MOTION MODELING WITH GEOMETRY AND QUAD-TREE LEAF MERGING

Reji Mathew[*†] *and David S. Taubman*[*]

[*] The University of New South Wales, Sydney, Australia
[†] National ICT Australia

ABSTRACT

Quad-tree structures are often used to model motion between frames of a video sequence. However, a fundamental limitation of the quad-tree structure is that it can only capture horizontal and vertical edge discontinuities at dyadically related locations. In this paper we seek to address this limitation by introducing geometry information into the nodes of a pruned quad-tree. We start with a typical optimally pruned quad-tree where each node is allowed to model motion. Then for each node in the tree, we consider augmenting the node's motion model with a linear geometry model. Experimental results show that the introduction of geometry information improves the performance of quad-trees in representing motion. Recent work into quad-tree representations have highlighted the benefits of leaf merging. In this paper we extend the leaf merging paradigm to incorporate both geometry and motion information, allowing the creation of regions that have both motion and geometry attributes.

Index Terms— Video coding, Motion compensation

1. INTRODUCTION

Quad-tree structures are often used to represent motion between frames of a video sequence, since they allow the motion field to be recursively subdivided into smaller regions with each region represented by its own motion model. For video coding applications, the quad-tree representation is attractive as it enables a Lagrangian cost function to be globally minimized using tree pruning strategies.

Recent work into quad-tree representations have commented on the sub-optimal nature of quad-trees due to their inability to exploit the dependence between neighboring leaf nodes with different parents. Similar statements can be made about other tree structures, such as those used in H.264. To rectify this performance loss, leaf merging has been proposed, where the possibility of jointly coding neighboring nodes is considered. For motion modeling, the gain achieved by merging has recently been reported in [1][2][3].

While merging has proved successful in exploiting the motion redundancy between nodes in a pruned quad-tree, discontinuities in the motion field, caused by boundaries of moving objects, remain difficult and expensive to model. At a fundamental level, the quad-tree structure itself can only capture horizontal and vertical edge discontinuities at dyadically related locations. This is separate to the structural redundancy of quad-trees, which is addressed by leaf merging. Our objective in this paper is to overcome both forms of redundancy, without sacrificing the computational advantages of quad-trees.

Our research is inspired by the work of [4] where a quad-tree coding algorithm is proposed for the efficient representation of images. In this scheme, each node is also associated with edge geometry, even if the relevant edge does not pass through the node itself. This allows leaf nodes to be merged into larger regions with a single simple edge geometry. In our work, we apply similar principles to augment our motion model with geometry attributes. As in [4], each node in the tree has an associated edge geometry, even if the relevant edge does not pass through the node itself. Each node is also associated with motion models, one for each side of the edge, although only one of them might actually be relevant, depending on the location of the edge. The key to making this representation efficient is leaf merging.

Previous work into incorporating edge information to tree structured motion models [5] have been carried out in the context of H.264 coding with no leaf merging. Our primary motive is to explore the introduction of geometry information in conjunction with leaf merging.

We start with a typical optimally pruned quad-tree where each node is allowed to model motion with a translational motion vector. We then visit each leaf node in a bottom-up manner and consider augmenting the leaf node's motion vector with a linear geometry model. To compensate for the high cost of signaling both geometry and motion parameters, nodes for which geometry is signalled borrow motion from their neighbors. In this way, for a given node only one of motion or geometry is actually signaled. The decision to augment a node's model with geometry is made only if there is an accompanying reduction in Lagrangian cost. We also consider further tree pruning by examining the possibility that the complex motion associated with a group of sibling nodes can be described by assigning a single geometry model to their parent. Finally we perform merging, incorporating both motion and geometry information in the merging process. The merging procedure is similar to that described in [3] and ultimately seeks to exploit redundancy that may exist between neighboring nodes that could not be exploited within the hierarchical structure of the original quad-tree.

2. MOTION COMPENSATION WITH GEOMETRY

In this paper, each frame is partitioned into a collection of blocks $\bar{\mathbf{b}}_i$ of variable size, following a quad-tree structure. We use the notation $\mathbf{b}_j$ to refer to all nodes in the tree, adding the bar, as in $\bar{\mathbf{b}}_i$ where we wish to specifically restrict our attention to leaf nodes. Each non-leaf node $\mathbf{b}_j$ at level k_j in the tree has four descendants at $k_j + 1$, some or all of which may be leaves. In other work [3], we have considered more general tree structures, but we limit our attention here to quad-trees for simplicity.

In our proposed model, each block $\bar{\mathbf{b}}_i$ has an associated simple edge geometry G_i and two motion model, $M_{i,1}$ and $M_{i,2}$, corresponding to disjoint regions $\mathcal{R}_{i,1}$ and $\mathcal{R}_{i,2}$, such that $\mathcal{B}_i \subseteq \mathcal{R}_{i,1} \cup \mathcal{R}_{i,2}$; where $\mathcal{B}_i$ refers to the spatial region corresponding to block $\bar{\mathbf{b}}_i$. In our implementation the motion models $M_{i,1}$ and $M_{i,2}$ can take on forward, backward or bi-directional modes, corresponding to one or two translational motion vectors each. In future work, we plan to also consider polynomial motion models, considering the benefit we

have found them to bring in [3]. Conceptually, each of the two motion models is applied separately to the entire block, after which a blending operation is used to combine the results in accordance with edge geometry, G_i.

In practice, our blending operation currently takes the motion compensated result associated with $M_{i,1}$ for all locations in $\mathcal{R}_{i,1} \cap \mathcal{B}_i$ and the motion compensated result associated with $M_{i,2}$ for all locations in $\mathcal{R}_{i,2} \cap \mathcal{B}_i$. In future work, however, we plan to extend this to include more careful consideration of foreground and background regions. With this simplified blending scheme, leaf nodes for which the edge G_i falls outside $\mathcal{B}_i$, are affected by only one of the two motion models, reducing the process to conventional motion compensation. In this way each node in the quad-tree is allowed to take on geometry and motion information, with the geometry influencing motion compensation only when the edge is located within the spatial region represented by the node.

Currently, we consider only simple straight line edge models for any given block. Each edge is described in terms of its intercepts with two points. In practice, these two points belong to the boundary of some block in which the edge geometry is signalled.

3. SIGNALLING GEOMETRY AND MOTION

Information about the augmented quad-tree is conveyed by a signalling scheme, whose parameters are the subject of our estimation algorithm. In this present section, we describe the properties of the signalling scheme. At each node in the tree, $\mathbf{b}_i$, we signal at most either a geometry model G_i or a motion model M_i. Currently, if a block $\mathbf{b}_i$ has a signalled motion model M_i, we consider that block to have no intersecting edge, so that only that one motion model is relevant for motion compensation. If $\mathbf{b}_i$ has a signalled geometry G_i, the motion models $M_{i,1}$and $M_{i,2}$ for regions on either side of the line, $\mathcal{R}_{i,1}$ and $\mathcal{R}_{i,2}$, are derived from neighboring nodes in the orthogonal direction to the line; we denote these neighbors as $\mathbf{b}_{i_1}$and $\mathbf{b}_{i_2}$.

For a node with signalled geometry, the neighboring nodes $\mathbf{b}_{i_1}$and $\mathbf{b}_{i_2}$, from which motion is borrowed, must be leaf nodes of equal or greater size or branch nodes of the same size. This policy allows for hierarchical decoding of the quad-tree structure, with the possibility for resolution scalability in the motion model, because the motion borrowed by a geometry signalled node $\mathbf{b}_i$ is recovered from nodes at the same or previous levels in the tree.

Since each of the neighboring nodes $\mathbf{b}_{i_1}$and $\mathbf{b}_{i_2}$ may in general have two distinct motion models of its own, our policy is to borrow that motion model from $\mathbf{b}_{i,1}$ (either $M_{i_1,1}$ or $M_{i_1,2}$) whose associated region has the greatest overlap with $\mathbf{b}_i$, and that motion model from $\mathbf{b}_{i,2}$ (either $M_{i_2,1}$ or $M_{i_2,2}$), whose region has the greatest overlap with $\mathbf{b}_i$. Note that the neighbors $\mathbf{b}_{i_1}$and $\mathbf{b}_{i_2}$ may each have a signalled geometry model or a signalled motion model. For example, if $\mathbf{b}_{i_1}$is signalled with motion model M_{i_1}, then the borrowed model $M_{i,1}$ is necessarily equal to M_{i_1}. If $\mathbf{b}_{i_1}$has geometry signalling, the borrowed model $M_{i,1}$ is one of $M_{i_1,1}$ or $M_{i_1,2}$, each of which is itself borrowed from elsewhere. In this way, signalled information propagates through the tree. Figure 1 shows one example of this. The motion for the left region of block $\mathbf{b}_2$ is borrowed from block $\mathbf{b}_1$ which could be a branch or leaf node. This borrowed motion then propagates to a region in block $\mathbf{b}_3$ and $\mathbf{b}_4$. The borrowing of motion is similar to merging but limited to geometry signalled nodes with the merge options being implicit.

In addition to nodes for which geometry is signalled and nodes for which motion is signalled, there are nodes for which a merge is explicitly signalled. These nodes derive their models (G_i, $M_{i,1}$ and $M_{i,2}$) from the neighboring node to which they are merged. We write $\mathcal{R}_i$ for the entire merged region, consisting of all blocks with which $\mathbf{b}_i$ is connected via merging. Similar to individual nodes described earlier, merged regions are also notionally associated with a simple edge geometry and two motion models. All blocks in the merged region share these same models.

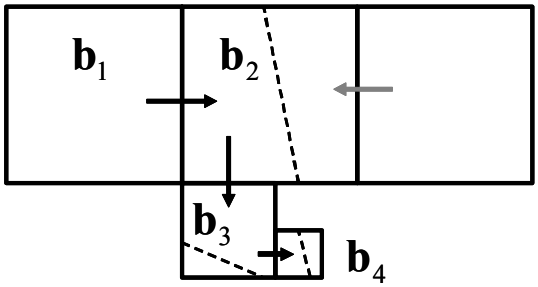

Fig. 1. *Propagation of motion information from node* $\mathbf{b}_1$ *to geometry-signalled nodes* $\mathbf{b}_2$, $\mathbf{b}_3$ and $\mathbf{b}_4$.

We follow the merging rules described in [3] and employ the same hierarchical coding strategy for the merged quad-tree. The merging rules identify a set $\mathcal{T}_{\mathbf{b}_i}$ of candidate merge targets for $\mathbf{b}_i$. This set is restricted to neighboring nodes of larger size, or of the same size but with a different immediate parent. For additional restrictions on $\mathcal{T}_{\mathbf{b}_i}$, the reader is referred to [3]. By limiting the size of $|\mathcal{T}_{\mathbf{b}_i}|$, we keep the cost of merge signalling as low as possible, thereby encouraging the likelihood that merging will prove favorable.

The geometry and motion for all nodes in a merged region $\mathcal{R}$ are inherited from a single node in the region, which we refer to as the anchor node, $\mathbf{b}_\mathcal{R}$. As described in [3] for hierarchical coding, the anchor node is the largest block in the merged region that appears first for a raster scan order. By pursuing this convention, the location of the anchor node does not need to be explicitly signaled but can be determined once all merge directions for nodes (at any given level in the tree) have been conveyed. Following the general principle outlined previously, the anchor node can have either a signalled geometry or a signalled motion model. If the anchor node $\mathbf{b}_\mathcal{R}$ has a signalled motion model then all merged nodes in the region inherit this motion. The assumption in this case is the notional edge geometry lies outside the merged region. If the anchor node has signalled geometry $G_\mathcal{R}$,we first find the borrowed motion models $M_{\mathcal{R},1}$ and $M_{\mathcal{R},2}$ for the anchor node, following the principles described above. These become the geometry and motion models adopted by all blocks in the region.

4. PARAMETER ESTIMATION

As hinted in the introduction, we start out by ignoring both the geometry and merging options, generating an quad-tree motion model which is optimally pruned in the rate-distortion sense to minimize a Lagrangian objective of the form $D + \lambda L$, where D denotes distortion in the sense of motion compensated residual energy and L denotes the total signalling cost for the model. This first step can find a globally optimal solution with relatively low computational cost, subject to the constraints imposed by the quad-tree motion modelling structure. From this point on, we make incremental adjustments only when these produce further reductions in $D + \lambda L$. These subsequent steps are necessarily greedy, so their order is important to overall performance. The first step, described in Section 4.1, considers in-

troducing geometry into selected nodes of the tree. The second step, considered in Section 4.2, is concerned with leaf merging.

4.1. Introducing Geometry

In order to determine the impact on our $D + \lambda L$ objective of signalling a node $\mathbf{b}_i$ with geometry, we first form a set $\mathcal{S}_i$ of all nodes that may be affected by the change. $\mathcal{S}_i$ includes neighboring nodes that have been signalled with geometry and borrow motion information from $\mathbf{b}_i$. Other nodes which are not immediate neighbors of $\mathbf{b}_i$ may also be affected through the propagation of motion borrowing described in the previous section and shown in Figure 1. Distortion calculations relating to introducing geometry to node $\mathbf{b}_i$ must include all nodes in $\mathcal{S}_i$ as all nodes in this set are impacted upon by the model of $\mathbf{b}_i$.

When geometry is introduced to $\mathbf{b}_i$ two motion models, for either side of the edge geometry, are borrowed from neighbors. However it is likely that alternative motion models than those provided by the neighbors may be more suitable. In order to encourage the selection of geometry nodes where appropriate, we consider the possibility of jointly optimizing motion models. This implies that the motion of neighbors may change in the process, which expands the set of nodes in $\mathcal{S}_i$ which may be affected by changes that we introduce. To attain alternative motion models, we do not employ a full motion search at any point; instead we simply re-use the original set of precomputed block motion vectors belonging to the member blocks to find a model of best fit.

To determine the orientation and position of the edge within a block, we begin with the gradient and rough location information produced by a simple edge detector. The line parameters are then allowed to vary within a defined small interval as we search for the best location. The coding cost associated with signalling geometry within a node is equal to the total number of bits required to encode two intercepts of the edge with the block's boundary. The two intercepts are coded without any prediction and therefore the bits required for a node of size $2^j \times 2^j$ is equal to $2(j+2)$ bits.

This introduction of geometry, one node at a time, is essentially a greedy process and therefore the order in which we visit nodes to consider geometry changes is important. In our implementation, we first consider changes which would further prune the tree by introducing geometry to current parent nodes and discarding their children. These nodes already represent cases where discontinuities in the motion field exit, so it is likely that a simple edge geometry at the parent level may be able to describe the more complex motion described by its children. Next we visit leaf nodes in an order which is determined by the energy of edges produced by the edge detector. Leaf nodes with higher edge energy are considered first, since they are more likely to be modified by this step.

4.2. Leaf Merging with Geometry and Motion

In our current work we follow the same merging procedure as that detailed in [3]. Merging is performed in a single raster-scan pass within each level of the quad-tree, starting with the lowest level, whose leaves are the smallest. To calculate the impact of merging $\mathbf{b}_i$ with one of the nodes $\mathbf{b}_j$ in its candidate merge set $\mathcal{T}_{\mathbf{b}_i}$, we first create a set $\mathcal{S}$ of all nodes that may be affected by the merge. Certainly $\mathcal{S}$ contains all nodes in $\mathcal{R}_j$ and $\mathcal{R}_i$, the merged regions to which $\mathbf{b}_j$ and $\mathbf{b}_i$ already belong. Of course, $\mathcal{R}_i$ may contain only the node $\mathbf{b}_i$ and $\mathcal{R}_j$ may contain only the node $\mathbf{b}_j$, but regions tend to grow as we proceed. $\mathcal{S}$ also includes any other nodes which borrow motion directly or indirectly from within $\mathcal{R}_j$ or $\mathcal{R}_i$. These are all nodes which are either signalled with geometry or members of merged regions whose anchor is signalled with geometry.

The Lagrangian cost of a potential merge is calculated by taking into account the distortion of all nodes in the affected set $\mathcal{S}$, the motion and/or geometry signalling cost for the merged region's anchor and the cost of signalling merge information. Merging is allowed to take place only if it reduces the overall Lagrangian cost. This ensures that each modification to the original rate-distortion optimally pruned quad-tree will only serve to reduce the overall Lagrangian functional $D + \lambda L$.

It is worth mentioning that the motion borrowing rules, outlined in Section 3, can make certain merge combinations illegal. In particular, it is not possible to create a merged region whose anchor node has signalled geometry, if the region itself contains one of the nodes from which the anchor is expected to borrow its motion.

When considering the merging of nodes $\mathbf{b}_i$and $\mathbf{b}_j$, two options are always considered for determining the model of the merged region, these being: a) using the model of block $\mathbf{b}_i$ for the potential merged region; and b) using the model of block $\mathbf{b}_j$ for the potential merged region. If $\mathbf{b}_i$ is a node signalling geometry, this geometry model is considered for the merged region. One condition that must be satisfied, however, is that the geometry of $\mathbf{b}_i$ can be signalled by the anchor node of the potential merged region. This means that the line modeling the edge geometry must pass through its boundary. If $\mathbf{b}_i$ signals motion, then this motion model is considered for the merged region. In the same way the model of $\mathbf{b}_j$ is also considered for the potential merged region, regardless of whether it is signalled with geometry or motion.

In addition to considering the models of $\mathbf{b}_i$and $\mathbf{b}_j$ for the merged region, a third option is considered if both $\mathbf{b}_i$and $\mathbf{b}_j$ signal motion. In this case a new motion model is formed by averaging the original motion vectors associated with all nodes in the new region. Therefore, when considering the merge possibilities for node $\mathbf{b}_i$, at most three models are considered for each merge target $\mathbf{b}_j \in \mathcal{T}_i$.

5. RESULTS

We now present results demonstrating the improvement in performance obtained by introducing geometry into the quad-tree and to the subsequent leaf merging process. Our results relate to a hierarchical coding of a quad-tree with 4 levels, corresponding to blocks of size 32×32 at the top most level and blocks of size 4×4 at the lowest level. One might suspect that hierarchical coding carries an unnecessary coding cost, since information is coded for all levels in the tree, including non-leaf nodes which are not involved in the actual motion compensation process. In our previous work [3], however, it has been shown that this performance penalty disappears in the presence of leaf merging, so that there is no significant benefit to invoking more complex spatially predictive coding schemes such as those found in H.264. This justifies our use of the same hierarchical coding strategy for the present study.

In Figure 2 we show the improvement in performance gained by introducing geometry information into an optimally pruned quad-tree motion model. The graph labelled "Pruned_Quadtree" refers to the performance of a conventional quad-tree motion model. The graph labeled "Geom" refers to the case in which geometry information is introduced in accordance with the procedure described in Section 4.1. The results presented in Figure 2 correspond to 50 frames of the CIF resolution Mobile and Calendar sequence at 15Hz. The bit rate for these graphs includes all signalled information for the quad-tree structure and model parameters. The motion compensated residual power, expressed in terms of PSNR, corresponds to the Y

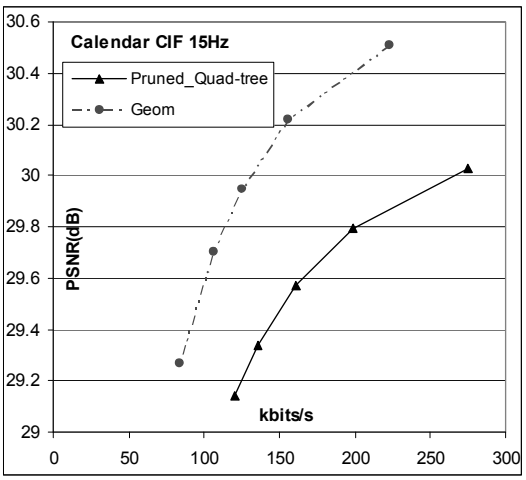

Fig. 2. *Performance gain accomplished by introducing geometry to an optimally pruned quad-tree model.*

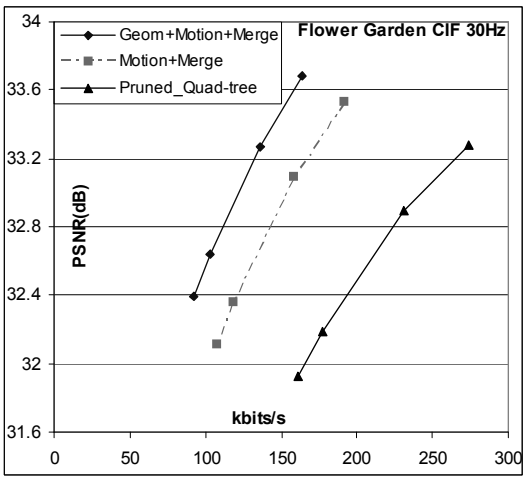

Fig. 4. *Comparison of merging performed on quad-tree models with and without geometry information, for the Flower Garden sequence.*

component. The gain achieved in modeling motion by introducing geometry is clear: typically 30% to 40% bit rate saving achieved.

Figure 3 shows the impact of introducing leaf merging to the pruned quad-trees, both with and without geometry. The graph labelled "Motion+Merging" refers to the application of merging to a quad-tree model where all nodes signal motion. This corresponds to the case reported in [3]. The graph labelled "Geom+Motion+Merge" relates to the case where merging is applied to the quad-tree model after the introduction of geometry information. Evidently, both geometry modeling and leaf merging play important roles in overcoming the short comings of conventional quad-tree motion, as foreshadowed in the introduction. Even though merging brings large improvements to motion-only models, the introduction of geometry yields an additional 10% to 20% reduction in bit-rate.

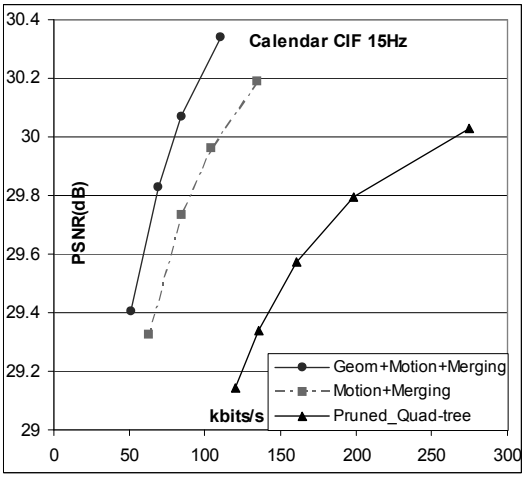

Fig. 3. *Comparison of merging with and without geometry information for the Mobile and Calendar sequence.*

Similar results for 50 frames of the CIF resolution Flower Garden sequence at 30Hz are shown in Figure 4. Once again, the presence of geometry information yields superior results compared to those achieved by motion merging alone; savings of about 10% to 15% in bit rate are evident in this case. An interesting observation from both Figures 3 and 4 is the tremendous improvement that merging provides when compared to the initial optimally pruned quad-tree. Previous results in [2] and [3] relate to uni-directional motion only; however, as demonstrated here, merging brings even greater savings for the case of bi-directional prediction. This is due to the higher cost associated with bi-directional motion signalling. When a block with a bi-directional motion model is merged into a region, two motion vectors are replaced with a single merge signal.

6. CONCLUSION

A fundamental limitation of the quad-tree structure is that it can only capture horizontal and vertical edge discontinuities at dyadically related locations. We address this limitation by introducing geometry information into the nodes of a pruned quad-tree. We also extend the leaf merging paradigm to incorporate both geometry and motion information, allowing the creation of regions that have both motion and geometry attributes, subject to rate-distortion optimization considerations. Incorporating both geometry and leaf merging allows two significant shortcomings of conventional quad-tree motion models to be simultaneously mitigated.

7. REFERENCES

[1] R. D. Forni and D. Taubman, "On the benefits of leaf merging in quad-tree motion models," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 858–861, September 2005.

[2] M. Tagliasacchi, M. Sarchi, and S. Tubaro, "Motion estimation by quadtree pruning and merging," *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 1861–1864, July 2006.

[3] R. Mathew and D. S. Taubman, "Hierarchical and polynomial motion modeling with quad-tree leaf merging," *Proc. IEEE Int. Conf. Image Processing*, pp. 1881–1884, October 2006.

[4] R. Shukla, P. Dragotti, M. Do, and M. Vetterli, "Rate-distortion optimized tree-structure compression algorithms for piecewise polynomial images," *IEEE Trans. Image Processing*, vol. 14, pp. 343–359, March 2005.

[5] E. M. Hung, R. L. D. Queiroz, and D. Mukherjee, "On macroblock partition for motion compensation," *Proc. IEEE Int. Conf. Image Processing*, pp. 1697–1700, October 2006.