AN APPARATUS FOR TRANSFORMATION PARAMETER ESTIMATION IN IMAGE REGISTRAITON USING TWO-DIMENSIONAL FEED FORWARD NEURAL ARCHITECTURE IMPLEMENTED WITH 2D SYSTOLIC ARRAY ARCHITECTURE

PREAMBLE OF THE DESCRIPTION:

THE FOLLOWING SPECIFICATION PARTICULARLY DESCRIBES THE INVENTION AND THE MANNER IN WHICH IT IS TO BE PERFORMED

ABSTRACT

Image registration is one of the basic image processing operations in remote sensing. With the increase in the number of images collected every day from different sensors, automated registration of multi-sensor/multi-spectral images has become an important issue. In this paper D-level DTCWT transform is computed, features from all levels of sub band are computed considering mutual information index and 2D feed forward neural network structure is trained to estimate the transformation parameters improving inter-pixel correlation and improving image registration accuracy. Proposed algorithms for image registration for 2D images demonstrate image registration accuracy in terms of mean absolute difference error and mean target registration error within 2% and 1% error limits. FPGA implementation of 2D FFNN structure is designed and implemented operating at 245 MHz frequency consuming less than 10% of device resources and with power dissipation of less than 1.5W.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

In this invention, novel algorithms for image registration considering DTCWT features combined with neural networks are designed, modeled and validated considering various test cases demonstrating improved registration accuracy. High speed architecture is designed for proposed 2D FFNN structure.

DESCRIPTION OF THE PRIOR ART

001. Image registration using neural networks was reported in [*Qian, Z., & Li, J. (1997)*] using Hopfield network for matching of features only. Feed forward, Gaussian-sigmoid and radial basis function based neural network were used detecting matching points for registration in the study work carried out by [*Wachowiak, M. P., Smolikova.,R., Zurada, J. M., & Elmaghraby, A. S. (2002)*]. In 2000 [*Elhanany, I., Sheinfeld, M., & Beck, A. (2000)*] proposed FFNN structure for estimating transformation parameters by processing DCT features for image registration. Based on the work reported in [*Elhanany, I., Sheinfeld, M., & Beck, A. (2000)*] rather than using DCT features, many other features such as Zernike moments [*Wu, J., & Xie, J. (2004)*], principal components [*Xu, A., Jin, X., & Guo, P. (2006)*] and kernel

components [*Xu, A., Jin, X., Guo, P., Bie, R. (2006)*] were also used to improve image registration accuracy using neural networks. Limitations of FFNN such as long duration learning process and data dependent performance has been overcome in [*Devin SAĞIRLIBAŞ, (2010)*] by using radial basis function neural network (RBFNN).

002. *Devin SAĞIRLIBAŞ* use neural network to estimate affine transformation parameters based on DCT and 2-D Principal Component Analysis (PCA). Features are extracted from the images to be registered using 2D DCT and 2D PCA methods; the trained Radial Basis Function Neural Network (RBFNN) is designed to process the features to generate the transformation parameters which are used to transform the input image to obtain the registered image. Training of the network is carried out based on features extracted from set of training data set. The advantage of RBFNN is that it can be a generalized structure for image registration and also is robust in the presence of noise. DCT features combined with RBFNN are found to be producing more accurate results compared with PCA and RBFNN results.

003. In the presence of noise in the input image PCA combined with RBFNN registration results were more accurate than DCT based methods. Video registration process has advantages over 2D image registration in terms of visibility of features and in addition the movement of objects can be easily be observed. The registration accuracy is dependent on feature extraction quality and completeness, thus preprocessing of images prior to registration plays a vital role [*Markelj, P. , Tomaževi ˇc, D. , Likar, B. , Pernuš, F., (2012)*]. Automated registration of video is based on iterative optimization methods that adjust transformation parameters and similarity metric [*D. Skerl, D. Tomazevic, B. Likar, and F. Pernus, (2006)*].

004. It is been reported in [*P. Thevenaz and M. Unser, (2000)*] that 100% registration has been achieved, however with real time setup for surveillance operations inaccuracies can occur with remote environment, dynamics in image content, deformation in objects, appearance of foreign objects and imaging system instability.

005. Video registration using neural networks have been reported in [*Liu, H., Yan, J., & Zhang, D. (2006); Yan, C., Ong, S. H., Ge, Y., Zhang, J., Teoh, S. H., & Okker, B. H. (2004), Zhang, J., Ge, Y., Ong, S., Chui, C., Teoh, S., & Yan, C. (2008)*]. The methods reported for video registration is based on neural network structure that were used to estimate distance functions between the surface of reference image to the surface of input image.

006. Multi Layered Perceptron (MLP) network were used with to perform registration in two stages: coarse and fine registration process [*Zhang, J., Ge, Y., Ong, S., Chui, C., Teoh, S., & Yan, C. (2008)*]. Liu et al. in 2006 have used MLP structure to estimate rotation and translation parameters from principal components. The MLP structure with only one hidden layer was able to accurately estimate the video surface differences.

007. The architecture presented in [*Li, Q., Gao, C., Wu, X., & Li, D. (2010)*] is used for fine registration and has only one output layer and is used to map video surfaces using coordinates of corresponding points.

008. In [*Ana Isabel Antunes Dias Rodrigues Gouveia, (2014)*] video registration is carried out 2D registration methods by carrying supervised training algorithm. The features from the spatial and frequency domain are combined and are given to the neural network architecture to estimate transformation parameters for image registration.

009. DWT feature extraction is faster than DCT feature extraction as the later process is operates on every 10 x 10 block size which also leads to blocking artifact errors instead of the entire image as in DWT processing.

010. The major limitations of DWT are shift variance and directional selectivity limited to 90^0 and 45^0. *Huizhong Chen and Nick Kingsbury*, () have used Dual Tree Complex Wavelet Transform (DTCWT) coefficients to align images by considering phase information of coefficients. The DTCWT front-end filter is shift invariance and directional selective and hence the registration algorithm is robust to local mean and contrast changes in images to be registered. Neural network based registration methods reported in literature use either DCT or Fourier or DWT or Zernike moments for transformation parameter estimation.

011. With DTCWT being shift invariant and DTCWT sub bands comprising of directional selective features, training the neural network to estimate transformation parameters using DTCWT features will have advantages of shift invariant as well as reduce computation complexity in registration process. Figure 1 presents the embodiment of the present invention. The two input images are transformed into wavelet domain to obtain the wavelet sub bands, from the wavelet sub bands features such as edges, curves, lines, vertices, points and intensity are determined from both the images. Based on the feature points matching of features between images are carried out. During the feature matching process the location of features are identified and transformation of source image is carried out.

012. The process of feature detection, similarity measurement of features and transformation of image is a iterative process is carried out until best optimum matching is achieved between source and target image. The transformation process carried out is in the wavelet domain, after registration the inverse wavelet transform is carried out to obtain the spatial domain registered image. In video registration one of the simplest methods for evaluating image registration is to measure voxel similarity which is the Sum of Squared intensity Differences (SSD) between the two images. For N voxels in the overlap domain $\boldsymbol{\Omega^T_{S,T}}$ the SSD is given as in Eq. (1)

$$\mathbf{SSD} = 1/N \sum_{\mathbf{x} \in \boldsymbol{\Omega^T_{S,T}}} | T(x) - S^T(x)|^2 \qquad (1)$$

013. 100% registration accuracies has been reported in literature with existing image registration techniques, however with real time setup for clinical operations inaccuracies can occur with clinical environment, dynamics in image content, deformation in organs, appearance of foreign objects and imaging system instability. Optimization process in automated registration process gets locked into local minimum leading to registration error. In addition to registration errors, errors caused by radiation therapists or radiation oncologists lead to inaccuracies as the video of source images are overlapped with target images. Image

processing software's have been developed to facilitate the errors caused by oncologists that process source images and target images that have been captured at various time instances.

014. Introduction of the DTCWT was made in [7][8], and showed which has desirable properties of approximate shift insensitive and good directionality. These properties will play a key role for many applications in analysis and synthesis, like denoising, deblurring, super-resolution, watermarking [9], segmentation [10] and pattern classification [11].

015. Traditional DWT can only exhibits the shift independence in its undecimated form, which is computationally inefficient, particularly in multiple dimensions. The directional selectivity of the DWT is poor because the separability cannot distinguish between the edge and ridge features on opposing diagonals. With conventional approach, to get optimal shift independence, mid-way location of the scaling basis functions of imaginary tree between those for real tree at each level of the transform is must and it was proposed achieving this by a delay of one sample between the same level filters in each tree, and then, for subsequent levels, by employing alternate odd and even length linear-phase filters.

016. In [12], Nick Kingsbury proposed novel approach to achieve optimal shift invariance [13] with only even length linear phase filters by highlighting the major limitations of the alternate even and odd length filter approach. 2D DTCWT produces directionally selective sub bands that can be used to capture features in the images for registration. Milad Ghantous et al. [14] have proposed complex wavelets for fast multimodal automatic image registration of IR and visible images. Pyramidal approach is considered for feature extraction, edge information at the lowest level sub band is used as matching criteria. The search criteria are refined at higher levels based on mutual information thus reducing the computational time for search operations. Accuracy in IR based on the proposed work is improved to 25% as compared with wavelet based registration technique.

017. Christopher et al [15] have presented novel approach for IR based on Wavelets. Gradients in both x and y directions estimated from the sub bands are used as feature points for feature matching and registration is performed achieving PSNR of 36 dB. In this paper a novel approach is proposed for IR based on DTCWT. DTCWT of images produce real part sub band and six complex sub bands with orientations of ($\pm 15^0$, $\pm 45^0$ and $\pm 75^0$). In this work, gradient computation is carried out in both x and y directions on the six sub bands foe feature selection, and mutual information is computed based on cross correlation function for feature selection from the real sub band.

018. The features from gradients and mutual information are used as feature correspondence for image registration. The Gaussian filtered image is decomposed using DTCWT 11,17 tap symmetric filters. The images are decomposed to three levels, with each level consisting of real part and imaginary part. The imaginary part consists of six orientation bands with ($\pm 15^0$, $\pm 45^0$ and $\pm 75^0$). The input image X is decomposed into eight sub bands as shown in Figure 2, the first stage consists of row processing, and the second stage consists of column processing filters. The butterfly structure is designed with sign inversion operation and 2's complement

operation to reduce computation complexity. The division operation by √2 is replaced with threshold operation.

019. Neural Networks (NN) have a large number of highly interconnected processing elements called neurons, which usually operate in parallel and in regular architectures. The collective behavior of neurons realizes a function for which they are being trained. The connection of neurons in a given pattern of interest defines neural network architecture. Every neuron is characterized by weights, biases and activation function.

020. In multilayer Perceptron (MLP) there are multiple neurons that are connected forming a structure consisting of three major layers: input layer, hidden layer and output layer. Figure 3 presents the structure of three layered perceptron also referred to as Feed Forward Neural Network (FFNN) Architecture. The FFNN structure comprises of input layer with N inputs represented as $\{P_1, ----P_N\}$ that are processed with S^1 neurons in the hidden layer, with each neuron processing N inputs. The input vectors that are connected to every neuron are multiplied by the synaptic weight, and the input vector multiplied by the weight matrix is accumulated in the adder module and the output is represented by n^1.

021. The activation function represented by f processes the intermediate output n^1 to generate the hidden layer output represented by a^1. The output layer also comprises of S^2 neurons with each neuron processing the hidden layer output a^1 to generate the final output represented by a^2.

OBJECT OF THE INVENTION

022. The primary object of the embodiments in the present invention is to provide a method for two dimensional FFNN architecture design with optimum use of CLB resources on FPGA for image registration applications with reduced time delay in latency and processing time.

023. These and other objects and advantages of the embodiments herein will become readily apparent from the following detailed description taken in conjunction with the accompanying drawings.

SUMMARY OF THE INVENTION

024. The present invention overcomes the limitations of the conventional techniques of 2D image registration by providing a methodology that combines two techniques. Specifically, the present invention uses landmark manifolds to identify features in both real and imaginary domain and estimates transformation parameters for image registration.

025. The advantages and additional features of the present invention will be detailed in the description which follows, and in part, will be apparent during further description, or may be learned by practicing the invention. The objectives and techniques described in this invention will be realized by the method particularly pointed out in the written description and the claims hereof as well in the associated diagrams listed.

026. In one embodiment of aerial image registration according to the present invention, image registration algorithm and methods are based on neural network that is trained to estimate transformation parameters considering DTCWT features.

027. To achieve these and other advantages and in accordance with the purpose of the invention, as embodied and broadly described, according to the present invention for registering input image with the reference image there are several steps, including identifying significant features from the real and imaginary sub bands of complex wavelets of both input image and reference image. Once the features have been identified, the method includes the process of computing transformation parameters considering the trained neural network structure that has been specifically trained to perform the task. The transformation parameters are then used to transform the input image.

028. Both the foregoing general description and the following detailed description are exemplary and the explanatory and are intended to provide further explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

029. Figure 1 Dual Tree Wavelet and Neural Network based IR Algorithm
030. Figure 2 DTCWT computation
031. Figure 3 Feed Forward NN Structure
032. Figure 4 Transformation parameter estimation
033. Figure 5 Two dimensional FFNN structure
034. Figure 6 PE of Systolic Architecture
035. Figure 7 Two-dimensional Systolic Array
036. Figure 8 shows initial data organization for systolic array
037. Figure 9 Systolic array operations during first clock cycle
038. Figure 10 Systolic array operations during second clock cycle
039. Figure 11 Systolic array operations during third clock cycle
040. Figure 12 Systolic array operations during fourth clock cycle
041. Figure 13 Systolic array operations during fifth clock cycle
042. Figure 14 Systolic array operations during sixth clock cycle
043. Figure 15 Systolic array operations during seventh clock cycle
044. Figure 16 Systolic array operations during eight clock cycle
045. Figure 17 Systolic array operations during tenth clock cycle
046. Figure 18 Systolic array operations during eleventh clock cycle
047. Figure 19 Systolic array operations during eighteenth clock cycle
048. Figure 20 Proposed systolic array architecture for 2DFFNN
049. Figure 21 Simulation results
050. Figure 22 Simulation results
051. Figure 23 Device utilization summary

052. Figure 24 XPower analyzer power report

DETALED DESCRIPTION OF THE PREFERRED EMBODIMENTS

053. The neural network architecture is trained to estimate the transformation parameters from the input that is being deformed with known transformation parameters. Once the network is trained, then the input image which needs to be registered is processed by the network estimates the transformation parameters that can be used for image registration. Figure 4 is the block diagram illustrating registration according to one or more aspects of the present invention. The process facilitates faster or rapid execution of video registration of source image as the neural network is trained to estimate the transformational parameters for all possible distortions in the input image. The block diagram depicts the basic system of image registration using wavelet sub bands and neural networks.

054. In Figure 4 the input image which is the deformed image is decomposed into wavelet sub bands, the sub bands are processed by the trained neural network structure to predict the transformational parameters. The transformational parameters estimated by the neural network are used in transformation of the source image sub band, further to which inverse wavelet transform is carried out to obtain the registered image. The training of neural network structure is carried out by taking the target image and deforming the target image with scaling, translation, rotation and shearing. The transformation parameters to deform the target image are set as the target to the neural network structure and the neural network is trained to generate the target parameters with the wavelet sub bands that are obtained from the deformed image.

055. Two Dimensional FFNN structure hereafter referred to as "**2DFFNN_1**" is shown in Figure 5. The M frames of LLL sub band with each frame of N x N rows and columns is first grouped into 4 x 4 sub image blocks in the LLL sub band. Each of the sub image blocks of 4 x 4 is reordered to 16 x 1 column vectors and is processed by the M arrays of three layered FFNN structure. The structure is termed as two dimensional as the column vector data representing DTCWT LLL sub band after being reordered is processed by all the M frames of FFNN structure that can generate optimum transformational parameters with increased correlation among pixels in multiple frames of video LLL sub band. The LLL sub band with size N x N x M is processed using M FFNN structure. With N x N frame being grouped into 4 x 4 sub images reordering of 4 x 4 block generates 16 x 1 column vector. Grouping N x N into 4 x 4 blocks generates $N^2/16$ sub images, reordering $N^2/16$ sub blocks gives generates 16 x $N^2/16$ matrix from each M frames that are processed by each of the FFNN structure in M array. The input layer for each of M array FFNN is 16, the two layered structure is designed with 32 hidden layer neurons and 6 output layer neurons. The network function selected for hidden layer and output layer is purelin function. The first output of every M array FFNN are represented as $\{O_k^1, O_k^2, \ldots\ldots\ldots O_k^M\}$, where the subscript represents output level and superscript represents the frame level. Eq. (2) for every output of FFNN structure.

$$R_k^1 = \sum_{i=1}^{3} a_i W_{k,i}^1 + b_k^1, \quad O_1^1 = f(R_1^1) \quad (2)$$

056. To compute the equation (2) with a two dimensional systolic array is proposed as in Eq. (3),

$$C_{mxn} = A_{mxk} \text{ X } B_{kxn} \quad \text{.....Eq.(3)}$$

057. Where A, B and C are the matrices with order m x k, k x n and m x n respectively. Each PE of systolic array computes the multiplication of elements and accumulates to the corresponding element and then elements will be passed to neighbour PE in the systolic array. First elements $a_{i,j}$ in row i of matrix a are injected first into PE as pipeline with the sequence of $a_{i,k}$ and the input time to the element of $b_{i+1,j}$is one time unit later than . Similarly, elements in column j of matrix B are injected first into PE as pipeline with the sequence of $B_{k,j}$and the input time to the element of the sequence of $B_{k,j+1}$is onetime unit later than $B_{k,j}$. The architecture of PE in this approach is shown in Figure 6 which performs the Multiplication and accumulation on data.

058. A systolic architecture is an arrangement of processors i.e. PE's in an array where data flows synchronously across the array between neighbours, usually with different data flowing in different directions. PE at each step takes input data from one or more neighbours (e.g. Left and Top), processes it and, in the next step, outputs results in the opposite direction (Right and Bottom). The two dimensional systolic Architecture is given in the Figure 7.

059. The array architecture given above takes input data in parallel into first PEs in the array and processes the Multiplication and Accumulation on them and then outputs result to the next level PEs of array. Systolic arrays do not lost their speed due to their connection like any other parallelism. Where, each cell (PE) is an independent Processor (CPU) and has its own registers and Arithmetic and Logic Units (ALUs) i.e. Multiplication and Accumulation unit. The cells share the information with their neighbours, after performing the necessary operations on the data.

The implementation of FFNN using proposed scheme of preprocessing involves the following steps:

a) Let 16-component vector be the input 1D signal,

X = [x0x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15]

b) The 16-component vector is convert into two dimensional matrix as shown Eq. (4)

$$P = \begin{bmatrix} x0 & x4 & x8 & x12 \\ x1 & x5 & x9 & x13 \\ x2 & x6 & x10 & x14 \\ x3 & x7 & x11 & x15 \end{bmatrix} \quad (4)$$

c) For an 4 × 4 input signal P, construct a 4 × 4 transformation matrix, L, using trained network weight matrices. The weight matrix L dimension after training coefficients value will be 8×8 matrix which has the same dimension of the input matrix after repeated-row preprocessing. The weight matrix L is show Eq. (5)

$$L=\begin{bmatrix} \frac{3}{5\sqrt{2}} & \frac{4}{5} & \frac{3}{5\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ \frac{-1}{20} & \frac{-3}{10\sqrt{2}} & \frac{9}{20} & \frac{1}{\sqrt{2}} & \frac{9}{20} & \frac{-3}{10\sqrt{2}} & \frac{-1}{20} & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{5\sqrt{2}} & \frac{4}{5} & \frac{3}{5\sqrt{2}} & 0 \\ \frac{9}{20} & \frac{-3}{10\sqrt{2}} & \frac{-1}{20} & 0 & \frac{-1}{20} & \frac{-3}{10\sqrt{2}} & \frac{9}{20} & \frac{1}{\sqrt{2}} \\ \frac{-1}{20} & \frac{-3}{10\sqrt{2}} & \frac{9}{20} & \frac{-1}{\sqrt{2}} & \frac{9}{20} & \frac{-3}{10\sqrt{2}} & \frac{-1}{20} & 0 \\ \frac{1}{10\sqrt{2}} & \frac{3}{10} & \frac{9}{10\sqrt{2}} & 0 & \frac{9}{10\sqrt{2}} & \frac{-3}{10} & \frac{-1}{10\sqrt{2}} & 0 \\ \frac{9}{20} & \frac{-3}{10\sqrt{2}} & \frac{-1}{20} & 0 & \frac{-1}{20} & \frac{-3}{10\sqrt{2}} & \frac{9}{20} & \frac{-1}{\sqrt{2}} \\ \frac{9}{10\sqrt{2}} & \frac{-3}{10} & \frac{-1}{10\sqrt{2}} & 0 & \frac{-1}{10\sqrt{2}} & \frac{3}{10} & \frac{9}{10\sqrt{2}} & 0 \end{bmatrix} \quad (5)$$

d) The second layer of the weight matrix can be used for computing final output.

e) During preprocessing instead of multiplying the repeated input stream with α, α is multiplied with column 2,4,6,8 of the inverse matrix so that the number of multiplier is reduced in the architecture.

f) As a result of the above step the input matrix P after preprocessing is as shown in (6),

$$P = \begin{bmatrix} x0 & x4 & x8 & x12 \\ x0 & x4 & x8 & x12 \\ x1 & x5 & x9 & x13 \\ x1 & x5 & x9 & x13 \\ x2 & x6 & x10 & x14 \\ x2 & x6 & x10 & x14 \\ x3 & x7 & x11 & x15 \\ x3 & x7 & x11 & x15 \end{bmatrix} \quad (6)$$

060. While applying input to systolic array, the first column of input is not padded with Zero as shown below

X0	X0	X1	X1	X2	X2	X3	X3

061. While the second column of input is padded with a zero and applied to the systolic array as shown below

0	X4	X4	X5	X5	X6	X6	X7	X7

062. While the third column of input is padded with two zeroes in succession and applied to the systolic array as shown below

0	0	X8	X8	X9	X9	X10	X10	X11	X11

063. While the fourth column of input is padded with three zeroes in succession and applied to the systolic array as shown below

0	0	0	X12	X12	X13	X13	X14	X14	X15	X15

064. Weights are scaled to 2^9 i.e. multiplied by 512 to get signed and unsigned filter coefficients which are represented using fixed integer arithmetic. The weight matrix L1 is as shown below.

$$L1=\begin{bmatrix} 217 & -18 & 0 & 162 & -25 & 25 & 230 & 230 \\ 409 & -76 & 0 & -76 & -108 & 108 & -108 & -108 \\ 217 & 162 & 0 & -18 & 230 & -230 & -25 & -25 \\ 0 & 256 & 0 & 0 & -362 & 0 & 0 & 0 \\ 0 & 162 & 217 & -18 & 230 & 230 & -25 & 25 \\ 0 & -76 & 409 & -76 & -108 & -108 & -108 & 108 \\ 0 & -18 & 217 & 162 & -25 & -25 & 230 & -230 \\ 0 & 0 & 0 & 256 & 0 & 0 & -362 & 0 \end{bmatrix}$$

065. The weight matrix L1 of each row are stored in ROM as shown below. The table 1 show how the first row of transformation matrix is stored in the ROM1

217
-18
0
162
-25
25

230
230

Table 1 shows the filter coefficients in ROM1

066. The second row of the FFNN matrix L1 is padded is with a 0 to achieve synchronization of operation with neighbouring PEs. The Table 2 shows the how the second row of matrix L1 is stored in the ROM2.

0
409
-76
0
-76
-108
108
-108
-108

Table 2 shows the filter coefficients in ROM2

067. The third row of the matrix L1 is padded is with a 0 in two successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 3 shows the how the third row of matrix L1 is stored in the ROM3

0
0
217

162
0
-18
230
-230
-25
-25

Table 3 shows the filter coefficients in ROM3

068. The fourth row of the matrix L1 is padded is with a 0 in three successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 4 shows the how the fourth row of matrix L1 is stored in the ROM4

0
0
0
0
256
0
0
362
0
0
0

Table 4 shows the filter coefficients in ROM4

069. The fifth row of the matrix L1 is padded is with a 0 in four successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 5 shows the how the fifth row of matrix L1 is stored in the ROM5.

0
0
0
0
0
162
217
18
230
230
-25
25

Table 5 shows the filter coefficients in ROM5

070. The six row of the matrix L1 is padded is with a 0 in five successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 6 shows the how the six row of matrix L1 is stored in the ROM6

0
0

0
0
0
0
-76
409
-76
-108
-108
-108
108

Table 6 shows the filter coefficients in ROM6

071. The seventh row of the matrix L1 is padded is with a 0 in six successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 7 shows how the third row of matrix L1 is stored in the ROM7

0
0
0
0
0

0
0
-18
217
162
-25
-25
230
-230

Table 7 shows the filter coefficients in ROM7

072. The eighth row of the matrix L1 is padded is with a 0 in seven successive memory location to achieve synchronization of operation with neighbouring PEs. The Table 8 shows how the eighth row of matrix L1 is stored in the ROM8

0
0
0
0
0
0
0
0

0
0
256
0
0
-362
0

Table 8 shows the filter coefficients in ROM8

073. Figure 8 shows the initial data organization for systolic array in which the pre-processed input are fed column wise into systolic array from left, also the second, third and fourth column of pre-processed input are appropriately zero padded while the first column is not padded with zero. The inverse transform matrix is fed row wise into systolic array from top, while first row of inverse transformation matrix is applied without zero padding with rest of the rows appropriately zero padded. Each Processing element has an accumulator which is indicated by 'axx' corresponding to PExx where xx represent of number of each accumulator as shown in figure 8.and also initially accumulator should be cleared.

074. During the first clock cycle, the pre-processed input and inverse transformation matrix filter is enters into PE11 and these inputs are multiplied and accumulated in a11, while rest of PE is zero. The figure 9 shows the systolic array operation during the first clock cycle.

075. During second clock cycle, PE11, PE12 gets filter coefficients from memory while PE11 and PE21 gets the preprocessed input while the other input to the PE12 and PE21 is the data from neighbouring PEs which were the input to the neighbouring PEs in the previous clock cycle. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The operation of systolic array is during second clock cycle is as shown in the figure 10.

076. During third clock cycle, PE11, PE12, PE13 gets next filter coefficients from memory while PE11, PE21,PE31 gets the preprocessed input while the other input to the PE12, PE13, PE21and PE31 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22 gets both the data from Neighbouring PE. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The operation of systolic array is during Third clock cycle is as shown in the figure 11.

077. During fourth clock cycle, PE11, PE12, PE13,PE14 gets next filter coefficients from memory while PE11, PE21,PE31,PE41 gets the pre-processed input while the other input to the PE12, PE13, PE21,PE31and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE32, PE23 gets both the data from Neighbouring PE . Once both the data is loaded into PE it performs the multiplication and accumulation operations. The operation of systolic array is during Fourth clock cycle is as shown in the figure 12.

078. During fifth clock cycle, PE11, PE12, PE13,PE14and PE15 gets next filter coefficients from memory while PE11, PE21,PE31,PE41 gets the pre-processed input while the other input to the PE12,PE13,PE14,PE15,PE21,PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE32, PE23, PE24, PE42, PE33, PE42 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. Since at the end of fourth clock cycle the filter coefficient flows out of the systolic array, they are fed back for reusing. The operation of systolic array is during fifth clock cycle is as shown in the figure 13.

079. During six clock cycle, PE11, PE12, PE13, PE14, PE15 and PE16 gets next filter coefficients from memory while PE11, PE21, PE31, PE41 gets the pre-processed input while the other input to the PE12, PE13, PE14, PE15, PE16, PE21, PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE32, PE23, PE24, PE25, PE42, PE33, PE42, PE43 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during sixth clock cycle is as shown in the figure 14.

080. During seventh clock cycle, PE11, PE12, PE13,PE14, PE15,PE16 and PE17 gets next filter coefficients from memory while PE11, PE21,PE31,PE41 gets the pre-processed input while the other input to the PE12,PE13,PE14,PE15,PE16,PE17,PE21,PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE32, PE23, PE24, PE25, PE26, PE32, PE33, PE34, PE35, PE42, PE43, PE44 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during seventh clock cycle is as shown in the figure 15.

081. During eighth clock cycle, PE11, PE12, PE13,PE14, PE15,PE16, PE17 and PE18 gets next filter coefficients from memory while PE11, PE21,PE31,PE41 gets the pre-processed input while the other input to the PE12,PE13,PE14,PE15,PE16,PE17, PE18, PE21,PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE32, PE23, PE24, PE25, PE26, PE27, PE32, PE33, PE34, PE35, PE36, PE42, PE43, PE44, PE45 gets both the data from Neighbouring

PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during eighth clock cycle is as shown in the figure 16. At the end of eighth clock cycle, the entire multiplication operation is completed in PE11.

082. During ninth clock cycle, PE12, PE13,PE14, PE15,PE16, PE17 and PE18 gets next filter coefficients from memory while PE21,PE31,PE41 gets the pre-processed input while the other input to the PE12,PE13,PE14,PE15,PE16,PE17, PE18, PE21,PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE23, PE24, PE25, PE26, PE27, PE28, PE32, PE33, PE34, PE35, PE36,PE37, PE42, PE43, PE44, PE45,PE46 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of ninth clock cycle, the multiplication operation is completely carried out in PE12, PE 21.

083. During tenth clock cycle PE13,PE14, PE15,PE16,PE17 and PE18 gets next filter coefficients from memory while PE31,PE41 gets the pre-processed input while the other input to the PE13,PE14,PE15,PE16,PE17,PE18,PE31 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE22, PE23, PE24, PE25, PE26, PE27, PE28, PE32, PE33, PE34, PE35, PE36,PE37,PE38,PE42, PE43, PE44, PE45,PE46,PE47 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during tenth clock cycle is as shown in the figure 17. At the end of tenth clock cycle, the multiplication operation is completely carried out in PE13, PE 22, PE31.

084. During eleventh clock PE14, PE15, PE16, PE17 and PE18 gets next filter coefficients from memory while PE41 gets the pre-processed input while the other input to the PE14,PE15,PE16,PE17,PE18 and PE41 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE23,PE24, PE25, PE26, PE27, PE28, PE32, PE33, PE34, PE35, PE36,PE37,PE38,PE42, PE43, PE44, PE45,PE46,PE47,PE48 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during eleventh clock cycle is as shown in the figure 18. At the end of eleventh clock cycle, the multiplication operation is completely carried out in PE14, PE 23, PE32,PE41.

085. During twelveth clock PE15, PE16, PE17 and PE18 gets next filter coefficients from memory while the other input to the PE15, PE16, PE17, PE18 is the data from neighbouring Processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE24, PE25, PE26, PE27, PE28, PE33, PE34, PE35, PE36,PE37,PE38,PE42, PE43, PE44, PE45,PE46,PE47,PE48 gets both the data from Neighbouring PEs. Once both

the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of twelveth clock cycle, the multiplication operation is completely carried out in PE15, PE 24, PE33,PE42.

086. During thirteenth clock PE16, PE17 and PE18 gets next filter coefficients from memory while the other input to the PE16, PE17, PE18 is the data from neighbouring processing elements which were the input to the neighbouring PEs in the previous clock cycle. While PE25, PE26, PE27, PE28, PE34, PE35 PE36,PE37, PE38, PE43, PE44, PE45,PE46,PE47,PE48 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of eleventh clock cycle, the multiplication operation is completely carried out in PE16, PE25, PE34,PE43. During fourteenth clock PE17 and PE18 gets next filter coefficients from memory while the other input to the PE17, PE18 is the data from the neighbouring processing elements which were the inputs to the neighbouring PEs in the previous clock cycle. While PE26, PE27, PE28, PE35 PE36, PE37, PE38, PE44, PE45, PE46, PE47, PE48 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of fourteenth clock cycle, the multiplication operation is completely carried out in PE17, PE26, PE35,PE44.

087. During fifteenth clock PE18 gets next filter coefficients from memory while the other input to the PE18 is the data from the neighbouring processing elements which were the inputs to the neighbouring PEs in the previous clock cycle. While PE27, PE28, PE36, PE37, PE38, PE45, PE46, PE47, PE48 gets both the data from Neighbouring PEs. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of fifteenth clock cycle, the multiplication operation is completely carried out in PE18, PE27, PE36,PE45. During sixteenth clock PE28, PE36, PE37, PE38, PE46, PE47, PE48 gets both the data from Neighbouring PEs which was the inputs to the neighbouring PEs during previous clock cycle. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing.

088. At the end of sixteenth clock cycle, the multiplication operation is completely carried out in PE28, PE37,PE46. During seventeenth clock PE38, PE47, PE48 gets both the data from Neighbouring PEs which was the inputs to the neighbouring PEs during previous clock cycle. Once both the data is loaded into PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. At the end of seventeenth clock cycle, the multiplication operation is completely carried out in, PE38, PE47.

089. During eighteenth clock PE48 gets both the data from Neighbouring PEs which was the inputs to the neighbouring PEs during previous clock cycle. Once both the data is loaded into

PE it performs the multiplication and accumulation operations. The filter coefficients flows out of systolic array is fed back for reusing. The operation of systolic array is during sixteenth clock cycle is as shown in the figure 19. At the end of eighteenth clock cycle, the multiplication operation is completely carried out in PE48. Proposed architecture of the systolic array for implementing FFNN is as shown below in Figure 20. The proposed Systolic array architecture basically performs the matrix multiplication of the pre-processed input with the weights of the FFNN matrix.

090. In the figure the basic element is Processing element which performs the multiplication and accumulation. The multiplication is performed between the pre-processed input and the FFNN which consist of trained weight elements. The element D represents the delay element which is used to delay the weight coefficient and pre-processed input while applying the data to the next PE in the architecture. First three clock cycles are required to initially fill up the systolic array. So the output from PE11 is available at the 10 clock, the output from PE21 and PE12 is available at 11 clock cycle, the output from PE31,PE22 and PE23 is available at 12 clock cycle, the output from PE41,PE32,PE23 and PE14 is available at 13 clock cycle, the output from PE42,PE33,PE24 and PE15 is available at 14 clock cycle, the output from PE43,PE34,PE25 and PE16 is available at 15 clock cycle, the output from PE44,PE35,PE26 and PE17 is available at 16 clock cycle, the output from PE45, PE36, PE27 and PE18 is available at 17 clock cycle, the output from PE46,PE37 and PE28 is available at 18 clock cycle, the output from PE47 and PE38 is available at 19 clock cycle, , the output from PE48 is available at 20 clock cycle.

091. The proposed embodiment discussed in this patent document is modelled using Verilog HDL and verified for its functionality. The functional correct model is synthesized and implemented on Virtex FPGA device. Synthesis is a process by which an abstract form of designed circuit behavior or register transfer level has been converted into design implementation i.e., in terms of logic gates. The synthesis of Verilog code has been carried out by Xilinx Synthesis Technology(XST) tool, which is part of Xilinx ISE software.

ADVANTAGES OF THE INVENTION

092. The preprocessed real and imaginary components of DTCWT output are applied to two dimensional FFNN blocks which is implemented using systolic array architecture. The systolic array perform matrix multiplication of pre-processed real and imaginary component with trained weight matrix which that are stored in ROM. Figure 21 and Figure 22 shows the simulation of FFNN output. The output of DTCWT is multiplied with weight matrix using systolic array architecture to estimate transformation parameters for image registration. During post-processing the transformation parameters of various bands are averaged and the final transformation parameters are estimated as shown in Figure 4. From the synthesis report (shown in Figure 23) it is observed that the design operates at the maximum frequency of 242.066 MHz. From power report shown in Figure 24 the total Quiescent power consumed is 1.045 W and total dynamic power consumed is 0.083 W, Interconnects and clocks consume

most of the power at a junction temperature of 25 degrees Celsius. Finally the total power consumed is 1.129 W.

WHAT IS CLAIMED IS:

1. An apparatus of registering remote sensing images of 2D to reduce imaging artifacts caused by movement of objects in the input image which comprises of the following steps:
    a. Providing two images of input image and reference image which are of multi-modal and consist of cross sectional features of the same medical object, the two images consist of two;
    b. Reference image has objects with known landmarks, features and feature positions in the 2D geometry, the input image is the deformed image or image captured using different method has the similar object as in the reference image but is time shifted, scaled, rotated, sheared and has also has features that are not in the reference image in addition to the features in the reference image;
    c. Transforming both images using complex wavelets or in particular dual tree complex wavelets to generate multiple sub bands of both input image and reference image, the DTCWT algorithm decomposes the 2D image into sub bands at different resolutions, the number of decomposition levels is set to D. For 2D the D-level decomposition results in low pass and high pass bands with only 32 x 32 pixels.
    d. Image registration is carried in two steps: feature extraction and transformation parameter estimation
    e. During training phase of neural network structure the network weights and biases are obtained by training the network using multiple data sets images that are obtained after DTCWT transformation. The trained network estimates the transformation parameters for a given input that is used to register input image with the reference image.
    f. During validation phase input image that were not part of training data set is considered for image registration using the algorithm proposed in the present invention
    g. Evaluation of registration algorithm is carried out by computing MSE and PSNR metrics considering reference image and registered image, input image and registered image and comparing them with MSE and PSNR obtained from reference image and input image.
    h. In addition to computing MSE and PSNR the transformation vectors estimated from NN is compared with actual transformation vectors and the corresponding error is measured to determine image registration algorithm accuracy.
2. The apparatus of **Claim 1**, wherein said image is transformed into wavelet sub bands is performed using 2D DTCWT that decomposes input data at every level into 6 high pass and 2 low pass bands. At every level of decomposition the data size is reduced by half.

For example if the input data size is 512 x 512 after first level of decomposition the data size of each of the 64 sub bands will be 256 x 256. The number of levels of decomposition is limited to D wherein for the example considered D is set to five and after five levels of decomposition there will be four low pass bands of size 16 x 16 and 6 high pass bands of size 16 x 16 at level 5, 6 high pass bands of size 32 x 32 at level 4, 8 high pass bands of size 64 x 64 at level 3, 6 high pass bands of size 128 x128 at level 2, 6 high pass bands of size 256 x 256 at level 1. In total there will be 2 low pass bands and 30 high pass bands. The apparatus in **Claim 1** requires low pass bands for image registration that is obtained after D levels of decomposition.

3. The apparatus in **Claim 1**, where in the DTCWT sub bands are used for estimating the transformation vectors is carried out using one of the neural network algorithms defined as "2DFFNN_1" that comprises of M (M is number of frames that have been obtained after DTCWT decomposition) three layered (input layer, hidden layer, output layer) Feed Forward Neural Network Structure with input layer having 16 vectors, hidden layer having 32 neurons and tansig activation function, output layer consisting of 16 neurons with tansig activation function. The FFNN structure is arranged into M-arrays that process each of the M frames of DTCWT to generate 16 transformation vectors.
4. The apparatus in **Claim 1**, where in the DTCWT sub bands are used for estimating the transformation vectors is carried out using another neural network structure defined as "2DFFNN_1" that comprises of three FFNN structure as defined in Fig. 5, that are arranged connected into two stage structure. Stage 1 consists of two FFNNs forming 2D network structure that process the two frames of low pass DTCWT sub band and generates 16 outputs from each FFNN structure. The second stage processes both vectors generated from stage 1 and generate 16 output vectors that represent transformation vectors for image registration.
5. The apparatus in **Claim 1**, where in the DTCWT sub bands are used for estimating the transformation vectors is carried out using another neural network structure defined as "2DFFNN_1" that comprises of multiple neurons that process the DTCWT sub bands to generate transformation vectors as shown in Fig. 4. Every neuron in the hidden layer is connected with four pixels from the two frames of DTCWT sub bands to generate one output; each of the hidden layer neuron output generated is processed by the output layer to generate the transformation vector that is required for image registration. With hidden layer processing data from both frames of DTCWT sub bands, the hidden layer is a two dimensional structure that converts 2D input data to 1D output vector, the output layer is a 1D structure.
6. The DTCWT sub bands that are being processed by the apparatus in **Claim 3 to Claim 5** are obtained by image reordering of DTCWT sub bands. DTCWT low pass sub bands are sub grouped into non-overlapping 4 x 4 blocks and each of the 4 x 4 blocks is scanned in zig-zag method to convert 4 x 4 sub image into 16 x 1 vector. For example if the frame size of DTCWT sub band is 16 x 16, they are grouped into 16 sub images each of 4 x 4

size and are reordered into 16 columns with each column comprising of 16 pixels. As there are two frames the reordered matrix will have two data inputs representing frame 1 and frame 2 each having 16 column vectors, with each column vector having 16 elements.

7. The neural network apparatus as described in **Claim 3 to Claim 5** processes the two data inputs independently. The reordered data of two frames form the input layer for the neural network structure 2DFFNN_1 that are processed by the two dimensional structures.
8. From the four transformation vectors estimated as in **Claim 5** at each level average of these four transformation vectors are computed from each level, further the average of transformation vectors from all levels are computed. The algorithm described in present invention generates two transformation vectors one from low pass sub band processing and the other from high pass sub band processing.
9. Registration of low pass DTCWT sub bands are performed by considering low pass processed transformation vectors as in **Claim 8** and high pass bands are registered by considering high pass processed transformation vectors as in **Claim 8**.
10. The proposed DTCWT NN architecture estimates transformation parameters considering features from all sub bands and both from real and imaginary sub bands as in Claim 9.
11. The 2DFFNN_1 structure as in Claim 3 to 5 is designed using systolic array structure that performs the data processing operation of transformation estimation with 2D systolic array structure.
12. The proposed Systolic array architecture basically performs the matrix multiplication of the pre-processed input with the DTCWT filter coefficients.
13. The multiplication is performed between the pre-processed input and trained weight matrix. The element D represents the delay element which is used to delay the filter coefficient and pre-processed input while applying the data to the next PE in the architecture.
14. First three clock cycles are required to initially fill up the systolic array. So the output from PE11 is available at the 10 clock, the output from PE21 and PE12 is available at 11 clock cycle, the output from PE31,PE22 and PE23 is available at 12 clock cycle.
15. The output from PE41,PE32,PE23 and PE14 is available at 13 clock cycle, the output from PE42,PE33,PE24 and PE15 is available at 14 clock cycle, the output from PE43,PE34,PE25 and PE16 is available at 15 clock cycle, the output from PE44,PE35,PE26 and PE17 is available at 16 clock cycle.

16. The output from PE45, PE36, PE27 and PE18 is available at 17 clock cycle, , the output from PE46,PE37 and PE28 is available at 18 clock cycle, the output from PE47 and PE38 is available at 19 clock cycle, , the output from PE48 is available at 20 clock cycle.
17. In this invention, DTCWT the structure designed is implemented on FPGA and is verified for its functionality. These tests were carried out to verify its successful operation and its possibility of implementation. It can be reported that that this structure achieves much lower bit error rates assuming reasonable choice of the bases function and method of computation.
18. The software reference model is developed is able to evaluate the result.
19. DTCWT-FFNN structure is able to provide better performance in in terms of MSE in comparison with all other image registration techniques with 2% improvement.
20. Hardware implementation of 2DFFNN structure is carried out using systolic array architecture. Hardware modeling of 2DFFNN structure is carried out using Verilog HDL.
21. The maximum operating frequency achieved for the design is 242.066 MHz, the total power consumed is watts 1.128W.