Total Recovery bits have to embed= 71 bits * total no. of blocks in an image for entire secret image

Table 1: Total length of information embedded

Parameters	Values
Image resize to	768 * 1024
Divide image into blocks having block size	8 * 8
Each row having blocks	1024 \ 8 = 128
Each column having blocks	768\8 = 96
Total number of blocks	128 * 96 = 12,288
Embedding length of information for one block	71
Total length of information embedded	12,288 * 71 = 8,72,448

3.2 Ideas of Secret Image Recovery

The second phase specifically involves,
The embedded information we have to extract to recover nearly lossless the secret image from the generated mosaic image.

Here, we will have to do totally inverse procedure that we done in mosaic image creation.For extraction will use Inverse Reverse Contrast Mapping (Inverse RCM).

To compute the original color values (r_i, g_i, b_i) of p_i from the new ones (r_i'', g_i'', b_i''),we use the formula which is inverse form of previous used equation (3),

$$c_i= 1\backslash q_c\ (c_i'' - \mu_c') + \mu_c (5)$$

4. Algorithmic Flow

The algorithm for mosaic image generation and secret image recovery process are given below in detail as algorithm 1 and algorithm 2 respectively.

Algorithm 1: Mosaic image creation

Input: a secret image *S*, a target image T, and a secret key K
Output: a mosaic image F.

Stage 1. Fitting the tiles of secret images into the blocks of target images.

Step-1. Change the sizes of target image Tand secret image S and make them identical. (Here we resize both to 768*1024); and divide the secret imageinto *n* tile images as well as the target image into *n* target blocks with eachbeing of equal size. (Here each block and tile is of 8*8 size)

Step-2. Compute the means and the standard deviations of each tile image and each target block for the three color channels, and compute accordingly the average standard deviations for each individual of them.

Step-3. According to values of average standard deviation obtained, keeping it in ascending order, sort the tile images and the target blocks in separate sets.; map in order the blocks in the sorted tile setto those in the sorted target setin 1-to-1 manner; and re-order the mappings according to the indices of the target images, new sequence named as L.

Step-4. According to L, create a mosaic image Fby fitting the tile images into the corresponding target blocks.

Stage 2. Performing color conversions of tile images to that of target blocks.

Step-5. Create a counting tableTBwith 256 entries, each with an index corresponding to a residual value (where, each residual value will be in the range of 0 to 255), and assign an initial value of zero to each entry.

Step-6. For each mapping, represent the means of tile image and target block, present at that particular mapping point, respectively, by eight bits; and represent thestandard deviation quotient qc by seven bits, where c = r, g, or b.

Step-7. For each pixel $p_i$in each tile image $T_i$of mosaic image Fwith color value $c_i$where c= r, g, or b, transform $c_i$into a new value c_i'' by (3); if c_i'' is not smaller than 255 or if it is not larger than 0, then change c_i'' to be 255 or 0, respectively.

Stage 3. Rotating the tile images with minimum RMSE.

Step-8.Compute the RMSE values of each color transformed tile image $T_i$in Fwith respect to its corresponding target block B_{ji}after rotating $T_i$into each of the directions θ=0°, 90°, 180° and 270°; and fix the rotation of $T_i$into the optimal direction θ° with the smallest RMSE value.

Stage 4. Embedding recovery information.

Step-9.For each tile image $T_i$in mosaic image F, construct a bit stream $M_i$for recovering T_i, in the way as described in section 3.1.4, including the bit-segments which encode the data items of (4):

1) The optimal rotation angle θ° of T_i;
2) The means of T_i and B_{ji} and
3) The related standard deviation quotients of all three color channels.

Step-10. Concatenate the bit streams M_i of all $T_i$in Fina raster-scan order to form a total bit stream M_t; generate a random key and use this secret key Kto encrypt $M_t$into another bit stream M_t'; and embed M_t' into Fby the reversible contrast mapping scheme.

Step-11. Obtain the final form of a secret-fragment-visible mosaic image F.

Algorithm 2: Secret image recovery

Input: a mosaic image Fand the secret key K.
Output: the secret image S.

Stage 1. Extracting the recovery information.

Step-1. Extract the bit stream M_t' using the reverse version ofscheme (inverse Reverse contrast mapping) used in an encoding previously.

Step-2. Decrypt the bit stream M_t' into $M_t$by secret key K.