

**WHAT IS CLAIMED IS:**

1. — A decoder comprising:  
circuitry configured to:  
 5        ~~receive a bitstream;~~  
       determine whether an exponential partitioning mode is enabled;  
       partition a block into a first region and a second region according to a curved line;  
           and  
       reconstruct pixel data of the block and using the curved line, the first region and  
 10        the second region being non-rectangular.
2. — The decoder of claim 1, wherein the exponential partitioning mode is signaled in the  
bitstream.
- 15 3. — The decoder of claim 1, wherein the curved line partitioning the block into the first region  
and the second region is characterized by a predefined template.
4. — The decoder of claim 1, wherein the curved line partitioning the block into the first region  
and the second region is characterized by a predefined coefficient value.  
 20
5. — The decoder of claim 1, wherein the exponential partitioning mode is available for block  
sizes greater or equal to 8x8 luma samples.
6. — The decoder of claim 1, wherein reconstructing pixel data includes computing a predictor  
 25        for the first region using an associated motion vector contained in the bitstream.
7. — The decoder of claim 1, further comprising:  
       an entropy decoder processor configured to receive the bitstream and decode the  
           bitstream into quantized coefficients;  
 30        an inverse quantization and inverse transformation processor configured to process the  
           quantized coefficients including performing an inverse discrete cosine transform;

a deblocking filter;  
 a frame buffer; and  
 an intra prediction processor.

5 8. — The decoder of claim 1, wherein the bitstream includes a parameter indicating whether the exponential partitioning mode is enabled for the block.

9. — The decoder of claim 1, wherein the block forms part of a quadtree plus binary decision tree.

10

10. — The decoder of claim 9, wherein the block is a non-leaf node of the quadtree plus binary decision tree.

11. — The decoder of claim 1, wherein the block is a coding tree unit or a coding unit.

15

12. — The decoder of claim 1, wherein the first region is a coding unit or a prediction unit.

13. — A method comprising:

receiving, by a decoder, a bitstream;

20

determining, by the decoder, whether an exponential partitioning mode is enabled;

determining, by the decoder, a curved line partitioning a block into a first region and a second region; and

reconstructing, by the decoder, pixel data of the block and using the curved line.

25 14. — The method of claim 13, wherein the exponential partitioning mode is signaled in the bitstream.

15. — The method of claim 13, wherein the curved line partitioning the block into the first region and the second region is characterized by a predefined template.

30

16. — ~~The method of claim 13, wherein the curved line partitioning the block into the first region and the second region is characterized by a predefined coefficient value.~~
17. — ~~The method of claim 13, wherein the exponential partitioning mode is available for block sizes greater or equal to 8x8 luma samples.~~
18. — ~~The method of claim 13, wherein reconstructing pixel data includes computing a predictor for the first region using an associated motion vector contained in the bitstream.~~
19. — ~~The method of claim 13, wherein the decoder comprises:  
an entropy decoder processor configured to receive the bitstream and decode the  
bitstream into quantized coefficients;  
an inverse quantization and inverse transformation processor configured to process the  
quantized coefficients including performing an inverse discrete cosine transform;  
a deblocking filter;  
a frame buffer; and  
an intra prediction processor.~~
20. — ~~The method of claim 13, wherein the bitstream includes a parameter indicating whether the exponential partitioning mode is enabled for the block.~~
21. — ~~The method of claim 13, wherein the block forms part of a quadtree plus binary decision tree.~~
22. — ~~The method of claim 21, wherein the block is a non-leaf node of the quadtree plus binary decision tree.~~
23. — ~~The method of claim 13, wherein the block is a coding tree unit or a coding unit.~~
24. — ~~The method of claim 13, wherein the first region is a coding unit or a prediction unit.~~

1. A video decoder comprising circuitry configured to:  
receive a coded video bit stream including a coded picture, the coded picture  
including a coded coding tree unit comprising multiple coding units and signaling  
information including a first index for determining a start point and a second index for  
determining an end point of a non-straight, non-rectangular boundary in the coding  
tree unit;  
decode the coded coding tree unit using the first index and second index by:  
determine, using the first index and second index, the start point and end  
point of the non-straight, non-rectangular boundary in the coding tree unit;  
generate first predictive pixel values in a first region on a first side of the  
non-straight, non-rectangular boundary;  
generate second predictive pixel values in a second region on a second  
side of the non-straight, non-rectangular boundary;  
smooth the first predictive pixel values and second predictive pixel  
values across the non-straight, non-rectangular boundary; and  
add residual pixel values to the smoothed first and second predictive  
pixel values.
2. The decoder of claim 1 wherein the non-straight, non-rectangular boundary is a curve.
3. The decoder of claim 1 wherein the picture depicts a curved object and the non-  
straight, non-rectangular boundary represents a boundary of the object.
4. The decoder of claim 1 wherein the coding tree unit is  $N \times N$ , wherein each coding unit  
is  $N/2 \times N/2$ , and  $N$  is one of 128, 64, or 32.
5. The decoder of claim 1 wherein the start point is on a first side of the coding tree unit  
a first offset distance from a first corner of the coding tree unit.
6. The decoder of claim 5, wherein the end point is located on a second side of the  
coding tree unit a second offset distance from a second corner of the coding tree unit.

7. A decoder comprising circuitry configured to:

receive a coded video bit stream including a coded picture, the coded picture including a block of coded pixels and signaling information including a first index for determining the start point of a non-straight non-rectangular partition and a second index for determining the end point of the non-straight non-rectangular partition in the block;

decode the block of coded pixels using the first index and second index by:  
determine, using the first index, a start point of a non-straight non-rectangular partition in the block, the start point being located on a first side of the block a first offset distance from a first corner of the block;

determine, using the second index, an endpoint of the non-straight non-rectangular partition in the block, the end point being located on a second side of the block, a second offset distance from a second corner of the block;

generate first predictive pixel values in a first region on a first side of the non-straight, non-rectangular partition;

generate second predictive pixel values in a second region on a second side of the non-straight, non-rectangular partition;

smooth the first predictive pixel values and second predictive pixel values across the non-straight non-rectangular partition; and  
add residual pixel values to the smoothed first and second predictive pixel values.

8. The decoder of claim 7 wherein the coded picture includes a block of intra-predicted pixels and wherein the decoder includes an intra-prediction processor for decoding the block of intra-predicted pixels.

9. The decoder of claim 7 wherein the block is an NxN coding tree unit comprising a plurality of coding units.