CLAIMS

1. A coding method , comprising:

determining whether the size of a current block is larger than a minimum allowed quadtree leaf node size, wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size or the minimum allowed quadtree leaf node size is not larger than a maximum allowed ternary tree root node size;

if the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying multi-type tree splitting to the current block .

2. The coding method according to claim 1 further comprising the steps of:

determining whether the current block of a picture is a boundary block;

wherein if the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying multi-type tree splitting to the current block comprises:

if the current block is a boundary block and the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying binary splitting to the current block, wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size.

3. The coding method according to claim 2, further comprising the steps of:

dividing an image into blocks, wherein the blocks comprising the current block;

wherein the applying binary splitting to the current block comprises:

applying binary splitting to the boundary block with a maximum boundary multi-type partition depth, wherein the maximum boundary multi-type partition depth is a sum of at least a maximum multi-type tree depth and a maximum multi-type tree depth offset, wherein the maximum multi-type tree depth is larger than 0.

4. The coding method according to claim 3, wherein the maximum multi-type tree depth is larger than 0 when applying the binary splitting to the boundary block.

5. The coding method according to any one of claims 1 to 4, wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size and the minimum allowed quadtree leaf node size is not larger than a maximum allowed ternary tree root node size.

6. The coding method according to any one of claims 1 to 5, wherein the applying multi-type tree splitting to the current block comprising applying ternary splitting to the current block, or applying binary splitting to the current block.

7. The coding method according to any one of claims 1 to 5, further comprising:

determining the maximum allowed binary tree root node size based on the minimum allowed quadtree leaf node size.

8. The coding method according to any one of claims 1 to 7, further comprising:

dividing an image into blocks, wherein the blocks comprising the current block;

wherein the applying multi-type-tree splitting to the current block comprises: applying multi-type-tree splitting to the current block of the blocks with a final maximum multi-type-tree depth, wherein the final maximum multi-type-tree depth is a sum of at least a maximum multi-type-tree depth and a maximum multi-type-tree depth offset, wherein the maximum multi-type-tree depth is larger than or equal to subtraction of Log2 value of minimum allowed transform block size from Log2 value of minimum allowed quadtree leaf node size, or the maximum multi-type-tree depth is larger than or equal to subtraction of Log2 value of minimum allowed coding block size from Log2 value of minimum allowed quadtree leaf node size.

9. The coding method according to claim 8, wherein the current block is a non-boundary block.

10. The coding method according to claim 8 or 9, wherein the maximum multi-type-tree depth offset is 0.

11. The coding method according to claim 8, wherein the block is a boundary block and the multi-type-tree splitting is binary splitting.

12. The coding method according to any one of claims 8 to 10, wherein the multi-type-tree splitting comprises ternary splitting.

13. A method of coding implemented by an encoding device, comprising:

determining whether the size of a current block is larger than a minimum allowed quadtree leaf node size;

if the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying multi-type-tree splitting to the current block;

wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size or the minimum allowed quadtree leaf node size is not larger than a maximum allowed ternary tree root

~~node size.~~

~~14. A decoding device, comprising:~~

~~circuitry configured to:~~

~~determine whether the size of a current block is larger than a minimum allowed quadtree leaf node size;~~

~~if the size of the current block is not larger than the minimum allowed quadtree leaf node size, apply multi-type tree splitting to the current block; wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size or the minimum allowed quadtree leaf node size is not larger than a maximum allowed ternary tree root node size.~~

~~15. An encoding device, comprising:~~

~~circuitry configured to:~~

~~determine whether the size of a current block is larger than a minimum allowed quadtree leaf node size;~~

~~if the size of the current block is not larger than the minimum allowed quadtree leaf node size, apply multi-type tree splitting to the current block; wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size or the minimum allowed quadtree leaf node size is not larger than a maximum allowed ternary tree root node size.~~

~~16. A non-transitory computer-readable storage medium storing programming for execution by a processing circuitry, wherein the programming, when executed by the processing circuitry, configures the processing circuitry to carry out the method according to any one of claims 1 to 13.~~

~~17. A computer program product comprising a program code for performing the method according to any one of claims 1 to 13.~~

1. A coding method, comprising:

determining whether the size of a current block is larger than a minimum allowed quadtree leaf node size;

on condition that the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying binary tree splitting to the current block based on a maximum allowed binary tree root node size, wherein the maximum allowed binary tree root node size is determined based on the minimum allowed quadtree leaf node size, and the minimum allowed quadtree leaf node size is not larger than the maximum allowed binary tree root node size.

2. The coding method according to claim 1 further comprising the steps of:

determining whether the current block of a picture is a boundary block;

wherein on condition that the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying binary tree splitting to the current block comprises:

on condition that the current block is a boundary block and the size of the current block is not larger than the minimum allowed quadtree leaf node size, applying binary splitting to the current block, wherein the minimum allowed quadtree leaf node size is not larger than a maximum allowed binary tree root node size.

3. The coding method according to claim 2, further comprising the steps of:

dividing an image into blocks, wherein the blocks comprising the current block;

wherein the applying binary splitting to the current block comprises:

applying binary splitting to the boundary block with a maximum boundary multi-type partition depth, wherein the maximum boundary multi-type partition depth is a sum of at least a maximum multi-type tree depth and a maximum multi-type tree depth offset, wherein the maximum multi-type tree depth is larger than 0.

4. The coding method according to claim 3, wherein the maximum multi-type tree depth is larger than 0 when applying the binary splitting to the boundary block.

5. The coding method according to any one of claims 1 to 4, further comprising:

dividing an image into blocks, wherein the blocks comprising the current block;

wherein the applying binary tree splitting to the current block comprises:

applying binary tree splitting to the current block of the blocks with a final maximum multi-type tree depth, wherein the final maximum multi-type tree depth is a sum of at least a maximum multi-type tree depth and a maximum multi-type tree depth offset, wherein the maximum multi-type tree depth is larger than or equal to subtraction of Log2 value of minimum allowed coding block size from Log2 value of minimum allowed quadtree leaf node size.

6. The coding method according to claim 5, wherein the current block is a non-boundary block.

7. The coding method according to claim 5 or 6, wherein the maximum multi-type tree depth offset is 0.

8. A decoding device, comprising:

circuitry configured to:

determine whether the size of a current block is larger than a minimum allowed quadtree leaf node size;

on condition that the size of the current block is not larger than the minimum allowed quadtree leaf node size, apply binary tree splitting to the current block based on a maximum allowed binary tree root node size, wherein the maximum allowed binary tree root node size is determined based on the minimum allowed quadtree leaf node size, and the minimum allowed quadtree leaf node size is not larger than the maximum allowed binary tree root node size.

9. An encoding device, comprising:

circuitry configured to:

determine whether the size of a current block is larger than a minimum allowed quadtree leaf node size;

on condition that the size of the current block is not larger than the minimum allowed quadtree leaf node size, apply binary tree splitting to the current block based on a maximum allowed binary tree root node size, wherein the maximum allowed binary tree root node size is determined based on the minimum allowed quadtree leaf node size, and the minimum allowed quadtree leaf node size is not larger than the maximum allowed binary tree root node size.

10. A non-transitory computer-readable storage medium storing programming for execution by a processing circuitry, wherein the programming, when executed by the processing circuitry, configures the processing circuitry to carry out the method according to any one of claims 1 to 7.

11. A computer program comprising a program code for causing a computer to perform the method according to any one of claims 1 to 7.

12. A decoder, comprising:

one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the decoder to carry out the method according to any one of claims 1 to 7.

13. An encoder, comprising:

one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the encoder to carry out the method according to any one of claims 1 to 7.

14. A non-transitory storage medium which includes the computer program according to claim 11, which when executed by a processor, cause the processor to perform the method according to any one of claims 1 to 7 to generate a bitstream.