Authors' responses to the comments of the paper

SMCB-E-01162002-0026

Replies to the Associate Editor:

Above all, I am very grateful for your efforts on my paper. I have carefully read and been thinking about all comments from you and the reviewers respectfully. I have tried hard to revise my paper and improve its quality according to your suggestions. The followings are answers to your comments and suggestions.

1. The whole paper is revised carefully to address the contributions of the paper clearly in many places. The second paragraph in the Introduction is rewritten to clarify the clues of my paper with respect to existing work.
2. More references are added and referred in the context.
3. All notations have been checked and defined clearly.
4. Some more words are added in Section 2.2.1 and Section 3.2 to address the meanings of the parameters. Their influences on system performance are discussed and clearly stated. Their reference values in average applications are also given in these sections as well as in Simulations.
5. Evaluation part, Section 3.3, of the paper addresses the computation efficiency, learning ability and incremental learning ability of the modified RFWR algorithm. The purpose is to show that the proposed modified RFWR is better in these three issues than the original RFWR, which is very important for dynamic multi-sensor data fusion applications. One of the important contributions of the paper is to propose a new cost function stated in Equation (10). We compare this with the original one to show better performance in learning. This comparison is useful to prove that the proposed modified RFWR has better learning efficiency than the original RFWR, thus is better for sensor fusion in dynamic environment. Concerning about the incremental learning ability. The incremental learning ability is accumulated with incremental training data in input space. The motivation of the paper is to propose a new kind of learning algorithm with incremental learning ability that is applicable for sensor fusion system in which the number and the sort of the sensor in the system may be variable in accordance with dynamic environment. Thus the paper provides performance evaluation on the algorithm we proposed to show that the modification of the original RFWR does not affect its incremental learning ability that is very important for dynamic fusion system. The paper is not trying to study how many ways or what are the other ways to gain incremental learning ability. One more paragraph is added in the end of

Section 3.3.3 to state the importance of the incremental learning ability on multi-sensor data fusion application.

6. I am sorry for my limited ability in written English. I have tried my best to improve presentation of the paper. I also asked two of my friends who have lived in English-spoken countries for more than 10 years to check the expressions of the paper. Hope the paper is more readable this time.

Replies to the Reviewer 1:

Above all, I am grateful to you for reviewing my paper and respect your comments very much. The following is the answer to each of your comments after I read them thoroughly several times.

1. Thanks for your positive comment on my paper.
2. The Reference of the paper is reorganized and many more important papers which relate to my research are listed. All reference papers are referred in the context. The relations between the RFWR and the RBF are clearly stated in the second paragraph of the Introduction.
3. There has no evidence that the RFWR is a type of neural network or a type of ellipsoidal basis function NN since the Schaal and Atkeson's pioneer work addressing RFWR. So I cannot state this in my paper. Actually, this is not so important since my paper emphasizes applications of RFWR in multi-sensor data fusion system rather than classification of RFWR.
4. Section 2.1 is modified to explain what the RFWR is more clearly. But I don't want to say RFWR is a kind of neural network since I cannot find evidence in literatures, though it's clear that RFWR is a kind of link net. The slash in the nodes of figure 1 means the linear relationship.
5. I have checked the whole paper and corrected the errors.
6. All your efforts are appreciated very much. I have checked the whole paper and corrected the errors.

Replies to the Reviewer 2:

Above all, I am very grateful to you for reviewing my paper. I appreciate your comments and respect your suggestions very much. The followings are the answers to each of your comments after I read them thoroughly several times.

1. I have checked the whole paper and corrected mistakes as possible as I can. Hope this time the paper is more readable.
2. The Introduction Section is carefully revised. Relevant techniques are reviewed and summarized more extensively with more literatures added in References.
3. I have done my best to improve the quality of the plots in the paper. Hope they are illustrative this time.
4. I have checked all variables in the paper and tried to define and explain them clearly, e.g. in Section 2.
5. Parameters used in the modified RFWR algorithm and simulations have been explained their meanings in details. Their normal values in applications are given. See following explanations of items 4 and 5. For more details, please check sections 2.2.1, 3.2 and 3.3.2.
6. See following explanations in items 9-13 for more details.
7. Many more relevant work has been added in References and referenced in context.

Specific:

1. b_{k}, b_{0}, k, P ,P^{n}, and e_{cv} are defined clearly in Sections 2.1, 2.2.2.
2. In Eq. (6), the cost function relates to y_{i}, which is the output in each individual receptive field. In Eq. (10), the cost function relates to y, which is the final weighted averaged output of all receptive fields. The difference of the idea of the two cost functions is obvious.
3. Thanks for your suggestion. We use *G*. to denote the gain factor.
4. More words are added to explain the meaning of the gain factor. Its usual value is clearly given in Section 3.3.2.
5. The meanings of w_{gen} and w_{prun} are explained in more details in section 2.2.1. More words are added to explain w_{a} and e_{a} in section 3.2. Their reference values used in simulations are given in section 3.3.2.
6. Figure 2 is corrected for its legend.
7. In fig 3 (c) and (d), ellipses represent receptive fields. I have added the statement in the last paragraph in page 10 in the revised paper. Due to space limitation of the paper, results from original RFWR are not given here. But the performance of the original RFWR and the modified RFWR are compared extensively in the paragraph below fig.3.
8. One of the important contributions of the paper is to propose a new cost function stated in

Equation (10). We compare this with the original one to show better performance in learning. This comparison is useful to prove that the proposed modified RFWR is better than the original RFWR for our tasks.

9. Concerning about the incremental learning ability. The incremental learning ability is accumulated with incremental training data in input space. The motivation of the paper is to propose a new kind of learning algorithm with incremental learning ability that is applicable for multi-sensor data fusion in dynamic environment. Thus the paper provides performance evaluation on the algorithm we proposed to show that the modification of the original RFWR does not affect its incremental learning ability that is very important for dynamic fusion system. The paper is not trying to study how many ways or what are the other ways to gain incremental learning ability. One more paragraph is added in the end of Section 3.3.3 to state the importance of the incremental learning ability on multi-sensor data fusion application.

10. Actually, the original RFWR cannot deal with sensor fusion task. This is the main inspiration of the research, which is stated in the third paragraph of Section 5. Tsai's method is a classical method in camera calibration and has been used widely in practice. So it is meaningful to compare performance of our method with Tsai's method.

11. Yes, but the modified RFWR alone (before fusion) has overall similar performance to that of Tsai's method which has been widely adopted in applications. This is actually a proof of effectiveness of the modified RFWR.

12. Tsai's method is of another category for calibration that leads to explicit descriptions of camera models. The model from Tsai's method is usually not able to fuse since each camera model obtained is independent. The proposed fusion-based calibration system is of a new kind of category that leads to no explicit parametric descriptions of the camera model. Calibration model of the camera are integrations of numerous pairs of linear and Gaussian kernel models in numerous receptive fields. Thus measurements from the fusion-based calibration model are the fused results from estimations on all different receptive fields obtained from the modified RFWR algorithm. An additional contribution of my paper is to propose a novel category for camera calibration techniques.

13. Comparisons between table 2 and 3 are just to show that the modified RFWR algorithm alone is also a good solution for camera calibration. We think this comparison is meaningful to evaluate the performance of the modified RFWR in camera calibration applications.

Replies to the Reviewer 3:

Above all, I am very grateful to you for reviewing my paper. I appreciate your comments on my paper and respect your suggestions very much. The followings are the answers to each of your comments and suggestions.

1.In real applications, *prior* knowledge about sensors involved in the system should be invoked, such as measurement performance of each sensor, measurement precisions (confidences) of each sensor across its range, etc. The weights of the weighted average are chosen according to *prior* confidences in the sensor measurements across its measurement ranges. The simple average scheme is just an exemplified scheme to be adopted in the paper for the fusion system. For the camera calibration case, it is well adapted. One could use more elegant fusion schemes for final results according to applications.

2.The meanings of the thresholds are explained in more details in Section 3.2. Their normal values are given in section 3.3.2.

3. The calibration strategy for the two-camera system based on the modified RFWR technique and the fusion scheme is actually a novel contribution to the camera calibration problem. Tsai's two-step camera calibration method is very famous, It is a very efficient technique for camera calibration and thus has served as a classical method for many years. Therefore it is normal, necessary and convincing to compare any new calibration methods with Tsai's method to evaluate their performances. This is also what we do for our method in our paper.

4. Spelling error in figure 2 has been corrected.

Replies to the Reviewer 4:

Thank your very much for your reviewing. I have revised the whole paper and hope to clarify the contribution of the paper more clearly. Multi-sensor data fusion in dynamic environment is a key problem to be studied in sensor fusion society. Developing a new fusion structure with the incremental learning ability is a good solution to this kind of problem. RFWR is a kind of learning scheme with incremental learning ability that can overcome many disadvantages from other incremental learning algorithms. But the idea it adopts for learning is not appropriate to the application in multi-sensor data fusion, since the learning in RFWR is emphasizing on individual receptive field that will leads to unbalanced measurements across a particular sensor's measurement ranges as well as among different sensors. Actually this research is motivated by the failure that we cannot get reasonable results by applying the normal RFWR in multi-sensor data fusion applications. Thus we modify the idea of learning for the normal RFWR to adapt to multi-sensor applications. We should state that the contributions of paper lie in:

a) We state that a fusion system with incremental learning ability is effective for dynamic sensor fusion problems;

b) We find that RFWR of the incremental learning ability is a good alternative in this category that can overcome the disadvantages of bias dilemma and negative interference from normal learning algorithm of incremental learning ability. Thus it is a good strategy for multi-sensor applications;

c) We state that direct application of the RFWR is not proper due to its computational complexity and its unbalanced measurements for sensors involved in the system. Thus we modify the learning idea that is adopted in the normal RFWR with the help of the idea of back propagation. We propose new cost function that is more efficient and more proper for dynamic sensor fusion applications. We evaluate the proposed new RFWR and show that the modified RFWR is more efficient in computation. Much more, all remarkable characteristics of the normal RFWR are meanwhile retained and somewhat improved. This is also a good extension for the RFWR method.

d) All these contributions are evaluated with a two-camera calibration system. We not only show that the proposed modified RFWR is effective for sensor fusion, but also propose a novel scheme for camera calibration that has a more flexible structure and more accurate measurements.

Incremental Learning with Balanced Updating on Receptive Fields for Multi-Sensor Data Fusion

Jianbo Su Jun Wang and Yugeng Xi

Institute of Automation
Shanghai Jiao Tong University
Shanghai, 200030, P.R.China

Correspondence Author: Dr. Jianbo Su
Tel: +86-21-62933694
Fax: +86-21-62933155
Email: jbsu@sjtu.edu

Abstract

This paper addresses applications of incremental learning ability from Receptive Field Weighted Regression (RFWR) algorithm in multi-sensor data fusion system. First, a new cost function is proposed based on the idea of back propagation (BP) to modify the original RFWR learning algorithm. This cost function emphasizes balanced learning among all receptive fields in addition to individual adjustments. With the proposed cost function and the BP learning algorithm, the computation efficiency of the modified RFWR is increased to a great extent while all remarkable advantages of the original RFWR are retained and somehow improved. All these features of the modified RFWR make it fit for applications in multi-sensor data fusion system. Thus a new fusion structure and algorithm with incremental learning ability is constructed based on the modified RFWR algorithm together with the weighted average algorithm. Experiments of a two-camera unified positioning system are implemented successfully to test the proposed computation structure and algorithms.

Keywords: Sensor Fusion, Incremental learning, Receptive field, Back Propagation.

1 Introduction

Frequently in practice, a multi-sensor fusion system needs to be upgraded by integrating additional sensors into the system to adapt to more complex environments and applications. Normally the structure and fusion algorithm of the fusion system should be designed from the very beginning for the upgrade, even if most of the sensors of the system are retained without any changes [2]. This inefficiency can be overcome if the fusion system has incremental learning ability [11]. With this ability, the structure of the fusion system is easy to be upgraded and only the added sensor needs to be trained before being included in the whole system.

Learning with spatially localized basis function [4,16,17] has been studied for many years in contrast to the learning with the global basis function [15]. A lot of applications have been accumulated ranging from robot control [6], chemical process modeling [7], nonlinear system estimation and control [8], to image coding [18], pattern recognition [9,12], etc. Incremental learning ability from local receptive-field is proved to be extremely useful for approximating unknown functional relationships between input and output data streams [11]. Among these, Schaal and Atkeson proposed a Receptive Field Weighted Regression (RFWR) algorithm with incremental learning ability in [1]. This algorithm is related to constructive learning [10] and local function approximation based on the well-known radial basis function networks. But with some particular nonparametric regression techniques involved, RFWR is more efficient for incremental function approximation in the sense that it is not necessary to store the training data and discard receptive fields after using them. In addition, it can overcome some difficulties occurring normally in the incremental learning tasks, especially the bias-variance dilemma [13] and the negative interference problems. Therefore the RFWR is able to deal with a sufficiently complex learning task and can be expected to have wide applications in many disciplines.

However, direct application of RFWR in the multi-sensor data fusion system is not practical. Although some techniques from nonparametric statistics, such as leave-one-out local cross validation and the stochastic approximation, improve the effectiveness of learning for RFWR, they contribute much to the computational complexity of whole learning process. Moreover, RFWR is a receptive field based learning system. Learning process is actually the process of updating size and shape of the receptive field describing model uncertainties. The idea of learning adopted in RFWR emphasizes only individual adjustment in each receptive field. If this learning scheme is utilized in multi-sensor data fusion, resultant sensor models by RFWR may unexpectedly have inconsistent measurements across their ranges and thus unbalanced contributions to the final fused result. Thus the learning algorithm in RFWR should be modified to fit for the characteristics of the multi-sensor fusion applications.

In this paper, a new cost function based on the idea of back propagation (BP) [5] for learning in RFWR is proposed so that balanced updates among all receptive fields are additionally reached. With the new cost function, back propagation algorithm is consequently involved for learning. These two modifications sharply reduce the computational complexity of the modified RFWR compared with that of the original RFWR. At the same time, all remarkable features of the RFWR, such as incremental learning ability and efficiency to approximating complex functions, are retained and improved. In addition, some tricks are also proposed to further improve the learning efficiency of the modified RFWR.

The modified RFWR algorithm is then served as an efficient tool for learning in multi-sensor data fusion problem, where learning ability has become extremely important, especially in dynamic unknown environment [3,21]. For a fusion system, its fusion structure is of the same importance as its fusion algorithm in the sense that both are tightly related to computation efficiency and final performance of the fusion system [20]. We show that the modified RFWR is inherently fit for sensor fusion problems not only in its learning ability but also in its computation structure. Combined with the weighted average strategy, a new computation paradigm is formed for multi-sensor data fusion system.

The rest of the paper is organized as follows. Section 2 briefly introduces the idea and algorithm of RFWR. Section 3 describes the ideas of the new cost function together with the back propagation learning algorithm. We will show that the incremental learning ability is retained and improved with the new cost function, while the computational complexity is decreased to a great extent. In Section 4, the computation structure of RFWR is extended to a new computing paradigm for sensor fusion applications by combining it with the weighted average scheme. In the new computation paradigm, the modified RFWR algorithm is used for learning and local fusion and the weighted average algorithm is used for final fusion. The success of the extended model is shown by its application in a unified two-camera positioning system described in Section 5. Conclusions are provided in Section 6.

2 Receptive Field Weighted Regression

2.1 Preliminaries

Receptive Field Weighted Regression (RFWR) algorithm is composed of two steps: 1) learning on the receptive field; and 2) building prediction from weighted average approach. Two models are involved in each receptive field. A linear model describes the input-output relations of the receptive field while a Gaussian function-based model describes the weight of the estimated output in this receptive field to the final estimation. For a training sample $(\mathbf{x}, y)$, assuming there are K receptive fields to be used to approximate function relations between $\mathbf{x}$ and y, the models of the *k*-th receptive field are:

$$\begin{cases} \hat{y}_k = (\mathbf{x} - \mathbf{c}_k)^T \mathbf{b}_k + b_{0,k} = \tilde{\mathbf{x}}^T \beta_k, & (1) \\ w_k = \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T \mathbf{D}_k (\mathbf{x} - \mathbf{c}_k)), & (2) \end{cases}$$

where k = 1, ..., K. Here, Equation (1) is a linear model, approximating relationship between input and output data in the k-th receptive field. $\hat{y}_k$ is the prediction of a query point $\mathbf{x}$ in the k-th receptive field. $\mathbf{c}_k$ is the location of the center of the k-th receptive field in input space. $\tilde{\mathbf{x}} = ((\mathbf{x} - \mathbf{c}_k)^T, 1)^T$, is the center-subtracted, augmented input vector. $\beta_k = (\mathbf{b}_k{}^T, b_{0,k})^T$, denotes the parameters of the locally linear model, composed of the coefficient vector $\mathbf{b}_k$ and the bias $b_{0,k}$ of the linear model. Equation (2) determines the size and shape of each receptive field in terms of a positive definite distance matrix $\mathbf{D}_k$ for the k-th receptive field. Normally $\mathbf{D}_k$ can be decomposed as $\mathbf{D}_k = \mathbf{M}_k^T \mathbf{M}_k$, in which $\mathbf{M}_k$ is an upper triangular matrix. The weight w_k corresponds to the activation strength of the k-th receptive field for a prediction $\hat{y}_k$. So a prediction $\hat{y}$ for a query point $\mathbf{x}$ is obtained from the normalized weighted sum of individual predictions $\hat{y}_k$ of all receptive fields:

$$\hat{y} = \frac{\sum_{k=1}^{K} w_k \hat{y}_k}{\sum_{k=1}^{K} w_k} \tag{3}$$

Figure 1 is the structure of RFWR, which shows the computation procedure of RFWR and the relations among models described by (1), (2) and (3).

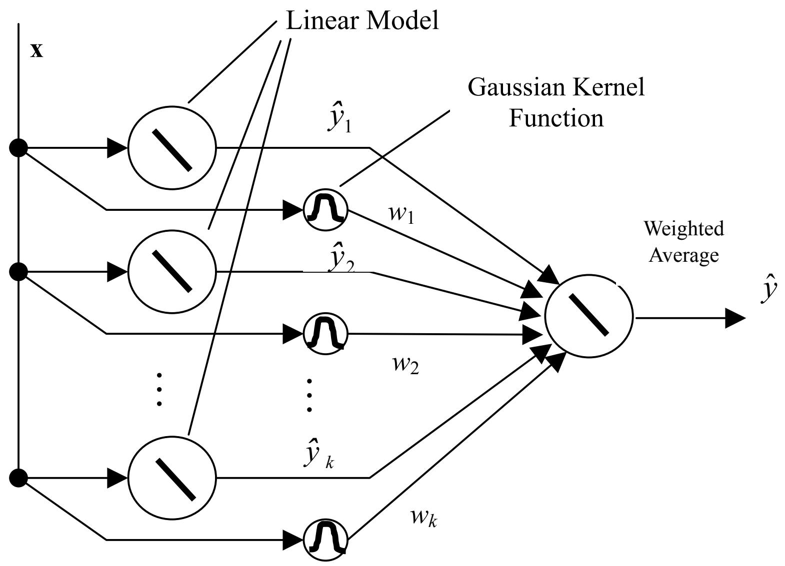


Figure 1. Structure plot of RFWR

2.2 Learning with RFWR

Learning task in RFWR includes three parts: 1) evolving receptive fields for approximation descriptions; 2) updating the linear model parameters $\boldsymbol{\beta}_k$, which is to learn the linear model for input-output relations in each receptive field; and 3) updating the distance matrix $\mathbf{D}_k$, or similarly, its decomposed matrix $\mathbf{M}_k$, which is to adjust the size and shape for each receptive field.

2.2.1 Evolving receptive field

A new receptive field is created if a training sample $(\mathbf{x}, y)$ does not activate any of the existing receptive fields by more than a threshold w_{gen}. When a new receptive field is created, the parameters related to the new receptive field are initialized. On the contrary, a receptive field is pruned if it overlaps another receptive field too much. The overlap can be detected when a training sample activates two receptive fields simultaneously more than a predefined threshold w_{prun}. We take the rule that the receptive field with the larger determinant of the distance matrix **D** is pruned. It is useful to note that w_{gen} and w_{prun} determine the overlap of the receptive fields. They could be chosen independently of a particular learning problem and should thus be considered constants of the algorithm and not open parameters. Empirical values for w_{gen} is 10%, and 80%~90% for w_{prun}.

2.2.2 Learning the linear model

Due to the linearity of $\boldsymbol{\beta}_k$ in the regression model in RFWR, $\boldsymbol{\beta}_k$ can be updated from a weighted regression in the batch form. Since each receptive field is updated in the same way, we drop the subscript *k* in the following discussion.

$$\boldsymbol{\beta} = (\mathbf{X}^{\mathbf{T}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{W}\mathbf{Y} = \mathbf{P}\mathbf{X}^{\mathbf{T}}\mathbf{W}\mathbf{Y} \tag{4}$$

where $\mathbf{X} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_p)^T$, $\mathbf{Y} = (y_1, y_2, \ldots, y_p)^T$, $\mathbf{W} = diag(w_1, w_2, \ldots, w_p)^T$, $\mathbf{P} = (\mathbf{X}^{\mathbf{T}}\mathbf{W}\mathbf{X})^{-1}$, and p is the number of training data points. According to [19], Equation (4) can be transformed into the recursive form given a training point $(\mathbf{x}, y)$:

$$\boldsymbol{\beta}^{n+1} = \boldsymbol{\beta}^{n} + w\mathbf{P}^{n+1}\tilde{\mathbf{x}}e_{cv}^{T} \tag{5}$$

where

$$\mathbf{P}^{n+1} = \frac{1}{\lambda}\left(\mathbf{P}^{n} - \frac{\mathbf{P}^{n}\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{T}\mathbf{P}^{n}}{\frac{\lambda}{w} + \tilde{\mathbf{x}}^{T}\mathbf{P}^{n}\tilde{\mathbf{x}}}\right), \quad e_{cv} = (y - \boldsymbol{\beta}^{n^T}\tilde{\mathbf{x}})$$

A forgetting factor λ is included here in order to gradually cancel contributions from previous data points.

2.2.3 Learning the shape and size of the receptive field

It is clear that the update of $\mathbf{M}_k$ cannot be implemented by using Equation (2) in a direct way. Therefore, a cost function is introduced assuming that there exists a batch of training data points:

$$J = \frac{1}{W}\sum_{i=1}^{p} w_i \left\| y_i - \hat{y}_i \right\|^2, \quad W = \sum_{i=1}^{p} w_i \tag{6}$$

However, training in terms of this cost function will result in over-fitting problem. Thus several techniques, such as leave-one-out local cross validation, etc., are involved, which results in the following form:

$$J = \frac{1}{W}\sum_{i=1}^{p} \frac{w_i \left\| y_i - \hat{y}_i \right\|^2}{(1 - w_i \widetilde{\mathbf{x}}_i^T \mathbf{P} \widetilde{\mathbf{x}}_i)^2} + \gamma \sum_{i,j=1}^{n} D_{ij}^{\ 2} \tag{7}$$

where $D_{i,j}$ is the element in distance matrix **D**. With the cost function described in (7), **M** can be adjusted by using gradient descent method with learning rate α:

$$\mathbf{M}^{n+1} = \mathbf{M}^n - \alpha \frac{\partial J}{\partial \mathbf{M}} \tag{8}$$

Computation of $\partial J / \partial \mathbf{M}$ in (8) can further be transformed to conduct in an incremental way by adopting the stochastic approximations, which are normally very computationally expensive [19].

3 New Cost Function for RFWR

3.1 New cost function

In RFWR, the shape and size of a receptive field is determined by its associated Gaussian kernel function as described in Equation (2). Thus evolving the shape and size of receptive field turns out to be updating of the Gaussian kernel function, which is further to update the decomposed distance matrix **M** as shown in Section 2. As mentioned before, update of the Gaussian kernel function has to resort to a suitable cost function because Gaussian kernel function does not describe the relationship between input $\mathbf{x}$ and prediction $\hat{y}_k$ directly. In RFWR, the cost function described by (6) or (7) is technically set up from the viewpoint of local learning, and learning process focuses on individual adjustment of each receptive field. This is unfortunately not appropriate for applications in multi-sensor data fusion since a trade-off among all the receptive fields should be achieved to implement the same and balanced reliability for all measurements of each sensor.

The output of Gaussian kernel function in (2) is the weight w_k that describes the contribution of estimation $\hat{y}_k$ from associated linear model to the final estimation $\hat{y}$. Updating the shape and size of receptive fields is basically to adjust the weight w_k. From figure 1, we can see that this work is very similar to that in training a neural network. Thus the idea of back propagation can be employed here and a similar cost function used in BP network can be borrowed:

$$J' = \frac{1}{2}\|y - \hat{y}\|^2, \quad where\ \hat{y} = \sum_{k=1}^{K} w_k \hat{y}_k \Big/ \sum_{k=1}^{K} w_k \ . \tag{9}$$

This cost function is to adjust weights w_k in order to minimize the bias between the actual output y and the prediction of RFWR $\hat{y}$. Moreover, according to the feature of RFWR, there exists the relationship between the weights w_k and local prediction bias $(y - \hat{y}_k)$ in the update process, which leads to the second cost function:

$$J'' = \sum_{i=1}^{K} w_i (y - \hat{y}_i)^2 \Big/ \sum_{i=1}^{K} w_i \tag{10}$$

This cost function emphasizes that weights w_k should be adjusted in terms of the local prediction bias so that the coordination among the adjustments of all the receptive fields can be achieved. By combining Equation (9) and (10), a new cost function is obtained and adopted in this paper:

$$J = J' + J'' = \frac{1}{2}\|y - \hat{y}\|^2 + \sum_{i=1}^{K} w_i (y - \hat{y}_i)^2 \Big/ \sum_{i=1}^{K} w_i \tag{11}$$

Equation (11) means that the update of all the receptive fields is dedicated to global prediction error and local prediction bias. This idea improves that of original RFWR and is more complete and reasonable. The new cost function in (11) focuses on the balance among all the receptive fields in addition to adjustments in individual fields, while cost function of the original RFWR in (6) or (7) only emphasizes on adjustments in individual fields. We believe that balance among all receptive fields is important because this means estimations from all receptive fields have identical contributions to the final result. This consideration is essential for the applications of the modified RFWR algorithm in multi-sensor fusion systems.

Update of the receptive fields with the new cost function is done by minimizing J with respect to **M**. After tedious mathematical derivations, we have

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{M}_j} &= \frac{\partial J'}{\partial \mathbf{M}_j} + \frac{\partial J''}{\partial \mathbf{M}_j} = (\hat{y} - y)\frac{\hat{y}_j J_2 - J_1}{J_2^2}\frac{\partial w_j}{\partial \mathbf{M}_j} + \frac{(y - \hat{y}_j)^2 J_2 - J_3}{J_2^2}\frac{\partial w_j}{\partial \mathbf{M}_j} \\ &= (\hat{y} - y)\frac{\hat{y}_j - \hat{y}}{J_2}\frac{\partial w_j}{\partial \mathbf{M}_j} + \frac{(y - \hat{y}_j)^2 - J_3/J_2}{J_2}\frac{\partial w_j}{\partial \mathbf{M}_j}\end{aligned} \tag{12}$$

where we define $J_1 = \sum_{k=1}^{K} w_k \hat{y}_k$, $J_2 = \sum_{k=1}^{K} w_k$, $J_3 = \sum_{i=1}^{K} w_i (y - \hat{y}_i)^2$. For the second equation in (12), we use the relation: $J_1 = \hat{y} J_2$, which is an equivalent form of Equation (3).

Equation (12) is efficient for iterating **M** in RFWR training. But normally the convergence speed for **M** is slow. Thus a gain factor G is introduced to improve the training speed and training precision, which leads to the following equation:

$$\frac{\partial J}{\partial \mathbf{M}_j} = \frac{\partial J'}{\partial \mathbf{M}_j} + \frac{\partial J''}{\partial \mathbf{M}_j} = G \cdot \left[(\hat{y} - y) \frac{\hat{y}_j - \hat{y}}{J_2} \frac{\partial w_j}{\partial \mathbf{M}_j} + \frac{(y - \hat{y}_j)^2 - J_3/J_2}{J_2} \frac{\partial w_j}{\partial \mathbf{M}_j} \right] \quad (13)$$

Computational complexity from (13) or (12) is far less than that of Equation (8). We will show this later by simulation comparisons.

3.2 Additional skills for the modified RFWR

The performance and computation efficiency of the modified RFWR can be further enhanced by the following two tricks in applications:

1) The effect of updating a receptive field can almost be neglected for a training data point if its corresponding weight is less than w_a, $0 < w_a < w_{gen}$. Of course this neglect is surely affect the training accuracy, thus the value of w_a is normally set as $w_a = (1\% \sim 5\%) w_{gen}$ according to applications and requirements for training accuracy. So we can judge which receptive field will be adjusted by comparing w_k with w_a in order to improve the training efficiency.
2) Considering the approximation error $|y - \hat{y}|$. If $|y - \hat{y}|$ is sufficiently small, say, smaller than a predefined threshold e_a: $|y - \hat{y}| < e_a$, we can only take the cost function in Equation (10) instead of that in Equation (11). This is because if $|y - \hat{y}|$ is small enough, the cost function in Equation (9) has already been approximately minimized. Thus further minimization of this cost function contributes very little to iterations of **M** but much to computational complexity. Hence the cost function in Equation (10) is the main part that should be minimized for updating **M**. This trick is useful and simplifies the computational complexity in many applications. Ignorance of little approximation error $|y - \hat{y}|$ is surely realted to the final training accuracy, thus the threshold e_a is usually determined according to practical learning task.

3.3 Evaluations of the modified RFWR

3.3.1 Comparisons of the computation efficiency

We first show the computation efficiency of the modified RFWR algorithm improved by the new cost function and the BP learning scheme over the original RFWR. We use a PC with CPU- Intel Celeron 366, OS-Windows 98, Simulation Tools-Matlab5.3 for all simulation calculations. In the simulation, we construct a RFWR model with 100 receptive fields and then use one training data point to finish one epoch of training by using RFWR and modified RFWR respectively. Figure 2 shows the result.

From figure 2, we can see that the more the receptive fields, the more time the modified RFWR saves over the original RFWR. This is especially important for training task of approximating complex functions since the more complex the learning task is, the more receptive fields it requires. Thus more time for learning can be saved.

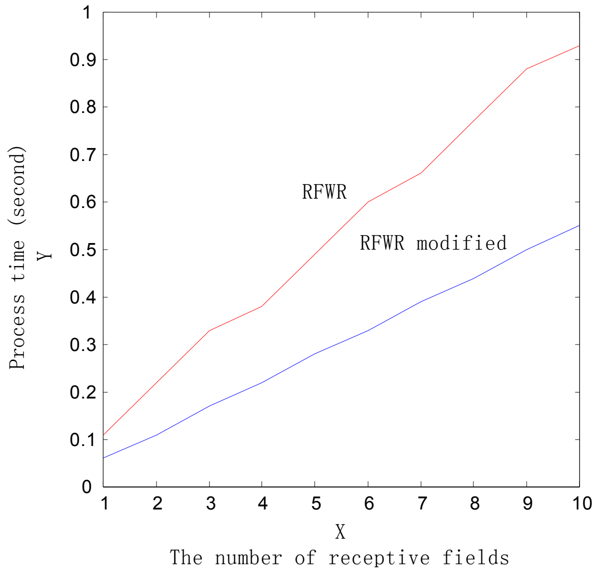


Figure 2: Comparison of training time

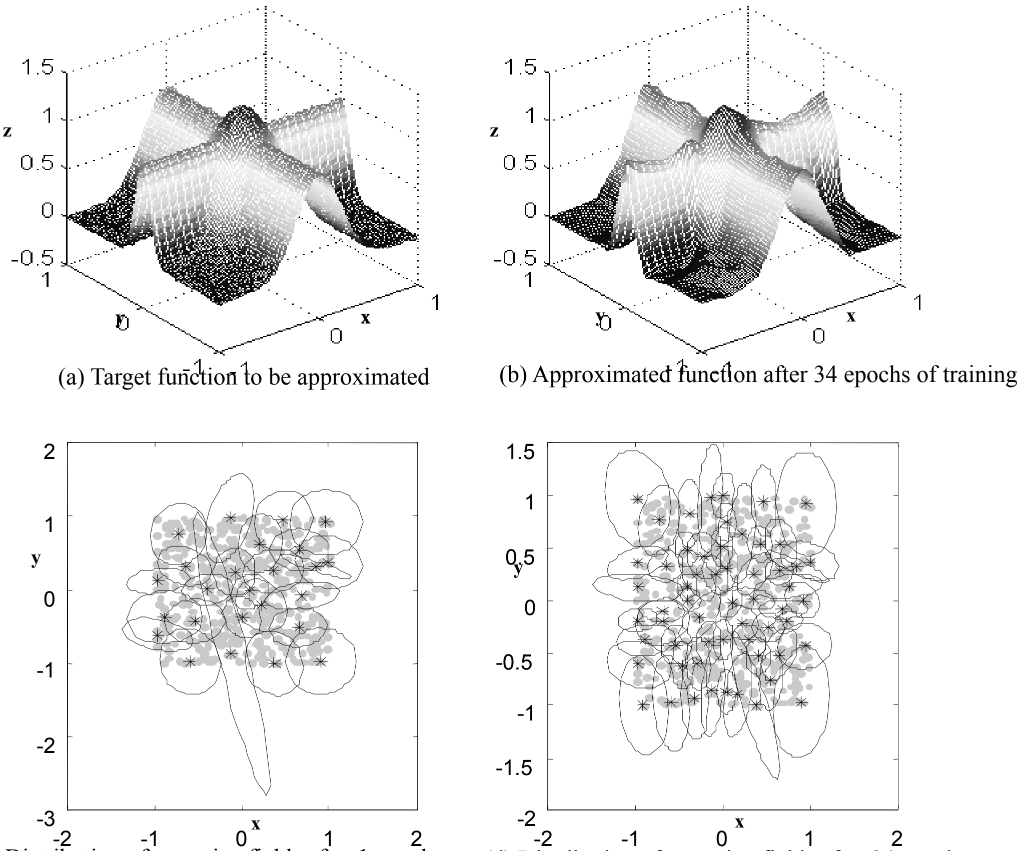
3.3.2 Comparison in the learning ability

The complex function used in [1] is adopted here to investigate the learning ability of the modified RFWR and compare with that of the normal RFWR.

$$z = \max\left\{e^{-10x^2}, e^{-50y^2}, 1.25e^{-5(x^2+y^2)}\right\} + N(0, 0.01), |x, y| \le 1 \tag{14}$$

We take this task to show the advantages of the modified RFWR over the original RFWR on the same training set. Similarly, 500 training samples are drawn uniformly from the input space. The test set consists of 1681 data points corresponding to the vertices of a 41x41 grid over a unit square. Corresponding output values are the exact function values. The approximation error is measured as a normalized mean squared error, *nMSE*. The initial values of the parameters of the modified RFWR in training are set to $\mathbf{M}^0$=5**I** (**I** is the identity matrix), $w_{gen} = 0.1$, $w_{prun} = 0.9$, $w_a = 0.001$, $e_a = 0.001$, *G*=100, and $\mathrm{P}^0 = \begin{bmatrix} 1/0.001^2 & 0 \\ 0 & 1/0.001^2 \end{bmatrix}$.

The simulation results are shown in Figure 3. Figure 3 (a) is the function to be approximated; figure 3 (b) is the approximate result after 34 epochs. Figure 3 (c) is the receptive field in input space after 1 epoch, and figure 3 (d) is the final receptive fields after 34 epochs (In figure 3(c) and (d), receptive fields are represented by the ellipses, "*" stands for the center of each receptive field and the training data is displayed by ".").



(a) Target function to be approximated (b) Approximated function after 34 epochs of training

(c) Distribution of receptive fields after 1 epoch (d) Distribution of receptive fields after 34 epochs

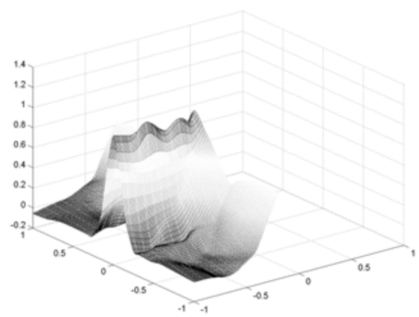
Figure 3. Simulation results

We compare the simulation results in Figure 3 with the results reported in [1]. With the modified RFWR, 25 receptive fields are created after 1 epoch illustrated in Figure 3(c), while with the original RFWR, only 16 receptive fields were created. This means that the modified RFWR has higher learning efficiency than the original one. After 34 epochs of training, the approximation error meets the training termination condition, $nMSE<0.02$ and 51 receptive fields are obtained. If it is trained with original RFWR, 50 epochs are needed before converged to the same *nMSE*, and 48 receptive fields are finally created. This means the convergence speed of the modified RFWR is faster than that of the original RFWR. Moreover, since the learning in the modified RFWR is conducted evenly in all receptive fields, more receptive fields are obtained for final results that have balanced convergence preciseness over all receptive fields. From the comparisons, we can conclude that the performance and learning ability of the modified RFWR with the new cost function and the BP algorithm is obviously better than that of the original RFWR.

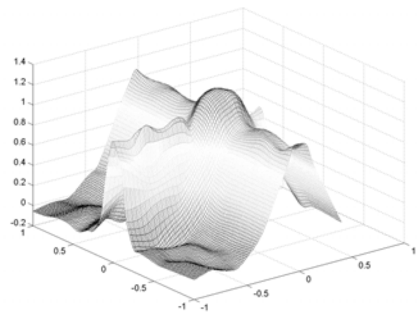
3.3.3 Incremental Learning Ability

Although the cost function and the learning algorithm in modified RFWR to update the kernel function is different from that in RFWR due to different cost function, incremental learning ability is retained. We show this by the following simulations.

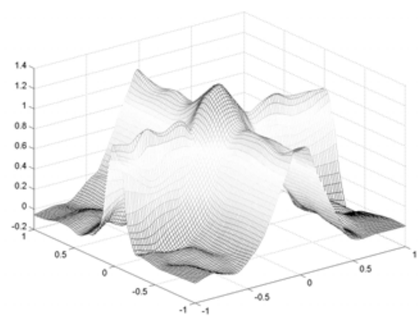
In this simulation, the input space is divided into three subspaces: $T_1 = \{-1.0 < x < -0.2\}$, $T_2 = \{-0.4 < x < 0.4\}$, and $T_3 = \{0.2 < x < 1\}$. They have overlaps in some parts. First the algorithm is used to train the function on T_1 only and tested on T_1 with convergence condition. The resulting approximated function is depicted in Figure 4 (a). Then the trained results from T_1 (including linear models, Gaussian kernel functions, and receptive fields created) are trained further on T_2 only and tested on $T_1 \cup T_2$. The resulting approximated function is plotted in figure 4 (b). It is seen that the functional relations in the input subspace T_1 is still well retained after it is trained in T_2. Finally the trained results from $T_1 \cup T_2$ are trained on T_3 only and tested on the whole space, i.e. $T_1 \cup T_2 \cup T_3$. Figure 4 (c) illustrates the final results of the approximated function. Through Fig. 4(a) to (c), we can see that when the modified RFWR is used to train a function, the results from former training spaces hold in consequent training spaces and are not required to be trained again. This exactly verifies the incremental learning ability of the modified RFWR.



(a) Training in T_1



(b) Training in T_2



(c) Training in T_3

Figure 4. The incremental learning ability with the modified RFWR

The incremental learning ability is especially important for applications in dynamic multi-sensor data fusion where structure of a multi-sensor system, thus the algorithm for data fusion, might be variable online. Thus efficient task implementations in different phases could be achieved by involving different sorts and numbers of the sensors in the sensor system.

4 Fusion system based on the modified RFWR

A RFWR model can be interpreted as a system composed of a set of experts [14]. Thus the final prediction can be regarded as the consensus result from all the experts as shown in Equation (3). This characteristic of RFWR is similar to that of a multi-sensor data fusion system. Moreover, RFWR presents a valid scheme to model an uncertain system and a RFWR model is used to predict the actual output of a system according to its input. Therefore, RFWR can be extended to be a good solution to implement fusion. Figure 5 shows the structure for a fusion system by combining the RFWR model and weighted average scheme.

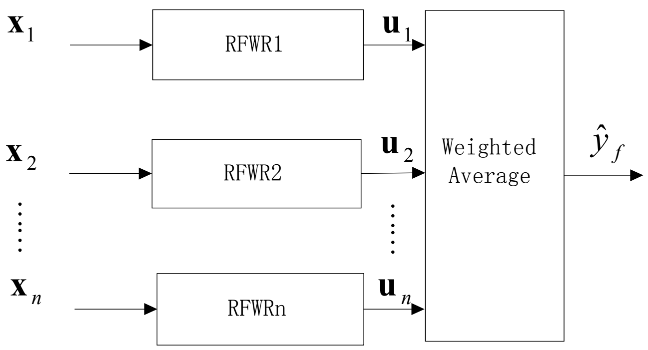


Figure 5: Structure of the fusion system

In Figure 5, $\mathbf{u}_i$ is defined as the uncertain description of the output of *i*-th RFWR model, that is, $\mathbf{u}_i = \{(y_1^i, w_1^i), (y_2^i, w_2^i), \ldots, (y_{K_i}^i, w_{K_i}^i)\}, i = 1,2,\ldots,n$, K_i is the number of receptive fields in the *i*-th RFWR model. Thus the fused output is obtained by

$$\hat{y}_f = \frac{\sum_{i=1}^{K_1} w_i^1 \hat{y}_i^1 + \sum_{i=1}^{K_2} w_i^2 \hat{y}_i^2 + \cdots + \sum_{i=1}^{K_n} w_i^n \hat{y}_i^n}{\sum_{i=1}^{K_1} w_i^1 + \sum_{i=1}^{K_2} w_i^2 + \cdots + \sum_{i=1}^{K_n} w_i^n} \tag{15}$$

If a RFWR model is considered as a team of experts, the fusion system based on RFWR model can be a set of several expert teams. In addition, the new cost function employed in the modified RFWR emphasizes balanced update among receptive fields. If a sensor in a multi-sensor system is represented by a RFWR model with many receptive fields, then balanced update of the receptive fields means the sensor has a consistent performance across its measurement range. A fusion system of several sensors will be modeled by several RFWR models, with each RFWR representing the model of each sensor. The final fused result of the fusion system is accomplished by the weighted average algorithm, which addresses contribution of each sensor via its corresponding weight. With these characteristics of RFWR and weighted average, the fusion model illustrated above can meet the requirements of a fusion system. Furthermore, it can enhance the performance of the fusion system, such as accuracy and robustness.

Since the modified RFWR proposed in the former sections retains the computation structure of the original RFWR and improves its learning efficiency, it is a better algorithm to be integrated into a fusion system. Thus the modified RFWR algorithm is used to realize the fusion system depicted in Fig. 5.

5 Experiments

Multi-camera systems can not only provide a larger quantity of visual information, but also improve the accuracy and reliability of the information needed [21]. Therefore it has attracted great attention since long ago. A two-camera vision system has been set up in our lab to form a stereovision system, with which the 3-D location of the object in the world coordinate system can be obtained from its projections in two 2-D image planes. Figure 6 shows the configuration of our system.

Figure 6. A two-camera vision system

The purpose of calibration model is to obtain measurements for control. Thus if we can develop a fusion system for the two-camera system that can provide measurements of the environment, the calibration model of the two-camera system is equivalently obtained even though we do not have explicit parametric descriptions of the camera models. Furthermore, the measurements from fusion system can be expected to be more accurate than that from either single one. So we use the modified RFWR model for learning the calibration model of the camera and use fusion algorithm for final measurement.

The fusion scheme presented in Section 4 is adopted here for the camera calibrations. We use the modified RFWR algorithm to learn the calibration model for each camera. Here we should point out that only modified RFWR can accomplish the fusion task because the modified RFWR has the modified cost function that emphasize on balance of receptive fields, which inherently meet requirements of multi-sensor data fusion.

Since the experiment is just to show the application of the modified RFWR in camera calibrations, we restrict ourselves to a 2-D to 2-D mapping calibration problem, which means the depth information of the object in the world coordinate system is omitted. Even though, the 2-D to 2-D calibration problem is still a complex nonlinear mapping that is sufficient to evaluate the performance of the fusion scheme and algorithm developed. The calibration problem with the fusion method is solved in two stages. The camera models are first trained and tested with sampling data individually. The final results are obtained from the fusion system with weighted average scheme.

To simplify the learning tasks, each camera employs two RFWR models, either of which is for mapping in x or y direction from the world coordinate system to the image plane:

$$\begin{cases} x_c^i = RFWR_x^i(x_w, y_w) \\ y_c^i = RFWR_y^i(x_w, y_w) \end{cases}, i = 1,2 \tag{16}$$

where (x_c^i, y_c^i) (i=1,2) is the location of the object in the i-th camera′s image plane and (x_w, y_w) is the location of the object in world coordinate space. The final fusion results are obtained from Equation (15).

For gathering the training data, the same procedure as in [22] is followed here so that 504 training samples and 252 test samples are obtained with the help of reference grids. The statistical error of the samples is illustrated in the Table 1.

Table 1. The statistical error of the samples (unit: mm)
(*ME*-Max Error, *MSE*-Mean Square Error)

Item	**Statistical error in X axis**		**Statistical error in Y axis**	
	ME	*MSE*	*ME*	*MSE*
Camera 1	0.613	0.106	0.661	0.096
Camera 2	0.609	0.084	0.617	0.104

The corresponding models are obtained via the training procedure with the modified RFWR algorithm. The training results are shown in Table 2, where RF# is the number of receptive fields obtained after 7 epochs.

To illustrate the advantage of the modified RFWR in modeling, we compare it with a well-known two-stage method proposed by Tsai in [22]. Results of Tsai’s method for the same two-camera unified calibration problem are shown in Table 3. Comparing the results in Table 2 with Table 3, we can see that in the training stage, all maximum errors (*ME*) and mean square errors (*MSE*) of modified RFWR are smaller than those of Tsai's method. In the test stage, most of the *ME*s from the modified RFWR are smaller except in the x direction of the 2nd camera's image plane and the two methods are of the same performance for *MSE*s. This means that the modified RFWR itself is a good solution for the camera calibration problem.

Table 2. Training result with the modified RFWR (unit: mm)
(*ME*-Max Error, *MSE*-Mean Square Error)

Items		**Camera 1**		**Camera 2**	
		RFWR_x^1	RFWR_y^1	RFWR_x^2	RFWR_y^2
RF#		121	55	140	92
Training Error	*ME*	3.3274	2.7429	3.8856	2.7995
	MSE	0.7689	0.9408	1.1632	0.5093
Test Error	*ME*	4.3420	4.2714	4.4406	3.4515
	MSE	2.5152	2.2641	2.9102	2.3275

Table 3. Training result of Tsai's method (unit: mm)
(*ME*-Max Error, *MSE*-Mean Square Error)

Items		**Camera 1**		**Camera 2**	
		x	*y*	*x*	*Y*
Training Error	*ME*	3.4551	2.9237	4.3765	3.2456
	MSE	1.9908	0.9465	2.9001	1.1647
Test Error	*ME*	5.4986	4.5826	4.3523	3.8192
	MSE	2.4941	2.6451	3.3570	1.6707

The fusion model can be achieved after training and testing stages described above, and used to get more accurate and robust measurement results. 50 samples are acquired randomly in the whole sample space and used to evaluate the performance of the fusion method. The final results are shown in Table 4. Results from Tsai's method are also provided as further comparisons.

In Table 4, it is shown that the performance of individual RFWR model is worse than that of Tsai's model. But with the fusion algorithm proposed, the fused results are much better than either individual one's as well as results from Tsai's method. This verifies that the fusion structure shown in Figure 5 and the fusion algorithm based on the modified RFWR algorithm are successful in multi-sensor fusion applications. And the fusion strategy from Equation (15) and the fusion model from Equation (16) are also effective for this application.

Table 4. Results of the fusion system based on the modified RFWR algorithm (unit: mm)
(*ME*-Max Error, *MSE*-Mean Square Error)

Items	**RFWR (Camera 1)**	**RFWR (Camera 2)**	**Fusion**	**Tsai （Camera 1）**	**Tsai （Camera 2）**
ME	2.8864	4.1606	**1.2890**	1.8361	3.2904
MSE	1.9998	3.0158	**0.8996**	1.7056	1.6774

6 Conclusions

An efficient algorithm with incremental learning ability is studied and introduced into the multi-sensor fusion system in this paper. We replace the cost function in RFWR that emphasizes balanced updating on receptive fields in addition to individual adjustments. This means, for multi-sensor data fusion applications, all measurements of a sensor have identical reliability across its range. With the modified cost function, idea of back propagation algorithm is introduced to RFWR for the updating of the receptive fields. Consequently the computation complexity is reduced to a great extend, while all other remarkable features, such as incremental learning ability, etc., are retained and improved. All these improvements for the RFWR algorithm make it more appropriate for its applications in multi-sensor fusion systems. Thus a fusion system with its structure and learning algorithm is developed to be endowed with incremental learning ability, which is very important for sensor fusion in dynamic environments. Experiments of a unified calibration process of a two-camera system and extensive comparisons of its performance with Tsai's method are provided to show the performance of the proposed fusion system and its successful application. Future work will lie in new applications of proposed algorithm and fusion system with incremental learning ability in dynamic environments.

Acknowledgment:

This work was partially supported by the National Natural Science Foundation of China under grant 69889501.

References

1. Schaal, S., and Atkeson, C. G. "Constructive incremental learning from only local information", *Neural Computation*, 1998, 10(8), pp.2047-2084.
2. Durrant-Whyte, H. F., "Elements of sensor fusion", *IEE Colloquium on Published*, 1991,

Page(s): 5/1-5/2.
3. Luo, R.C., and Kay, M. G., "Multisensor integration and fusion in intelligent systems", *IEEE Transaction on Systems, Man and Cybernetics*, 1989, 19(5), pp. 901-931.
4. Poggio, R., and Girosi, F., "Regularization algorithms for learning that are equivalent to multilayer networks", *Science,* 1990, 246(4945), pp.978-982.
5. Rumelhart, D.E., Hinton, G. E., and Williams, R. J. "Learning representations by back-propagating errors", *Nature*, 1986, 323, pp. 533–536.
6. Atkeson, C. G., and Schaal, S., "Memory-based neural networks for robot learning", *Neurocomputing*, 1995, 9, pp.243-269.
7. Schmitz, G.P.J. and Aldrich, C., "Neurofuzzy modeling of chemical process systems with ellipsoidal radial basis function neural networks and genetic algorithms", *Computers and Chemical Engineering*, 1998, 22, pp. S1001-1004.
8. Cannon, M., and Slotine, J. E., "Space-frequency localized basis function networks for nonlinear system estimation and control", *Neurocomputing*, 1995, 9(3), pp.293-342.
9. Carpenter, G. A., and Grossberg, S. "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics, and Image Processing*, 1987, 37, pp.54-115.
10. Frean, M., "The upstart algorithm: A method for constructing and training feedforward neural networks", *Neural Computation*, 1990, 2, pp.198-209.
11. Fritzke, B., "Incremental learning of locally linear mappings", In: *Proceedings of the International Conference on Artificial Neural Networks*, Paris, France, Oct.9-13, 1995.
12. Furlanello, C., and Giuliani, D., "Combining local PCA and radial basis function networks for speaker normalization", In: Girosi, F., Makhoul, J., Manolakas, E., & Wilson, E. (Eds.), *Proceedings of the 1995 IEEE Workshop on Neural Networks for Signal Processing* V, pp.233-242. New York: IEEE.
13. Geman, S., Bienenstock, E., and Doursat, R., "Neural networks and the bias/variance dilemma", *Neural Computation*, 1992, 4, pp.1-58.
14. Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E., "Adaptive mixtures of local experts", *Neural Computation*, 1991, 3, pp.79-87.
15. Pao, Y.H., Park, G.H. and Sobajic, D. J., "Learning and generalization characteristics of the random vector functional link net", *Neurocomputing* 1994, 6, pp.163-180.
16. Looney, C.G., "Pattern Recognition Using Neural Networks", Oxford University Press, NY, 1997, pp.258-264.
17. Hunt, K.J., Haas, R. and Murray-Smith, R., "Extending the functional equivalence of radial basis function networks and fuzzy inference systems", *IEEE Trans. On Neural Networks*, 1996, 7(3), pp.776-781.
18. Olshausen, B. A., and Field, D. J., "Emergence of simple-cell receptive field properties by learning a sparse code for natural images", *Nature*, 1996, 381, pp.607-609.

19. Ljung, L., and Söderström, T. "Theory and practice of recursive identification", Cambridge, MIT Press, 1986.
20. Wang, J., Su, J.B., and Xi, Y.G., "COM-based software architecture for multisensor fusion system", *International Journal of Information Fusion*, 2001, 2(4), pp.261 – 270.
21. Do, Yongtae, "Application of neural network for stereo-camera calibration", *Proceedings of the 1999 IEEE/RJS International Conference on Intelligent Robots and Systems*, pp.2719-2722.
22. Tsai, R.Y., "An efficient and accurate camera calibration technique for 3D machine vision." *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pp.364-374.