(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 October 2006 (19.10.2006)

PCT

(10) International Publication Number
WO 2006/108863 A2

(51) International Patent Classification:
H04N 7/26 (2006.01) *H04N 7/46* (2006.01)

(21) International Application Number:
PCT/EP2006/061563

(22) International Filing Date: 12 April 2006 (12.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
05075870.5 13 April 2005 (13.04.2005) EP

(71) Applicant *(for all designated States except US)*: **THOMSON LICENSING** [FR/FR]; 46 Quai Alphonse Le Gallo, F-92100 Boulogne-Billancourt (FR).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **FRANCOIS, Edouard** [FR/FR]; 18 Allee Du Locar, F-35890 Bourg des Comptes (FR). **VIERON, Jérome** [FR/FR]; 15 Rue Frédéric Cournet, F-35000 Rennes (FR). **MARQUANT, Gwenaelle** [FR/FR]; 1, rue du Général de Gaulle, F-35340 Liffre (FR). **BURDIN, Nicolas** [FR/FR]; 4, Square Armand de la Rouerie, F-35700 Rennes (FR). **LOPEZ, Patrick** [FR/FR]; 4 Allée du Clos de la Merceraye, F-35450 Livré Sur Changeon (FR).

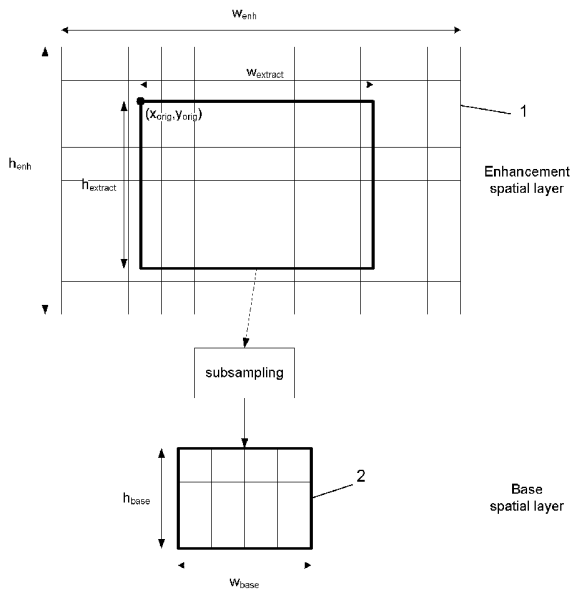
(74) Agent: **KOHRS, Martin**; THOMSON, European Patent Operations, 46 Quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: PROCESS FOR SCALABLE CODING OF IMAGES



(57) Abstract: The process, comprising an inter layer residual prediction mode, to code a current macroblock of an en hancement layer from a prediction macroblock of a base layer, a macroblock being made up of elementary blocks, the prediction macroblock being obtained through an upsampling of the base layer (3) is characterised in that the upsampling is made at an elementary block level and in that the phase of the top left sample within an elementary block depends on the location of the elementary block within the picture.

WO 2006/108863 A2

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Process for scalable coding of images

The invention relates to a process for coding/decoding of images, more specifically scalable video coding/decoding where several spatial video formats need to be encoded/decoded.
This process enables the encoding of two spatial resolutions, low and high layers, where the factor between pictures width and height of the two successive spatial layers can be different vertically and horizontally and does not necessarily equal 2 and where pictures of the higher resolution level can contain parts, for example picture borders, that are not present in corresponding pictures of the low resolution level.
This process is fully generic, that is, any inter-layer size ratio and any cropping window can be handled. It addresses both the "spatial scalable coding using cropped areas" (CE9) and "non dyadic scalability" (CE10).

We consider two successive spatial layers, a low layer, considered as base layer, and a high layer, considered as enhancement layer. Figure 1 shows relations between the enhancement layer 1 and the base layer 2. Width and height of enhancement layer pictures are defined respectively as w_{enh} and h_{enh}. Base layer pictures dimensions are defined as w_{base} and h_{base}. Base layer pictures are a subsampled version of sub-pictures of enhancement layer pictures, of dimensions $w_{extract}$ and $h_{extract}$, positioned at coordinates (x_{orig} , y_{orig}) in the enhancement layer pictures coordinates system. In figure 1, the enhancement and base layer pictures are divided in macroblocks. Upsampling factors between base layer pictures and extraction pictures in enhancement layer are respectively defined as $\alpha_{horiz} = w_{extract} / w_{base}$ and $\alpha_{vertic} = h_{extract} / h_{base}$ for horizontal and vertical dimensions.
Parameters (x_{orig} , y_{orig} , $w_{extract}$, $h_{extract}$) completely define the geometrical relations between high and low layer pictures. In a standard dyadic scalable version, these parameters are equal to (0,0,2.w_{base},2.$h_{extract}$).

The problem to solve is the encoding/decoding of the macroblocks of the enhancement layer knowing the decoded base layer. Figure 2 illustrates the macroblock overlapping between the upsampled base layer picture 3 , in dashed lines, and the enhancement layer picture, in solid lines. A macroblock of enhancement layer may have either no base layer

corresponding block, on borders of the enhancement layer picture, either one or several base layer corresponding macroblocks. Consequently a specific managing of the inter layer prediction is necessary.

An aim of the invention is to solve the aforementioned problems.
Its object is a process for scalable coding of images with a F1 format corresponding to an enhanced layer and with a F2 format having a lower resolution than F1, corresponding to a base layer, comprising an inter layer residual prediction mode to code a current macroblock of the enhancement layer from a prediction macroblock of the base layer, a macroblock being made up of elementary blocks, the prediction macroblock being obtained through an upsampling of the base layer, characterised in that the upsampling is made at an elementary block level and in that the phase of the top left sample within an elementary block depends on the location of the elementary block within the picture.
According to a particular embodiment, the size of an elementary block is 4 x 4 pixels.
According to a particular embodiment, the filter used for the upsampling is a 6-tap filter.
According to a particular embodiment, the upsampling is a 4/3 upsampling corresponding to 4 samples for 3 original pixels.
According to a particular embodiment, the phase, expressed as the relative position of the pixel, for 3 successive elementary blocks, is 0, ½ and ¼.

Other objects and features of the invention will become understood from the following description with reference to the accompanying drawings which represent :
- figure 1, relations between the enhancement layer and base layer,
- figure 2, macroblock overlapping between upsampled base layer picture and enhancement layer picture,
- figure 3, authorized blocks for inter layer motion vector prediction,
- figure 4, localization of a high layer pixel in the base layer,
- figure 5, 4x4 blocks b of a 8x8 block B
- figure 6, 8x8 blocks B of a prediction macroblock,
- figure 7, oversampling of the 4x4 blocks in the case of a 4/3 ratio.

In the sequel, according to the semantic defined further, we will name:

- scaled_base_column = x_{orig}
- scaled_base_line = y_{orig}
- scaled_base_width = $w_{extract}$
- scaled_base_height = $h_{extract}$

Inter layer prediction is only possible for macroblocks fully embedded in the scaled base layer window, corresponding to the grey-colored area in figure 3, that is, macroblocks whose coordinates (MB_x,Mb_y) respect the following conditions:

MB_x >= (scaled_base_column + 15) / 16 and

MB_x < (scaled_base_column + scaled_base_width) / 16

And

MB_y >= (scaled_base_ line + 15) / 16 and

MB_y < (scaled_base_ line + scaled_base_ height) / 16

Inter-layer motion prediction

A high layer macroblock can exploit inter-layer prediction using scaled base layer motion data, using either "BASE_LAYER_MODE" or "QPEL_REFINEMENT_MODE", as in case of dyadic spatial scalability. In case of using one of these two modes, the high layer macroblock is reconstructed with default motion data deduced from the base layer. These modes are only authorized for high layer macroblocks having corresponding base layer blocks, i.e. grey-colored area in figure 3 where bold line represents the upsampled base layer window 3.

- base-layer prediction macroblock construction

As in [JSVM1.0], these macrob lock modes indicate that motion/prediction information including macroblock partitioning are directly derived from the base layer. The process consists in constructing a prediction macroblock, *MB_pred*, inheriting motion data from base layer. When using "BASE_LAYER_MODE" mode, the macroblock partitioning as well as the reference indices and motion vectors are those of the prediction macroblock

MB_pred. "QPEL_REFINEMENT_MODE" is similar, but a quarter-sample motion vector refinement is achieved.

The process to derive *MB_pred* works in three steps:

- for each 4x4 block of *MB_pred*, inheritance of motion data from the base layer motion data
- partitioning choice for each 8x8 block of *MB_pred*
- mode choice for *MB_pred*

These 3 steps are described in the following sections.

- 4x4 block inheritance

For each 4x4 block *b* of the macroblock, the inheriting process consists in considering the center point (x_c, y_c) of the block, computing its base layer luma location (*xB*, *yB*), and inheriting motion data from the base layer block containing the pixel (*xB*, *yB*). Figure 4 represents the localisation of a high layer pixel in the base layer.

The detailed motion data inheritance process for *b* is the following:

Identification of base layer corresponding macroblock, 8x8 block and 4x4 block :

- Let (*x0,y0*) be the luma coordinates of the upper left pixel of *b*. The base luma location (xB, yB) of the pixel (1,1) of *b* in the base layer is derived as follows :

$$\begin{cases} xB = \dfrac{(x0 - \text{scaled_base_column} + 1) * \text{BasePicWidthInSamples}_L}{\text{scaled_base_width}} \\ yB = \dfrac{(y0 - \text{scaled_base_line} + 1) * \text{BasePicHeightInSamples}_L}{\text{scaled_base_height}} \end{cases}$$

with

$\text{BasePicWidthInSamples}_L = 16*(\text{BasePicWidthInMbsMinus1}+1)$

$\text{BasePicHeightInSamples}_L = 16*(\text{BasePicHeightInMbsMinus1}+1)$

- the base corresponding macroblock is defined as the base layer macroblock containing pixel (xB, yB). Its coordinates (mbxB, mbyB) are given by the following equations:

$$\begin{cases} mbxB = xB/16 \\ mbyB = yB/16 \end{cases}$$

- the base corresponding 8x8 block is defined as the base layer 8x8 block, belonging to the base corresponding macroblock, and containing pixel (xB, yB). Its coordinates (b8xB, b8yB) in the base corresponding macroblock are defined by the following equations:

$$\begin{cases} b8xB = (xB - mbxB*16)/8 \\ b8yB = (yB - mbyB*16)/8 \end{cases}$$

- the base corresponding 4x4 block is defined as the base layer 4x4 block containing pixel (xB, yB). Its coordinates (b4xB, b4yB) in the base corresponding macroblock are defined by the following equations:

$$\begin{cases} b4xB = (xB - mbx*16)/4 \\ b4yB = (yB - mby*16)/4 \end{cases}$$

Motion data inheritance :

- IF the base corresponding macroblock is intra, THEN 4x4 block is set as intra
- ELSE for each list *listx* (*listx*=0 or 1), the 4x4 block gets the reference index and motion vector from the base corresponding 4x4 block

- 8x8 block inheritance

Once each 4x4 block has been treated, a merging process is applied to merge reference indices (1 per list) and motion vectors of the 8x8 block it belongs to. In the following, 4x4 blocks of a 8x8 block are identified as indicated in figure 5, blocks b1 and b2 for the top, b3 and b4 for the bottom, from the left to the right.

For each 8x8 block *B*, the following process is applied:

- IF the four 4x4 blocks have been classified as intra blocks, *B* is considered as intra 8x8 block.
- ELSE, reference indices and partitioning mode are chosen as follows:

  Reference indices choice (for assigning same indices to each 4x4 block)

  - for each list *lx*
    - IF no 4x4 block uses this list, no reference index and motion vector of this list are set to *B*
    - ELSE
      - reference index $r_B(lx)$ for *B* is computed as the minimum of the existing reference indices of the 4 4x4 blocks:

$$r_B(lx) = \min_{b \in \{b_1, b_2, b_3, b_4\}} (r_b(lx))$$

      - IF ($r_{b1}(lx)$!= $r_B(lx)$)
        - $r_{b1}(lx) = r_B(lx)$
        - IF ($r_{b2}(lx)==r_B(lx)$) $mv_{b1}(lx) = mv_{b2}(lx)$
        - ELSE IF ($r_{b3}(lx)==r_B(lx)$) $mv_{b1}(lx)=mv_{b3}(lx)$
        - ELSE IF ($r_{b4}(lx)==r_B(lx)$) $mv_{b1}(lx)=mv_{b4}(lx)$
      - IF ($r_{b2}(lx)$!= $r_B(lx)$)
        - $r_{b2}(lx) = r_B(lx)$
        - IF ($r_{b1}(lx)==r_B(lx)$) $mv_{b2}(lx) = mv_{b1}(lx)$
        - ELSE IF ($r_{b4}(lx)==r_B(lx)$) $mv_{b2}(lx)=mv_{b4}(lx)$
        - ELSE IF ($r_{b3}(lx)==r_B(lx)$) $mv_{b2}(lx)=mv_{b3}(lx)$
      - IF ($r_{b3}(lx)$!= $r_B(lx)$)
        - $r_{b3}(lx) = r_B(lx)$
        - IF ($r_{b4}(lx)==r_B(lx)$) $mv_{b3}(lx) = mv_{b4}(lx)$
        - ELSE IF ($r_{b1}(lx)==r_B(lx)$) $mv_{b3}(lx)=mv_{b1}(lx)$
        - ELSE IF ($r_{b2}(lx)==r_B(lx)$) $mv_{b3}(lx)=mv_{b2}(lx)$
      - IF ($r_{b4}(lx)$!= $r_B(lx)$)
        - $r_{b4}(lx) = r_B(lx)$
        - IF ($r_{b3}(lx)==r_B(lx)$) $mv_{b4}(lx) = mv_{b3}(lx)$

- ELSE IF ($r_{b2}(lx)==r_B(lx)$) $mv_{b4}(lx)=mv_{b2}(lx)$
- ELSE IF ($r_{b1}(lx)==r_B(lx)$) $mv_{b4}(lx)=mv_{b1}(lx)$

Choice of partitioning mode

- Two 4x4 blocks are considered as identical if their motion vectors are identical. The merging process is applied as follows:
  - IF b_1 is identical to b_2 and b_3 is identical to b_4 THEN
    - if b_1 is identical to b_3 then BLK_8x8 is chosen
    - else BLK_8x4 is chosen
  - ELSE IF b_1 is identical to b_3 and b_2 is identical to b_4 THEN BLK_4x8 is chosen
  - ELSE BLK_4x4 is chosen

- prediction macroblock mode choice

A final process is achieved to determine the *MB_pred* mode. In the following, 8x8 blocks of the macroblock are identified as indicated in figure 6, blocks B1 and B2 for the top, B3 and B4 for the bottom, from the left to the right.

Two 8x8 blocks are considered as identical blocks if:

- One or both of the two 8x8 blocks are classified as intra blocks

or

- Partitioning mode of both blocks is BLK_8x8 and reference indices and motion vectors of list0 and list1 of each 8x8 block, if they exist, are identical.

The mode choice is done using the following process:

- IF all 8x8 blocks are classified as intra blocks, THEN *MB_pred* is classified as INTRA macroblock
- ELSE, *MB_pred* is an INTER macroblock

Remaining intra 8x8 blocks are enforced to be Inter 8x8 blocks

- 8x8 blocks classified as intra are enforced to INTER blocks with 8x8 partitioning. Their reference indices and motion vectors are computed as follows. Let B_{INTRA} be such a 8x8 block.

  for each list *listx*

  - IF none of the other 8x8 block uses this list, no reference index and motion vector of this list is assigned to B_{INTRA}
  - ELSE, the following steps are applied:
    - reference index *r*(*listx*) is computed as the minimum of the existing reference indices of the other 8x8 blocks:

$$r(\text{listx}) = \min_{B \in \{B_1, B_2, B_3, B_4\}} (r_B(\text{listx}))$$

    - mean motion vector mv_{mean}(*listx*) of the 8x8 blocks having the same reference index *r*(*listx*) is computed
    - *r*(*listx*) is assigned to B_{INTRA} and each 4x4 block of B_{INTRA} is enforced to have *r*(*listx*) and mv_{mean}(*listx*) as reference index and motion vector.

Mode choice

- The choice of the partitioning mode for *B* is achieved. Two 8x8 blocks are considered as identical if their Partitioning mode is 8x8 and reference indices and motion vectors of list0 and list1 of each 8x8 block, if they exist, are identical. The merging process is applied as follows:
  - if *B1* is identical to *B2* and *B3* is identical to *B4* then
    - if *B1* is identical to *B3* then MODE_16x16 is chosen.
    - else MODE_16x8 is chosen.
  - else if *B1* is identical to *B3* and *B2* is identical to *B4* then MODE_8x16 is chosen.
  - else MODE_8x8 is chosen.

- motion vectors scaling

A motion vector rescaling is finally applied to every existing motion vectors of the prediction macroblock *MB_pred*. Motion data $mv=(d_x,d_y)$ is scaled in the $mv_s=(d_{sx},d_{sy})$ using the following equations:

$$\begin{cases} d_{sx} = [d_x . \alpha_{horiz}] \\ d_{sy} = [d_y . \alpha_{vertic}] \end{cases}$$

Using the semantic described further, the actual formulas are as follows :

$$\begin{cases} d_{sx} = \dfrac{2 * d_x * \text{scaled_base_width} + \text{sign}[d_x] * \text{BasePicWidthInSamples}_L}{2 * \text{BasePicWidthInSamples}_L} \\ d_{sy} = \dfrac{2 * d_y * \text{scaled_base_height} + \text{sign}[d_y] * \text{BasePicHeightInSamples}_L}{2 * \text{BasePicHeightInSamples}_L} \end{cases}$$

Inter-layer texture prediction

Inter layer texture prediction is based on the same principles as inter layer motion prediction. It is only possible for macroblocks fully embedded in the scaled base layer window (grey-colored area in figure 3).

For Intra texture prediction, the interpolation filter is applied across transform blocks boundaries. For residual texture prediction, this process only works inside transform blocks (4x4 or 8x8 depending on the transform).

The process at the decoder works as follows. Let *MB* be a high layer texture macroblock to be interpolated. Texture samples of *MB* are derived as follows:

- let (xP , yP) be the position of the upper left pixel of the macroblock in the high layer coordinates reference.
- a base layer prediction array is first derived as follows:
  - the corresponding quarter-pel position (x4, y4) of (xP , yP) in the base layer is computed as:

$$\begin{cases} x4 = 4 * xP * \text{BasePicWidthInSamples}_L / \text{scaled_base_width} \\ y4 = 4 * yP * \text{BasePicHeightInSamples}_L / \text{scaled_base_height} \end{cases}$$

  - the integer-pel position (xB , yB) is then derived as:

$$\begin{cases} xB = x4 >> 2 \\ yB = y4 >> 2 \end{cases}$$

- the quarter-pel phase is then derived as:

$$\begin{cases} px = x4 - xB << 2 \\ py = x4 - y4 << 2 \end{cases}$$

- the base layer prediction array corresponds to the samples contained in the area (xB-8, yB-8) and (xB+16, yB+16). The same filling process, as used in the dyadic case, is applied to fill samples areas corresponding to non existing or non available samples (for instance, in case of intra texture prediction, samples that do not belong to intra blocks).

- the base layer prediction array is then upsampled. The upsampling is applied in two steps : first, texture is upsampled using the AVC half pixel 6-tap filter; then a bilinear interpolation is achieved to build the quarter pel samples, which results in a quarter-pel interpolation array. For intra texture, this interpolation crosses block boundaries. For residual texture, interpolation does not cross transform block boundaries.

- then, prediction sample pred[x, y] at each position (x, y), x=0..N-1,y=0..N-1, of the high layer block is computed as:

pred[x, y] = interp[xI , yI]

with

$$\begin{cases} xI = px + (4 * x * \text{BasePicWidthInSamples}_L / \text{scaled_base_width}) \\ yI = py + (4 * y * \text{BasePicHeightInSamples}_L / \text{scaled_base_height}) \end{cases}$$

interp[xI , yI] is the quarter-pel interpolated base layer sample at position (xI, yI)

- inter-layer intra texture prediction

A given macroblock *MB* of current layer can exploit intra layer residual prediction only if co-located macroblocks of the base layer exist and are intra macroblocks.

For generating the intra prediction signal for high-pass macroblocks coded in I_BL mode, the corresponding 8x8 blocks of the base layer high-pass signal are directly de-blocked and interpolated, as in case of 'standard' dyadic spatial scalability. The same padding process is applied for deblocking.

- inter-layer residual prediction

A given macroblock *MB* of current layer can exploit inter layer residual prediction only if co-located macroblocks of the base layer exist and are not intra macroblocks.

At the encoder, the upsampling process consists in upsampling each elementary transform block, without crossing the block boundaries. For instance, if a MB is coded into four 8x8 blocks, four upsampling processes will be applied on exactly 8x8 pixels as input.

The interpolation process is achieved in two steps : first, the base layer texture is upsampled using the AVC half pixel 6-tap filter; then a bilinear interpolation, is achieved to build the quarter pel samples. Interpolated high layer samples The nearest quarter pel position is chosen as the interpolated pixel.

Figure 7 shows how the 4x4 blocks are oversampled in the case of a 4/3 ratio. The number of pixels per block in the original picture is 4x4. After the 4/3 oversampling, this number becomes 6x6, 6x5, 5x6 or 5x5 depending on the location of the block within the picture. It can be observed that this number of pixels inside an oversampled 4x4 block follows a regular periodic pattern. In the 4/3 case, the pattern is : {6, 5, 5}

The phase of the top left pixel of each, expressed as the position of this pixel relatively to original grid is given by another periodic pattern. In the 4/3 case, the pattern is : {0, 1/2, 1/4}.

CLAIMS

1 Process for scalable coding of images with a F1 format (1) corresponding to an enhanced layer and with a F2 format (2) having a lower resolution than F1, corresponding to a base layer, comprising an inter layer residual prediction mode to code a current macroblock of the enhancement layer from a prediction macroblock of the base layer, a macroblock being made up of elementary blocks, the prediction macroblock being obtained through an upsampling of the base layer (3), characterised in that the upsampling is made at an elementary block level and in that the phase of the top left sample within an elementary block depends on the location of the elementary block within the picture.

2 Process according to claim 1, characterized in that the size of an elementary block is 4 x 4 pixels.

3 Process according to claim 1, characterized in that the filter used for the upsampling is a 6-tap filter.

4 Process according to claim 1, characterized in that the upsampling is a 4/3 upsampling corresponding to 4 samples for 3 original pixels.

5 Process according to claim 4, characterized in that the phase, expressed as the relative position of the pixel, for 3 successive elementary blocks, is 0, ½ and ¼.

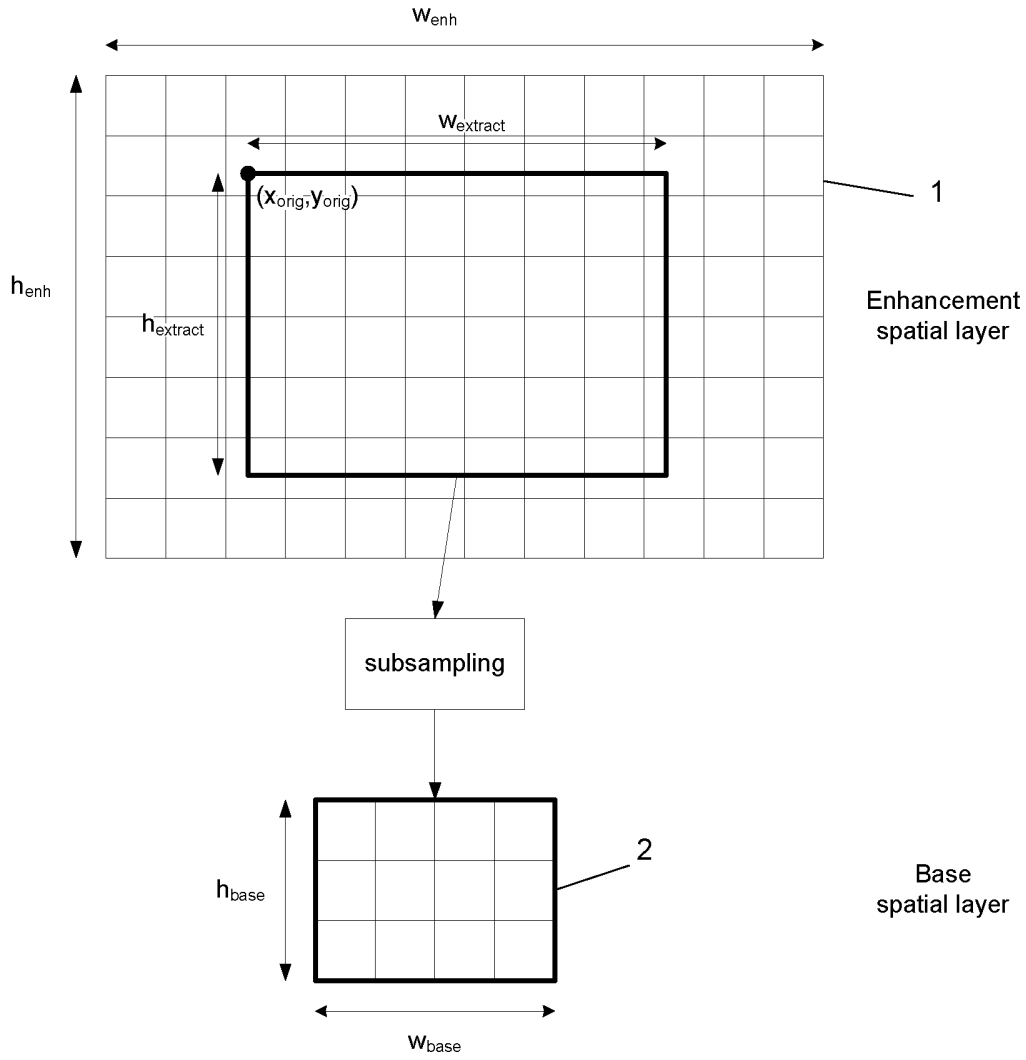

w_{enh}
$w_{extract}$
(x_{orig},y_{orig})
1
h_{enh}
$h_{extract}$
Enhancement spatial layer
subsampling
2
h_{base}
Base spatial layer
w_{base}

FIG 1

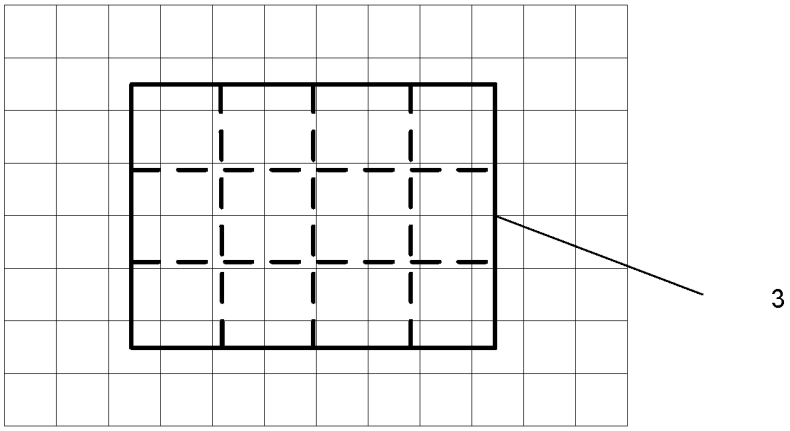

3

FIG.2

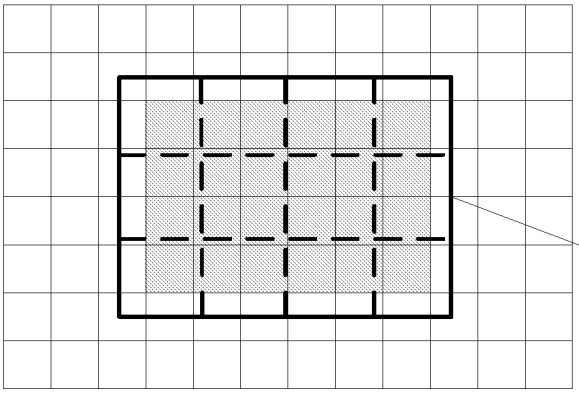

3

FIG 3

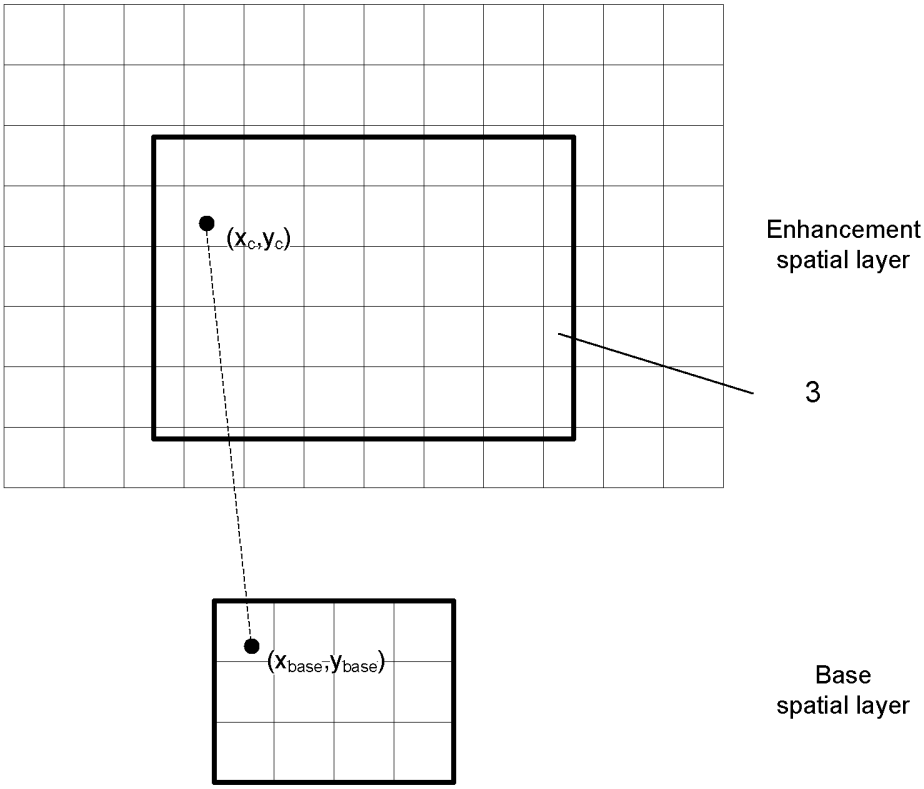
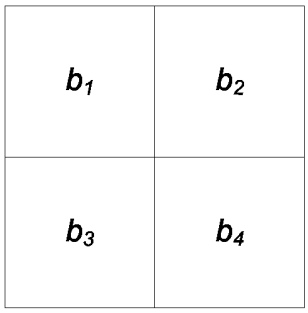

(xc,yc)
Enhancement spatial layer
3
(xbase,ybase)
Base spatial layer

FIG.4



8x8 block *B*

FIG.5

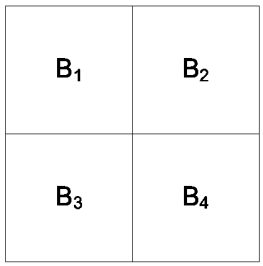


FIG.6

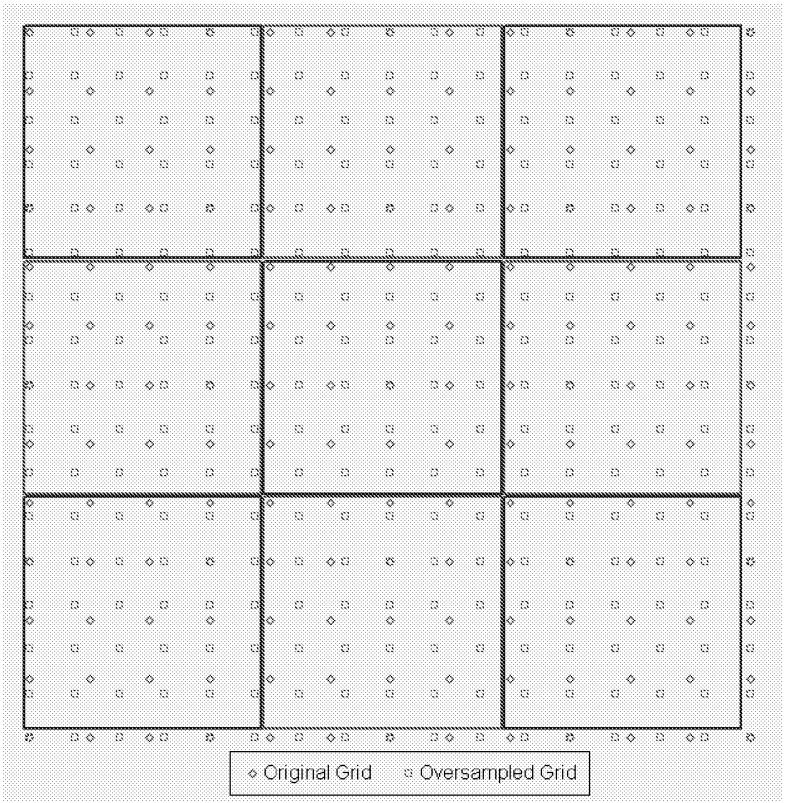


FIG.7