work focusses on video messages with humans as receivers, its worth looking at rate-distortion theory involving models with better-understood characteristics.

[10] Adapted from [64], Figure 1.

3.2. Rate-Distortion Theory

Consider the information source to be a random variable X. The goal is to represent realizations of sequences $X^n = (X_1, X_2, \ldots, X_n)$ with independent and identically distributed elements using as few bits as possible [16]. We assume that the sample space $\mathcal{X}$ contains finitely many *symbols*, though formulations of rate-distortion theory on continuous random variables exist. A sequence of n symbols shall be encoded by an *index* of 2^{nR} bits, constructed by a $(2^{nR}, n)$ *code pair* of functions that is defined as follows:[11]

$$f_n : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\} \qquad g_n : \{1, 2, \ldots, 2^{nR}\} \to \hat{\mathcal{X}}^n$$

These correspond to the blocks in a simplified communication system model:

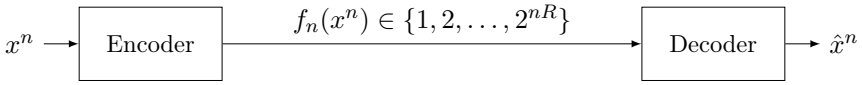

Figure 14: Rate-distortion encoder-decoder model.[12]

The sequence $\hat{x}^n = g_n(f_n(x^n)) \in \hat{\mathcal{X}}$ is an approximation of the input sequence $x^n \in \mathcal{X}^n$. An element-wise distortion function d is utilized to evaluate this reconstruction. Popular choices include the squared difference d_{SD} and the Hamming distance d_H:

$$d : \mathcal{X} \times \widehat{\mathcal{X}} \to \mathbb{R}^+ \qquad d_{SD}(x, \hat{x}) = (x - \hat{x})^2 \qquad d_H(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases}$$

Pointwise distortions can be extended to sequences in multiple natural ways:

$$d_{\text{mean}}(x^n, \hat{x}^n) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i) \qquad d_\infty(x^n, \hat{x}^n) = \max_{i=1,\ldots,n} d(x_i, \hat{x}_i)$$

The *distortion* D of code (f_n, g_n) on X given d is defined as the expected distortion[13]:

$$D = E\left[d(X^n, g_n(f_n(X^n)))\right] = \sum_{x^n} p(x^n) d(x^n, g_n(_n(x^n)))$$

A rate-distortion pair (R, D) is *achievable* if a sequence (f_n, g_n) of $(2^{nR}, n)$ codes exists such that:

$$\lim_{n \to \infty} E\left[d\left(X^n, g_n(f_n(X^n))\right)\right] \leq D$$

The rate-distortion function $\mathcal{R}(D)$ is the infimum of the rate R such that (R, D) is in the closure of the set of achievable rate-distortion pairs of source for a given D on source X. Its calculation yields a theoretical bound for the compressibility of X in the rate-distortion sense [16]. Unfortunately, neither describing a video source using random variables nor modeling human perception as a distortion function is easy. This prevents us from achieving the most prominent results of rate-distortion theory in our application of foveated video coding, but Shannon's theory is an essential reference in this field and provides valuable context on the goal of bitrate optimization.

[11] 10.7 and 10.8 in [16]
[12] Adapted from [16], Figure 10.2.
[13] 10.9 and 10.10 in [16]

4. Video Coding

Video coding as a discipline builds upon traditional signal- and image processing with the goal of creating video data representations for specific purposes. A prominent concern is to achieve high compression rates with *sufficient* reconstruction quality, e.g., for streaming and storage in consumer applications. Requirements and challenges depend on the setting and differ largely e.g., in cinematic, industrial, and scientific videography. This section introduces concepts of modern video coding and presents relevant aspects of the state of the art, though the subject is too broad to aspire for completeness.

4.1. Motivation

For the scope of this work, we consider a raw video to be a sequence of matrices

$$f = (f_i)_{i=1,\dots k} \in \mathcal{C}^{m \times n}$$

called *frames*, with a *resolution* of $m \times n$ *pixels*. For now, we assume either $\mathcal{C} = \mathcal{C}_{\text{gs}} := \{0, \dots, 2^8 - 1\}$ or $\mathcal{C} = \mathcal{C}_{\text{rgb}} := \mathcal{C}_{\text{gs}}^3$, which are natural choices to represent either grayscale or color values. Enhancements over this baseline in terms of compression are inevitable, as the file size of a video is the product of the resolution, the number of frames, and the *bit depth* per pixel. One hour of plain 4K video with 30fps at 24bpp requires circa 2.6 terabytes of data.

4.2. Hybrid Video Coding

The predominant class of *hybrid video codecs* [15, 50, 70, 91] extends concepts from still image codecs, such as JPEG [80], to motion pictures. Their approach is to *predict* parts of the next frame based on previously available data and to represent the residual error in terms of transform domain coefficients. These are subsequently quantized to discrete values, which induces compression loss but also accounts for a major share of data reduction. Entropy coding further reduces the size of the resulting bitstream. Codec-independent post-processing, such as multiplexing with audio streams, commonly finalizes the process. A coarse overview of the mechanics is given below.

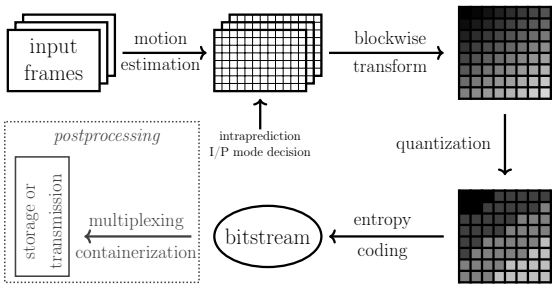

Figure 15: General schematic of a hybrid video encoder.

We will now give an overview of available techniques without confining ourselves to one particular codec implementation.

4.2.1. Frequency Domain Representations

A certain degree of spatial correlation is typical in visual media, as pixels in close proximity often have similar color values. Natural images and videos tend to contain regions with smooth color gradients. The key to exploiting this property for compression is to store information about relative variation instead of absolute color values for each individual pixel. The Fourier transform [11] provides the canonical approach to this kind of analysis.

Transform Considerations The discrete Fourier transform is defined as:

$$\mathrm{DFT}[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi \frac{kn}{N}} \quad \text{for } 0 \leq k \leq N-1$$

This operator maps a signal x of N complex elements to an equal-length sequence of Fourier coefficients. It is a discretization in both the spatial and the frequency domain of the continuous Fourier transform, which is given as:

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2i\omega t} dt$$

Though widely used in signal analysis applications, the Fourier transform was ousted in the compression domain by variants of the discrete cosine transform. The DCT was proposed by Ahmed et al. [3, 4]. The following definition is according to [37].

$$\mathrm{DCT}[k] = \alpha_k \sum_{n=0}^{N-1} x[n] \cos \frac{(2n+1)\pi k}{2N} \tag{1}$$

$$\alpha_0 = \sqrt{\frac{1}{N}} \quad \alpha_k = \sqrt{2N} \text{ for } 1 \leq k \leq N-1$$

Fast DCT Calculation We now deduce how to express the DCT in terms of the DFT. This clarifies the relationship of the transforms and serves as a means for efficient computation through fast Fourier transform implementations [37]. The inverse DCT of a sequence d is given as:

$$x[n] = \sum_{k=0}^{N-1} \alpha_k d[k] \cos \frac{(2n+1)\pi k}{2N} \quad \text{for } 0 \leq n \leq N-1$$

For $0 \leq N \leq \frac{N}{2} - 1$ define an auxiliary sequence:

$$\tilde{x}[n] = x[2n] \qquad \tilde{x}[N-n-1] = x[2n+1]$$

Split the sum in (1) into even and odd terms:

$$\begin{aligned}\mathrm{DCT}[k] &= \alpha_k \left(\sum_{n=0}^{\frac{N}{2}-1} x[2n] \cos \frac{(4n+1)\pi k}{2N} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \cos \frac{(4n+3)\pi k}{2N} \right) \\ &= \alpha_k \left(\sum_{n=0}^{\frac{N}{2}-1} \tilde{x}[n] \cos \frac{(4n+1)\pi k}{2N} + \sum_{n=0}^{\frac{N}{2}-1} \tilde{x}[N-n-1] \cos \frac{(4n+3)\pi k}{2N} \right)\end{aligned}$$

substitute $n' := N - n - 1$; utilize periodicity and symmetry:

$$\begin{aligned}&= \alpha_k \left(\sum_{n=0}^{\frac{N}{2}-1} \tilde{x}[n] \cos \frac{(4n+1)\pi k}{2N} + \sum_{n'=\frac{N}{2}}^{N-1} \tilde{x}[n'] \cos \frac{(4N-1-4n')\pi k}{2N} \right) \\ &= \alpha_k \sum_{n=0}^{N-1} \tilde{x}[n] \cos \frac{(4n+1)\pi k}{2N} \\ &= \mathrm{Re} \left(\alpha_k e^{-i2\pi \frac{k}{N}} \sum_{n=0}^{N-1} \tilde{x}[n] e^{-i2\pi \frac{kn}{N}} \right) \\ &= \mathrm{Re} \left(\alpha_k W_N^k \, \mathrm{DFT} \left(\tilde{x}[n] \right) \right)\end{aligned}$$

where we use the common abbreviation $W_N = e^{-i2\pi/N}$.

The DCT for Images A 2D version of the DCT can be defined as follows, which is a natural extension required for processing image data of $M \times N$ pixels:

$$\mathcal{DCT}[u,v] = \alpha_u \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] \cos \frac{(2m+1)\pi u}{2M} \cos \frac{(2n+1)\pi v}{2N}$$

with the following normalization coefficients:

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}}, & \text{for } u = 0 \\ \sqrt{\frac{2}{M}}, & \text{for } 1 \le u \le M-1 \end{cases} \qquad \alpha_v = \begin{cases} \frac{1}{\sqrt{N}}, & \text{for } v = 1 \\ \sqrt{\frac{2}{M}}, & \text{for } 1 \le v \le N-1 \end{cases}$$

We will now discuss the two components required to unfold the DCT's potential for compression, namely quatization and entropy coding.

4.2.2. Quantization

The representation in terms of DCT coefficients does not induce a data reduction per se, as the bijective transform results in the same amount of transform coefficients as there are pixels in the input image. However, it allows removing information on certain frequency bands through quantization, which is customarily implemented as a division[14]:

$$\overline{\mathcal{DCT}}(u,v) = \operatorname{sgn}\{\mathcal{DCT}(u,v)\}\,\frac{|\mathcal{DCT}(u,v)| + f(Q_s)}{Q_s}$$

Here, Q_s controls the quantization step size, $f(Q_s)$ adjusts the behavior near zero.

Static Quantization Band-dependent choices of Q_s are realized, e.g., through standardized tabular values, which are empirically optimized for perceptual appeal. These base values are *scaled* to adjust the overall bitrate. A prototypical example is JPEG, in which quantization according to the tables utilized by the IJG [80] coarsens accuracy in high frequency coefficients by enforcing larger denominators for high values of u and v. This aims to exploit the fact that the human eye is indifferent to high frequencies above the perceivable resolution.

Dynamic and Adaptive Quantization On the one hand, video codecs aim to improve the efficiency in the quantization, e.g., by avoiding divisions [49]. Quantization is, on the other hand, a main driving factor in data reduction, thus a promising leverage point for improvements. This led to the development of heuristics which aim to maximize data reduction with acceptable perceptual impact, such as the variance-based adaptive quantization (VAQ) in x264[15], briefly described in [77][16]and [1].

$$\Delta QP = D \times (\log(V_{\text{block}}) - \log(V_{\text{frame}}))$$

This approach calculates an offset for x264's blockwise quantization parameter QP according to the difference between the logarithmized block variance V_{block} and the overall frame variance V_{frame}. This causes *smoother* blocks to receive a finer quantization, as the human visual system is more sensitive to minute differences in flat regions [1]. The scale factor D is chosen empirically, and the logarithms are required to achieve a linear relationship between the variance difference and the quantization granularity.

The VAQ approach depends only on the current frame, but more recent proposals also incorporate e.g. inter-frame dependencies such as *motion*, which is discussed in Section 4.3. These techniques are arguably rather *rate control* methods than modifications of quantization itself. Details on the quantization in HEVC are given in [70][17], and the fundamental concept of adapting default values according to heuristics is still utilized in state of the art codecs [59].[18]

[14] According to (7) in [49].

[15] `https://code.videolan.org/videolan/x264/commit/b59440f09b7eb7e6f30c1131d56843ee92e3751d`

[16] Mailing lists are at least persistent references for not formally published open source contributions.

[17] Chapter 6.3: Quantization and De-quantization

[18] Chapter 7.12: Reconstruction and dequantization

4.2.3. Block Partitioning

Video codecs subdivide frames into spatial hierarchies. The largest entities are called macroblocks, superblocks or coding tree units (CTU), depending on the project lingo [50, 67]. They are commonly square shapes and have increased in size over the last codec generations, covering up to 128×128 pixels in the current AV1 standard [15]. CTUs are spatially subdivided for independent processing in terms of transform and prediction blocks, as shown below for a ten-way tree structure.

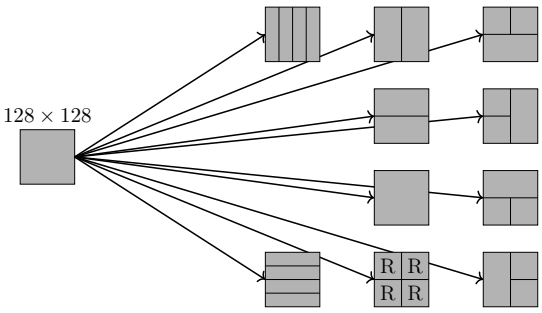

Figure 16: Partitioning tree in AV1 with a recursive option.[19]

The goal is to adapt these partitions to frame content to exploit repetitive textures through *prediction*. Such schemes are an extension of simpler block-based image codecs, e.g., JPEG [80], in which 8×8 pixel blocks are the only level of subdivision and processing.

Block Boundary Issues Coarsely quantized DCT coefficients with low reconstruction accuracies induce a distortion that manifests itself clearly through discontinuities at block boundaries. The resulting artifacts are easily visible in smooth areas, but they also affect more structured regions with higher frequencies, as shown in the enlarged patches.

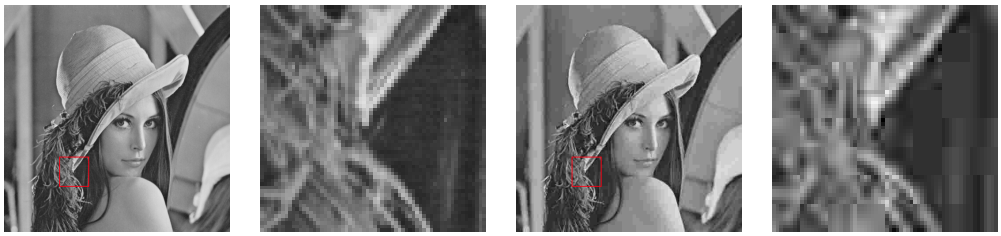

Figure 17: Lena: Original and JPEG compressed with patch sample markers.[20]

The JPEG algorithm is suitable for this visualization, as the artifacts are mainly caused by DCT coefficient quantization and not a result of a mixture of distortions as in modern video formats. Figure 33 depicts similar distortions in the periphery induced by our foveated video codec implementation.

[19] Adapted from [15], Figure 1.

[20] Source: `lena512.bmp`, *Standard Test Images*, compiled by Mike Wakin, University of Michigan [79].

4.3. Inter Prediction

4.3.1. Frame Types

Frames in temporal proximity within an uncut scene usually differ only slightly in terms of content. *Inter prediction* [65] techniques aim to exploit similarities between frames. The approach is to model a frame as closely as possible in terms of already decoded data. This reduces the coding complexity of that particular frame. Instead of expressing the whole content, it suffices to express the difference between the prediction and the original in terms of a residual or error signal. A categorization into *frame types*, based on the kind of references used in the prediction, typically includes the following:

- **I**-frames: *intra-* or key frames are self contained and do not reference other frames.
- **P**-frames: *predicted* frames that reference previously decoded frames.
- **B**-frames: *bidirectionally* predicted frames which also reference future frames.

These types are arranged into so called groups of pictures (GOP), which form an independend entity that requires no outside frames for decoding. The arrows shown in Figure 18 indicate the direction in which predictions are made: P_1 receives only a *forward* prediction from I_1, whereas B_1 and B_2 additionally receive *backwards* predictions from P_1. Recent codecs may utilize more intricate schemes, for example the multi-layered approach in AV1 [15], though their effect is content-dependent [14].

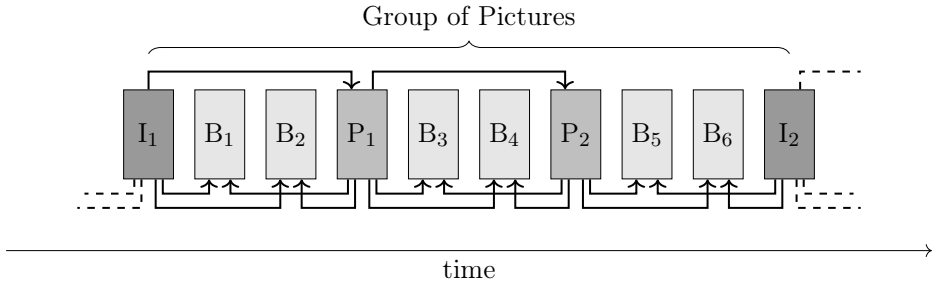

Figure 18: Multilayer inter prediction scheme.[21]

The frame sequence above is presented in *display order*. Encoded frame representations are usually permuted for storage and transmission according to their prediction dependencies to reduce buffering requirements on the decoder. A noteworthy parameter is the *GOP size*, which determines how many frames are inter predicted between two I-frames. Instead of fixed values, modern codecs [94, 96] implement upper and lower limits in combination with a scene cut detector. Long inter-predicted sequences can be beneficial when there is little temporal change, while the I-frames still contain roughly the same content [65]. We now discuss concrete approaches to inter prediction.

[21] Adapted from [65], Figure 16.1.

4.3.2. Inter Prediction Techniques

The tradeoff in synthesizing content from temporally adjacent frames is to gauge the reconstruction error with the overhead of parameterizing the underlying operation.

Replenishment An early method proposed for television systems in 1969 is *Conditional Replenishment* [52], in which only pixels changes that exceed a threshold are encoded and transferred. This requires a reference picture to be kept and updated on both the encoder and the decoder side. This primitive approach is superseded by the capabilities of modern codecs, which exploit geometric relationships between frame contents.

Block Matching and Motion Compensation Motion in the simplest sense is understood as a spatial displacement of some discernable entity along the temporal dimension of a video. Given a frame f_{src} that is to be encoded using inter prediction, one aims to identify the corresponding region in a reference frame. The situation can be illustrated as follows:

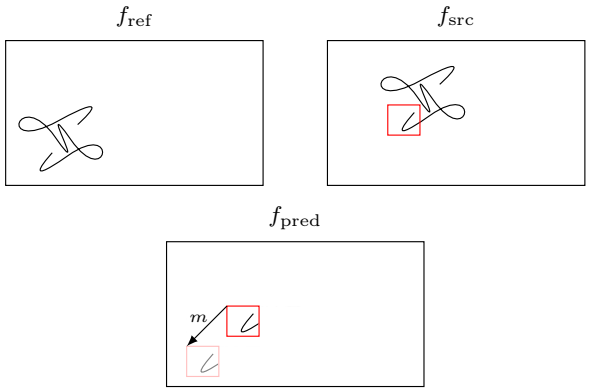

Figure 19: Block based motion compensation.

Identifying the counterpart to the red block in f_{src} in the reference can be formulated as the solution of a minimization problem, compactly written as:

$$m = \arg\min_{t} d(f_{\text{src}}[b], f_{\text{ref}}[b+t]))$$

It involves a distance function d defined on equal-sized $u \times v$ matrices of color values cropped from the respective frames at location b and $b+t$. The *motion vector* m is a solution in the sense of the distance function. A popular choice for d is the normalized cross-correlation [65], which can be formulated in terms of matrix indices as follows:

$$C(b,t) = \frac{\sum_{j=0}^{u}\sum_{k=0}^{v} f_{\text{src}}(j+b_x, k+b_y) f_{\text{ref}}(j+b_x+t_x, k+b_y+t_y)}{\sqrt{\sum_{j=u}^{u}\sum_{k=0}^{v} f_{\text{src}}(j+b_x, v+b_y)^2}\sqrt{\sum_{j=u}^{u}\sum_{k=0}^{v} f_{\text{ref}}(j+b_x+t_x, v+b_y+t_y)^2}}$$

Under ideal conditions, m alone suffices to describe the examined region in f_{pred}.

State of the Art in Inter Prediction Current encoders consider more elaborate transforms than just translations. Among others, AV1 utilizes the following techniques [15]:

- It employs the *dynamic motion vector referencing scheme* introduced in [23] and analyzes a spatially and temporally larger neighborhood than previous generation codecs. The comparison in [15] explicitly mentions VP9 [53], a last-generation codec developed by Google, compared to which AV1 increases the possible reference frames from 3 to 7. Combinations of predictions from multiple past and/or future frames are applied to minimize the residual depending on the frame content.

- The codec optionally utilizes *global and locally adaptive warped motion compensation*, as proposed by Parker et al. [56]. The global motion model works on a frame level and is intended to compensate camera movement, while the local compensation allows affine and even homographic transformations on a block level. This method is only applied if it achieves superior performance relative to plain translations.

- Interpolated prediction [39], where at pixel i, j a mask[22] $m(i, j) \in [0, 1]$ is used to blend two predictors p_1, p_2 linearly: $p(i, j) = m(i, j)p_1(i, j) + (1 - m(i, j))p_2(i, j)$

The increasing benefit of inter prediction techniques over the generations of video standards is clearly visible when comparing the compressed frame sizes of the same input video using default codec options, which produce visually indistinguishable results.

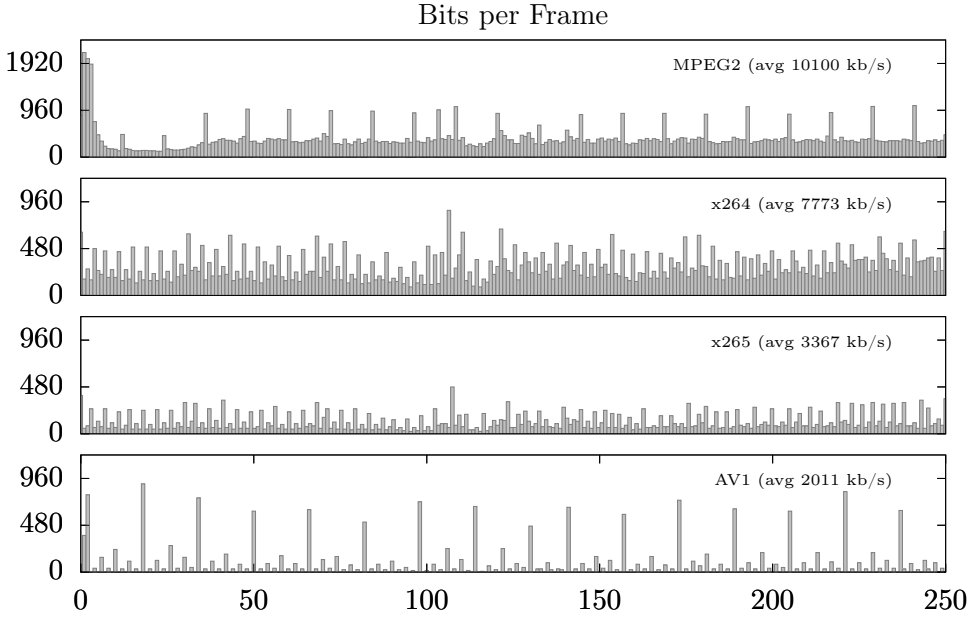

Figure 20: Compressed frames for src #04, VQEG JEG dataset, see Fig. 32.

[22] E.g. *wedge* codebook shapes: Fig. 5 in [15].

4.4. Intra Prediction

The ongoing development of inter prediction techniques naturally led to diminishing margins for further improvements. It is currently a promising branch of research to focus on the temporally independently coded I-frames [39], as already indicated by the bar ratios of the more recent codecs in Figure 20. Though the primary goal of intra coding is to reduce the size of these, intra techniques can also be utilized within predicted frames.

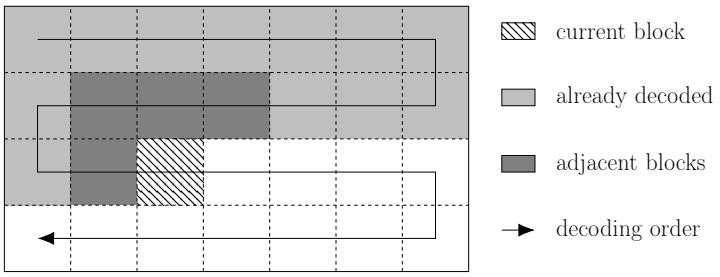

Figure 21: Conceptual data schematic of intra prediction.

Inter prediction can only rely on already decoded blocks within the same frame. A rudimentary schematic is given in Figure 21. The hatched block is to be predicted in terms of the already decoded blocks, whereby the adjacent ones constitute promising candidates. As for inter predictions, the parameterizations of the predictors have to utilize fewer bits than what the prediction saves in terms of the transform-coded residual.

The HEVC codec [70] allows intra predictions on all supported transform block sizes. Directional structures in the current block can be interpolated from adjacent pixels in 33 pre-defined angles, so-called *angular modes* with a displacement accuracy of $\frac{1}{32}$ pixels [42]. Alternatively, HEVC provides means to reconstruct planar surfaces lacking significant edges as the average of a horizontal and a vertical linear interpolation[23].

More recent developments include exploiting correlations between color channels [74], more precisely predicting chroma gradients from the luma channel. In certain scenarios, e.g., when encoding artificial renderings, it is possible to achieve performance gains with surprisingly simple methods. AV1 [15] introduced the possibility to define color palettes of up to 8 colors per block, which are assigned to pixels via an index that is subsequently entropy coded. Alternatively, the codec utilizes a novel *block copy* mode, that is especially beneficial on repetitive textures.

I-frames as a Necessity The previous discussions and Figure 20 suggest that I-frames are only sensible at scene cuts, when substantial, non-predictable content changes occur. In between, longer inter predicted sequences seem beneficial, but there are practical reasons for a certain I-frame frequency. They enable fast access to frames at random times, as used in seeking, by limiting the number of frame dependencies that have to be decoded first. In addition, they allow to fully recover from damaged data and packet loss at the next I-frame, and allow to join live streams in a timely manner.

[23] Equations (4.18), (4.19) and (4.20) in [70].

4.5. Color Models and Spaces

Color models [30] define representation systems for visually perceivable color impressions. Several types, tailored towards specific purposes, exist, in which concrete *color spaces* [38] specify the actually representable values. These constitute the *gamut* of a color space. The additive RGB color models [68] are based on the human visual system's trichromaticity. Capturing and replicating relevant wavelengths of the visible spectrum motivates the development of camera sensors and display technology.

Color spaces naturally intersect in large parts of their gamut. In some cases, the conversions between two spaces are as simple as a linear coordinate transform[24]. As most color spaces do not cover the whole perceivable spectrum, there exist cases of irrepresentable color elements. Rounding errors caused by conversions between spaces may introduce visible distortions. To this day, `sRGB` [6] is one of the most prevalent definitions and endorsed by multiple standardization bodies. It has been criticized for its small gamut and is not suited for high dynamic range imagery, which is a major topic in current developments and standardization efforts [34].

An RGB representation has been shown to require significantly more bits per pixel at the JPEG algorithm's visually lossless threshold than alternatives [51]. De facto standard in compression is the `YUV` [30] derived `YCbCr` space [20]. It stores one luminance and two chrominance channels, which allows an apt data reduction through subsampling.

4.6. Chroma Subsampling

Chroma subsampling is an image compression method that is physiologically inspired through the receptor distributions on the retina. The human eye is equipped with vastly more brightness sensitive rods than color-sensitive cones, as discussed in Section 1. This property is directly exploitable in the `YCbCr` color space, whereby the chroma channels `Cb` and `Cr` are only stored at a reduced resolution, while the luminance `Y` channel is retained. The common $X : Y : Z$ notation is a historical remnant from analog television [58]:

- X indicates the horizontal luma sampling reference.
- Y indicates the `Cb`/`Cr` horizontal sampling.
- Z Either equal to Y, or, if $Z = 0$, indicates a $2 : 1$ vertical subsampling.

The Z component originally indicated the horizontal Cr sampling rate, as vertical subsampling was not intended in times of line-based cathode ray tube displays[25].

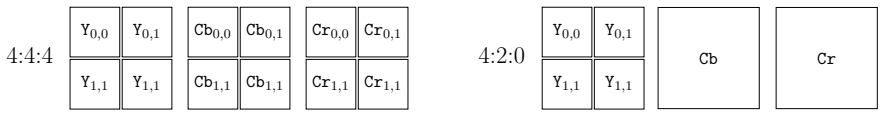

Figure 22: `YCbCr` pixels in full and chroma subsampled representations.

[24] E.g. equations (71) in [20].
[25] Further context on this notation in [58].

4.6.1. Entropy Coding

So far, the discussed methods create representations of a given input video that relied largely on quantized transform coefficients and predictor parameterizations. These are subsequently *entropy coded* in order to create an efficient bitstream representation. This class of lossless compression methods is used to assign a specific bit pattern to each input *symbol*, in our case elements of the set of actually occurring DCT coefficients and predictor parameters in the compressed video.

Since its introduction in the same fundamental paper as rate-distortion theory, the Shannon-Fano entropy code [17, 64] has been improved and extended. Huffman coding [28] guarantees minimal redundancy, which is not always achieved by the original algorithm. This, in turn, has been superseded by arithmetic coding [93] and geared specifically towards video applications, as in *context-based adaptive binary arithmetic coding* (CABAC), which is utilized in H264 and HEVC [69]. Modern entropy coders allow frequent updates the relevant alphabet and parallelization as in Daala's *Multi-Symbol Entropy Coder* [76] and circumvent licensing problems that affected arithmetic coding.

4.7. Computational Requirements as a Limiting Factor

This section provided an overview of contemporary video coding techniques. As per the thesis's title, we're interested in the special case of real time streams, which imposes a strict timeliness requirement on the encoding process. We now consider the total CPU

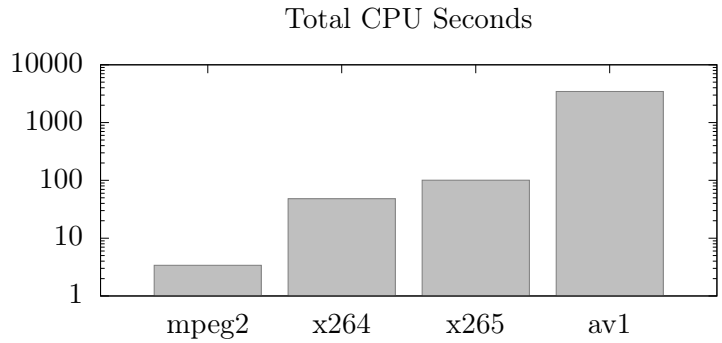

Figure 23: Encoding times for src #04, VQEG JEG dataset.

seconds required to encode the same-ten second FHD videos that were investigated in Figure 20 with sensible default codec parameters[26]. Mind the logarithmic scale of the plot. Though this is no solid statistical comparison of encoder performances, it shows an undeniable trend that the more sophisticated a codec, the more expensive its application gets. While MPEG2 had a throughput of 73 frames per second, AV1 merely managed 0.072 fps, which is far below real time requirements. Some aspects, e.g., predictions from future frames, are inapplicable if the required buffering would violate timing constraints. We will now investigate foveation as an alternative approach to data reduction in realtime streaming applications.

[26]Details on the parameterizations are given in Appendix A.

5. Adaptive Coding and Foveation

The key idea to improve video coding for real time streaming applications is to exploit the relative acuity of the human eye through a *foveated* coding scheme. After establishing the required context in terms of previous work, we present a system that implements foveation by providing a combining a modern video codec with feedback from an eye-tracker, and evaluate its performance in terms of perceptual quality versus bitrate by means of a lab experiment.

A Classification Spatially adapting visual media to improve coding performance is no novelty in the literature. We distinguish between *static* and *dynamic* approaches. In the former, a medium is processed once, and presented to arbitrarily many observers at an arbitrary point of time in the future, whereas dynamic approaches steadily adapt the medium to the current requirements while one or multiple observer are watching.

5.0.1. Static Adaptive Coding

Wang et al. proposed an algorithm for wavelet-based image compression in a paper titled "Embedded Foveation Image Coding" [84]. The nomenclature is partially ambiguous in this domain, especially when reviewing older literature. We interpret *foveation* in the sense that it requires real time adaption according to an observer's gaze, while older contributions are often solely capable of static region of interest (ROI) coding. The algorithm allows variable bitrate coding by truncating the bitstream at any position. By design, it thereby preserves more details in the *relevant* regions of the encoded image. The authors mention the problem of determining points of interest and the two general classes of solutions: The first option is to use interactive methods and probe users in experiments, either through eye-tracking or by self-reported saliency.

The alternative is algorithmic saliency estimation, which constitutes an entire field of research on its own [13]. Despite them being aware of the problem, it remains unclear how the saliency data was generated in this experiment. Results are given as printed examples of selected images together with their bitrates, alongside the author's own subjective impression of the visual appeal for two wavelet algorithms.

In the same year, other members of the LIVE lab published an approach to "Foveated Video Compression with Optimal Rate Control" [45]. The paper discusses a rate-distortion optimization for ROI video coding, which is implemented by adjusting the blockwise quantization granularity in a H.263 encoder. To this end of optimization, a *foveal* signal to noise ratio (FSNR), based on a locally weighted mean square error, is introduced. The idea of real time foveated video coding is presented, but not implemented. By modern standards, the test videos with a resolution of 288×352 pixels are far outdated, and the frame rates of as low as 15 fps would be deemed unacceptable in most applications.

The same group proposed "Foveation Scalable Video Coding with Automatic Fixation Selection" [86], which extends the work on fractal images compression to videos, and reiterates previous contributions on DCT based video coding. Point of interest selection is realized through a rudimentary face detector.

Lee et al. [43] present an approach to video coding with an audio-visual notion of attention. Regions of interest are defined through the canonical correlation [72] of visual and audio features, resulting in a so-called *cross-modal energy* formulation. The spatial and temporal location of events in both domains, e.g., the trajectory of a noisy vehicle driving by, are assumed to be of interest to an observer. This attention model relies on high-level recognition and association of mixed stimuli.

In their implementation, video frames are sliced along H.264 macroblock boundaries, which are then quantized according to their distance to the most relevant block. The authors point out that this approach is capable of improving coding efficiency without significant subjective quality degradations, especially in high-resolution sequences.

A limitation is imposed by content variations, as distinct visual and acoustic stimuli guide saliency in ways that are hard to predict at a low level: The authors state that participants regarded impairments on faces outside of the region of interest as much more disturbing than impairments on other background regions. The subjective effects are further detailed in an additional lab study [44], with a total of fifteen participants and a dataset of 12 videos. In the second experiment, a majority of observers preferred the traditionally encoded video with uniformly distributed impairments over the ROI coding. As the background distortions were allegedly severe, this does not contradict the validity of the approach as a whole. The finding rather suggests that either an observer's tolerance for distortions in the background is lower than the authors expected, or that their definition of regions of interest needs to be improved.

Boccignone et al. [10] present a Bayesian approach that combines high-level face detection with low-level visual cues, such as abrupt events, to steer the region of interest in videos. Instead of implementing or modifying a video codec, the authors chose to low-pass filter video frames according to their saliency prediction with respect to the maximal frequency, which is detectable by the human eye. The resulting video is passed to a DCT based MPEG-4 baseline encoder, which is then able to represent in terms of fewer high-frequency components as opposed to the original frames.

Videos were assessed using a single stimulus, absolute category rating scheme as well as in a pairwise comparison experiment according to the ITU-T Degradation Category Rating [35] protocol. Comparisons between videos compressed on the basis of either only low or high-level features show a clear preference for the latter approach in terms of higher MOS ratings. The authors conducted one of the larger-scale experiments in this domain with 200 participants in total, however, the content is limited to 10 source videos, out of which only three are reported on in detail. For these, under presumably ideal conditions for their algorithm, they claim that bitrate savings of up to 36.2% in comparison to the original are possible without any reduction of perceptual quality.

Our group experimented with ROI coding for JPEG compression by adapting the quantization strength for each 8×8 pixel block individually [26]. A crowd study resulted in an average 11% benefit in terms of bits per pixel over standard JPEG at the same perceptual quality. The saliency data for this experiment was taken from the dataset presented in [40]. This was an early successful experience with ROI coding that motivated further research, including this work.

Static ROI coding approaches generally suffer from the limitation that the encoded images or videos can not be adapted to unexpected viewing behavior. This limits the amount of degradations that can be introduced in less relevant regions and thus the bitrate savings that can be achieved, if user satisfaction has to be assured as well outside of the region of interest.

5.0.2. Dynamic Adaptive Coding

Girod [21] discussed the perceptual impact of eye movements and the potential benefits of exploiting its peculiarities for video coding already in the late eighties. He suggested a hypothetical system for *coder control by eye movements*, with an eye tracker on the receiver side.

$$t = t_{\text{eyetracker}} + 2t_{\text{transmission}} + t_{\text{encoding}}$$

At the time, he deemed the "usefulness of the approach limited" due to the delay t between an eye movement and an image update: Technical advances of the last 30 years, improving all delay constituents, made the idea practically feasible today.

Bulla et al. present a system for foveated video conferencing [12], which utilizes a Viola-Jones object detector [78] to identify and track faces on the encoder side, thus circumventing the issue altogether. They showcase the PSNR of the luma plane for the rectangular region of interest, and report average bitrate savings of around 50% for a "conventional" setting, according to the author's own perceptual evaluation.

Arndt and Antons [7] experimented with foveated coding. Their study provides insight into the effects of radius choices for the quality region as well as the influence of the quality difference between fore- and background. It remains partially unclear how their system is implemented. The sequence diagram in Figure 1 describes the process as if they were continuously streaming a background video in low quality, on top of which a high-quality crop is inserted at the current fixation point. It is not specified how this cropped is compressed for transmission over the network. The original source of this diagram could possibly provide more context, but the cited thesis is not publicly available.

The work of Illahi et al. [31] is closely related to ours. They present a streaming framework that utilizes foveation in the context of cloud gaming and evaluate its performance for different genres of video games. We aim to investigate a similar approach, but for the live streaming of natural videos.

Further Literature Foveation is currently popular in the virtual reality community [57]. The approach generally benefits from increasing screen sizes, as these allow to present a larger area in the peripheral regions of an observer's field of view, which is taken to the extreme in VR applications. However, the focus is not on video compression, but rather on the optimization of the costly rendering process by reducing the quality in the peripheral field of view, while maintaining a high level of user satisfaction.

A certain level of interest from the industry is indicated by numerous patents on similar approaches, as held for example by Google [83, 98], whereby the topic is arguably relevant outside of academia.

6. FFoveated: A Framework for Foveated Video Coding

We implemented a software framework for foveated video coding to approach research and experimentation on this topic in a structured and streamlined manner. It facilitates rapid prototyping on the entire pipeline, from gaze information processing and the usage of different codec configurations to the setup and execution of entire user studies. Additionally, this section provides insight into the practical aspects of video application development.

6.1. Usecase and Requirements

From a coarse perspective, the frameworks' purpose is to encode and display a given video in a foveated fashion. This entails multiple design choices on a more detailed level. The major requirements are summarized in the following list:

i) **Video sources:** Our framework shall permit various kinds of video input formats.

ii) **Encoders:** Exchanging encoders shall be possible through a well-defined interface.

iii) **Gaze data:** It shall be possible to incorporate updates on eye fixations in real time.

iv) **Parameterization:** Changes of e.g., the foveal radius shall be easy to apply.

v) **Rendering:** A video player shall be implemented for fine control over latencies.

vi) **Interaction:** Participants shall be able to react to stimulus presentations.

Some of these items are recurring problems in video engineering or require extensive library support to be solved efficiently. We hence decided to build upon the `FFmpeg` project's [19] `libav*` library collection, thus the ajar name `FFoveated`. The goal of this implementation is not the development of a novel codec itself, but the extension, utilization, and evaluation of existing algorithms for our purposes.

6.2. Overview

The required functionality can be subdivided into a client-server architecture, as illustrated below in Figure 24. On the left side, an input medium in the form of a *file* on disk is handled through a *source decoder*, which is necessary for reproducibility when utilizing video databases in repeated experiments. The alternative is to pass video input directly from a *live source*, such as a webcam, as would be the case in practice.

The *foveated encoder* block represents the core component of the system. It emits a compressed video stream that is sent to the client-side for decoding and playback. For the purpose of our lab studies, all of these components were running on the same host, but could be decoupled at this point for actual network streaming.

The dashed arrows around the *observer* block shall indicate data flow in the form of user interaction. While a stimulus is being presented, an *eye tracker* constantly sends feedback about the current fixation point back to the encoder, which then incorporates these updates in future frames. At last, the rightmost block represents an interactive

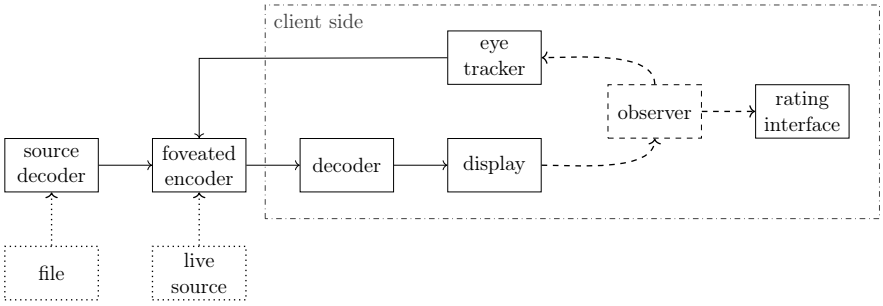

Figure 24: Schematic block diagram of FFoveated.

rating interface. FFoveated itself is generally agnostic to the assessment methodology. For example those presented in Section 2.2 can be implemented with little effort.

6.3. Implementation Details

The `libavcodec` API differentiates strictly between compressed and uncompressed data and provides two structs to represent instances of either. In short, these contain:

- `AVPacket`[27]: A buffer of compressed data and with little metadata for decoding.
- `AVFrame`[28]: Raw audio or video data with extensive metadata for e.g. presentation.

Starting from a *file* in the pipeline above, there are implicit steps involved to before obtaining decodable `AVPacket`s.

6.3.1. Container Formats and Multiplexing

Videos are commonly stored in *containers*, which mainly serve the purpose of multiplexing them with audio tracks, subtitles, and metadata for storage within a single file. Prevalent examples are the file format of MPEG-4, usually denoted with a `.mp4` extension, which is defined in Part 14 of the corresponding ISO standard [33], and the open-source `.mkv` Matroška [71] media container. By adding an index structure around the media content, they also provide functionality for fast seeking and a certain degree of error resilience in case of corrupted payload data. FFmpeg implements multiplexing and demultiplexing of various filetypes through its `libavformat` library.

Uncompressed video data is customary in quality-related databases to avoid artifacts. While expensive, storage is feasible for practically relevant durations, but the bitrate of uncompressed UHD videos exceeds the transfer speeds of current storage devices.

The solution in FFoveated is to utilize input videos with visually lossless compression, which are then decoded in main memory. Demultiplexing of packets from a containerized input file is implicitly done before passing its contents to the source decoder in Figure 24. When reading raw data directly from disk, these two steps are replaced by merely mapping bytes from the input file to the appropriate `AVFrame` buffers.

[27] Defined in `libavutil/frame.h`
[28] Defined in `libavcodec/avcodec.h`

6.3.2. The Decoding-Encoding-Decoding Cycle

Codec interfaces vary in terms of initialization and API peculiarities. However, as their core functionality is similar, FFmpeg provides a unified API covering all concrete implementations through a collection of wrappers in `libavcodec`. Communication between the client implementation and codec instances is handled mainly through the following functions[29]:

```
avcodec_send_packet(...)      avcodec_receive_packet(...)
avcodec_send_frame(...)       avcodec_receive_frame(...)
```

These are asynchronous and do not block execution upon the following conditions:

- The encoder (decoder) can not provide a packet (frame) and needs more data.
- The encoder (decoder) buffer is full and can not store another frame (packet).

Instead, they return a value indicating the current status. The interplay of source decoder, foveated encoder and presentation decoder when requesting an `AVFrame` for displaying from the latter is depicted as a finite automaton below in Figure 25.

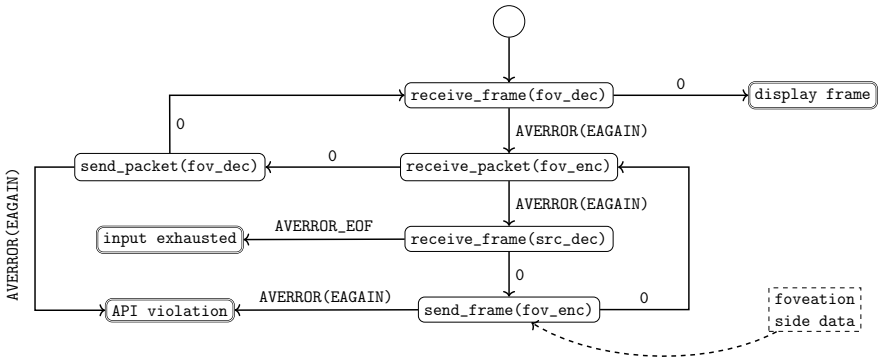

Figure 25: DED cycle, function names omit the `avcodec-` prefix.

A single-threaded application following this model, though conceptually appealing, turned out inadequate in early versions of the FFoveated implementation. The worst-case scenario occurs right on the first frame, as the request for data *fails* on all three codec instances. This situation requires them to be supplied with each successor's output sequentially, starting with disk reads at the very beginning.

Multithreading To meet the strict real-time requirements, FFoveated divides the workload of i) reading and demultiplexing ii) decoding the source video iii) foveated encoding iv) decoding the foveated stream for presentation and v) rendering and user interaction by spawning an individual thread for each task. Synchronization between these is implemented through blocking FIFO buffers as depicted in Figure 26.

[29] Defined in `libavcodec/avcodec.h`

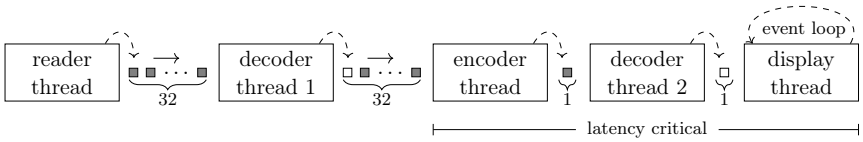

Figure 26: Inter-thread buffering in FFoveated.

The first two stages of the pipeline are equipped with buffers capable of fitting 32 `AVpackets` or `AVframes`, which get filled when opening a new input file before rendering is started. They're chosen to be large enough to resist, e.g., small delays when reading the input video from disk. The latency-critical section spans the range from encoding a new frame according to the observer's current gaze to its presentation on the screen. In order to keep synchronization tight while allowing parallelism in foveated coding, we connect the threads through blocking buffers with a capacity of only one element each.

6.4. Passing Foveation Data to Encoders

Foveation requires codec-specific implementations in `libavcodec`, which thus needs to be equipped with a possibility to allow fixation updates for each frame. Such *optional* data, which is not generally required by video codecs operations, is handled through the `AVFrameSideData` struct and associated functions. These provide a thin wrapper to enqueue pointers to such structs in an array referenced by the `side_data` member of an `AVFrame` instance.

```
typedef struct AVFrameSideData {
    enum AVFrameSideDataType type;
    uint8_t *data;
    int      size;
    AVDictionary *metadata;
    AVBufferRef *buf;
} AVFrameSideData;
```

Figure 27: Defined in `libavutil/frame.h`

Upon receiving a frame, an encoder instance decides based on the `type` field how to adequately handle provided side data. We extended the corresponding `enum` with a new entry to uniquely label foveation data:

```
enum AVFrameSideDataType {
  [...],
   AV_FRAME_DATA_FOVEATION_DESCRIPTOR
}
```

Figure 28: `libavutil/frame.h`

Should `FFoveaded` be linked against unpatched `libav*` shared objects, the encoder wrappers will ignore and free side data of unknown type, which preserves backward compatibility by providing unfoveated encoding.

Introducing a new `AVFrameSideDataType` is the only codec-independent patch required in `FFmpeg`. Further changes on specific wrapper implementations in `libavcodec` tangent solely that codec and are independent of each other.

```
typedef struct AVFrameSideData {
    enum AVFrameSideDataType type;
    uint8_t *data;
    int      size;
    AVDictionary *metadata;
    AVBufferRef *buf;
} AVFrameSideData;
```

Figure 29: `libavutil/frame.h`

Readers familiar with the `FFmpeg` project will recognize an analogy in purpose regarding the `AV_FRAME_DATA_QP_TABLE_DATA` and `AV_FRAME_DATA_REGIONS_OF_INTEREST` side data types. The former is deprecated and not part of `libavutil`'s default API anymore; the latter is meant to carry `AVRegionOfInterest` structs, which define rectangular bounding boxes within a frame and quality offset.

We chose not to utilize these types for preserving the original intent behind the API, and to delegate decisions on concrete foveation implementations to the respective encoders. This avoids the necessity to deal with disparities between codecs, e.g., different macroblock sizes, at the cost of minor code duplications should codec interfaces coincide. Each side data instance is indirectly referenced through an `AVFrameSideData` struct, as defined above in Figure 29. The data `FFoveated` passes on to encoder instances is a heap-allocated array of four `float`s, containing the following information:

- x-coordinate, relative to the frame width, which corresponds to $[0, 1]$.
- y-coordinate, relative to the frame height, which corresponds to $[0, 1]$.
- σ: standard deviation of the Gaussian to derive the quality offset map.
- δ: maximal quality offset.

6.5. Rendering and Interaction

The SDL [63] library is used to display the `AVFrames` that result from decoding the foveated video in the second rightmost block in Figure 26. Besides access to graphics primitives, it allows platform-independent event handling and interaction.

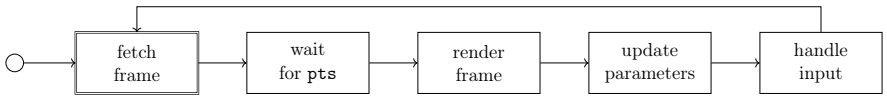

Figure 30: FFoveated: Event loop.

Timing is managed in the main event loop in the display thread. It fetches a new frame from its input queue and renders it according to the frame's *presentation timestamp*. Afterward, in our experimental setup, δ is adjusted, and potential user input is handled. If the input queue signals an end of file, the control flow breaks out of the loop again.

6.6. Foveated Video Coding Using x264

We now investigate how the codec-specific components of `FFoveated` can be implemented based on the widely-used open-source encoder `x264` [94]. It conforms to the H.264 [91] standard, which is supported on virtually any modern device and belongs to the class of hybrid, block-based transform-domain video codecs introduced in Section 4.2. The goal is to enforce a compression scheme corresponding to the spatial structure of the fovea.

Rate control The mechanism responsible for maximizing video quality during encoding under a given set of constraints is called *rate control* [61]. These constraints can affect various aspects of the encoding process. Typical examples include limits on the (average) bitrate, the buffer size available to the encoder or the latency, after which a given input frame has to be returned in compressed form. In H.264 [50] this involves choices on

a) group-of-picture level, affecting all frames included in that GOP

b) frame level, affecting all macroblocks within that frame

c) macroblock level, affecting all transform blocks within that macroblock

Rate control determines values for the quantization parameter (qp) at each level, which ultimately determines the granularity of the DCT coefficient quantization. Depending on the situation, `x264` offers a number of different *modes* [48, 50, 60, 61], through which the encoder determines these choices:

Mode		Description
constant qp	(CQP)	qp depends solely on frametype
constant rate-factor	(CRF)	additionally incorporates motion compensation
average bitrate	(ABR)	optimization towards target bitrate, in 1 or 2 passes
constant bitrate	(CBR)	potentially wasteful

Table 1: Rate control modes in `x264`.

A categorization for the first three entries in Table 1 is *variable bitrate* (VBR) modes. We now briefly discuss these options while eliminating candidates that are suboptimal for live streaming:

CQP mostly relevant in research, e.g., when investigating details the transform domain coefficient quantization itself, but underperforms in practical applications as the bitrate is very dependant on the content. CBR can be wasteful, as it includes superfluous high-frequency information in easy-to-encode frames to reach its target, instead of falling below the desired bitrate. This is contrary to the goal of achieving low bitrates in live streaming, but at least guarantees not to exceed them. The `x264` developers generally discourage users from applying the ABR model, and for practical purposes also from applying CQP [48].

The constant rate factor approach modulates a nominal qp value through a heuristic, which enforces higher quantizer values for macroblocks with large motion vector magnitudes [50, 95]. This results in coarser quantization steps and fewer details, ultimately exploiting the peculiarity of the human visual system to be less sensitive to details in fast-moving scenes. Conversely, slow-moving scenes receive a higher bit budget, as the eye is capable of discerning finer details in these.

Offset Function The quantization parameter qp in x264 ranges from 0 to 51. A higher qp results in a coarser quantization, thus a lower bitrate and reduced visual quality. We incorporate eye-tracking data through an offset $\bar{q}$ to qp.

$$\bar{q}(x, y) = \delta\left(1 - \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right)\right) \tag{2}$$

This offset can be calculated for a macroblock at position (x, y) given a fixation point (x_0, y_0), a standard deviation σ, and a scale factor δ. A two dimensional Gaussian is a natural choice to model the spatial quality distribution given the circular shape of the fovea centralis [81]. As depicted in Fig. 31, our function of choice does not inflict a quantization penalty on the fixation point at (x_0, y_0), where $\bar{q} = 0$.

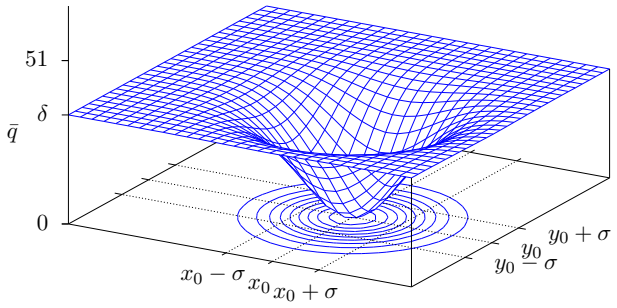

Figure 31: Spatial distribution of the quality offset function $\bar{q}$.

We chose to set σ to 2.5° of visual angle. This is based on the characteristics of the retina, but merely an educated approximation of an optimal choice, as both the relationship between the qp value and its perceptual impact, as well as the dependency on the concrete video content, are hard to control and optimize for. Under these premises, the question is how to choose δ such that the bitrate is minimized without introducing overly-disturbing artifacts in the peripheral regions.

x264 Encoder Configuration - Preset, Tune In x264, so-called *preset*s provide generally useful collections of settings that work well for broad application scenarios. These can be further tailored towards specific usecases by choosing a *tune* flag and overridden in terms of individual parameter choices through *user settings*. For our real-time streaming scenario, we chose the *ultrafast* preset and enabled the *zerolatency* tuning option. Descriptions of these and details on their consequences are available in the x264 documentation [94] and would be too verbose for this thesis.

As the names suggest, this configuration is suitable for real-time encoding, which disables many intense operations at the cost of a higher bitrate. The most noteworthy consequence is that B-frames are disabled, as predictions relative to future frames are impossible when using causal signal source without a buffer.

User Settings Specific adaptations include setting the *GOP size limit* to only three frames, which is short in comparison to the default of 250. This enforces frequent I-frames, which prevents quantization errors from accumulating over time. Furthermore, it serves as a mitigation to packet-loss-induced stalling during network streaming. Ultimately, we have to override the adaptive quantization mode defined in the preset by setting the *aq-mode* option to 1, which not only enables a variance-based heuristic, but also allows us to add our offset as defined by the function $\bar{q}$. Our implementation does not break the H.264 specification, thus the videos can be displayed with a standard-conforming player.

7. Perceptual Evaluation and Performance Quantification

7.1. Experimental Setup

We conducted a user study in a controlled lab environment to evaluate the performance of our implementation. This section describes the experimental setup, data source, acquisition procedure, and the structure of the resulting data.

7.2. Lab and Hardware

We utilized a color-calibrated HP Z31x UHD screen in a room that was illuminated solely by artificial light sources, in order to avoid inconsistencies caused by daylight changes. Gaze data was gathered using an SMI ka Red250mobile eye-tracker. The hardware device was mounted on a stand below the screen. This eye-tracker allows participants to freely move their head within a certain range as they observe the presented stimuli, while unobtrusively capturing fixation data.

Driver compatibility issues forced us to externalize the interaction with the eye tracker dongle to a notebook provided by the original vendor. Communication between that machine and the one running FFoveated was accomplished within the SMI library itself. It provides a server-client model, in which the driver application opens a listening TCP socket, providing the client with the means to fetch updates. The two machines were connected directly through a gigabit network, which introduces only negligible latencies relative to the video framerate.

7.3. Data Source

We utilized the source files taken from the VQEG JEG Hybrid [9] dataset in this study. These 10 videos of 10 seconds each have been recorded at a resolution of 1920×1080 pixels at a framerate of 25 fps. This dataset was originally intended for the development of video quality assessment algorithms and thus provides a suitable test set for our application. Foremost, the source files exhibit sufficiently high quality. This is required for being able to attribute distortions to FFoveated, which would be impossible if the source files were already contaminated with such.

The VQEG dataset contains diverse scene types that might affect gaze behavior in various ways, exposing the benefits and shortcomings of our approach. It includes content ranging from cartoons, with mostly flat, monochrome areas through sports videos with fast action in small regions to clips with a high dynamic range, scene cuts, and camera movements. An overview is given below in Figure 32.

Figure 32: Screenshots sampled from the VQEG JEG dataset.

The videos were presented *centered* on the screen in their native full HD resolution.

Stimulus Presentation and Quality Decay Lab studies are costly and labor-intensive. The premises and availability of hardware equipment restricted this experiment to a single participant at a time, which had to be instructed and supervised individually. We thus aimed to keep the assessment strategy as effective and economical as possible, to obtain the most meaningful results for the invested effort.

The experimental procedure is based on the concept of a just noticeable distortion, as presented in Section 2.2.4, with an added temporal component that involves adapting the presented stimuli over time. After an introduction to the task and the calibration of the eye-tracker, we present each of the 10 source videos contained in the VQEG-JEG dataset for a total of 10 *repetitions*. During each *repetition*, the maximal *qp* offset δ is increased by $\delta_\uparrow$ every Δf frames. Initially, at the beginning of the first *repetition* of each *source*, δ is set to 0, ergo no foveation is applied. Increasing δ over time leads to a bitrate reduction in the periphery.

rep	1	2	3	4	5	6	7	8	9	10
$\delta_\uparrow$	10	5	3	2	2	1	1	1	1	1
$\delta_\downarrow$	25	20	17	15	10	8	8	5	5	5
Δf	25	25	25	25	25	50	50	50	50	50

Table 2: Parameterization of the assessment procedure.

Interaction and User Feedback The sole *interaction* a participant can perform in this experiment is a button press that indicates that he or she perceived a visual distortion in the currently displayed video. Following such an event, the *repetition* is stopped, and a neutral black screen is displayed for one second. This serves as feedback to the observer and is standard practice to establish a common onset in between stimulus presentations. If there are still repetitions left to present for the current source, δ is reduced by $\delta_\downarrow$; otherwise, the program advances to the next source or exits at the end of the experiment.

The parameter δ is modified and carried on in between repetitions of the same source video, while $\delta_\uparrow, \delta_\downarrow$, and Δf are updated according to Table 2. If a participant does not interact during a repetition, δ is kept and modified according to the parameters of the next repetition. This approach enforces a rapid introduction of degradations during the first few repetitions of each source video. As $\delta_\uparrow$ and Δf are adapted over time, the participants can spend attention to increasingly minute distortions.

The idea is to approach the individual δ quickly in the beginning, and then ever-more slowly, in order to obtain precise results. We zero in on the participant's individual point of just noticeable distortion for that particular video. A benefit of this repetitive linear search is that we can display various levels of quality within a single repetition while performing a relatively well-defined assessment task that does not require participants to have an extensive mental model of a video quality scale readily available. During our experiments, we gathered the fixation points and the δ values that were used to encode each frame and as well as the information on participant interactions.

7.4. Results and Discussion

Our lab study was conducted with 10 participants, totaling at a number of 1000 displayed video repetitions, in which 734 interactions were observed. This section provides insight into the collected data and compares the results to related works in the literature, after exemplarily visualizing the foveated video that our method produces.

Figure 33: Frame sampled from participant #4, source #2, repetition #4.

The scene in Figure 33 contains rapid action confined to a relatively small region on the left side of the court, with darkened and indistinct surroundings on the stadium ranks. These are ideal conditions for the application of foveated video, as the observer focusses only on a small region, unaware of the distortions in the periphery.

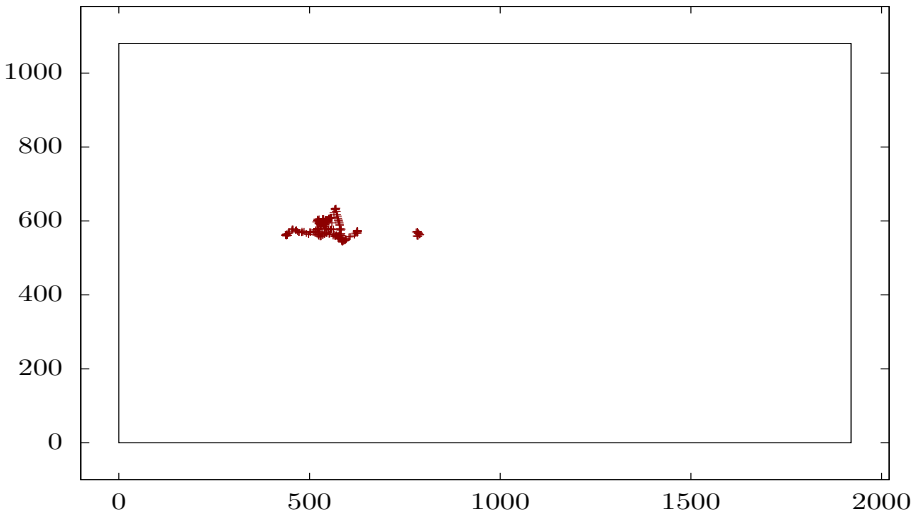

Figure 34: Pixel fixation points of the whole repetition, corresponding to Figure 33.

7.4.1. Average Bitrate Savings

Tables 3 and 4 contain the average bitrate savings at the 25% and 10% point of just noticeable distortion, which was calculated by taking all observed *interactions* into account. More precisely, the reported JND δ value for each source video is calculated as the corresponding percentile of all the δ values used in the compression at the time of an interaction for that particular source video over all participants and repetitions.

We compare the video bitrates using *all* recorded gaze paths. Videos based on partial paths that were recorded up to the n-th frame and discontinued due to a participant interaction are compared with videos that were also cut at the n-th frame. The only difference is that foveation was disabled by setting $\delta = 0$; the remaining codec parameters were equal. This allows utilizing all 1000 runs for maximally diverse gaze patterns in the comparison, as these may have an impact on the performance of the compression.

src	*int.*	JND δ	br_0 (kbit/s)	br_{JND} (kbit/s)	Reduction
#1	72	20	6411.9	2167.7	66.19%
#2	75	25	4406.0	1381.7	68.64%
#3	80	18	5082.2	1564.0	69.22%
#4	70	20	8181.6	2663.3	67.44%
#5	75	20	8749.1	1945.5	77.76%
#6	74	18.25	12931.0	4619.7	64.27%
#7	72	17.25	3926.9	1303.7	66.80%
#8	75	14.5	5486.3	1545.8	71.82%
#9	71	19	8721.7	2700.0	69.05%
#10	70	20	5900.0	1909.7	67.63%
avg	73.5	19.2	6979.7	2180.1	68.88%

Table 3: Average bitrate savings at the 25% JND.

The columns in both Table 3 and 4 contain the following information:

- *src*: Source file number, as displayed in Figure 32.
- JND δ: Maximal *qp* offset at the point of just noticeable distortion.
- br_0, br_{JND}: Bitrates without foveation ($\delta = 0$) and with foveation at the JND.
- Saving: The percentage of data discarded through foveation at the JND.

Although the 25% JND is widely used [47], this choice likely overestimates the performance of our method due to high δ values in early repetitions. We therefore also report savings at the 10% JND, at which only 10% of the participants expressed that noticeable distortions were present. With this stricter, less distortion-forgiving definition, we still achieve an *average bitrate saving of* 62.76% in comparison to the unfoveated baseline.

src	JND δ	br_0 (kbit/s)	br_{JND} (kbit/s)	Reduction
#1	18	6411.9	2343.4	63.45%
#2	22	4406.0	1484.0	66.41%
#3	14.9	5082.2	2207.1	56.57%
#4	18	8181.6	2861.4	65.02%
#5	15	8749.1	2528.4	71.10%
#6	12	12931.0	6066.3	54.08%
#7	14	3926.9	1475.2	62.43%
#8	10.4	5486.3	1885.7	65.62%
#9	14	8721.7	3456.3	60.37%
#10	16	5900.0	2208.8	62.56%
avg	15.43	6979.7	2651.7	62.76%

Table 4: Average bitrate savings at the 10% JND.

7.4.2. Interaction Events

To get a better understanding of the *quality decay* over time, we plot the interaction events of all possibly displayed frames for all 10 repetitions in Fig. 35. Each vertical bar indicates the beginning of a new repetition, i.e., a restart of the video sequence. Colors encode unique participants. An ×-marker is placed for each *interaction*. Upward and downward triangles denote the δ-range per user, from the lowest to the highest possible value within that repetition. The grey lines indicate the .25 and the .1 JND for this source over all participants and repetitions.

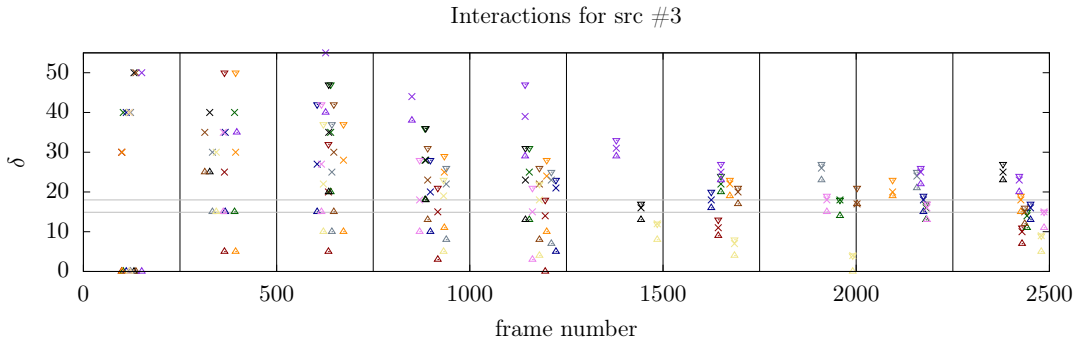

Figure 35: Interactions for source #3.

The rapid initial increases in δ, likely in combination with a certain reaction time required by the participants to recognize distortions, leads to a bias due to which the initial participant interaction reports are over-exaggerated. However, throughout the repetitions, the reported δ values shift towards lower, more plausible values, and the participants agree up to a certain difference in subjective sensitivity.

7.5. Performance Comparison to Existing Works

Direct comparison to the literature is difficult, as the anchoring of the point of comparison via the JND is novel, and the field has not established common benchmarking grounds. However, the results of the existing studies paint a similar picture as our results. We summarize how the reported numbers were selected from the results reported in the respective papers:

Contribution	Comparative Bitrate Saving	Description
Arndt and Antons [7]	ca. 63%	A study that reports mean opinion scores of pairs of foveation radii and background quantization parameters. We report the bitrate reduction according to Table 1 for the combination of a 190px radius with a qp of 32, which is the first setting with a noticeable decrease in quality, according to Fig. 3a.
Illahi et al. [31, 32]	ca. 42%	User study on foveated video for cloud-rendered video game streaming. We report the bitrate reduction according to Fig. 13, when comparing the bitrates for QOmax=0 with QOmax=8. The domain shift towards interactive gaming and the compression of artificial renderings instead of natural videos likely affects the user-reported QoE in comparison to our study.
Our work:	62.76% at the 10% JND.	

7.6. Outlook and Future Work

The surprisingly good experimental results motivate further work on research into foveated real-time video coding. A follow-up task would be to verify our findings in larger-scale studies. Higher resolutions and screen sizes should, as mentioned earlier, further improve bitrate gains, since it is possible to reduce the quality in a larger share of each frame. We expect the limiting factor to be the increase in encoding time; this limitation can be alleviated with task-specific hardware implementations.

As the human eye is more sensitive to abrupt contrast changes and motion in peripheral vision [81], we assume that the discernibility of changes in coarsely quantized blocks can be reduced by simple post-processing such as blurring. There is room for improvement regarding the reaction speed when longer saccades occur; the problem might be mitigated by higher frame rates.

Eye Tracking for User Filtering The gaze patterns depicted in Figure 36 are consistent with the presumable region of interest on the back and forth moving cheetah in src #4. Another participant's gaze for the same source video is displayed in Figure 37. In this scatterplot, larger saccades occur much more frequently, and they often terminate in background regions, leading to fixation groups in arguably uninteresting regions.

We attribute this to an ambitious participant, who is eagerly trying to spot distortions in the outer regions, analogous to the behavior presented in [5]. Depending on the assessment task, this may be undesirable, when more natural user behavior is expected. To our knowledge, eye tracking has not been used for participant post-filtering in image or video quality assessment databases.

Prediction of Just Noticeable Distortions Foveated coding allows us to make additional assumptions about the location of visible distortions, as the quality decreases radially around the current fixation point. Comparing gaze paths between runs of the same participant as well as between participants shows surprisingly different patterns and none that allowed to establish an obvious connection between gaze and distortion visibility. The size of our data collection is currently a limiting factor that prevents the application of, e.g., machine learning based methods, which might be able to predict noticeable distortions from irregularities in the observer's gaze. This would be beneficial, as it allows us to steer the bitrate without requiring disruptive user interventions.

Approximative Eye-Tracking in Crowdsourcing Experiments Quality assessment studies are often realized through crowd sourcing, in order to gather numerous opinions on a large number of media items. The dependence on lab-studies is a downside of eye-tracking. As foveated coding arguably does not require pixel-level precision, it seems promising to carry out further testing through approximative eye-tracking, e.g., on the basis of [97]. This is directly applicable, e.g., in the scenario of webcam-based video telephony and entails all the practically relevant obstacles, which are easily overlooked in lab experiments.

We presented these ideas in a workshop paper on the subject [89].

7.7. Fixation Scatterplots

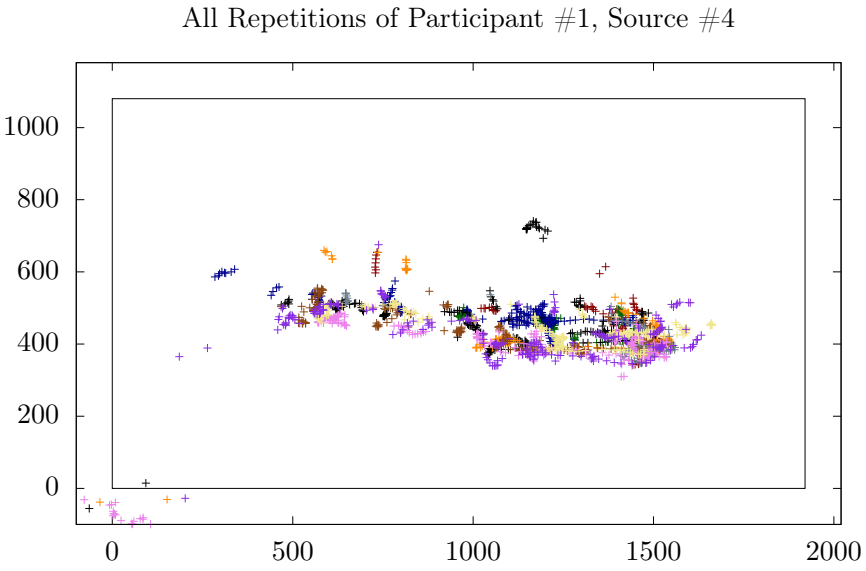

Figure 36: Expected behavior.

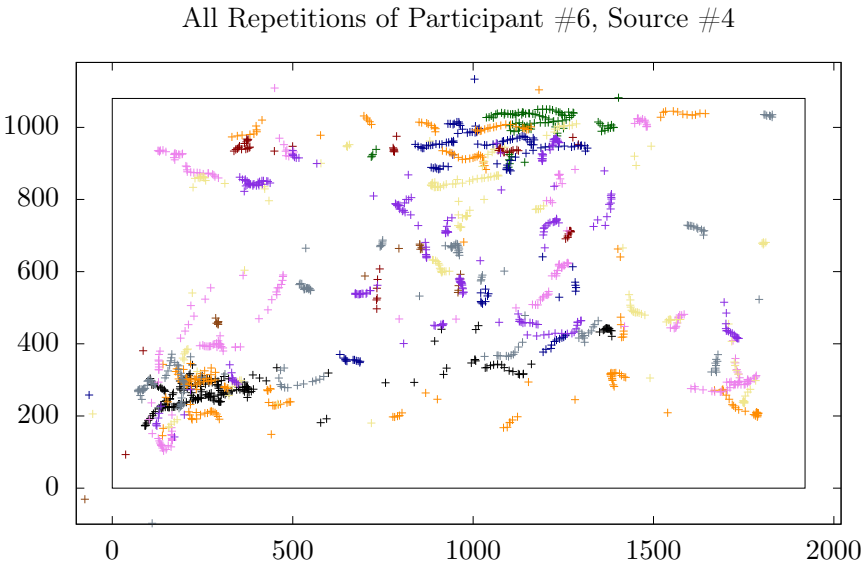

Figure 37: Explorative participant, eager to spot distortions.

8. Contributions

The main achievements in the course of this master's thesis were the following:

- `FFoveated`, a framework for prototyping and research on foveated video coding.
- A pilot study with 1000 data points on video presentations to identify the JND.

The results show that our implementation achieves bitrate savings that are comparable to the state of the art in the literature.

The Thurstonian scale reconstruction with multiple choices in Section 2.2.3 is an unpublished elaboration of an approach that came up during a group retreat, ca. 2015.[30]

8.1. Resulting Publications

Parts of this thesis were published previously in the following papers:

- Oliver Wiedemann et al. "Foveated Video Coding for Real-Time Streaming Applications". In: *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2020

- Oliver Wiedemann and Dietmar Saupe. "Gaze Data for Quality Assessment of Foveated Video". In: *ACM Symposium on Eye Tracking Research and Applications*. ACM. 2020

Open access and/or Author's version PDFs are available on my personal website[31].

[30] Credit for the idea is due to my supervisor Dietmar Saupe.
[31] `https://oliver-wiedemann.net`

Acknowledgments

I would like to express my gratitude to the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for financial support within the SFB-TRR 161, Project A05 (Project-ID 251654672).

I would like to thank my supervisors Prof. Dr. Dietmar Saupe and Prof. Dr. Bastian Goldlücke for always having a sympathetic ear and for their invaluable advice on the issues I encountered while working on this thesis. Due to Dietmar's support, I was able to get a conference paper published even before I held a Bachelor's degree, and more were to follow during my Master studies. I have a lot to be thankful for, and I'm looking forward to continue working with him in the years to come.

I thank my colleagues and co-supervisors Dr. Vlad Hosu and Dr. Hanhe Lin for the many fruitful discussions we had over a cup of tea. I would particularly like to thank Franz Hahn, who was not only up to discussing even the most unorthodox ideas, but also became a close friend of mine over the years.

I would like to thank my parents for their unconditional support, without which I would be nowhere near of where I am today.

Last but not least, I would like to express my most sincere thankfulness to my girlfriend, Lena. She always manages to cheer me up when I am frustrated and in self-doubt, and provided unfailing patience and encouragement through this thesis's writing.

List of Figures

Appendices

A. FFmpeg Sample Parameters

The video sequences utilized in Figures 20 and 23 have been created using the following `Archlinux` package versions:

- `ffmpeg version n4.2.3`
- `x264 3:0.159.r2999.296494a-1`
- `x265 3.4-1`
- `aom 2.0.0-1`

The FFmpeg input specification is identical for all encodings. To keep the notation concise, we provide an incomplete program call, followed by the specific output parameters that have to be appended to create either version:

```
ffmpeg -f rawvideo -vcodec rawvideo -s 1920x1080 -r 25 -pix_fmt yuv420p
    -i src04_1920x1080p25.yuv -c:v
```

The following suffixes create MPEG2, H.264 and HEVC compressed videos:

```
mpeg2video -b:v 10M mpeg2.mp4
```

```
libx264 x264.mp4
```

```
libx265 x265.mp4
```

The `libaom` AV1 encoder is at this point in time best utilized in a two-pass configuration, which requires separate FFmpeg invocations for the analysis and encoding runs:

```
libaom-av1 -strict experimental -b:v 2M -pass 1 -an -f mp4 /dev/null
```

```
libaom-av1 -strict experimental -b:v 2M -pass 2 av1.mp4
```

References

[1] Anna Abrahamsson. "Variance Adaptive Quantization and Adaptive Offset Selection in High Efficiency Video Coding". Examensarbete. Uppsala Universitet, 2016.

[2] Laura Acqualagna et al. "EEG-based classification of video quality perception using steady state visual evoked potentials (SSVEPs)". In: *Journal of neural engineering* 12.2 (2015).

[3] Nasir Ahmed. "How I came up with the discrete cosine transform". In: *Digital Signal Processing* 1.1 (1991), pp. 4–5.

[4] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. "Discrete cosine transform". In: *IEEE transactions on Computers* 100.1 (1974), pp. 90–93.

[5] Hani Alers, Lennart Bos, and Ingrid Heynderickx. "How the task of evaluating image quality influences viewing behavior". In: *2011 Third International Workshop on Quality of Multimedia Experience*. IEEE. 2011, pp. 167–172.

[6] Matthew Anderson et al. "Proposal for a standard default color space for the internet—srgb". In: *Color and imaging conference*. Vol. 1996. 1. Society for Imaging Science and Technology. 1996, pp. 238–245.

[7] Sebastian Arndt and Jan-Niklas Antons. "Enhancing video streaming using real-time gaze tracking". In: *in Proc. ISCA/DEGA Workshop on Perceptual Quality of Systems*. 2016.

[8] David Atchison and George Smith. *Optics of the human eye*. Butterworth Heinemann, 2000.

[9] Marcus Barkowsky et al. "Subjective experiment dataset for joint development of hybrid video quality measurement algorithms". In: *Third Workshop on Quality of Experience for Multimedia Content Sharing QoEMCS*. 2012, pp. 1–4.

[10] Giuseppe Boccignone et al. "Bayesian integration of face and low-level cues for foveated video coding". In: *IEEE Transactions on Circuits and Systems for Video Technology* 18.12 (2008), pp. 1727–1740.

[11] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986.

[12] Christopher Bulla, Christian Feldmann, and Martin Schink. "Region of interest encoding in video conference systems". In: *International Conferences on Advances in Multimedia*. 2013, pp. 119–124.

[13] Zoya Bylinskii et al. *MIT Saliency Benchmark*.

[14] Di Chen et al. "Multi-reference video coding using stillness detection". In: *Electronic Imaging* 2018.2 (2018), pp. 156–1.

[15] Yue Chen et al. "An overview of core coding tools in the AV1 video codec". In: *Picture Coding Symposium (PCS)*. IEEE. 2018, pp. 41–45.

[16] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2006.

[17] Robert M Fano. *The transmission of information*. Massachusetts Institute of Technology, Research Laboratory of Electronics, 1949.

[18] Gustav Theodor Fechner. *Elemente der Psychophysik*. Leipzig: Breitkopf und Härtel, 1860.

[19] *FFmpeg*. Accessed 10.02.2020. URL: `https://ffmpeg.org`.

[20] Adrian Ford and Alan Roberts. "Colour space conversions". In: *Westminster University, London* 1998 (1998), pp. 1–31.

[21] Bernd Girod. "Eye movements and coding of video sequences". In: *Visual Communications and Image Processing: Third in a Series*. Vol. 1001. International Society for Optics and Photonics. 1988, pp. 398–405.

[22] Bernd Girod. "What's wrong with mean-squared error?" In: *Digital images and human vision* (1993), pp. 207–220.

[23] Jingning Han, Yaowu Xu, and James Bankoski. "A dynamic motion vector referencing scheme for video coding". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 2032–2036.

[24] David M Hoffman and Dale Stolitzka. "A new standard method of subjective assessment of barely visible image artifacts and a new public database". In: *Journal of the Society for Information Display* 22.12 (2014), pp. 631–643.

[25] Vlad Hosu et al. "KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment". In: *arXiv preprint arXiv:1910.06180* (2019).

[26] Vlad Hosu et al. "Saliency-driven image coding improves overall perceived JPEG quality". In: *Picture Coding Symposium (PCS)*. IEEE. 2016, pp. 1–5.

[27] Vlad Hosu et al. "The Konstanz natural video database (KoNViD-1k)". In: *2017 Ninth international conference on quality of multimedia experience (QoMEX)*. IEEE. 2017, pp. 1–6.

[28] David A Huffman. "A method for the construction of minimum-redundancy codes". In: *Proceedings of the IRE* 40.9 (1952), pp. 1098–1101.

[29] Quan Huynh-Thu and Mohammed Ghanbari. "Scope of validity of PSNR in image/video quality assessment". In: *Electronics letters* 44.13 (2008), pp. 800–801.

[30] Noor A Ibraheem et al. "Understanding color models: a review". In: *ARPN Journal of science and technology* 2.3 (2012), pp. 265–275.

[31] Gazi Illahi, Matti Siekkinen, and Enrico Masala. "Foveated video streaming for cloud gaming". In: *19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2017, pp. 1–6.

[32] Gazi Karam Illahi et al. "Cloud Gaming with Foveated Video Encoding". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16.1 (2020), pp. 1–24.

[33] ISO. *ISO/IEC 14496-14:2020: Information technology — Coding of audio-visual objects — Part 14: MP4 file format*. URL: http://www.iso.ch/standard/791110.html.

[34] *ITU-R Recommendation BT.2020-2: Parameter values for ultra-high definition television systems for production and international programme exchange*. ITU-T, 2015.

[35] *ITU-T Recommendation P.910 - Subjective video quality assessment methods for multimedia applications, International Telecommunication Union*. ITU-T, 2008.

[36] *ITU-T Recommendation P.913 - Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment*. ITU-T, 2016.

[37] Anil K Jain. *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[38] George H Joblove and Donald Greenberg. "Color Spaces for Computer Graphics". In: *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*. 1978, pp. 20–25.

[39] Urvang Joshi et al. "Novel inter and intra prediction tools under consideration for the emerging AV1 video codec". In: *Applications of Digital Image Processing XL*. Vol. 10396. International Society for Optics and Photonics. 2017, 103960F.

[40] Tilke Judd et al. "Learning to predict where humans look". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 2106–2113.

[41] Brian Keelan. *Handbook of Image Quality: Characterization and Prediction*. CRC Press, 2002.

[42] Jani Lainema et al. "Intra coding of the HEVC standard". In: *IEEE transactions on circuits and systems for video technology* 22.12 (2012), pp. 1792–1801.

[43] Jong-Seok Lee, Francesca De Simone, and Touradj Ebrahimi. "Efficient video coding based on audio-visual focus of attention". In: *Journal of Visual Communication and Image Representation* 22.8 (2011), pp. 704–711.

[44] Jong-Seok Lee, Francesca De Simone, and Touradj Ebrahimi. "Subjective quality evaluation of foveated video coding using audio-visual focus of attention". In: *IEEE Journal of Selected Topics in Signal Processing* 5.7 (2011), pp. 1322–1331.

[45] Sanghoon Lee, Marios S Pattichis, and Alan C Bovik. "Foveated video compression with optimal rate control". In: *IEEE Transactions on Image Processing* 10.7 (2001), pp. 977–992.

[46] Hanhe Lin, Vlad Hosu, and Dietmar Saupe. "KADID-10k: A large-scale artificially distorted IQA database". In: *Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2019, pp. 1–3.

[47] Weisi Lin and C-C Jay Kuo. "Perceptual visual quality metrics: A survey". In: *Journal of Visual Communication and Image Representation* 22.4 (2011), pp. 297–312. DOI: 10.1017/j.jvcir.2011.01.005.

[48] *Making sense out of x264 rate control methods*. Accessed 03.04.2020. URL: https://mailman.videolan.org/pipermail/x264-devel/2010-February/006934.html.

[49] Henrique S Malvar et al. "Low-complexity transform and quantization in H. 264/AVC". In: *IEEE Transactions on circuits and systems for video technology* 13.7 (2003), pp. 598–603.

[50] Loren Merritt and Rahul Vanam. *x264: A high performance H. 264/AVC encoder*. 2006. URL: http://neuron2.net/library/avc/overview_x264_v8_5.pdf.

[51] Nathan Moroney and Mark D Fairchild. "Color space selection for JPEG image compression". In: *Journal of Electronic Imaging* 4.4 (1995), pp. 373–382.

[52] FW Mounts. "A Video Encoding System With Conditional Picture-Element Replenishment". In: *Bell System Technical Journal* 48.7 (1969), pp. 2545–2554.

[53] Debargha Mukherjee et al. "The latest open-source video codec VP9-an overview and preliminary results". In: *2013 Picture Coding Symposium (PCS)*. IEEE. 2013, pp. 390–393.

[54] *Objective perceptual quality assessment of video quality: Full reference television*. ITU-T, 2004.

[55] Antonio Ortega and Kannan Ramchandran. "Rate-distortion methods for image and video compression". In: *IEEE signal processing magazine* 15.6 (1998), pp. 23–50.

[56] Sarah Parker et al. "Global and locally adaptive warped motion compensation in video compression". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 275–279.

[57] Anjul Patney et al. "Towards foveated rendering for gaze-tracked virtual reality". In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), p. 179.

[58] Charles Poynton. "Chroma subsampling notation". 2002. URL: http://vektor.theorem.ca/graphics/ycbcr/Chroma_subsampling_notation.pdf (visited on 10/03/2020).

[59] Peter de Rivaz and Jack Haughton. "Av1 bitstream & decoding process specification". In: *The Alliance for Open Media* (2018), p. 182.

[60] Werner Robitza. *CRF Guide: Constant Rate Factor in x264, x265 and libvpx*. Accessed 03.04.2020. URL: https://slhck.info/video/2017/02/24/crf-guide.html.

[61] Werner Robitza. *Understanding Rate Control Modes*. Accessed 03.04.2020. URL: https://slhck.info/video/2017/03/01/rate-control.html.

[62] Simon Scholler et al. "Toward a direct measure of video quality perception using EEG". In: *IEEE transactions on Image Processing* 21.5 (2012), pp. 2619–2629.

[63] *SDL: Simple Direct Media Layer*. Accessed 21.12.2019. URL: https://libsdl.org.

[64] Claude E Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[65] Yun Q Shi and Huifang Sun. *Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards, 2nd edition*. CRC press, 2008.

[66] Andrew Stockman, Donald IA MacLeod, and Nancy E Johnson. "Spectral sensitivities of the human cones". In: *JOSA A* 10.12 (1993), pp. 2491–2521.

[67] Gary J Sullivan et al. "Overview of the high efficiency video coding (HEVC) standard". In: *IEEE Transactions on circuits and systems for video technology* 22.12 (2012), pp. 1649–1668.

[68] Sabine Süsstrunk, Robert Buckley, and Steve Swen. "Standard RGB color spaces". In: *Color and Imaging Conference*. Vol. 1999. 1. Society for Imaging Science and Technology. 1999, pp. 127–134.

[69] Vivienne Sze and Madhukar Budagavi. "High throughput CABAC entropy coding in HEVC". In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (2012), pp. 1778–1791.

[70] Vivienne Sze, Madhukar Budagavi, and Gary J Sullivan. "High efficiency video coding (HEVC), Integrated Circuits and Systems". In: *Cham: Springer* (2014).

[71] *The Matroska Media Container*. Accessed 10.02.2020. URL: `https://matroska.org`.

[72] Bruce Thompson. *Canonical correlation analysis: Uses and interpretation*. 47. Sage, 1984.

[73] Louis L Thurstone. "A law of comparative judgment." In: *Psychological review* 34.4 (1927), p. 273.

[74] Luc Trudeau, Nathan Egge, and David Barr. "Predicting chroma from luma in AV1". In: *2018 Data Compression Conference*. IEEE. 2018, pp. 374–382.

[75] Kristi Tsukida and Maya R Gupta. *How to analyze paired comparison data*. Tech. rep. University of Washington, 2011.

[76] Jean-Marc Valin et al. "Daala: Building a next-generation video codec from unconventional technology". In: *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2016, pp. 1–6.

[77] *variance-based psy adaptive quantization in x264*. Accessed: 02.07.2020. URL: `https://mailman.videolan.org/pipermail/x264-devel/2012-July/009403.html`.

[78] Paul Viola and Michael J Jones. "Robust real-time face detection". In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[79] Mike Wakin. *Standard Test Images*. 2003. URL: `https://www.ece.rice.edu/~wakin/images/`.

[80] Gregory K. Wallace. "The JPEG still picture compression standard". In: *Communications of the ACM* 34.4 (1991). URL: `https://www.ijg.org/files/Wallace.JPEG.pdf`.

[81] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates, 1995. ISBN: 9780878938537. URL: `https://foundationsofvision.stanford.edu`.

[82] Haiqiang Wang et al. “VideoSet: A large-scale compressed video quality dataset based on JND measurement”. In: *Journal of Visual Communication and Image Representation* 46 (2017), pp. 292–302.

[83] Haohong Wang and Khaled Helmi El-Maleh. *Region-of-interest coding with background skipping for video telephony*. US Patent 8,693,537. Apr. 2014.

[84] Zhou Wang and Alan C Bovik. “Embedded foveation image coding”. In: *Transactions on Image Processing (TIP)* 10.10 (IEEE, 2001), pp. 1397–1410.

[85] Zhou Wang and Alan C Bovik. “Mean squared error: Love it or leave it? A new look at signal fidelity measures”. In: *IEEE signal processing magazine* 26.1 (2009), pp. 98–117.

[86] Zhou Wang, Ligang Lu, and Alan C Bovik. “Foveation scalable video coding with automatic fixation selection”. In: *IEEE Transactions on Image Processing* 12.2 (2003), pp. 243–254.

[87] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[88] Oliver Wiedemann. “Local No-Reference Image Quality Assessment Using Convolutional Neural Networks”. Bachelor’s Thesis. University of Konstanz, 2018.

[89] Oliver Wiedemann and Dietmar Saupe. “Gaze Data for Quality Assessment of Foveated Video”. In: *ACM Symposium on Eye Tracking Research and Applications*. ACM. 2020.

[90] Oliver Wiedemann et al. “Foveated Video Coding for Real-Time Streaming Applications”. In: *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2020.

[91] Thomas Wiegand et al. “Overview of the H.264/AVC video coding standard”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 13.7 (2003), pp. 560–576.

[92] Stefan Winkler. “Analysis of public image and video databases for quality assessment”. In: *IEEE Journal of Selected Topics in Signal Processing* 6.6 (2012), pp. 616–625.

[93] Ian H Witten, Radford M Neal, and John G Cleary. “Arithmetic coding for data compression”. In: *Communications of the ACM* 30.6 (1987), pp. 520–540.

[94] *x264*. Accessed 07.01.2020. URL: `https://www.videolan.org/developers/x264.html`.

[95] *x264-/H.264-Technik: Quantizer und Constant Rate Factor (CRF)*. Accessed: 14.05.2020. URL: `https://encodingwissen.de/codecs/x264/technik/`.

[96] *x265 Documentation*. Accessed 10.05.2020. URL: `https://x265.readthedocs.io/en/default/index.html`.

[97] Pingmei Xu et al. “Turkergaze: Crowdsourcing saliency with webcam based eye tracking”. In: *arXiv preprint arXiv:1504.06755* (2015).

[98] Liuyang Yang. *Region of interest video coding*. US Patent 6,490,319. Dec. 2002.

[99] Emin Zerman et al. "The relation between MOS and pairwise comparisons and the importance of cross-content comparisons". In: *Electronic Imaging* 2018.14 (2018), pp. 1–6.