An optimal quad-tree-based motion estimation and motion compensated interpolation scheme for video compression

*Guido M. Schuster and †Aggelos K. Katsaggelos

* Advanced Technologies Research Center
Carrier Systems, 3COM
Mount Prospect, IL 60056
Tel: (847) 222-2486
Fax: (847) 222-0573
Email: Guido_Schuster@mw.3com.com

† Northwestern University
Department of Electrical and Computer Engineering
McCormick School of Engineering and Applied Science
Evanston, IL 60208-3118
Tel: (847) 491-7164
Fax: (847) 491-4455
Email: aggk@ece.nwu.edu

Abstract

In this paper we propose an optimal quad-tree (QT)-based motion estimator for video compression. It is optimal in the sense that for a given bit budget for encoding the displacement vector field (DVF) and the QT segmentation, the scheme finds a DVF and a QT segmentation which minimizes the energy of the resulting displaced frame difference (DFD). We find the optimal QT decomposition and the optimal DVF jointly using the Lagrangian multiplier method and a multilevel dynamic program. We introduce a new very fast convex search for the optimal Lagrangian multiplier λ^*, which results in a very fast convergence of the Lagrangian multiplier method. The resulting DVF is spatially inhomogeneous since large blocks are used in areas with simple motion and small blocks in areas with complex motion. We also propose a novel motion compensated interpolation scheme which uses the same mathematical tools developed for the QT-based motion estimator. One of the advantages of this scheme is the globally optimal control of the tradeoff between the interpolation error energy and the DVF smoothness. Another advantage is that no interpolation of the DVF is required since we directly estimate the DVF and the QT-segmentation for the frame which needs to be interpolated. We present results with the proposed QT-based motion estimator which show that for the same DFD energy the proposed estimator uses about 25% fewer bits than the commonly used block matching algorithm. We also experimentally compare the interpolated frames using the proposed motion compensated interpolation scheme with the reconstructed original frames. This comparison demonstrates the effectiveness of the proposed interpolation scheme.

EDICS: IP 1.12 Image Sequence Processing

List of Figures

List of Tables

1 Introduction

Since the bandwidth of uncompressed video is usually much larger than the available channel capacity, video compression is one of the enabling technologies behind the multimedia revolution. Video lends itself to compression because of the high perceptual, temporal and spatial redundancies inherent in a natural scene. The most common approach to exploit the temporal redundancy is motion compensated prediction. All current video coding standards [1, 2, 3, 4] are motion compensated video coders (MCVC), where the current frame is predicted using a previously reconstructed frame and motion information, which needs to be estimated.

The motion estimation problem for video coding differs from the general motion estimation problem. There are several reasons for this. First, for video coding, the performance of a motion estimator can be assessed using a distortion measure, whereas for the general motion estimation problem, such measures are very hard to define. Second, for video coding, the displacement vector field (DVF) has to be transmitted and therefore the minimization of the distortion should be constrained by the available number of bits to encode the DVF, whereas for the general motion estimation problem, the constraints are used to enforce certain desirable properties of the DVF, such as smoothness. Third, for video coding, the resulting DVF can be arbitrary, i.e., it does not have to correspond to the real motion in the scene, as long as the distortion is minimized for the available bit rate. The goal of the general motion estimation problem is in strong contrast to this, since it is to find the two dimensional projection onto the image plane of the real world three dimensional motion. Since the three dimensional motion is usually unknown, the performance of the scheme needs to be evaluated by inspecting the resulting motion field and making a subjective judgment of its quality. Another evaluation scheme results if the estimated DVF is used to interpolate a known frame, against which the interpolated frame is then compared. Clearly the better the DVF describes the real motion, the closer the interpolated frame matches the known frame. We will use this technique in Sec. 8 to describe the performance of the proposed motion estimator.

Before we go on to describe previous attempts to find the DVF which results in the smallest energy, subject to a given rate constraint, we would like to point out that this is part of the bigger optimal video compression problem. In optimal video compression, we try to allocate the available bit rate among motion, segmentation and residual error, such that the resulting distortion is minimize, subject to a given rate constraint. Readers interested in this topic are referred to [5, 6, 7]. Note that in [6] we use motion estimator proposed in this paper for an optimal video compression scheme.

In [8], the problem of rate-constrained motion estimation is considered and the optimal bit allocation condition for a strictly convex and everywhere differentiable multivariate rate distortion function is derived. It is applied to the problem of optimal bit allocation between the DVF and the displaced frame difference (DFD) and a rate-constrained, region-based motion estimator is introduced. The derivation in [8] is true to the idea of rate distortion theory, where certain models are assumed and then solutions are derived based on these models. In this paper, we follow the paradigm of operational rate distortion theory, where an efficient scheme is derived and its decisions are analyzed such that the resulting overall approach is optimal [9, 10, 11, 5]. Therefore, we do not assume knowledge of a convex and everywhere differentiable multivariate rate distortion function, but instead we deal with finite sets of admissible motion vectors.

This paper is organized as follows: In Sec. 2 we formulate the problem and discuss some of the previous solution attempts. In Sec. 3 we introduce the necessary notation and the underlying assumptions. In Sec. 4 we introduce the definition of an efficient scanning path for an arbitrary QT decomposition. We propose a recursive procedure to generate such a scanning path and we derive a necessary and sufficient condition for the procedure to result in an efficient scanning path. We propose a special scanning path, based on a Hilbert curve, which guarantees that the resulting blocks are square. In Sec. 5 we formulate the motion estimation problem and show how the Lagrangian multiplier method can be used to find the optimal solutions on the convex hull of the operational rate distortion curve. We propose a very fast convex search, which is based on a second order Bezier curve, to find the optimal Lagrangian multiplier. In Sec. 6 we develop a fast algorithm to solve the relaxed problem using dynamic programming (DP). In Sec. 7 we propose a motion compensated interpolation scheme which is based on a similar mathematical formulation as the QT-based motion estimator. The proposed scheme directly segments the frame which has to be interpolated and therefore no DVF interpolation is necessary. In Sec. 8 we compare the proposed QT-based motion estimator with block matching and evaluate the interpolation scheme by comparing interpolated frames with reconstructed original frames. Finally, in Sec. 9, we summarize the paper.

2 Problem formulation

All current video coding standards are based on a fixed segmentation of the frame into blocks of a given size. The main advantages of this are simplicity of the algorithm and the fact that no segmentation information needs to be transmitted. The only exception to this is H.263 when the “Advanced Prediction Mode” is used. Then there is the option to split the 16×16 block into four 8×8 blocks. The main disadvantage of a fixed and arbitrary segmentation is that it can not accurately represent real objects, and hence the representation of the scene is not as compact as it could be. Clearly the denser DVF the more complicated the motion it can represent. The problem, however, in this case is that in many regions of the scene, most of the motion vectors are identical and therefore not needed to describe the motion accurately.

Hence a natural DVF is inherently inhomogeneous, in the sense that for regions with simple motion, one motion vector is sufficient, whereas in regions with complex motion the DVF should be denser. If the scene is segmented into blocks of different sizes, then a compact representation of the DVF can be achieved. Since the segmentation information of the frame needs to be transmitted, an efficient method should be used to encode it.

The quad-tree (QT) data structure [12] is commonly used to decompose a given frame into blocks of different sizes, since it enables an efficient representation of the resulting decomposition. A QT starts with a square block with side length a power of two. This block can be split into four equally sized square sub-blocks, and only one bit is required to encode the split or lack of it. Clearly this splitting can then be recursively applied to the sub-blocks until a sub-block is of dimensions 1×1. Hence the entire segmentation of the frame can be represented by a tree structure, where every node has four children.

The QT structure is an efficient way of segmenting the frame into blocks of different sizes. It is therefore very attractive for the efficient representation of an inhomogeneous DVF. In [13, 14, 15], the QT structure is used to compactly represent the DVF. In [13] the dense motion field is represented efficiently using higher order motion models, and the spatial density of the applicability of these models is encoded using a QT. In [14] the temporal gradient of a pel-recursive motion estimation algorithm is QT encoded and in [15] a QT-based motion estimator is proposed which finds the best QT in the rate distortion sense. Note that the algorithm proposed in [15] is a direct application of the more general scheme proposed in [16, 17]. The motion estimator we propose is more general and more efficient than the scheme in [15], since it finds the optimal QT decomposition and the optimal DVF, given that the DVF is first order differential pulse code modulation (DPCM) encoded. The presented algorithm is a generalization of the scheme presented in [16, 17], since the optimal QT decomposition can be found even when there are dependencies between the QT leafs.

Note that the variable block size based decomposition of the frame is a good compromise between the overly complex object-oriented approaches [18], and the overly simple fixed block size based schemes. The object-oriented approaches suffer from the fact that it is in general very complex to accomplish joint segmentation and motion estimation, while the fixed block size schemes suffer from the inability to represent complex motion. In addition, the overhead paid for the segmentation of a QT-based scheme is much lower than the bit rate required to transmit the fine segmentation of an object-oriented scheme, although it is still substantial considering the fact that a fixed segmentation based scheme does not require any segmentation information to be transmitted.

We propose a new QT-based motion estimator which results in an optimal tradeoff between the bit rate required for the DVF and the resulting DFD energy. The proposed scheme is based on variable block sizes and therefore the resulting DVF is inhomogeneous. Depending on the available bit rate, the density of the DVF changes, such that for a high bit rate the DVF is quite dense, whereas for a low bit rate the DVF is quite sparse. Since the inhomogeneous DVF represents the real motion quite well, we also propose a motion compensated interpolation scheme which uses a similar mathematical formulation as the proposed motion estimator. The resulting scheme has the advantage that it offers globally optimal control of the tradeoff between the smoothness of the DVF and the resulting interpolation error energy.

3 Notation and assumptions

In this section we introduce the necessary notation and the underlying assumptions. Consider Fig. 1, where a frame is segmented using variable block sizes and the black curve indicates the feature of interest. The QT shown in Fig. 2 is used to represent this segmentation. As can be seen in this figure, the QT data structure decomposes a $2^N \times 2^N$ image (or block of an image) down to blocks of size $2^{n_0} \times 2^{n_0}$. This decomposition results in an $(N - n_0 + 1)$-level hierarchy ($0 \leq n_0 \leq N$), where all blocks at level n ($n_0 \leq n \leq N$) have size $2^n \times 2^n$. This structure corresponds to an inverted tree, where each $2^n \times 2^n$ block (called a *tree node*) can either be a *leaf*, i.e., it is not further subdivided, or can branch into four $2^{n-1} \times 2^{n-1}$ blocks, each a *child* node. The tree can be represented by a series of bits that indicate termination by a leaf with a "0" and branching into child nodes with an "1".

Let $b_{l,i}$ be the block i at level l, and the children of this block are therefore $b_{l-1,4*i+j}$, where $j \in [0, 1, 2, 3]$. The complete tree is denoted by $\mathcal{T}$ and a tree node is identified by the ordered pair (l, i) (see Fig. 3). This ordered pair is called the index of the tree node. Each leaf of $\mathcal{T}$ represents a particular block in the segmented frame. For future convenience, let the leafs be numbered from zero to the total number of leafs in the QT ($N_{\mathcal{T}}$) minus one, from left-to-right and hence in increasing order of the measure $4^{l-n_0} * i$ (this ordering of the leafs is indicated by italic numbers in Fig. 3).

Let $m_{l,i} \in M_{l,i}$ be the motion vector for block $b_{l,i}$, where $M_{l,i}$ is the set of all admissible motion vectors for block $b_{l,i}$. Let $s_{l,i} = [l, i, m_{l,i}] \in S_{i,l} = \{l\} \times \{i\} \times M_{l,i}$ be the local state for block $b_{l,i}$, where $S_{l,i}$ is the set of all admissible state values for block $b_{l,i}$. Let $x = [l, i, m] \in X = \bigcup_{l=N}^{n_0} \bigcup_{i=0}^{4^{N-l}-1} S_{l,i}$ be the global state and X the set of all admissible state values. Finally let $x_0, \ldots, x_{N_{\mathcal{T}}-1}$ be a global state sequence, which represents the left-to-right ordered leaves of a valid QT $\mathcal{T}$.

The DFD energy $D(x_0, \ldots, x_{N_{\mathcal{T}}-1})$ is the sum of the individual block energies $d(x_j)$, that is,

$$D(x_0, \ldots, x_{N_{\mathcal{T}}-1}) = \sum_{j=0}^{N_{\mathcal{T}}-1} d(x_j). \tag{1}$$

This is true since each block in the current frame is predicted using one motion vector which points to a block in a previously reconstructed frame. Note that in the rest of the paper we will use the terms *DFD energy* and the *distortion* between the original and predicted frames, interchangeably.

As pointed out in the introduction, the DVF exhibits large spatial correlation, and hence we encode it using a first order DPCM. In other words, we allow for a first order dependency between the QT-leaves along the scanning path. Because of the variable block sizes, we add the constraint that blocks which belong to the same parent, need to be scanned in sequence. We discuss the selection of an efficient scanning path in detail in Sec. 4.

Based on the first order DPCM assumption, the DVF rate $R(x_0, \ldots, x_{N_\mathcal{T}-1})$ can be expressed as follows,

$$R(x_0, \ldots, x_{N_\mathcal{T}-1}) = \sum_{j=0}^{N_\mathcal{T}-1} r(x_{j-1}, x_j), \tag{2}$$

where $r(x_{j-1}, x_j)$ is the block rate which depends on the motion vectors of the current and previous blocks, since the difference between these two motion vectors is entropy encoded.

4 Efficient scanning path

In this section we consider the problem of finding an efficient scanning path for an arbitrary QT decomposition of a frame. A scanning path is a rule which defines in what order the different blocks of a frame are visited. Since the QT decomposition can change from frame to frame, each frame might require a different scanning path. We propose a procedure for inferring the scanning path from a given QT decomposition. For this procedure, we derive a necessary and sufficient condition which guarantees that consecutive blocks along the scanning path are always neighboring blocks, that is, they have a common edge. This results in an efficient encoding of the DVF using a first order DPCM scheme along the scanning path.

4.1 Efficiency

We propose the following definition of an efficient scanning path.

Definition 1 *A scanning path is efficient if it connects only neighboring blocks, i.e., blocks which share an edge, and if it visits each block once and only once.*

The motivation behind this definition comes from the observation that an efficient scanning path will lead to highly correlated successive blocks, since such blocks will always be neighboring blocks. As stated above, the goal is to encode the DVF using a first order DPCM scheme along the scanning path, therefore a high correlation between successive blocks will result in an efficient encoding of the DVF. Note that the recursive raster scan, which has been traditionally used in the QT decomposition, is not efficient, as can be seen from Fig. 4d.

As pointed out above, the segmentation changes from frame to frame, therefore no fixed scanning path can be employed. On the other hand, no additional information should have to be transmitted to indicate which scanning path was used. Therefore, a good scanning path must be inferable from the transmitted QT and it has to be efficient in the above defined sense. To simplify the search for a good scanning path, we require that child blocks, which belong to the same parent block, are scanned in sequence. This is the same condition we introduced in the previous section. In the following section we propose a procedure to infer a scanning path from the transmitted QT decomposition and derive a necessary and sufficient condition for the procedure to result in an efficient scanning path.

4.2 Procedure

Assume that the frame is completely segmented into blocks of the smallest size, i.e., $2^{n_0} \times 2^{n_0}$. The QT representation of this segmentation is a complete tree, where all leafs are of level n_0. Further assume that the scanning path for this decomposition is known to the encoder and decoder. The scanning path for any other QT decomposition is defined recursively, by merging four consecutive blocks along the scanning path at the lower level to form the blocks at the higher level. Note that with this procedure, the resulting blocks do not need to be square, although at level n_0 the blocks are square. An example of this recursive definition is given in Fig. 11, where the level n_0 scanning path is a third order Hilbert scan, which we will introduce in Sec. 4.3.

This recursive definition of the scanning path results in a scanning path that is inferable from the QT decomposition and the scanning path of the fully decomposed QT. The following lemma establishes a necessary and sufficient condition on the level n_0 scanning path, so that the scanning path created with the proposed recursive procedure is always efficient.

Lemma 1 *If the level n_0 scanning path is efficient, and the above recursive procedure is used for the generation of a scanning path for any QT decomposition, then the resulting scanning path is efficient.*

Proof: necessary condition: if it is violated, then the completely decomposed tree (down to level n_0) would not result in an efficient scanning path since the level n_0 scanning path is not efficient.

Proof: sufficient condition (by induction): The level n_0 scanning path is efficient. Consider an efficient scanning path. Pick any connected sub-path visiting four blocks. Since the overall scanning path is efficient, the sub-path is efficient too. By merging the four blocks of the sub-path into a single block, the neighboring blocks of the four merged blocks are also neighboring blocks of the new single block. Hence the new scanning path is efficient. ■

4.3 Efficient scanning path based on a Hilbert curve

In the previous section we have established a procedure and a necessary and sufficient condition for generating efficient scanning paths. Clearly there are many different efficient scanning paths, since there are many different efficient level n_0 scanning paths. As already mentioned, a drawback of the proposed procedure is that the blocks resulting from the merging procedure may not be square, although the blocks at level n_0 are square. In this section we propose an efficient scanning path which results in square blocks, and hence in the kind of QT decomposition displayed in Fig. 1.

The efficient scanning path we developed is based on a Hilbert curve, also called a Peano curve [19]. A Hilbert curve has a certain order and the first order Hilbert curve can have one of four orientations, denoted by 0, 1, 2 and 3. Figure 5 shows the four first order Hilbert curves with the associated second order Hilbert curves drawn beneath them. Note that each second order Hilbert curve consists of four first order Hilbert curves connected by a dotted line and scaled by a factor of $\frac{1}{2}$.

We propose the generalization of this relationship between first and second order Hilbert curves using the following algorithm, which generates a Hilbert curve of order N and shows the close relationship between a QT and a Hilbert curve. Let $O_{l,i} \in [0,1,2,3]$ be the orientation (scanning order of the blocks $b_{l-1,4*i+j}, j \in [0,1,2,3]$) of block $b_{l,i}$ and let $(\cdot)_4$ denote the modulo 4 operation. We propose to create a Hilbert curve of order N in a Top-Down fashion by calling the function "orient(l,i)" below with the following parameters: orient(N,0).

```
orient($l, i$)
    $O_{l-1,4*i+0} = (5 - O_{l,i})_4$; $O_{l-1,4*i+1} = O_{l,i}$; $O_{l-1,4*i+2} = O_{l,i}$; $O_{l-1,4*i+3} = 3 - O_{l,i}$;
    if $(l - 1 > 1)$
        orient($l - 1, 4 * i + 0$); orient($l - 1, 4 * i + 1$); orient($l - 1, 4 * i + 2$); orient($l - 1, 4 * i + 3$);
```

This recursion needs to be initialized by the desired orientation of the first order Hilbert curve, say $O_{N,0} = 0$ (Fig. 5a). The resulting orientations at each QT level l, $(N \geq l \geq 1)$ correspond to a Hilbert curve of order $N - l + 1$, hence at level one, the orientations form a Hilbert curve of order N. In Fig. 6 the recursive generation of a third order Hilbert curve is shown. This curve was generated by orient(3,0).

Hilbert curves have been used in image and video processing as scanning paths on the pixel level in the luminance domain for lossless coding [20] and lossy coding [21]. They have also been used as a scanning path for the coefficients in the transform domain [22]. In all these cases, the fact that an image, which is scanned according to a Hilbert curve, creates an one-dimensional representation of the image which is more correlated than a raster scan, is exploited.

Having defined the relationship between a Hilbert curve and a QT and discussed some properties of the Hilbert curve, it is clear that when the level n_0 scanning path is a Hilbert curve of order $N - n_0$, then the resulting scanning path is efficient and the resulting blocks are square, regardless of the QT decomposition. Besides resulting in square blocks, the Hilbert scan has also the advantage of being spatially non-disruptive and hence tends to preserve local pixel correlations better than a raster scan.

5 Optimal QT-based motion estimator in the rate distortion sense

The motion estimation problem for video coding can be stated as the following constrained optimization problem,

$$\min_{x_0,\ldots,x_{N_{\mathcal{T}}-1}} D(x_0,\ldots,x_{N_{\mathcal{T}}-1}), \qquad \text{subject to:} \quad R(x_0,\ldots,x_{N_{\mathcal{T}}-1}) \leq R_{max}, \tag{3}$$

where $D(x_0,\ldots,x_{N_{\mathcal{T}}-1})$ and $R(x_0,\ldots,x_{N_{\mathcal{T}}-1})$ are defined by Eqs. (1) and (2), respectively. In other words, we try to find the inhomogeneous DVF which results in the smallest DFD energy for a given maximum bit rate for encoding the DVF. Clearly the optimal state sequence $x_0^*, \ldots, x_{N_{\mathcal{T}}-1}^*$ identifies the optimal QT decomposition and the optimal DVF, since each state value x_j contains the block size, its location and the associated motion vector.

This constrained discrete optimization problem is generally very hard to solve. Since we deal with a finite number of QTs and admissible motion vectors, it can clearly be solved by an exhaustive search. The time complexity for such an exhaustive search is extremely high. We develop a lower bound for the number of different QTs. Let z_k be the number of QTs at level k, $n_0 < k \leq N$. Clearly $z_{n_0+1} = 2$, since the QT has either no children or four children. Furthermore, $z_{n_0+2} = 1 + z_{n_0+1}^4 \geq z_{n_0+1}^4 = 2^4$, where the 1 stands for the unsplit QT and the $z_{n_0+1}^4$ represents its four children, each of them can take on z_{n_0+1} QT decompositions. Taking this one step further, $z_{n_0+3} = 1 + z_{n_0+2}^4 \geq z_{n_0+2}^4 \geq z_{n_0+1}^{4^2} = 2^{4^2}$. Therefore, by induction, $2^{4^{N-n_0-1}}$ is a lower bound for the number of different QTs, where $(N - n_0) \geq 1$.

To find a lower bound on the total time complexity we consider the case when the QT is completely decomposed, i.e., the image is segmented into 4^{N-n_0} blocks of size $2^{n_0} \times 2^{n_0}$. We further assume that the cardinality $|S_{n_0,i}|$ of all admissible local states $S_{n_0,i}$ of the leafs is the same. Hence an exhaustive search, using only this single QT, requires $|S_{n_0,i}|^{4^{N-n_0}}$ comparisons. Since we have only considered one QT, this is a lower bound, $\Omega_E(|S_{n_0,i}|^{4^{N-n_0}})$, on the time complexity for all QTs, where E stands for exhaustive search. We will show that the upper bound of the proposed algorithm is significantly smaller than this lower bound for an exhaustive search. Note that when we use the term time complexity, we refer to the number of comparisons necessary to find the optimal solution. This does not include the time spent to evaluate the operational rate distortion functions, since this strongly depends on the implementation.

5.1 The Lagrangian multiplier method

As we have pointed out, an exhaustive search is not a reasonable approach. We propose to use the Lagrangian multiplier method [23] to solve the constrained problem of Eq. (3). The idea behind this method is to transform the "hard" constrained optimization problem into a family of "easy" unconstrained problems, which can be solved efficiently. This transformation is achieved by creating a new objective function, the Lagrangian cost function $(J_\lambda(\cdot))$. It is the sum of the original objective function and the constraint, where the constraint is weighted by the Lagrangian multiplier λ,

$$J_\lambda(x_0, \ldots, x_{N_{\mathcal{T}}-1}) = D(x_0, \ldots, x_{N_{\mathcal{T}}-1}) + \lambda * R(x_0, \ldots, x_{N_{\mathcal{T}}-1}). \tag{4}$$

The main theorem [23] of the Lagrangian multiplier method states that if there is a λ^* such that

$$[x_0^*, \ldots, x_{N_{\mathcal{T}}-1}^*] = \arg \min_{x_0, \ldots, x_{N_{\mathcal{T}}-1}} J_{\lambda^*}(x_0, \ldots, x_{N_{\mathcal{T}}-1}) \tag{5}$$

leads to $R(x_0, \ldots, x_{N_{\mathcal{T}}-1}) = R_{max}$, then $x_0^*, \ldots, x_{N_{\mathcal{T}}-1}^*$ is also an optimal solution to (3). The main shortcoming of the Lagrangian multiplier method comes from the fact that only solutions which belong to the convex hull of the operational rate distortion curve (or in this case, the rate energy curve) can be found. Since for the proposed scheme, the convex hull solutions tend to be dense (see Sec. 8), this is not a problem in practice. Note that the dual problem, which can be stated as follows,

$$\min_{x_0,\ldots,x_{N_{\mathcal{T}}-1}} R(x_0,\ldots,x_{N_{\mathcal{T}}-1}), \quad \text{subject to:} \quad D(x_0,\ldots,x_{N_{\mathcal{T}}-1}) \leq D_{max}, \tag{6}$$

can be solved with exactly the same technique using the following relabeling of function names, $R(x_0,\ldots,x_{N_{\mathcal{T}}-1}) \leftarrow D(x_0,\ldots,x_{N_{\mathcal{T}}-1})$, $D(x_0,\ldots,x_{N_{\mathcal{T}}-1}) \leftarrow R(x_0,\ldots,x_{N_{\mathcal{T}}-1})$.

Having stated the main theorem of the Lagrangian multiplier method, there are two problems left to address: first, how to find the optimal λ^* of Eq. (5) and second, how to solve the unconstrained problem of Eq. (5) optimally for an arbitrary λ. In the next section we propose a very fast convex search to find λ^* and in Sec. 6 we introduce a dynamic programming algorithm to find the optimal solution to problem (5).

5.2 Very fast convex search based on a Bezier curve

In this section we introduce a very fast search algorithm for the optimal λ^* needed in the Lagrangian multiplier method [24]. It is well known that the rate is a non-increasing function of λ and hence the bisection method is a straightforward choice for finding the optimal λ^*. The bisection method is an iterative scheme and hence for every iteration, the solution to the unconstrained problem needs to be found. Therefore the time complexity of the entire algorithm is directly proportional to the number of iterations. It is therefore desirable to have as few iterations as possible in finding a good λ.

The situation at hand is shown in Fig. 7. The proposed iterative search makes use of the fact, that for a differentiable rate distortion curve, the derivative is equal to $\frac{-1}{\lambda}$. Assume that the bracketing points $P_0 = (D_0, R_0)$, $P_2 = (D_2, R_2)$ and the derivatives at these points, d_0 and d_2, respectively, are given. The goal is to approximate the convex hull between P_0 and P_2 as accurately as possible, so that this approximation can be used to estimate the derivative d_{max} at R_{max}. This estimate of the derivative $\hat{d}_{max}$ is then used to generate the new $\lambda = \frac{-1}{\hat{d}_{max}}$ for the next iteration.

In [25] a fast convex search is used, which is based on the λ theorem of the Lagrangian multiplier method [23]. The graphical interpretation of the λ theorem is, that given two points on the convex hull and their respective λs, the slope of the line connecting the two points lies between the slope implied by the λs of the points. The search in [25] can be interpreted as approximating the convex hull with a straight line between P_0 and P_2, which is not the best possible approximation since the knowledge of d_0 and d_2 (the negative inverses of the previous λs) is not used.

A good approximation to the convex hull is a function which goes through the given bracketing points P_0 and P_2 and matches the derivatives d_0 and d_2 at these points. Also the approximation should be convex, differentiable, and in closed form. A parabola would be an ideal choice but it has only three degrees of freedom which is in general not enough to match four independent requirements (P_0, P_2, d_0, d_2). A second order Bezier curve [26] is a parabola which can be tilted in any direction in the plane, hence it has an additional degree of freedom and can therefore match the four independent requirements. A second order Bezier curve is of the following parametric form,

$$P(u) = (1-u)^2 * P_a + 2 * (1-u) * u * P_b + u^2 * P_c. \tag{7}$$

By setting $P_a = P_0$ and $P_c = P_2$, two of the four conditions are immediately satisfied, i.e., $P(0) = P_0$ and $P(1) = P_2$. The point P_b can be chosen so that the two derivative conditions are satisfied. The derivative of the second order Bezier curve is,

$$d(u) = \frac{(u-1) * R_0 + (1 - 2 * u) * R_b + u * R_2}{(u-1) * D_0 + (1 - 2 * u) * D_b + u * D_2}. \tag{8}$$

Therefore

$$d_0 = d(0) = \frac{R_b - R_0}{D_b - D_0} \tag{9}$$

and

$$d_2 = d(1) = \frac{R_2 - R_b}{D_2 - D_b}, \tag{10}$$

which leads to $P_b = (D_b, R_b)$ where,

$$D_b = \frac{(R_0 - d_0 * D_0) - (R_2 - d_2 * D_2)}{d_2 - d_0}, \tag{11}$$

and

$$R_b = \frac{d_2 * (R_0 - d_0 * D_0) - d_0 * (R_2 - d_2 * D_2)}{d_2 - d_0}. \tag{12}$$

As can be seen in Fig. 7, P_b is the point where the two tangents of P_0 and P_2 intersect.

Since the goal is to find $\hat{d}_{max}$, the derivative of $P(u)$ has to be evaluated at R_{max}. Hence u_{max} has to be found first which leads to $P(u_{max}) = (D_{max}, R_{max})$. It is found by solving the quadratic equation Eq. (7), to obtain,

$$u_{max}^{\pm} = -\frac{R_b - R_0}{(R_2 - R_b) - (R_b - R_0)} \pm \sqrt{\frac{(R_b - R_0)^2}{((R_2 - R_b) - (R_b - R_0))^2} + \frac{R_{max} - R_0}{(R_2 - R_b) - (R_b - R_0)}}, \tag{13}$$

where u_{max} is picked such that $1 \geq u_{max} \geq 0$. When $(R_2 - R_b) - (R_b - R_0)$ equals zero, $u_{max} = (R_{max} - R_0)/(2 * (R_b - R_0))$. Hence the estimate of the derivative at R_{max}, is given by $\hat{d}_{max} = d(u_{max})$.

The search for the optimal λ^* follows the bisection method, but instead of setting the new λ equal to the average of the old λs, the new λ is set equal to $\frac{-1}{\hat{d}_{max}}$. Clearly the proposed search uses all information available (the two bracketing points, the derivatives at the two bracketing points and the convexity property of the rate distortion curve), to find the new λ in the bisection iteration. Therefore we expect this search to outperform other bisection schemes, which do not use as much information for selecting the new λ. As mentioned above, the straightforward approach is to use the arithmetic mean for the new λ. Another popular scheme uses the geometric mean and the fast convex search proposed in [25] selects the new λ according to a straight line approximation of the rate distortion curve.

Clearly, the speed of convergence depends on the particular rate distortion curve, the required accuracy, the initial starting points and the desired rate. In order to test the four different searches in a controlled fashion, we use the well known [27] rate distortion curve $R(D) = -0.5\ln(D)$, $0 < D < 1$ of a discrete Gaussian source, which consists of independent and identically distributed Gaussian random variables with zero mean and unit variance, and the distortion measure is the mean squared error. Note that this function is neither a straight line nor a quadratic, so neither the line based search nor the proposed search should have an inherent advantage. We define the accuracy A as the size of the absolute error between the resulting rate R_{real} and the desired rate R_{max} in percent of R_{max}, hence $A = |R_{max} - R_{real}|/R_{max}$. As initial bracket, we select $D_0 = 0.01$ and $D_2 = 0.99$, which results in $R_0 = -0.5\ln(D_0) = 2.3$, $\lambda_0 = 2D_0 = 0.02$, and $R_2 = -0.5\ln(D_2) = 0.005$, $\lambda_2 = 2D_2 = 1.98$. We randomly select R_{max} between R_0 and R_2, (in other words, R_{max} is uniformly distributed between R_0 and R_2), and run each of the four search algorithms until a solution is found which is within the specified accuracy. We run this experiment for $100'000$ different $R_{max}s$ and in Table 1 we report the average number of iterations required for each scheme using four different accuracies, 10%, 1%, 0.1% and 0.01%. As can be seen from Table 1, the proposed search significantly outperforms all other searches by a factor ranging from two to four. Note that all four algorithms show the expected logarithmic speed of convergence, in that increasing the accuracy exponentially, only increases the number of required iterations linearly. Further notice that the gain of the geometric mean and the line approximation over the arithmetic mean is mostly in how quickly they get within 10% of the solution. From then on out, all three, the arithmetic mean, the geometric mean and the line approximation require about 3 iterations to increase the accuracy by a magnitude.

The proposed search is in strong contrast to this. Not only is it within 10% accuracy twice as fast as the nearest competitors, but it also requires only 0.5 iterations, on average, to increase the accuracy by a magnitude. These results, based on a particular rate distortion function, are also reflected in our experiments with the proposed optimal QT motion estimator, where the iterative search which is based on the second order Bezier approximation exhibits a very fast convergence and hence we call it the "very fast convex search".

6 Optimal solution of the unconstrained problem by multilevel dynamic programming

6.1 Multilevel trellis

The form of the objective function of the optimization problem of Eq. (5) suggests that dynamic programming (DP) should be used to find the optimal solution efficiently. To be able to employ forward DP (the Viterbi algorithm), a DP recursion formula needs to be established. A graphical equivalent of the DP recursion formula is a trellis where the admissible nodes and the permissible transitions are explicitly indicated. Consider Fig. 8 which represents the multilevel trellis for a 32×32 image block ($N = 5$), with a QT segmentation developed down to level 3 ($n_0 = 3$, block size 8×8). The QT structure is indicated by the white boxes with the rounded corners. These white boxes are not part of the trellis used for the Viterbi algorithm but indicate the set of admissible state values $S_{l,i}$ for the individual blocks $b_{l,i}$. The black circles inside the white boxes are the nodes of the trellis (i.e., the state values $s_{l,i}$). Note that for simplicity, only two trellis nodes per QT node are indicated, but in general, a QT node can contain any number of trellis nodes. The auxiliary nodes, start and termination (S and T) are used to initialize the DPCM and to select the path with the smallest cost.

Each of the trellis nodes represents the prediction of the block it is associated with using a different motion vector. Since the state of a block is defined to contain its level and number within that level (which identifies the blocks size and its position in the frame), and its motion vector, each of the nodes contains the distortion occurring when the associated block is predicted using the node's motion vector.

As can be seen in Fig. 8, not every trellis node can be reached from every other trellis node. By restricting the permissible transitions, we are able to force the optimal path to select only valid QT decompositions. Such valid decompositions are based on the fact that at level l, block $b_{l,i}$ can replace four blocks at level $l-1$, namely $b_{l-1,4*i+0}, b_{l-1,4*i+1}, b_{l-1,4*i+2}$ and $b_{l-1,4*i+3}$. As we will see later in this section, the QT encoding cost can be distributed recursively over the QT so that each path picks up the right amount of QT segmentation overhead.

Assume that no QT segmentation is used and the block size is fixed at 8×8. In this case, only the lowest level of the trellis in Fig. 8 is used. The transition costs between the trellis nodes would be the rate required to encode the differences between consecutive motion vectors along the scanning path weighted by the Lagrangian multiplier λ. Assume now that the next higher level, level 4, of the QT is included. Clearly the transition cost between the trellis nodes of level 3 stay the same. In addition, there are now transition costs between the trellis nodes of level 4 and also transition cost from trellis nodes of level 3 to trellis nodes of level 4 and vice versa, since each cluster of four blocks at level 3 can be replaced by a single block at level 4. The fact that a path can only leave and enter a certain QT level at particular nodes results in paths which all correspond to valid QT decompositions. Note that every QT node in a path is considered a leaf of the QT which is associated with this path.

In this example, a tree of depth 3 has been used to illustrate how the multilevel trellis is built. For QTs of greater depth, a recursive rule has been derived which leads to the proper connections between the QT levels. Since the Viterbi algorithm is a forward dynamic programming scheme, the optimal solution is found by going through the multilevel trellis from left to right. At any epoch of the Viterbi algorithm, the shortest path is found, over the set of all admissible nodes of the previous epoch (say $F_{l,i}$), to every node in the set of all admissible nodes at the current epoch (say $T_{l,i}$).

These two sets, at each epoch, the "from" set $F_{l,i}$ and the "to" set $T_{l,i}$ can be constructed recursively from the sets of the admissible state values $S_{l,i}$ for each block $b_{l,i}$. Consider Fig. 9 which shows a QT node and its four children. The $T_{l,i}$ set contains all the trellis nodes for which a transition from the left is permissible and $F_{l,i}$ contains all the trellis nodes for which a transition to the right is permissible. Clearly for the top QT level N, $T_{N,0} = S_{N,0}$ and $F_{N,0} = S_{N,0}$, which is the initialization of the recursion. As we will see in Sec. 6.2, only the sets at level n_0 will be used in the Viterbi algorithm since they define the permissible transitions which in turn enforce the QT structure. These sets can be generated recursively in a Top-Down fashion by calling the function "sets(l,i)" below with the following parameters: sets(N,0).

```
sets$(l, i)$
    $T_{l-1,4*i+0} = T_{l,i} \cup S_{l-1,4*i+0}$; $F_{l-1,4*i+0} = S_{l-1,4*i+0}$; $T_{l-1,4*i+1} = S_{l-1,4*i+1}$; $F_{l-1,4*i+1} = S_{l-1,4*i+1}$;
    $T_{l-1,4*i+2} = S_{l-1,4*i+2}$; $F_{l-1,4*i+2} = S_{l-1,4*i+2}$; $T_{l-1,4*i+3} = S_{l-1,4*i+3}$; $F_{l-1,4*i+3} = F_{l,i} \cup S_{l-1,4*i+3}$;
    if $(l - 1 > n_0)$
        sets$(l-1, 4*i+0)$; sets$(l-1, 4*i+1)$; sets$(l-1, 4*i+2)$; sets$(l-1, 4*i+3)$;
```

In the presented multilevel trellis, the nodes of the respective blocks hold the information about the distortion occurring when the associated block is predicted using the motion vector of the node. The rate needed to encode the difference between the motion vectors is incorporated into the transition cost between the nodes (weighted by the Lagrangian multiplier λ), but so far, the rate needed to encode the QT decomposition has not been addressed.

Since the Viterbi algorithm will be used to find the optimal QT decomposition, each node needs to contain a term which reflects the number of bits needed to split the QT at its level. Clearly, trellis nodes which belong to blocks of smaller size have a higher QT segmentation cost than nodes which belong to bigger blocks.

When the path includes only the top QT level N, then the QT is not split at all, and only one bit is needed to encode this. Therefore its segmentation cost, $A_{N,0}$, equals one. For the general case, if a path splits a given block $b_{l,i}$ then a segmentation cost of $A_{l,i} + 4$ bits has to be added to its overall cost function, since by splitting block $b_{l,i}$, 4 bits will be needed to encode whether the four child nodes of block $b_{l,i}$ are split or not. Since the path only visits trellis nodes and not QT nodes, this cost has to be distributed to the trellis nodes of the child nodes of block $b_{l,i}$. How the cost is split among the child nodes is arbitrary since by the recursive definition of the $F_{l,i}$ and $T_{l,i}$ sets, every path which visits a sub-tree rooted by one child node, also has to visit the other three sub-trees rooted by the other child nodes. Therefore the path will pick up the segmentation cost, no matter how it has been distributed among the child nodes. Since the splitting of a node at level $n_0 + 1$ leads to four child nodes at level n_0, which can not be split further, no segmentation cost needs to be distributed among its child nodes.

These segmentation costs can be generated recursively in a Top-Down fashion by calling the function "seg(l, i)" below with the following parameters: seg(N,0). The recursion needs to be initialized with $A_{N,0} = 1$.

```
seg($l, i$)
    $A_{l-1,4*i+0} = A_{l,i} + 4$; $A_{l-1,4*i+1} = 0$; $A_{l-1,4*i+2} = 0$; $A_{l-1,4*i+3} = 0$;
    if $(l - 1 > n_0)$
        seg$(l-1, 4*i+0)$; seg$(l-1, 4*i+1)$; seg$(l-1, 4*i+2)$; seg$(l-1, 4*i+3)$;
```

Note that in the above function, the segmentation cost is distributed along the leftmost child. As mentioned before, any other assignment of the segmentation cost will lead to the same result. Furthermore, since in the Lagrangian cost function the rate and distortion are merged by adding the rate, weighted by the Lagrangian multiplier, to the distortion, the segmentation rate also needs to be weighted by λ.

The recursion involved in the assignment of the encoding cost is illustrated in Fig. 10. Having established the multilevel trellis, the forward DP algorithm can be used to find the optimal state sequence $x_0^*, \ldots, x_{N_\mathcal{T}-1}^*$ which will minimize the unconstrained problem (5). The Viterbi algorithm simply finds the shortest path from S to T, where the distance is measured as the sum of the node distortions $d(x_j)$ plus the sum of the weighted segmentation and DVF encoding rates, $\lambda * r(x_{j-1}, x_j)$. Hence the Viterbi algorithm finds the optimal solution to the following minimization problem,

$$\min_{x_0,\ldots,x_{N_\mathcal{T}-1}} \left(\sum_{j=0}^{N_\mathcal{T}-1} \left(d(x_j) + \lambda * r(x_{j-1}, x_j)\right) \right), \tag{14}$$

which is equivalent to the unconstrained problem (5).

6.2 Dynamic programming recursion formula

In the previous section, the multilevel trellis has been established which is a graphical equivalent of the DP recursion formula which is derived in this section. Again, each node x_j of the multilevel trellis contains $d(x_j)$, the distortion (DFD energy) resulting by predicting the corresponding block $b_{l,i}$ using the motion vector $m_{l,i}$, which together with the level l and block number i defines the node. As shown in the previous section, each node also contains $R^{SEG}(x_j) = A_{l,i}$, which is the recursively distributed QT encoding cost. In addition to that, the transition cost $R^{DVF}(x_{j-1}, x_j)$ is the number of bits required for the encoding of the DPCM error between node x_{j-1} and node x_j, where $x_{j-1} \in F_{n_0,j-1}$ and $x_j \in T_{n_0,j}$. The dependency between the blocks is expressed by this term and it is the reason for using DP to solve the optimization problem. Note that $r(x_{j-1}, x_j)$ in Eqs. (2) and (14) equals,

$$r(x_{j-1}, x_j) = R^{SEG}(x_j) + R^{DVF}(x_{j-1}, x_j). \tag{15}$$

Let $j_\lambda(x_{j-1}, x_j)$ be the Lagrangian block cost function which is defined as follows,

$$j_\lambda(x_{j-1}, x_j) = d(x_j) + \lambda * r(x_{j-1}, x_j). \tag{16}$$

Hence the total Lagrangian cost function can be calculated as follows,

$$J_\lambda(x_0, \ldots, x_{N_\mathcal{T}-1}) = \sum_{j=0}^{N_\mathcal{T}-1} j_\lambda(x_{j-1}, x_j). \tag{17}$$

The goal of DP is to find the optimal state sequence $x_0^*, \ldots, x_{N_\mathcal{T}-1}^*$ which will minimize Eq. (5), which is equivalent to minimizing Eq. (17). Let $j_{\lambda,k}^*(x_k)$ be the minimum Lagrangian cost up to and including epoch k,

$$j_{\lambda,k}^*(x_k) = \min_{x_0,\ldots,x_{k-1}} \sum_{j=0}^{k} j_\lambda(x_{j-1}, x_j), \tag{18}$$

where the epochs count the blocks $b_{n_0,k}$, hence $4^{N-n_0} - 1 \geq k \geq 0$, and $x_{-1} = S$, the auxiliary starting node. The motivation behind this definition is the fact that $j_{\lambda,4^{N-n_0}}^*(T)$ is the minimum cost required to traverse the trellis from (S) to (T), which is equivalent to minimizing $J_\lambda(\cdot)$ in Eq. (17). From Eq. (18) follows that,

$$\begin{aligned} j_{\lambda,k+1}^*(x_{k+1}) &= \min_{x_0,\ldots,x_k} \sum_{j=0}^{k+1} j_\lambda(x_{j-1}, x_j) && (19)\\ &= \min_{x_k} \left\{ \min_{x_0,\ldots,x_{k-1}} \left(\sum_{j=0}^{k} j_\lambda(x_{j-1}, x_j) + j_\lambda(x_k, x_{k+1}) \right) \right\} && (20)\\ &= \min_{x_k} \left\{ \min_{x_0,\ldots,x_{k-1}} \left(\sum_{j=0}^{k} j_\lambda(x_{j-1}, x_j) \right) + j_\lambda(x_k, x_{k+1}) \right\}, && (21) \end{aligned}$$

which results in the DP recursion formula,

$$j_{\lambda,k+1}^*(x_{k+1}) = \min_{x_k} \left(j_{\lambda,k}^*(x_k) + j_\lambda(x_k, x_{k+1}) \right). \tag{22}$$

Having established the DP recursion formula, the forward DP algorithm can be used to find the optimal state sequence. Note how the $F_{n_0,i}$ and $T_{n_0,i}$ sets are used to establish the connections between the layers of the multilevel trellis. The utilization of these sets constrains the admissible paths to represent a valid QT.

1. Initialization: for all $x_0 \in T_{n_0,0}$ do,

   $$j_{\lambda,0}^*(x_0) = j_\lambda(S, x_0), \tag{23}$$

   where S represents the starting node, i.e., an auxiliary node representing the convention for starting the DPCM. A pointer, $i(x_k)$ is introduced, which is used to keep track of the optimal path,

   $$i(x_0) = S. \tag{24}$$

2. Recursion: for $k = 0, \ldots, 4^{N-n_0} - 2$ and for all $x_{k+1} \in T_{n_0,k+1}$ do,

$$j^*_{\lambda,k+1}(x_{k+1}) = \min_{x_k \in F_{n_0,k}} (j^*_{\lambda,k}(x_k) + j_\lambda(x_k, x_{k+1})), \tag{25}$$

$$i(x_{k+1}) = \arg \min_{x_k \in F_{n_0,k}} (j^*_{\lambda,k}(x_k) + j_\lambda(x_k, x_{k+1})), \tag{26}$$

3. Termination:

$$j^*_{\lambda,4^{N-n_0}}(T) = \min_{x_{4^{N-n_0}-1} \in F_{n_0,4^{N-n_0}-1}} (j^*_{\lambda,k}(x_{4^{N-n_0}-1}) + j_\lambda(x_{4^{N-n_0}-1}, T)), \tag{27}$$

where T is an auxiliary terminal node used to find the minimum of all optimal cost functions at epoch $4^{N-n_0} - 1$, since for our purpose, $j_\lambda(x_{4^{N-n_0}-1}, T) = 0$. Again the back pointer is calculated

$$i(T) = \arg \min_{x_{4^{N-n_0}-1} \in F_{n_0,4^{N-n_0}-1}} (j^*_{\lambda,k}(x_{4^{N-n_0}-1}) + j_\lambda(x_{4^{N-n_0}-1}, T)). \tag{28}$$

4. Backtracking: the pointer i is used to backtrack the optimal state sequence.

$$\begin{aligned} \text{initialize:} &\quad k = N_{\mathcal{T}}, \quad x^*_k = T \\ \text{do:} &\quad k \leftarrow k-1, \quad x^*_k = i(x^*_{k+1}) \\ \text{until:} &\quad x^*_k = S. \end{aligned}$$

This leads to the optimal state sequence $x^*_0, \ldots, x^*_{N_{\mathcal{T}}-1}$.

In Fig. 10, a QT of depth 4 is displayed and the optimal state sequence is indicated which leads to the segmentation shown in Fig. 11. Note that the resulting scanning path is efficient and the segmentation cost along the optimal path adds up to 13 bits, which is the number of bits needed to encode this QT decomposition. The bit stream for this QT decomposition is "1010000011001". The time complexity of DP is $O_{DP}(4^{N-n_0} * |S_{n_0,i}|^2)$ which is significantly smaller than the lower bound for the exhaustive search $\Omega_E(|S_{n_0,i}|^{4^{N-n_0}})$. Note that we again have assumed that all the $S_{l,i}$ sets are of the same cardinality.

7 Motion compensated interpolation

In this section we propose a scheme for motion compensated interpolation which uses a similar mathematical formulation as the optimal QT-based motion estimator. Besides being a very general problem in digital video research, motion compensated interpolation can be used in the context of very low bit rate video coding to increase the displayed frame rate. In very low bit rate coding, it is very common to down-sample the original sequence in the temporal direction. In fact the most popular frame rates are 7.5 and 10 frames per second, or in other words, only every fourth or third frame of the original 30 frames per second sequence is encoded.

The down-sampling results in some jerkiness since the human observer starts to notice the single frames. The displayed frame rate can be increased by interpolating missing frames using motion compensated interpolation. The problem with motion compensated interpolation is that if an error is made, it will be clearly visible in the sequence since human observers are very sensitive to irregularities along the temporal direction. As pointed out before, the QT-based motion estimator results in a motion vector field which is very close to the actual field and therefore a similar mathematical formulation can be used for the motion compensated interpolation.

7.1 Regular grid approach

The idea behind the proposed motion compensated interpolation scheme can be explained by considering Fig. 12. In it the left most frame $\hat{f}_0$ and the right most frame $\hat{f}_N$ are the available frames (reconstructed frames at the decoder). The frames between these two frames are the ones we wish to interpolate. For example, in Fig. 12, frame 1 is the frame being interpolated. The most common approach for motion compensated interpolation is to find the DVF between the left most and right most frames and then use this motion information to interpolate the frames in between. One problem with this approach is that the DVF needs to be spatially interpolated for the interpolated frames, since one cannot guarantee that the DVF will fall onto grid points in these frames. In the proposed approach, we avoid this problem of an irregular grid since the frame to be interpolated is segmented (instead of the left most or right most frame as in traditional approaches), using a QT decomposition and the DVF is found for this frame.

7.2 Motion compensated interpolation

As depicted in Fig. 12, we assume non-accelerated motion (also called linear motion since the trajectory is a straight line). By segmenting the frame which has to be interpolated, we can assign a motion vector m_j to each block $b_{l,i}$ in the segmented frame. The motion vector is with respect to the closest previous frame (determined by the temporal resolution) in the sequence, even if the previous frame is not one of the reconstructed frames (this will become clearer later on). The QT segmentation and the motion vector, which are found using a procedure described in Sec. 7.3, are then used to interpolate block $b_{l,i}$ of the k-th frame as follows,

$$b_{l,i}(x,y) = \frac{N_2 * \hat{f}_0(x - N_1 * m_{j,x}, y - N_1 * m_{j,y}) + N_1 * \hat{f}_N(x + N_2 * m_{j,x}, y + N_2 * m_{j,y})}{N_1 + N_2}, \ \forall\, (x,y) \in b_{l,i}, \tag{29}$$

where $m_{j,x}$ and $m_{j,y}$ are the x and y coordinates of the motion vector m_j of block $b_{l,i}$, and N_1 and N_2 are the temporal distances of the frame which is being interpolated from frames $\hat{f}_0$ and $\hat{f}_N$, respectively (see Fig. 12). This linear interpolation leads to a fading effect so that the transmitted frame closer to the interpolated frame has a higher weight than the transmitted frame which is farther away.

7.3 Operational rate distortion formulation

Using the above interpolation formula, it is clear that the quality of the interpolated frame only depends on the QT segmentation and the DVF. As we pointed out in the previous section, the QT-based motion estimator results in very good DVFs and hence we employ a similar mathematical formulation to find the QT segmentation and the DVF such that the motion compensated interpolation results in a visually pleasing sequence. To this end we define a block interpolation error (distortion) of block $b_{l,i}$ using motion vector m_j (which is contained in node x_j) as follows,

$$d_j(x_j) = \sum_{(x,y)\in b_{l,i}} \left(\hat{f}_0(x - N_1 * m_{j,x}, y - N_1 * m_{j,y}) - \hat{f}_N(x + N_2 * m_{j,x}, y + N_2 * m_{j,y})\right)^2 . \tag{30}$$

This formula expresses the fact that the scheme looks for the same block in $\hat{f}_0$ and $\hat{f}_N$ which must pass through the block under consideration in the interpolated frame. This is also indicated in Fig. 12. As we pointed out before, the motion vector m_j is with respect to the previous frame in the sequence, hence it needs to be scaled by the temporal distances N_1 and N_2 in the above formula. Again the frame distortion is defined as the sum of the distortions of the blocks used in the QT decomposition, hence,

$$D(x_0, \ldots, x_{N_{\mathcal{T}}-1}) = \sum_{j=0}^{N_{\mathcal{T}}-1} d(x_j). \tag{31}$$

Intuitively it seems clear that if we find a QT and a DVF such that the frame distortions are minimized, then the interpolated frame should be as good as possible under the linear motion assumption. This, however, is not the case since the smallest frame distortion can be achieved by using one motion vector for each of the smallest possible blocks in the QT decomposition. These motion vectors would be selected so that the individual block distortions are minimized, which in turn results in the smallest frame distortion. The resulting DVF, however, would be quite noisy, since the ultimate goal is to minimize the above distortion. Clearly, minimizing only the frame distortion will not result in a good interpolation, since we did not take into account the smoothness of the resulting DVF. It is well known that a real DVF is quite smooth, except at object boundaries and in uncovered regions. Therefore it is common in motion estimation not to minimize the DFD energy alone, but to regularize the problem by adding a term to the objective function which measures the smoothness of the DVF. This can be easily achieved in the proposed framework, since the rate necessary to transmit the DVF is known by the QT-based motion estimator. As we have pointed out before, a smooth DVF results is a small rate, whereas a noisy DVF requires a high bit rate. This follows from the first order DPCM employed to encode the DVF. Additionally, the Hilbert scan enforces the smoothness of the DVF in all dimensions, which is one of the advantages of a Hilbert scan over a raster scan. Therefore we propose to find the DVF and the QT decomposition as the solution to the following regularized optimization problem,

$$\min_{x_0,\ldots,x_{\mathcal{T}-1}} (D(x_0, \ldots, x_{\mathcal{T}-1}) + \lambda * R(x_0, \ldots, x_{\mathcal{T}-1})), \tag{32}$$

where λ is the regularization parameter. Clearly, this is exactly the same optimization problem (see Eq. (5)) which we have been solving iteratively for different λs to find the optimal DVF and QT decomposition for the optimal QT-based motion estimator. Hence we can find the optimal solution to this regularized optimization problem using the approach of Sec. 6.

8 Experimental results

In this section we discuss a specific implementation of the proposed QT-based motion estimator and present experimental results. We compare the rate distortion performance of the proposed motion estimator with block matching, and compare the results of the interpolation scheme with the original frames.

8.1 Implementation of the optimal QT-based motion estimator

As in test model four (TMN4) [28] of H.263 we assign one bit per block to indicate if the motion vector of this block is zero, since this is the most common event. When the motion vector is non-zero, we encode the motion vector difference between the current and the previous block using the entropy table of TMN4. Efficient motion estimation is particularly important for very low bit rate video coding, since the encoding of the DVF can take up to 100% of the available bit rate. The main format for very low bit rate video coding is the quarter common intermediate format (QCIF). Therefore we select QCIF video sequences, which are of dimensions 176 × 144, for the presented experiments. We have developed the Hilbert curve based scanning path for frames which are of dimension $2^i \times 2^i$ pixels, where i is a positive integer. Clearly we have to modify the Hilbert scan for QCIF sequences. Figure 13 shows the modified Hilbert scan which we use for the optimal motion estimator, where the smallest block size is 8×8 ($n_0 = 3$). Note that the square of length 128 in the left lower corner in Figure 13 is still scanned using a perfect Hilbert curve of order 4. The rest of the image is then extended in a pattern which imitates the Hilbert scan. The resulting tree representation of the entire frame is therefore not a simple QT, but a pruned QT, where the blocks which do not belong to the tree structure are pruned off. Note that the presented algorithm does not depend on the number of children, hence it works equally well with a pruned QT as it does with a full QT or for any tree structure which represents image regions.

The modified Hilbert scan which we use for the motion compensated interpolation scheme where the smallest block size is 4×4 ($n_0 = 2$), can recursively be derived using the rule introduced in Sec. 4.3 and the modified Hilbert scan of level 3 which is displayed in Figure 13.

8.1.1 Recursive generation of admissible motion vector sets

From a theoretical point of view, every possible motion vector of block $b_{l,i}$ should be included in the set $M_{l,i}$, which is the admissible motion vector set for block $b_{l,i}$. This means that for a typical search window of ±15.5 pixels and an accuracy of 0.5 pixel, $|M_{l,i}|$ = 63*63 =3969, which is quite large. Most of these motion vectors, however, are not likely candidates for the optimal path, since they do not correspond well to the real motion in the scene and therefore they lead to a high distortion and a high rate. These motion vectors can be found by performing block matching since they will result in a high matching error.

To make the optimization process faster, the prior knowledge about these motion vectors is taken into account. Even though this might be complicated in general, it is easily achieved in the presented framework of DP by reducing the set $M_{l,i}$ of admissible motion vectors of block $b_{l,i}$.

We propose the following scheme for the recursive generation of admissible motion vector sets. An initial motion vector search is conducted for the $2^{n_0} \times 2^{n_0}$ blocks at level n_0 by using block matching with integer accuracy. The K integer motion vectors which lead to the best prediction are kept. Then the set $M_{n_0,i}$ is defined as the set which contains the K top integer motion vectors plus their half pixel neighbors.

After the set of admissible motion vectors has been defined for the bottom level (n_0), the sets of admissible motion vectors for higher level blocks are defined recursively. A block $b_{l,i}$ only includes a motion vector in $M_{l,i}$ if this motion vector has been selected by all of its child nodes. This leads to the fact that for small blocks, many motion vectors are considered but the bigger the block, the smaller the number of vectors associated with it. This reflects the well known fact that small block sizes lead to small energy in the DFD but not very consistent motion vector fields, whereas bigger blocks lead to consistent vector fields, but the energy in the DFD can be quite high [29]. Our experiments have shown that for $K = 10$ this restriction of the search space does not lead to a performance loss but increases the speed of the algorithm significantly.

8.1.2 Experiments

In these experiments we compare the QT-based optimal motion estimation scheme with the fixed block size (16 × 16) based scheme of TMN4. In the first experiment, we use two frames from the "Mother and Daughter" sequence and compare the performance of the block matching and the proposed algorithm visually. In the second experiment, we use three different sequences, with different amount of motion, at three different frame rates and compare the performance of the block matching algorithm and the proposed scheme on an average basis.

Visual Comparison As pointed out before, we use the same encoding technique for the DVF as TMN4 and hence the difference in performance stems from the optimal tradeoff between the rate necessary to encode the inhomogeneous DVF (QT segmentation and motion vector differences) and the resulting energy in the DFD (the distortion). For the first experiment, we use the 176-th and 180-th frames of the QCIF sequence “Mother and Daughter”, which are the leftmost and right most frames in Fig. 18, respectively. Note that we skip three frames, which corresponds to a resulting frame rate of 7.5 frames/second (we assume the original sequence was recorded at 30 frames/second). We only use the luminance part of the sequence to perform the experiments, but the presented theory also covers the case when the chrominance distortion is included in the distortion measure. It therefore provides an elegant approach for fusing the information provided by different channels.

The block distortion $d(x_j)$ can be defined arbitrarily and hence it can contain contributions from the luminance channel as well as from the two chrominance channels (or from R, G and B channels). In the most general form, the block distortion can be written as follows,

$$d(x_j) = \phi(d^Y(x_j), d^{Cb}(x_j), d^{Cr}(x_j)), \tag{33}$$

where $\phi(\cdot)$ is an arbitrary function and the superscripts Y, Cb and Cr indicate the distortions in the luminance and chrominance channels. One popular choice for the function $\phi(\cdot)$ is a weighted sum,

$$d(x_j) = d^Y(x_j) + \alpha * d^{Cb}(x_j) + \beta * d^{Cr}(x_j), \tag{34}$$

since for this definition of the block distortion, the frame distortion is also the weighted sum of the luminance and chrominance frame distortions. Clearly the selection of an appropriate α and β has to be done experimentally. Note that by defining the block distortion as above, the inhomogeneous DVF is found optimally, using the information from all three channels. Nevertheless, in the presented results, we set α and β to zero.

We use the mean squared error (MSE) as the distortion measure, and employ the peak signal to noise ratio (PSNR) to express its magnitude in dB, (PSNR $= 10 * \log_{10}(255^2/\text{MSE})$). We intra code the 176-frame using TMN4 and a quantizer step size of 10. The resulting PSNR for this frame is 33.85 dB . Then the original 180-th frame is used to find the DVF. First we employ the TMN4 block matching scheme which uses the sum of the absolute error and favors the zero motion vector by reducing its error by a constant amount of 100. The resulting bit rate for the DVF and the DFD energy are listed in row "TMN4" in Table 2. The predicted frame and the DVF are displayed in Fig. 14.

Now the QT-based optimal motion estimator is run, where first the maximum rate is set equal to the TMN4 rate, i.e., $R_{max} = 470$ bits. We call this experiment "matched Rate" in Table 2 and the resulting rate and distortion are listed in the corresponding row. Note that for the same rate, the total distortion is reduced significantly, or in other words, a better prediction is achieved. Besides outperforming the TMN4 motion estimator in the objective sense, the proposed QT-based scheme also outperforms the TMN4 motion estimator subjectively. This is due to the inhomogeneous representation of the motion by means of the QT, which enables the QT-based motion estimator to spend more bits in areas with complex motion, i.e., small block sizes are used, and fewer bits in areas with simple motion, i.e., large block sizes are used. The better prediction performance is apparent from the resulting predicted frame and the DVF which are displayed in Fig. 15. For example, the prediction of the left eye and the shirt collar of the mother and the right corner of the frame in the background, is clearly better in the proposed approach. Recall that the

n_0 level is scanned by the modified Hilbert scan shown in Fig. 13. To generate the scanning path for an arbitrary QT decomposition, the procedure introduced in Sec. 4.2 is used. In Fig. 16 the resulting overall scanning path which corresponds to the QT decomposition displayed in Fig. 15 is shown.

For the second part of the first experiment, the DFD energy of the TMN4 run is matched by setting $D_{max} = 30.6$ dB PSNR. We call this experiment "matched Dist." in Table 2 and the resulting rate and distortion are listed in the corresponding row. Note that for the same distortion, the bit rate is reduced significantly (about 30%). The resulting predicted frame and the DVF are displayed in Fig. 17. Again, even though the DFD energy is the same as in Fig. 14, the predicted frame is of higher quality. The same explanation applies as before, that is, the implicit DVF smoothing along the Hilbert scan results in a good DVF. In addition, the inhomogeneous structure of the DVF fits a real DVF better and hence a better representation of the real DVF can be achieved.

Average Comparison While the previous experiment lends itself for visualization, it is not sufficient to establish the performance of the proposed scheme. For that purpose we run a second experiment, where we use three different sequences, "Miss America", a low motion sequence, "Mother and Daughter" a medium motion sequence, and "Foreman" a high motion sequence. All sequences are 200 frames long, except the "Miss America" sequence, which is 150 frames long. In very low bit rate (VLBR) video coding, the most common frame rates are 7.5, 10 and 15 frames/second. This corresponds to skipping 3, 2 or 1 frames of the original sequence, where we assume that the original sequence was recorded at 30 frames/second. Clearly, the more frames skipped, the harder the motion estimation problem. The setup of this experiment is basically identical to the previous experiment, except that we encode many more frames of three different sequences at three different frame rates. For example, for the "Foreman" sequence at 10 frames/second, we first run the TMN4 block matching algorithm which results in a rate and a distortion for every third frame. This rate distortion pair is obtained by first intra coding the frame which is used to predict the current frame from, using a quantizer step size of 10, and then predicting the current frame using the motion vectors from the TMN4 block matching algorithm. The recorded rate for the current frame is then the bits used to encode the DVF and the distortion of the current frame is the MSE between the original frame and the predicted frame. Having this rate distortion pair for every frame in the temporally downsampled sequence, we then run the optimal QT based motion estimator, in the matched rate and the matched distortion mode. Doing this for all the sequences at all the frame rates results in Table 3. Clearly the proposed motion estimator outperforms the TMN4 block matching algorithm in the rate distortion sense. For the matched rate case, the average increase in PSNR (decrease in DFD energy) is about 0.4 dB and for the matched distortion case the average decrease in rate is about 25%. Note that Table 3 also shows that we have very good control over the rate distortion tradeoff with the proposed scheme, since we can hit any given rate distortion pair with very high accuracy. We have discussed the time complexity of the multilevel dynamic programming in Section 6.2 and introduced a fast recursive algorithm for the selection of the admissible motion vector sets in Section 8.1.1. Using these algorithms on our Sparc 20 requires on average 11.6 seconds, whereas the TMN4 block matching scheme requires 5.2 seconds. These 11.6 seconds are the result of three iterations (on average , due to the very fast convex search from Section 5.2) of the multilevel dynamic program, which takes 1.6 seconds per iteration, and the filling of the trellis, which takes 6.8 seconds.

8.2 Motion compensated interpolator

The compressed QCIF sequence, "Mother and Daughter" (luminance only) is again used for this experiment. This sequence has been compressed by a video compression scheme proposed in [6]. In this scheme it is possible to fix the frame distortion for each frame, and we set it to 34.0 dB PSNR. The compressed frame rate of the sequence is 15 frames per second (fps). For the presented experiments, we down-sample the 15 fps sequence temporally by skipping every second frame. Clearly, this results in a new sequence with a frame rate of 7.5 fps. We use the proposed motion compensated interpolation scheme to increase this frame rate back to 15 fps. We then compare the reconstructed frames with the interpolated frames visually and with the use of the PSNR metric.

For the implementation of the proposed motion compensated interpolation algorithm, the same QT-based motion estimator is used as in the experiments presented in the previous section. Based on experiments, we select λ 0.01. In addition, the smallest level of the QT are 4×4 blocks, hence n_0 is equal to 2, which is different from the QT based optimal motion estimator presented in the previous section where n_0 was equal to 3.

The sequence, when interpolated to 15 fps, is much more pleasing to a human observer than the 7.5 fps sequence, since it is less jerky. To illustrate the interpolation results, we display a sequence of three reconstructed frames, i.e., consecutive frames of the 15 fps sequence and compare them with the same frames resulting from the interpolation scheme. Clearly the first and last frame are identical, whereas the interpolated middle frame should be as close as possible to the middle frame of the reconstructed sequence.

In Fig. 18 the 176-th, 178-th and 180-th reconstructed frames are displayed. In Fig. 19 the 176-th and 180-th reconstructed frames are displayed on the left and on the right, respectively. The 178-th frame is interpolated using the proposed motion compensated interpolation scheme. Clearly the interpolated frame is very similar to the encoded frame. Recall that the interpolation is done at the decoder and is completely independent from the encoder. The PSNR of the interpolated frame is 33.3 dB which is 0.7 dB lower than that of the reconstructed frame. On the other hand, to transmit the reconstructed frame, 3254 bits where necessary. In Fig. 20 the QT segmentation and the DVF are displayed together with the interpolated frame.

9 Summary

In this paper we presented an optimal QT-based motion estimator which is based on the Lagrangian multiplier method and multilevel DP. The main feature of this estimator is that the QT-segmentation and the DVF are found jointly and optimally. The inhomogeneous representation of the DVF results in an efficient encoding of the DVF and in DVF estimates which are close to the real DVF. We exploited this fact by proposing a motion compensated interpolation scheme which uses a similar mathematical formulation as the QT-based motion estimator. This scheme allows for the globally optimal control of the tradeoff between the interpolation error energy and the DVF smoothness.

We encode the inhomogeneous DVF using a first order DPCM along the scanning path. We defined the notion of an efficient scanning path and introduced a procedure, and a necessary and sufficient condition for that procedure, to infer an efficient scanning path from a given QT decomposition. We further proposed a special efficient scanning path which is based on a Hilbert curve and results in square blocks.

We formulated the motion estimation problem as a constrained optimization problem, which we solve using the Lagrangian multiplier method. We proposed a new very fast convex search to find the optimal Lagrangian multiplier λ^*, which is based on a second order Bezier curve.

Finally, we presented results of the proposed QT-based motion estimator and compared them with block matching. The results clearly demonstrated that the proposed scheme outperforms block matching significantly in an objective, as well as, a subjective sense. We also presented results of the proposed motion compensated interpolation scheme by comparing the interpolated frames with the reconstructed original frames. Again the results show that the interpolation scheme is very accurate, since the reconstructed frames and the interpolated frames are almost identical.

BIOGRAPHICAL SKETCH

Guido M. Schuster received the Ing HTL degree in Elektronik, Mess- und Regeltechnik in 1990 from the Neu Technikum Buchs (NTB), Buchs, St.Gallen, Switzerland. At the NTB, he was awarded the gold medal for academic excellence, and was also the winner of the first annual Landis & Gyr fellowship competition.

Dr. Schuster received the M.S. and Ph.D. degrees, both in Electrical Engineering, from Northwestern University, Evanston, Illinois, in 1992 and 1996, respectively. In 1996 he joined the Network Systems Division of U.S. Robotics in Mount Prospect (now the Carrier Systems Business Unit of 3COM), Illinois, where he co-founded the Advanced Technologies Research Center.

Dr. Schuster has filed and holds several patents in fields ranging from adaptive control over video compression to Internet telephony. He also is the co-author of the book "Rate-Distortion based Video Compression", published by Kluwer Academic Publishers (1997). His current research interests are Operational Rate-Distortion Theory and Networked Multimedia.

Aggelos K. Katsaggelos received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1979 and the M.S. and Ph.D. degrees both in electrical engineering from the Georgia Institute of Technology, Atlanta, Georgia, in 1981 and 1985, respectively.

In 1985 he joined the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL, where he is currently professor, holding the Ameritech Chair of Information Technology. During the 1986-1987 academic year he was an assistant professor at Polytechnic University, Department of Electrical Engineering and Computer Science, Brooklyn, NY. His current research interests include image and video recovery, video compression, motion estimation, boundary encoding, computational vision, and multimedia signal processing. Dr. Katsaggelos is a Fellow of the IEEE, an Ameritech Fellow, a member of the Associate Staff, Department of Medicine, at Evanston Hospital, and a member of SPIE. He is a member of the Steering Committees of the *IEEE Transactions on Medical Imaging* and the *IEEE Transactions on Image Processing*, the IEEE Technical Committees on Visual Signal Processing and Communications, Image and Multi-Dimensional Signal Processing, and Multimedia Signal Processing, the Technical Chamber of Commerce of Greece and Sigma Xi. He has served as an Associate editor for the *IEEE Transcations on Signal Processing* (1990-1992), an area editor for the journal *Graphical Models and Image Processing* (1992-1995), and he is currently the editor-in-chief of the *IEEE Signal Processing Magazine.* He is the editor of *Digital Image Restoration* (Springer-Verlag, Heidelberg, 1991), co-author of *Rate-Distortion Based Video Compression* (Kluwer Academic Publishers, 1997), and co-editor of *Recovery Techniques for Image and Video Compression,* (Kluwer Academic Publishers, 1998, forthcoming). He has served as the General Chairman of the 1994 Visual Communications and Image Processing Conference (Chicago, IL), and he will serve as the technical program co-chair of the 1998 IEEE International Conference on Image Processing (Chicago, IL).

References

[1] "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbits/s." International Standard ISO/IEC IS-11172, Oct. 1992.

[2] "Generic coding of moving pictures and associated audio." International Standard ISO/IEC IS-13818, Nov. 1994.

[3] ITU-T Recommendations H.261, *Video codec for audivisual services at p × 64 kbits,*.

[4] Expert's group on very low bit rate video telephony, ITU-Telecommunication standardization sector, *Draft Recommendation H.263.*

[5] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression, Optimal Video Frame Compression and Object Boundary Encoding.* Kluwer Academic Publishers, 1997.

[6] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation among segmentation, motion and residual error," *IEEE Transactions on Image Processing*, vol. 6, pp. 1487–1502, Nov. 1997.

[7] G. M. Schuster and A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference," *IEEE Journal on selected areas in communications*, vol. 15, Dec. 1997.

[8] B. Girod, "Rate-constrained motion estimation," in *Proceedings of the Conference on Visual Communications and Image Processing*, vol. 2308, pp. 1026–1034, SPIE, 1994.

[9] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[10] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory*, vol. 35, pp. 299–315, Mar. 1989.

[11] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Transactions on Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[12] H. Samet, "The quad tree and related hierarchical data structures," *ACM computing surveys*, vol. 16, pp. 188–260, June 1984.

[13] H. Nicolas and C. Labit, "Region-based motion estimation using deterministic relaxation schemes for image sequence coding," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 265–268, 1992.

[14] M. R. Banham, J. C. Brailean, C. L. Chan, and A. K. Katsaggelos, "Low bit rate video coding using robust motion vector regeneration in the decoder," *IEEE Transactions on Image Processing*, vol. 3, pp. 652–665, Sept. 1994.

[15] J. Lee, "Optimal quadtree for variable block size motion estimation," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 480–483, Oct. 1995.

[16] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 2661–2664, May 1991.

[17] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Transactions on Image Processing*, vol. 3, pp. 327–331, May 1994.

[18] H. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, vol. 1, pp. 117–138, Oct. 1989.

[19] B. B. Mandelbrot, *The fractal geometry of nature.* San Francisco: W. H. Freeman and Company, 1983.

[20] J. A. Provine and R. M. Rangayyan, "Lossless compression of peanoscanned images," *Journal of Electronic Imaging*, vol. 3, pp. 176–181, Apr. 1994.

[21] B. Moghaddam, K. J. Hintz, and C. V. Steward, "Space-filling curves for image compression," in *Automatic Object Recognition*, vol. 1471, pp. 414–421, SPIE, 1991.

[22] T. Ebrahimi, F. Dufaux, I. Moccagatta, T. G. Campbell, and M. Kunt, "A digital video codec for medium bitrate transmission," in *Proceedings of the Conference on Visual Communications and Image Processing*, vol. 1605, pp. 2–15, SPIE, 1991.

[23] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399–417, 1963.

[24] G. M. Schuster and A. K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263," in *Proceedings of the Conference on Visual Communications and Image Processing*, pp. 784–795, SPIE, Mar. 1996.

[25] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, pp. 160–175, Apr. 1993.

[26] F. S. Hill, *Computer graphics.* Macmillan Publishing Company, 1990.

[27] R. G. Gallager, *Information theory and reliable communication.* Wiley, 1968.

[28] Expert's Group on Very Low Bitrate Visual Telephony, *Video Codec Test Model, TMN4 Rev1.* ITU Telecommunication Standardization Sector, Oct. 1994.

[29] M. Bierling, "Displacement estimation by hierarchical blockmatching," in *Proceedings of the Conference on Visual Communications and Image Processing*, vol. 1001, pp. 942–951, SPIE, 1988.

[30] S. Calzone, K. Chen, C.-C. Chuang, A. Divakaran, S. Dube, L. Hurd, J. Kari, G. Liang, F. H. Lin, J. Muller, and H. K. Rising, III, "Video compression by mean-corrected motion compensation of partial quadtrees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 86–96, Feb. 1997.

	10%	1%	0.1%	0.01%
arithmetic mean	4.4	7.3	10.5	12.8
geometric mean	3.2	6.2	9.5	13.8
line approx.	3.4	6.3	9.6	12.9
Bezier approx.	1.7	2.2	2.7	3.2

Table 1: Comparison between four different approaches to select the new λ in the bisection scheme. The values in the table indicate the average number of iterations necessary to converge within a given percentage of accuracy, where that percentage is shown at the top of each row.

	Rate (bits)	Dist. (PSNR)
TMN4	470	30.6
matched Rate	472	31.3
matched Dist	344	30.7

Table 2: Comparison between the TMN4 motion estimation algorithm and the proposed optimal motion estimator using the luminance values of the 176-th and 180-th frames of the QCIF sequence “Mother and Daughter”.

	Miss America						Mother and Daughter						Foreman					
	TMN4		matched R		matched D		TMN4		matched R		matched D		TMN4		matched R		matched D	
	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D
15 f/s	219	36.9	219	37.1	159	36.9	236	32.6	238	32.8	161	32.6	724	26.8	724	27.6	525	26.8
10 f/s	264	36.4	264	36.7	208	36.4	294	32.2	295	32.4	215	32.2	854	25.8	853	26.4	680	25.8
7.5 f/s	294	36.0	294	36.3	238	36.0	344	31.8	343	32.1	258	31.8	937	25.0	940	25.6	779	25.0

Table 3: Comparison between the TMN4 motion estimation algorithm and the proposed optimal motion estimator on an average basis. The comparison uses the luminance values of the the QCIF sequences ”Miss America”, “Mother and Daughter” and ”Foreman” at three different frame rates, 7.5, 10 and 15 frames/second and the table displays the average rate distortion performance for the TMN4 run and the matched rate and the matched distortion run of the optimal QT based motion estimator

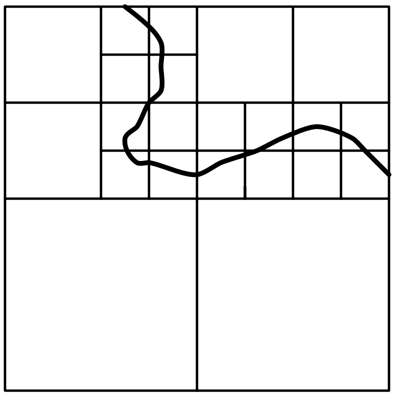


Figure 1: Frame segmented by a quad-tree.

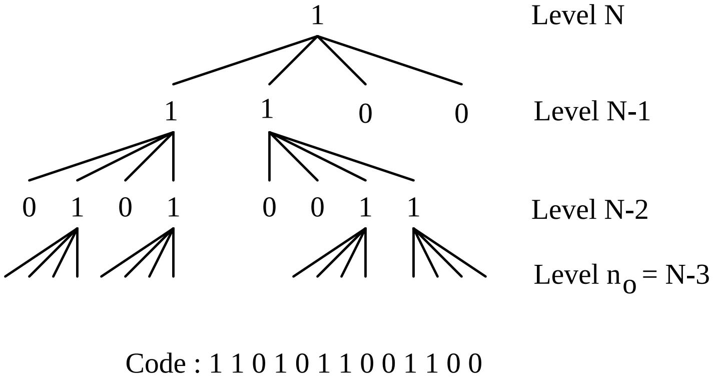


Figure 2: Quad-tree representation of the frame.

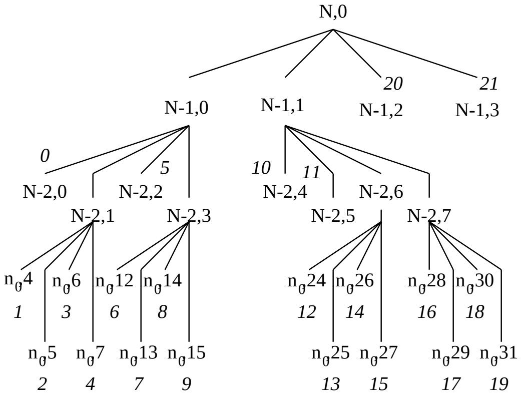

N,0
20
21
N-1,0
N-1,1
N-1,2
N-1,3
0
5
10
11
N-2,0
N-2,2
N-2,4
N-2,6
N-2,1
N-2,3
N-2,5
N-2,7
$n_0$4 $n_0$6 $n_0$12 $n_0$14 $n_0$24 $n_0$26 $n_0$28 $n_0$30
1 3 6 8 12 14 16 18
$n_0$5 $n_0$7 $n_0$13 $n_0$15 $n_0$25 $n_0$27 $n_0$29 $n_0$31
2 4 7 9 13 15 17 19

Figure 3: Quad-tree notation. Each node is identified by the ordered pair (l, i), where l is the level and i the number within that level. Also the leafs are numbered from zero to the total number of leafs in the QT minus one.

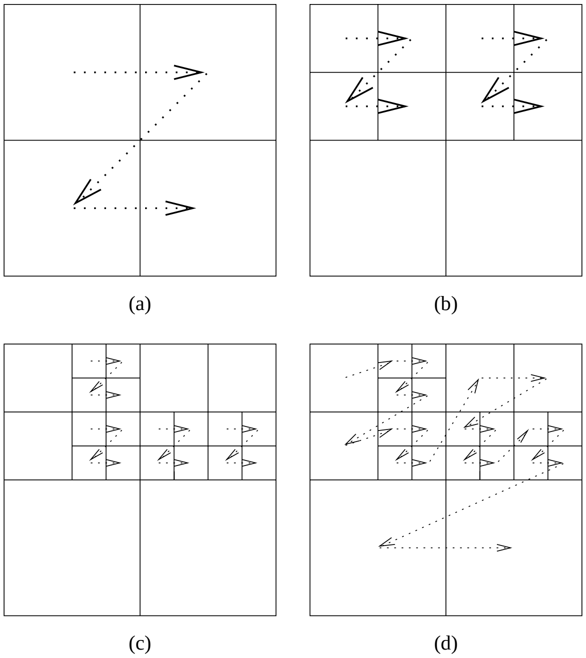
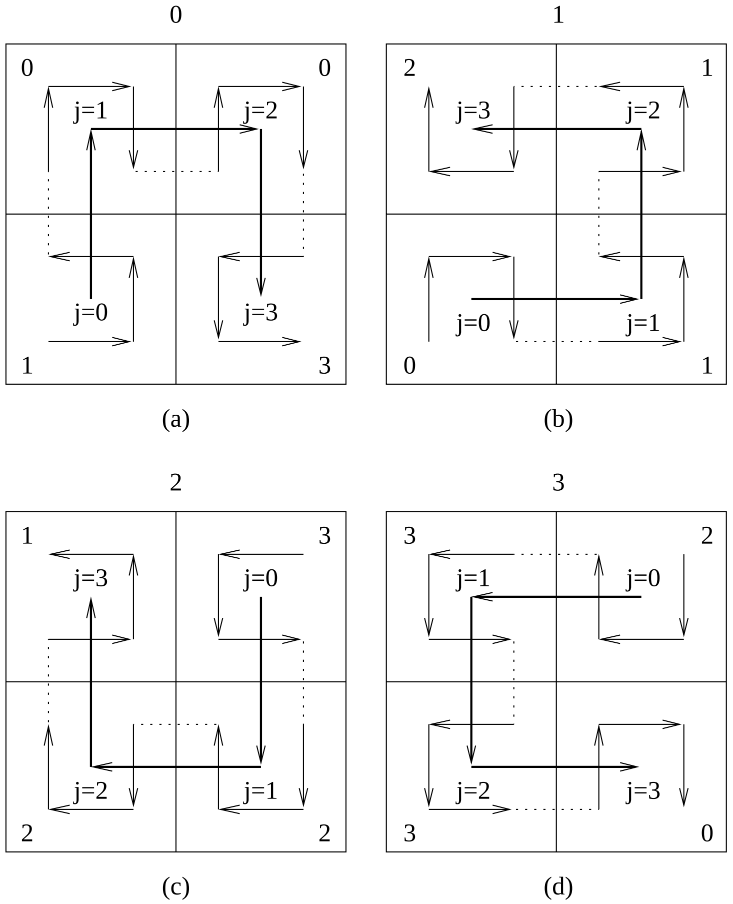


Figure 4: Recursive raster scan for a QT decomposition. Figures (a) through (c) show the raster scan at different levels of the QT and figure (d) shows the resulting overall scanning path.



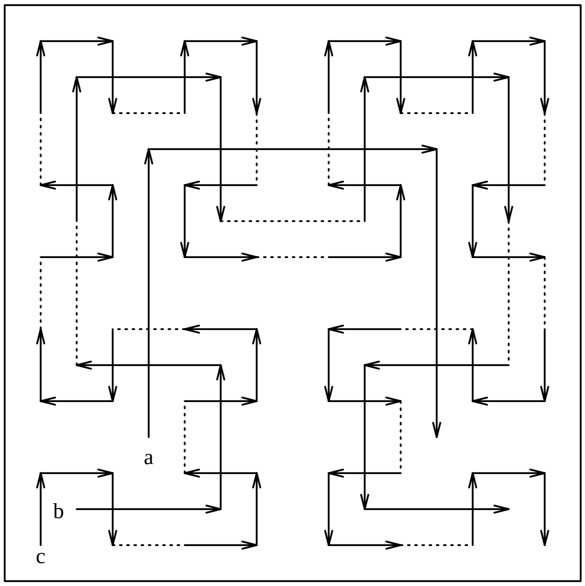


Figure 5: The four possible first order Hilbert curves (-) with the respective second order Hilbert curves underneath (-). The orientations $O_{l,i}$ are shown on top of the squares and the orientations $O_{l-1,4*i+j}$ are shown in the corners of the smaller squares.

Figure 6: Recursive Hilbert curve generation: a) first order Hilbert curve of orientation 0, b) second order Hilbert curve, c) third order Hilbert curve

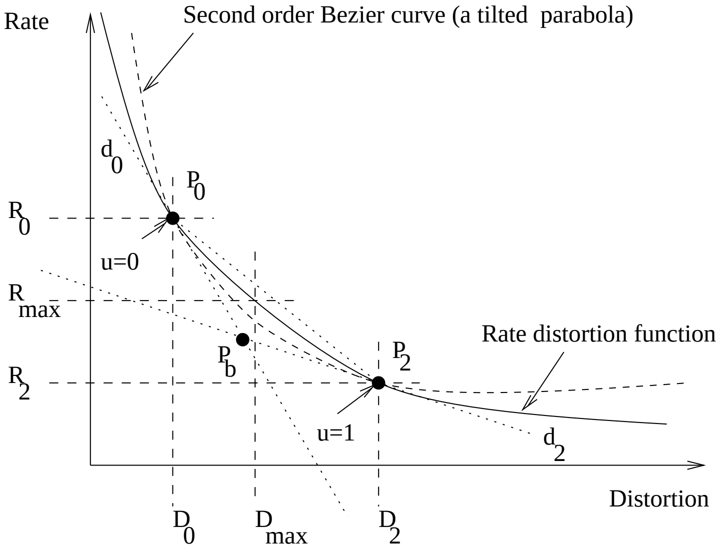


Figure 7: Continuous rate distortion curve and the second order Bezier approximation between two bracketing points.

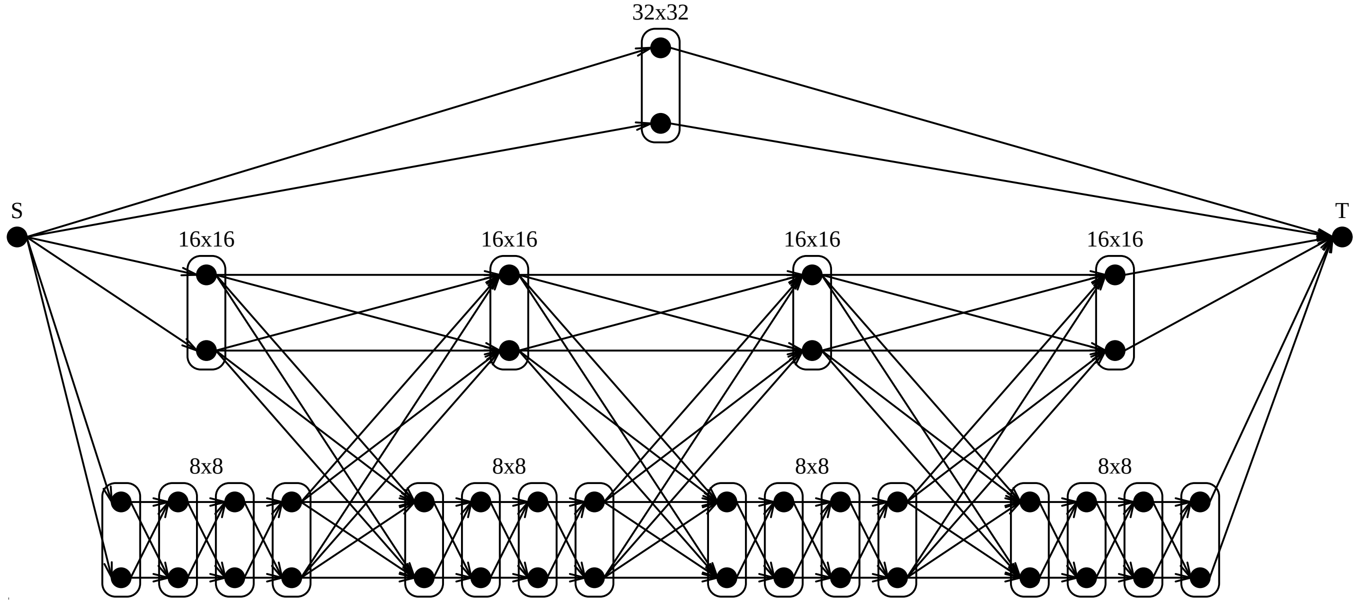


Figure 8: The multilevel trellis for $N = 5$ and $n_0 = 3$.

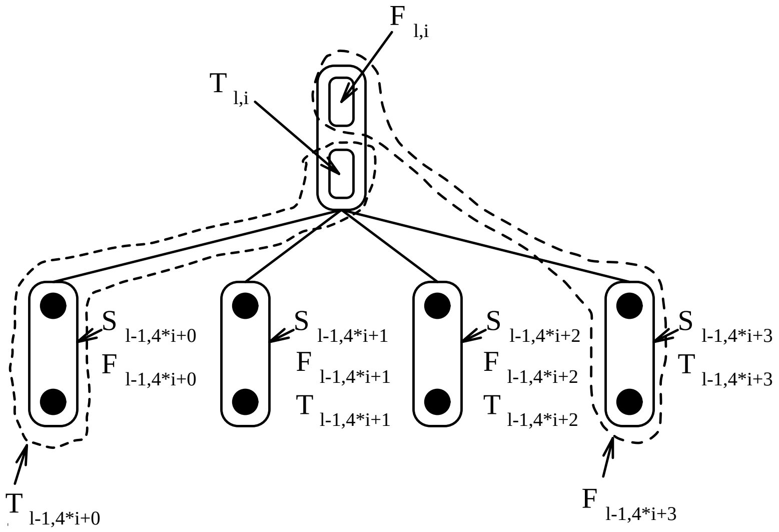


Figure 9: The recursive rule for the generation of the $T_{l,i}$ and $F_{l,i}$ sets.

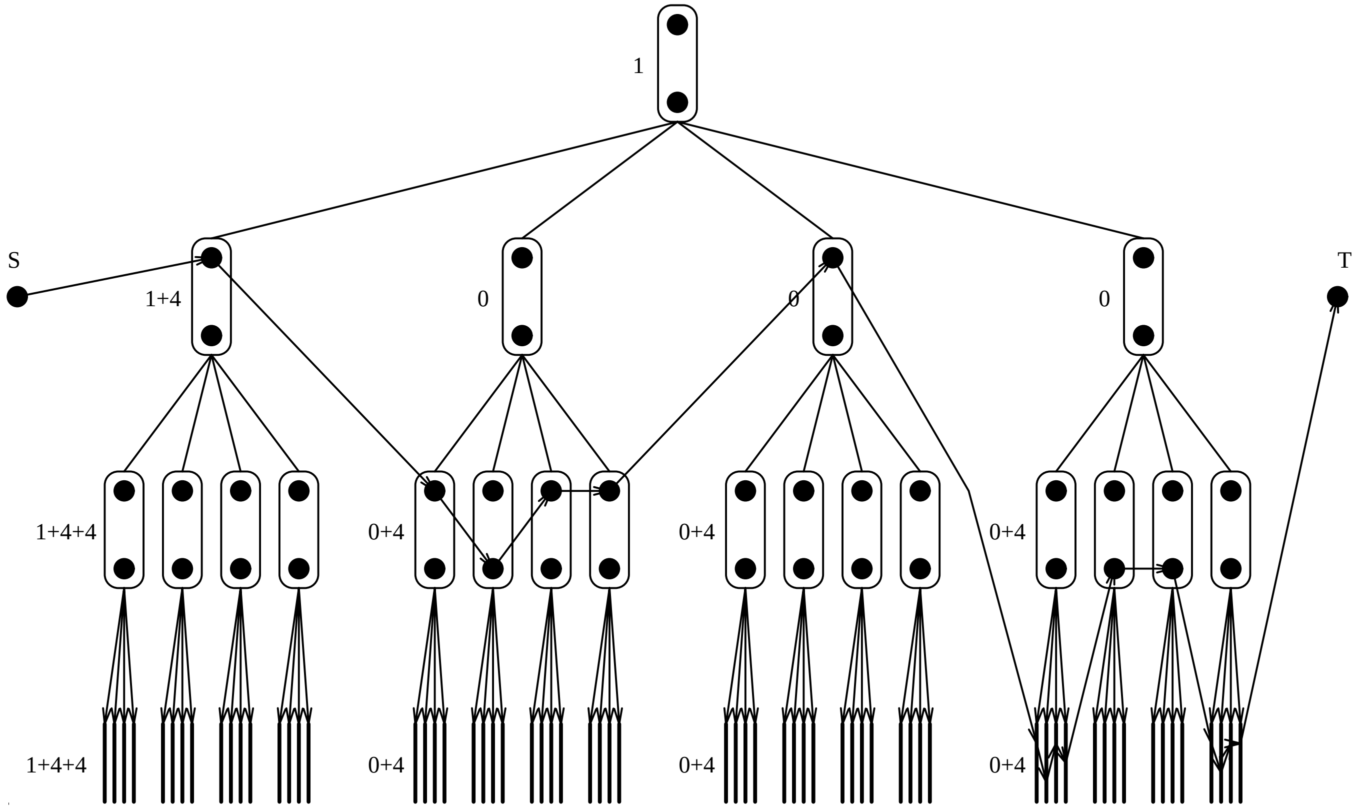


Figure 10: The recursive distribution of the quad-tree encoding cost among the trellis nodes for a quad-tree of depth 4 and an optimal path.

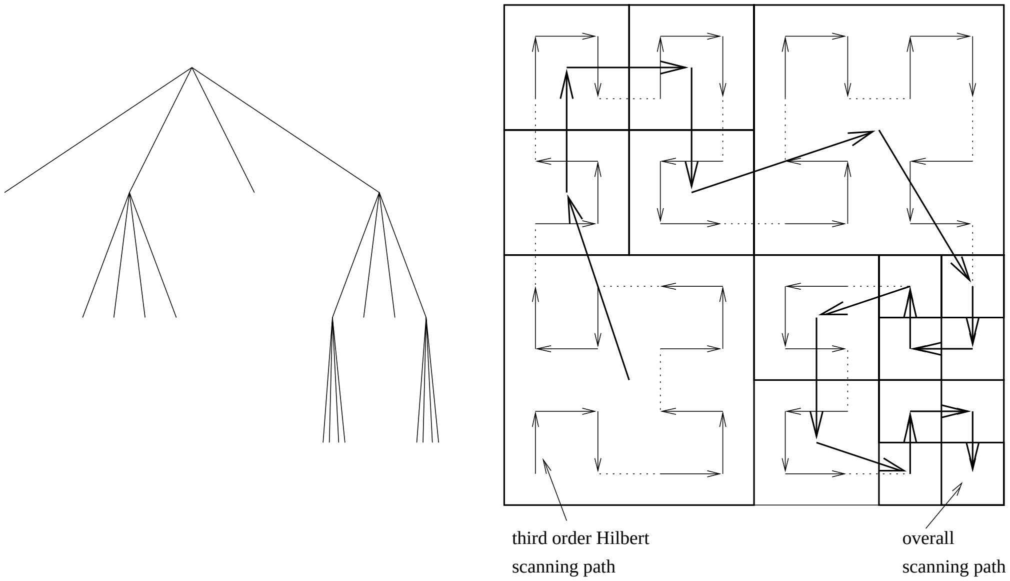


Figure 11: Quad-tree decomposition corresponding to the optimal state sequence. The left figure shows the graphical representation of the QT which corresponds to the frame decomposition in the right figure. The right figure shows the level n_0 scanning path, which is a third order Hilbert curve, and the resulting overall scanning path.

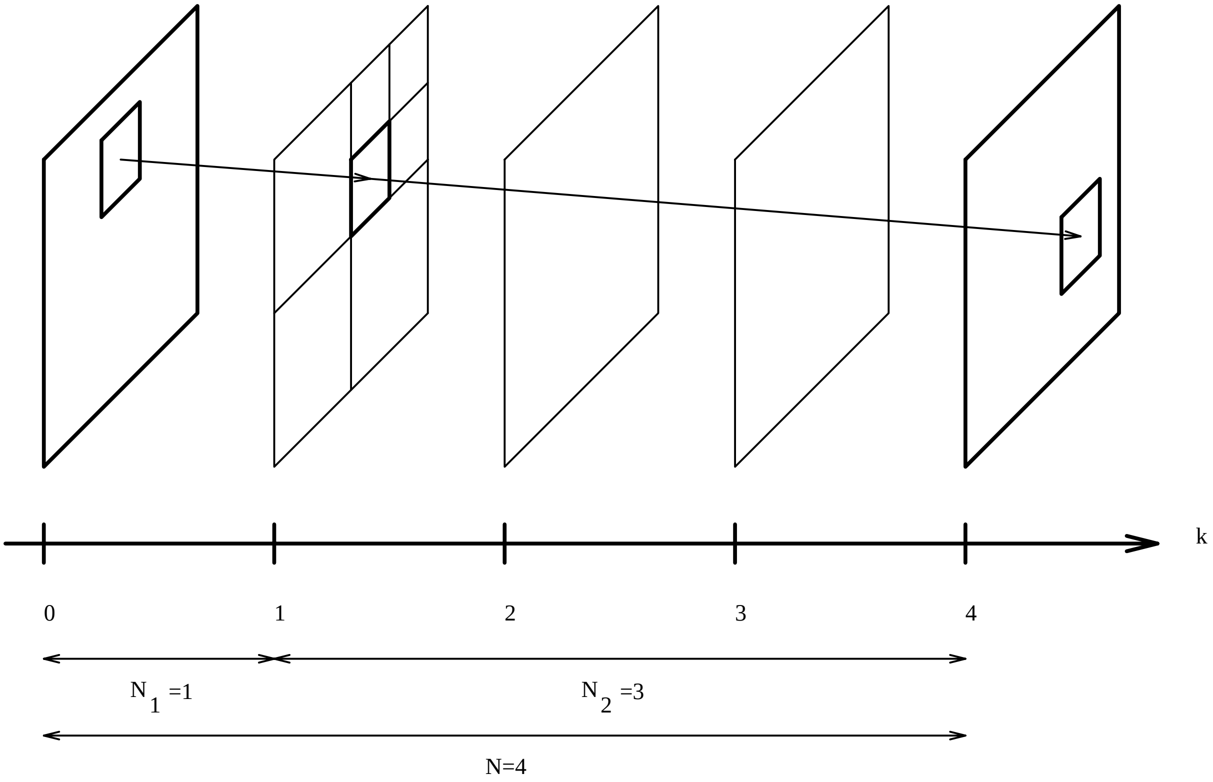


Figure 12: Motion compensated interpolation of the first frame using the two transmitted frames, which are the left most and the right most frame.

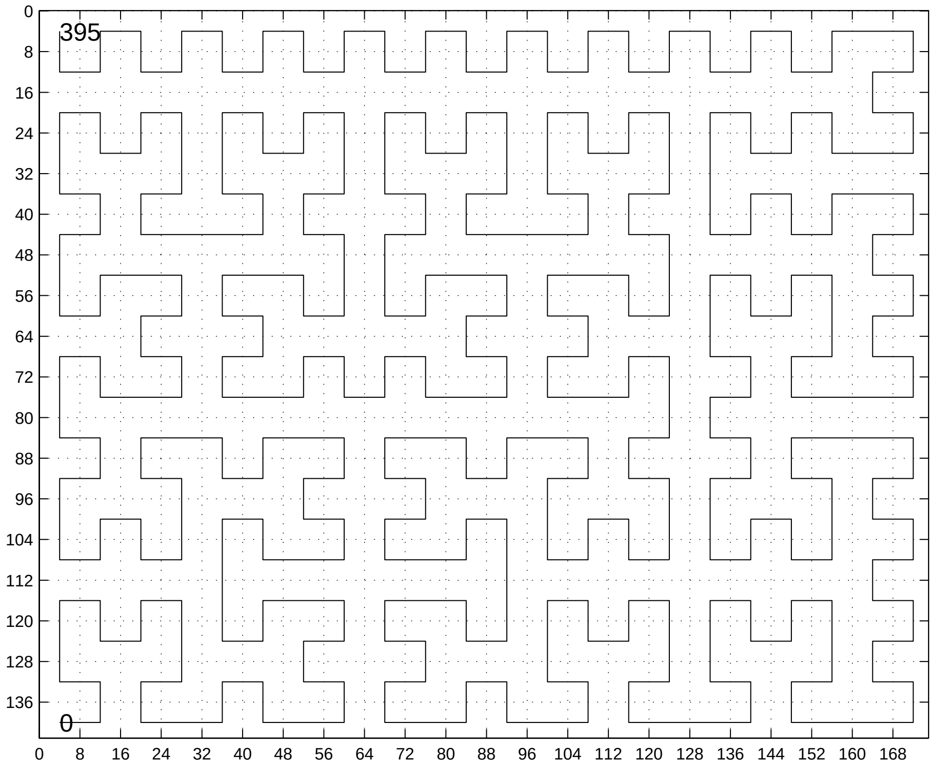


Figure 13: Modified Hilbert scan for level $n_0 = 3$ of a QCIF frame.

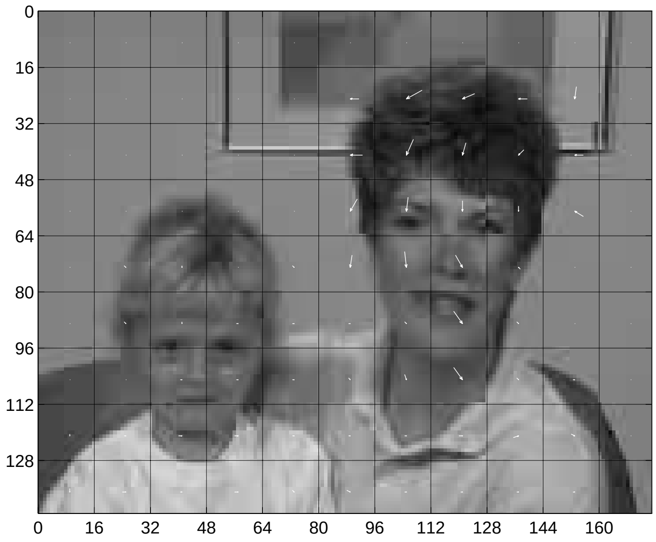


Figure 14: The predicted frame and the DVF for TMN4 block matching.

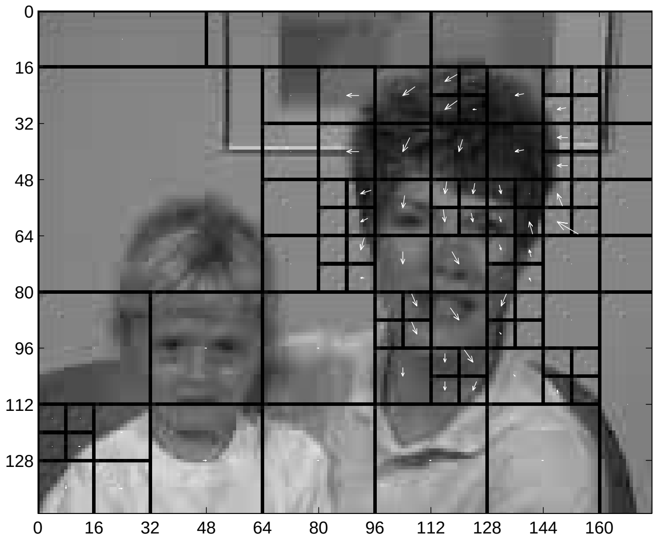


Figure 15: The predicted frame and the DVF for the optimal QT-based motion estimator, when the rate matches the TMN4 rate.

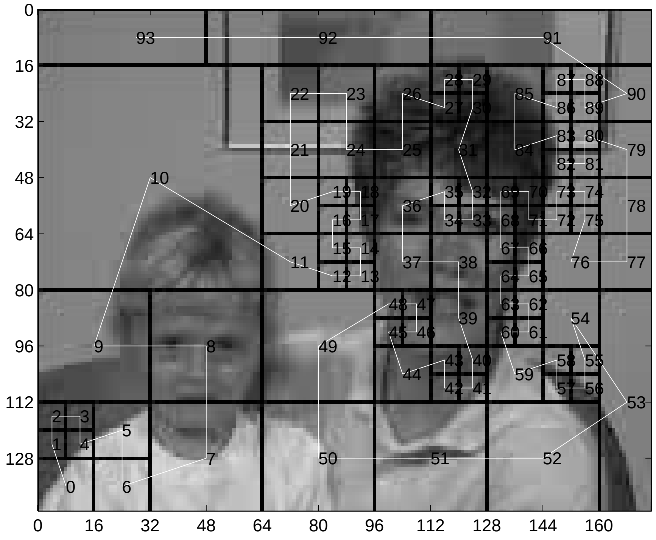


Figure 16: The overall scanning path.

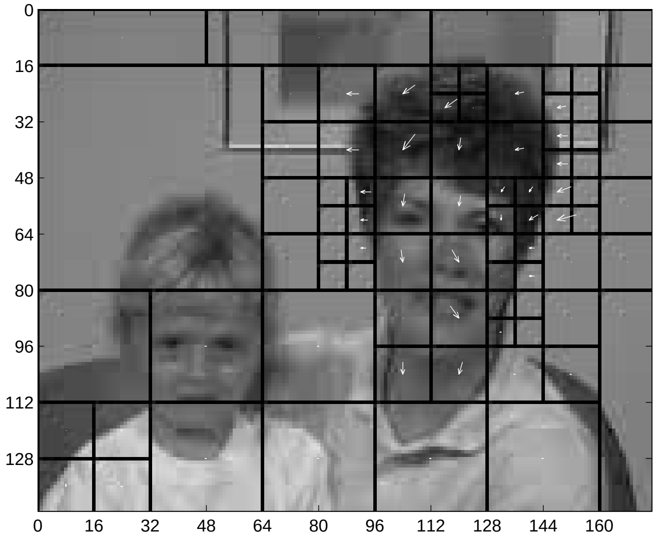


Figure 17: The predicted frame and the DVF for the optimal QT-based motion estimator, when the distortion matches the TMN4 distortion.

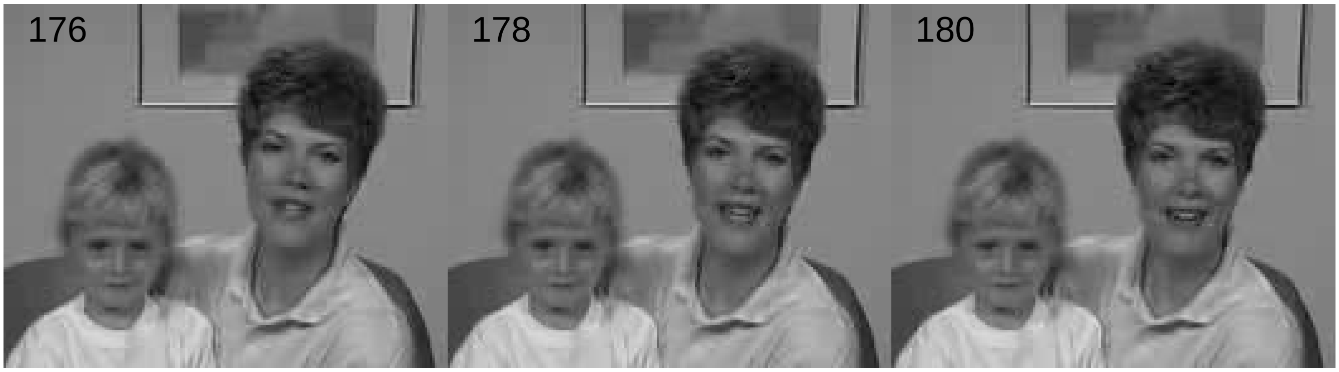


Figure 18: The 176-th, 178-th and 180-th reconstructed frames of the "Mother and Daughter" sequence. The PSNR of each frame is 34.0 dB and it takes 3254 bits to transmit the middle frame.

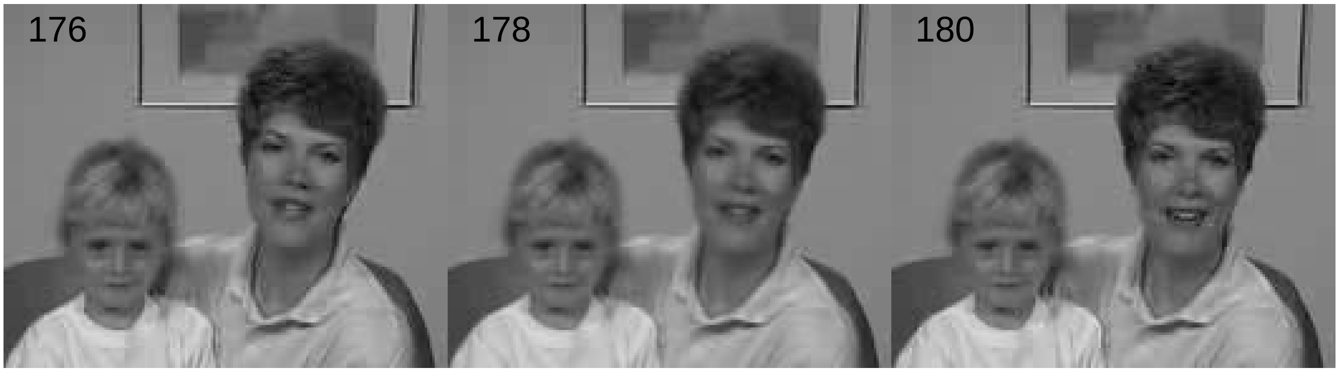


Figure 19: The 176-th and 180-th reconstructed frames of the "Mother and Daughter" sequence are displayed on the left and on the right, respectively. The frame in the middle is the interpolated 178-th frame, its PSNR is 33.3 dB.

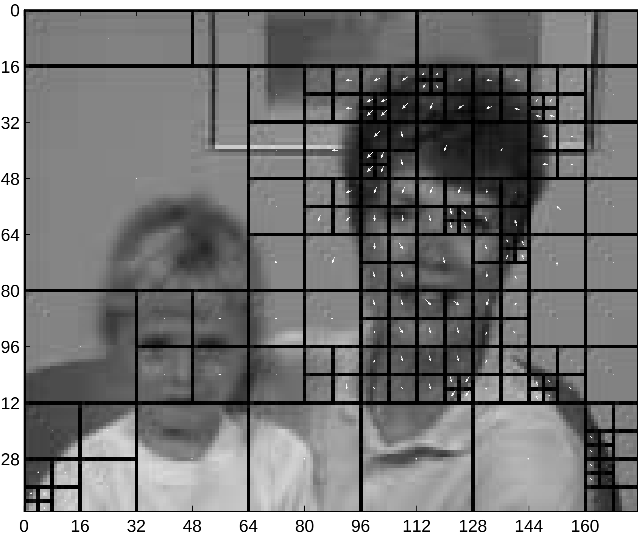


Figure 20: The QT segmentation and the DVF of the 178-th interpolated frame of the "Mother and Daughter" sequence.