```
function MAIN(Field Images f_im, Overlap Range mor, Uncertainty uc, Field Order f_order)

  create f_local than will contain relative position to neighbor field, correlation, and its #neighbor field

    f_im ← Denoising f_im using ANISOTROPIC_DIFFUSION
    [height, width] ← row and column size of f_im
    [row_size, col_size] ← row and col size of f_order

  for each f_im
    if f_im(i) amd f_im(i + 1) exists
           [tx, ty, corr]
             ← TRANSLATION(f_im(i) , f_im(i + 1), uc, mor) in vertical direction
           f_trans ← [i, i + 1, tx, ty, corr ]
    end if
    if f_im(i) amd f_im(i + 1) exists
           [tx, ty, corr]
             ← TRANSLATION(f_im(i) , f_im(i
             + col_size), uc, mor) in horizontal direction
           f_trans ← [i, i + col_size, tx, ty, corr ]
    end if
  end for

    f_local ← OPTIMIZED_PATH(f_local, f_trans, f_order);

  for each f_im
        f_global ← merge local x, y to neighbor field untill it reaches center
  end for
  Stitchedimage ← based on f_global merge field images
  return Stitchedimage
end function

function TRANSLATION(Field Image im1, Field Image im2, Uncertainty uc, Overlap Range mor)
    [height, width] ← row and column size of im1

  create threshold th1 of im1 using TRIANGLE method
  create threshold th2 of im2 using TRIANGLE method
  create maximum dislocation dis_loc of the image
  dis_loc ← constant value of maximum dislocation
    p_im1 ← im1(:, width − (width ∗ mor + width ∗ uc ∗ 0.01 − 1): width)
    p_im2 ← im1(:, 1: width ∗ mor + width ∗ uc ∗ 0.01)

    [bin_im1, bin_im2, condition] ← DETECTION(p_im1, p_im2, th1, th2)

  if neither p_im1 and p_im2 has an object
     start ← 1
  else
        start ← overlap depth where both p_im1 & p_im2 contain object
  end if
  for depth_x = start : width ∗ mor
    for depth_y= −dis_loc: dis_loc
```

```
                weight1 ← p_im1 − mean(p_im1)
                weight2 ← p_im2 − mean(p_im2)
                $corr \leftarrow \sum\sum weight1 * weight2 / \sqrt{\sum\sum weight1^2 * \sum\sum weight2^2}$
            end for
            if corr >= MaxCorr
                MaxCorr ← corr
                tx ← depth_x
                ty ← depth_y
            end if
        end for
        if neither p_im1 and p_im2 has an object
                MaxCorr ← MaxCorr − 2
        elseif the percentage of overlapping pixels on partial binary image is smaller than 0.4
                MaxCorr ← MaxCorr − 1
        end if
end function


function DETECTION(Field Image im1, Field Image im2, Threshold th1, Threshold th2)

        condtion

    create gray level image bin_im1 ← (im1 > th1)
    create gray level image bin_im2 ← (im2 > th2)

    if bin_im1 & bin_im2 contains element with value 1, then
            condition ← 1
    else
            condition ← −1
    end if

    return bin_image1, bin_image2, condition
end function

function OPTIMIZED_PATH(f_local, f_trans, f_order)

    create group that will contain fields in group
        create group_labels that will contain group label of each field

    numb=1
    for each field in f_order
        Find neighbor field with maximum correlation
        if both fields is not labeled in any group
            create new group and label two fields in new group
        else if one field is labeled to group and the other is not
            include the nonlabelled field to group of labeled fields
        else
            merge two groups into one
        end if
    end for

    for each group
        Find neighbor field from other group with maximum correlation
        Merge two groups into one
    end for
    f_local(1 , :) ← [1,1,0,0];
    for each field
        idx ←Find field that is connected to current field
        idx_idx ← Find the index of f_trans that contain currentfield and idx field
        f_local(idx , :) ← [f_trans(idx_idx , 3) , f_trans(idx_idx , 4) , f_trans(idx_idx , 5) , current field];
    end for
    return f_local
end function
```