Joint Motion and Geometry Modeling with Quad-tree Leaf Merging

Reji Mathew†* and David S. Taubman*

* The University of New South Wales  † National ICT Australia

Introduction

Quad-tree schemes are often used for representing motion between frames of a video sequence.

However a quad-tree structure can only capture horizontal and vertical edge discontinuities at dyadically related locations.

Also, quad-tree structures are unable to exploit redundancy between neighboring leaf nodes with different parents.

Leaf merging can improve the performance of quad-tree schemes, by considering joint coding of neighboring nodes.

Our Contributions

We continue to explore the introduction of geometry information, in conjunction with leaf merging, for quad-tree motion representations.

• General framework for joint modeling of geometry and motion.

• Tree pruning and leaf merging with joint models.

• Polynomial motion models and flexible merge options

• Improved boundary geometry detection
Modeling of motion between frames of a video sequence is critical for video coding. Consider the case of multiple foreground objects in a frame with different types of motion.

A quad-tree representation is attractive as it lends itself to direct rate distortion optimization, allowing a Lagrangian cost function to be globally minimized using tree pruning strategies.

\[ D + \lambda \cdot L = \sum_{\text{leaf nodes}} D_k + \lambda \cdot \sum_{\text{leaf nodes}} L_k \]

A quad-tree structure allows variable size blocks to take on separate motion models. At points of discontinuity in the motion field it may be more cost effective to use smaller block sizes while for areas with more continuous motion, larger blocks can be used.

Pruned quad-tree structure, composed of a hierarchical set of branch and leaf nodes.
**Limitations of Quad-Tree Motion Modeling**

A fundamental limitation of the quad-tree structure is that it can only capture horizontal and vertical edge discontinuities at dyadically related locations.

In general, this causes a greater number of leaf nodes to be produced at the vicinity of boundaries.

Moving object boundaries therefore are expensive to model.

In the left figure, hierarchical coding of the block at the motion boundary requires:
- 7 Leaf Nodes
- 2 Branch Node

\[ 7 + 2 = 9 \text{ motion models} \]

The quad-tree representation does not consider joint optimization of neighboring nodes. Therefore redundancy that may exist between spatial neighbors are not exploited.

In the figure, two neighboring nodes are represented by respective motion models of \( M_A \) and \( M_B \); each of which incurs its own model coding cost.

Joint optimization seeks to explore if the two nodes can be jointly coded by a single motion model \( M_{A+B} \), subject to the rate-distortion cost function.
We consider introducing geometry information where

- the geometry is modeled as a line; and,
- motion for either side of the line is modeled by two separate motion models \( M_1 \) and \( M_2 \).

Models \( M_1 \) and \( M_2 \) can refer to

- Forward Only,
- Backward Only or
- Bi-directional prediction

- And can describe motion using Translation, Linear or Affine flows

Motion compensation for leaf nodes with joint geometry and motion representation is performed by applying each of the two motion models separately to the entire block, and then combining the results in accordance with the geometry.
Detection of Boundary Geometry

To locate discontinuities in the motion field, we perform edge detection on a motion compensated residue image $\Delta$.

We observe this to be better than performing edge detection in the image domain since not all edges in an image correspond to boundaries in the motion field.

$$\Delta = |I - P_f| + |I - P_b|$$

$P_f$: block based forward prediction

$P_b$: block based backward prediction

Motion Models

At the lowest level (Level 0) in the quad-tree, motion is represented by pure translation. At all other levels in the hierarchy, we allow for motion models with 2, 4 or 6 parameters, corresponding to translation, linear & affine flows.

To create linear and affine motion models, we reuse translational motion vectors that have been computed for descendent nodes located lower down in the hierarchy.

Linear and affine models are represented by 2 and 3 motion vectors respectively, at nominal locations within a block.
Tree Pruning and Leaf Merging

**Step 1**: Populate every node of a quad-tree structure with motion and geometry information.

**Step 2**: Perform tree pruning based on joint motion and geometry models.

**Step 3**: Perform leaf merging using available motion and geometry models.

The anchor node signals the line boundary and the two motion models on either side. Other nodes in the merged region signal only merge direction.
**Leaf Merging with Geometry and Motion**

Consider **joint optimization** of a given node with a spatially neighboring node to determine if both nodes can be described by a common set of motion and geometry parameters.

Merge allowed to occur only if a reduction in Lagrangian cost $D + \lambda L$ is achieved.

Merged regions with joint models  
Region with only motion

**Geometry and Motion Models for Merged Regions**

For a new merged region, two options always considered
- Model of current block, and
- Model of neighboring block

New models considered, if both blocks describe only motion,
- New motion only model
- New joint model

New models for a merged region are determined using pre-computed motion and geometry estimates for each block and its sub-blocks. No new exhaustive search is conducted.
Results

Motion is communicated using differential coding with hierarchical prediction; Geometry is communicated by signaling the two intercept points of the edge geometry with the block boundary. These two points are coded without prediction.

PSNR results as a function of the average number of independent leaf regions present per frame. Merging reduces the number of pieces that describe motion.

Examples of merged regions with geometry and motion attributes.