Internal Use Only
非公開

TR - H - 123

0020

MVHBF:
A Network that Approximates Multi-Valued, Vector-Output Mappings

Masahiko Shizawa

1995. 1. 30

ATR人間情報通信研究所

〒619-02 京都府相楽郡精華町光台2-2 ☎ 0774-95-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-774-95-1011
Facsimile: +81-774-95-1008

MVHBF*: A Network that Approximates Multi-Valued, Vector-Output Mappings

(*MVHBF=Multi-Valued HyperBasis Function network)

Masahiko Shizawa
ATR Human Information Processing Research Laboratories

Outline

0. Introduction (multi-valued mappings in inverse problems).

1. A mathematical formalism: multi-valued, vector-output function approximation.

2. The basic structure: h -valued, Scalar-output MVRN.

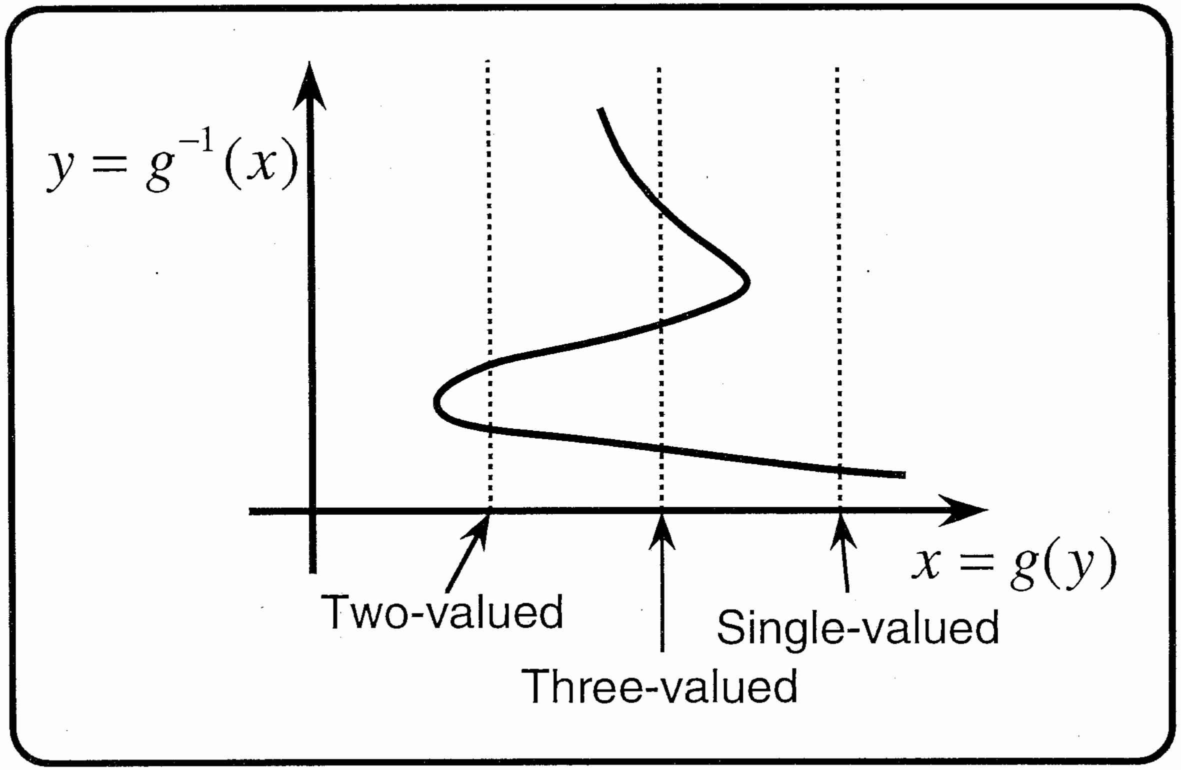
3. MVHBF: A function approximation network that combines unsupervised and supervised learning for multi-valued function approximation.

4. Applications (computer vision, inverse kinematics, sensor fusion, biological information processing, e.t.c.).

5. Conclusions and future research directions.

Presentation at ATR Symposium on Face and Object Recognition, Jan. 17, 1995.

Inverse models of a forward single-valued mapping is generally a multi-valued mapping.

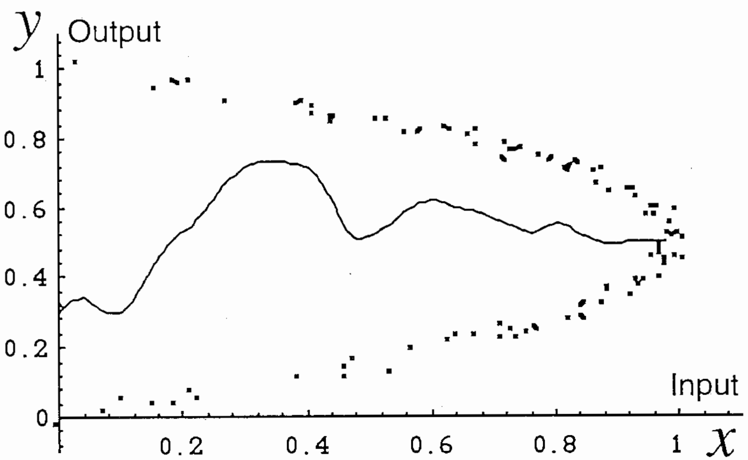


Inverse of C^1 mapping $g : y \mapsto x$

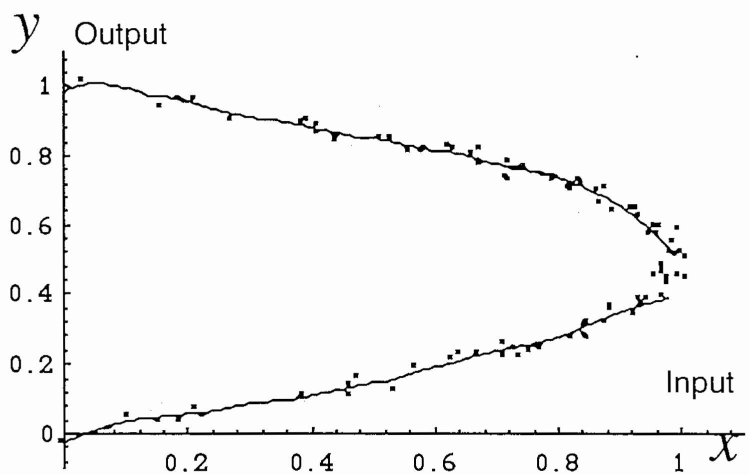
∴ Learning of inverse models from examples must be a multi-valued function approximation.

Need of multi-valued function approximation: function approximation of inverse of a quadric function

$$x = (y - 0.5)^2 \rightarrow y = \pm\sqrt{x} + 0.5$$



(a) approximation by cubic spline



(b) approximation by cubic spline version of two-valued MVRN

- 3 -

Applications of multi-valued function approximation network

1. A supervised learning technique

1.1 Learning in computer vision (learning of vision modules from examples).

1.2 Learning of inverse kinematics.

1.3 Learning in direct sensor-actuator coordination.

1.4 Learning in sensor fusion problems (*e.g.* multiple target detection).

1.5 Learning in nonlinear inverse problems in general.

2. A tool for modeling and analyzing numerical data

2.1 Reconstruction of complex overlapping surfaces from sparsely distributed data.

2.2 General tool for data analysis and modeling.

3. A tool for modeling biological information processing

3.1 Learning of multiple and ambiguous perception and perceptual transparency.

3.2 Model of multiple overlapping transparent surfaces perception.

3.3 Model of consciousness: choosing or deciding one output out of h possible outputs.

A mathematical formalism of h-valued, vector-output mapping

Logical representation

$$\boldsymbol{y} = f_1(\boldsymbol{x}) \vee \boldsymbol{y} = f_2(\boldsymbol{x}) \vee \cdots \vee \boldsymbol{y} = f_h(\boldsymbol{x})$$

Logical disjunction of h mappings

Algebraic representation

$$\left(\boldsymbol{y} - \boldsymbol{f}_1(\boldsymbol{x})\right) \otimes \left(\boldsymbol{y} - \boldsymbol{f}_2(\boldsymbol{x})\right) \otimes \cdots \otimes \left(\boldsymbol{y} - \boldsymbol{f}_h(\boldsymbol{x})\right) = \boldsymbol{0}$$

Direct algebraic representation of h-valued function

$\otimes$ denotes Kronecker's tensor product. $x \in R^n, y \in R^m$

Network representation

Multi-Valued Function Approximation
MVRN, MVHBF, MVNN, MVxxx, ...

h -valued, scalar-output MVRN

Direct representation of h -valued scalar function

$$\Lambda^{(h)}(\boldsymbol{x}, y) \equiv \prod_{k=1}^{h}(y - f_k(\boldsymbol{x})) = F_1^{(h)}(\boldsymbol{x}) + yF_2^{(h)}(\boldsymbol{x}) + \cdots + y^{h-1}F_h^{(h)}(\boldsymbol{x}) + y^h = 0$$

h mappings: $y = f_j(\mathbf{x}) \quad (j = 1, 2, \cdots, h)$

The equation of h -fold hypersurface $= h$ -degree algebraic equation with respect to y.

Linearization: elementary symmetric polynomials

$$F_1^{(h)}(\boldsymbol{x}) = (-1)^h f_1(\boldsymbol{x}) f_2(\boldsymbol{x}) \cdots f_h(\boldsymbol{x}),$$

$$F_2^{(h)}(\boldsymbol{x}) = \sum_{i_1=1}^{h} \sum_{i_2=i_1+1}^{h} f_{i_1}(\boldsymbol{x}) f_{i_2}(\boldsymbol{x}),$$

$$F_h^{(h)}(\boldsymbol{x}) = -\sum_{i=1}^{h} f_i(\boldsymbol{x})$$

Standard Regularization of h -Valued Function

$$E^{(h)}[F_1^{(h)}, F_2^{(h)}, \cdots, F_h^{(h)}] = \underbrace{\sum_{i=1}^{N} \{\Lambda^{(h)}(\boldsymbol{x}_{(i)}, y_{(i)})\}^2}_{\text{Error term}} + \underbrace{\sum_{k=1}^{h} \lambda_k \left\| S_k F_k^{(h)} \right\|^2}_{\text{Smoothness term}}$$

Euler-Lagrange equation

$$\frac{\delta E^{(h)}[F_1^{(h)}, F_2^{(h)}, \cdots, F_h^{(h)}]}{\delta F_k^{(h)}} = 0 \quad \longrightarrow$$

$$\sum_{i=1}^{N} (y_{(i)})^{k-1} \Lambda^{(h)}(\boldsymbol{x}, y_{(i)}) \delta(\boldsymbol{x} - \boldsymbol{x}_{(i)}) + \lambda \hat{S} S F_k^{(h)}(\boldsymbol{x}) = 0$$

Solution of E-L equation

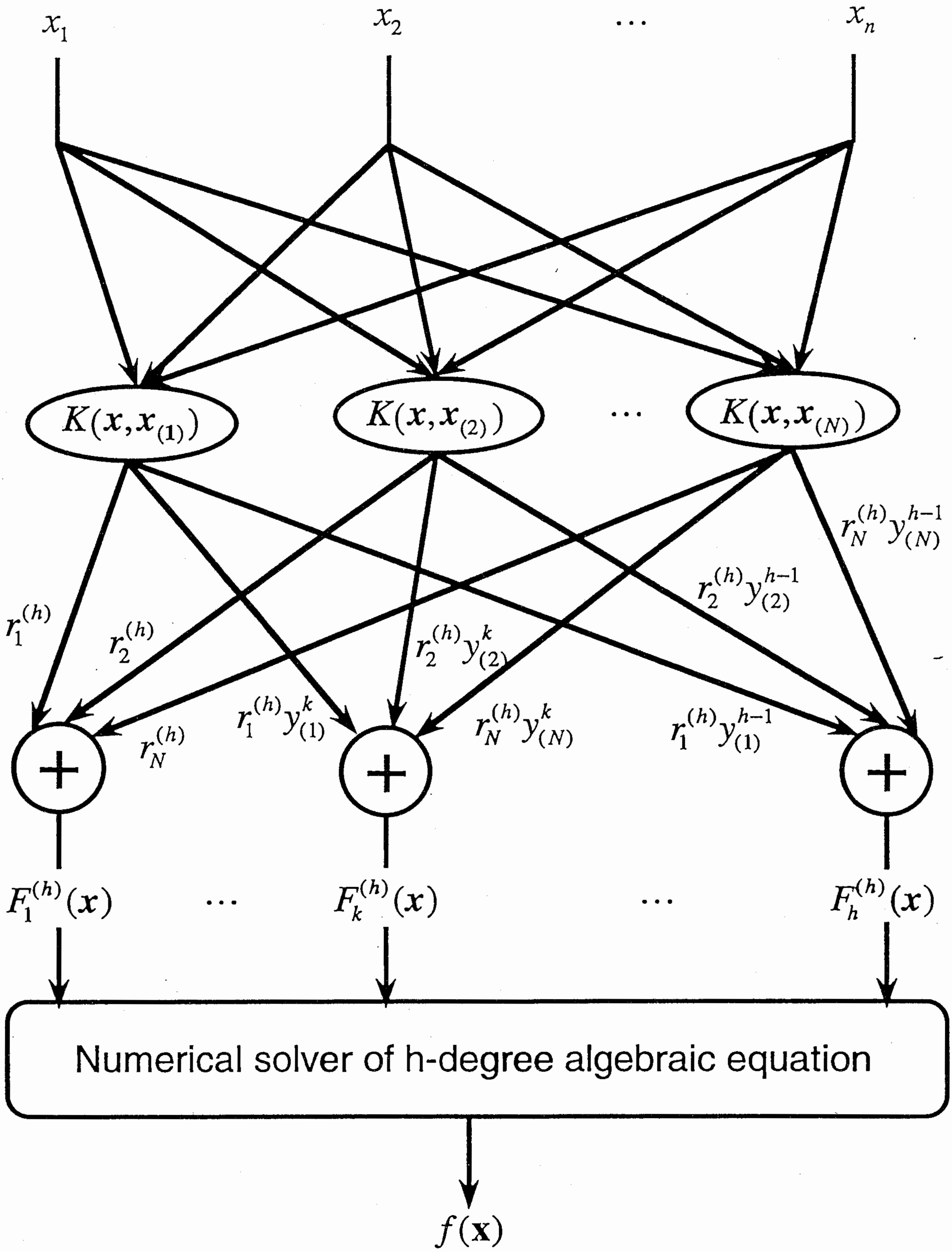
$$F_k^{(h)}(\boldsymbol{x}) = -\lambda^{-1} \sum_{i=1}^{N} (y_{(i)})^{k-1} \Lambda^{(h)}(\boldsymbol{x}_{(i)}, y_{(i)}) K(\boldsymbol{x}, \boldsymbol{x}_{(i)})$$

Multi-Valued Regularization Network

$$F_k^{(h)}(\boldsymbol{x}) = \sum_{i=1}^{N} r_i^{(h)} (y_{(i)})^{k-1} K(\boldsymbol{x}, \boldsymbol{x}_{(i)})$$

Weight parameters:

$$r_i^{(h)} = -\lambda^{-1} \Lambda^{(h)}(\boldsymbol{x}_{(i)}, y_{(i)})$$



x_1
x_2
$\cdots$
x_n
$K(\boldsymbol{x},\boldsymbol{x}_{(1)})$
$K(\boldsymbol{x},\boldsymbol{x}_{(2)})$
$\cdots$
$K(\boldsymbol{x},\boldsymbol{x}_{(N)})$
$r_N^{(h)} y_{(N)}^{h-1}$
$r_2^{(h)} y_{(2)}^{h-1}$
$r_1^{(h)}$
$r_2^{(h)}$
$r_2^{(h)} y_{(2)}^{k}$
$r_1^{(h)} y_{(1)}^{k}$
$r_N^{(h)} y_{(N)}^{k}$
$r_1^{(h)} y_{(1)}^{h-1}$
$r_N^{(h)}$
$F_1^{(h)}(\boldsymbol{x})$
$\cdots$
$F_k^{(h)}(\boldsymbol{x})$
$\cdots$
$F_h^{(h)}(\boldsymbol{x})$
Numerical solver of h-degree algebraic equation
$f(\mathbf{x})$

Learning algorithms of Multi-Valued Regularization Networks

N-dimensional linear system:

$$\boldsymbol{K}^{(h)}\boldsymbol{r}^{(h)} + z^{(h)} = \boldsymbol{0}$$

$$\boldsymbol{K}^{(h)} = (K_{ij}^{(h)}) = \left(\left\{ \sum_{k=1}^{h} (y_{(i)} y_{(j)})^{k-1} \right\} K(\boldsymbol{x}_{(i)}, \boldsymbol{x}_{(j)}) + \lambda \delta_{ij} \right),$$

$$\boldsymbol{r}^{(h)} = (r_1^{(h)}, r_2^{(h)}, \cdots, r_N^{(h)})^T$$

$$z^{(h)} = (\{y_{(1)}\}^h, \{y_{(2)}\}^h, \cdots, \{y_{(N)}\}^h)^T$$

Dimension of the linear system is invariant to multiplicity h.

Multi-Valued HyperBF Network (MVHBF)

(Scalar-valued MVHBF)

Basis functions: multivariate Gaussians of arbitrary covariances

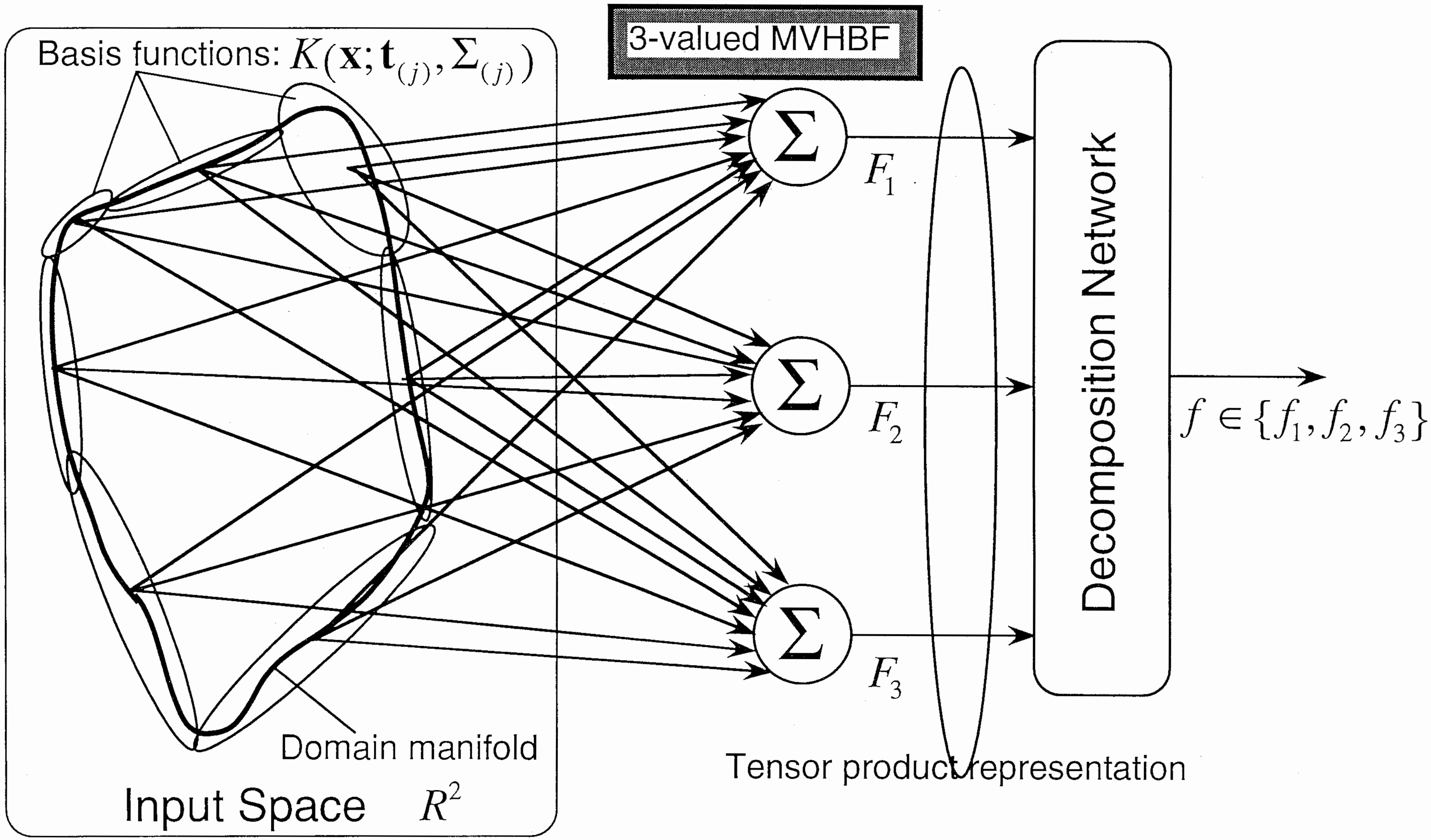
$$K_{(j)}(\mathbf{x}, \mathbf{t}_{(j)}, \Sigma_{(j)}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma_{(j)}}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{t}_{(j)})^T \Sigma_{(j)}^{-1} (\mathbf{x} - \mathbf{t}_{(j)}) \right\}$$

$$(j = 1, 2, \cdots, M)$$

Intermediate representation: $F_k^{(h)}(\mathbf{x}) = \sum_{j=1}^{M} \tilde{r}_{k,j} K_{(j)}(\mathbf{x}, \mathbf{t}_{(j)}, \Sigma_{(j)}) \quad (k = 1, 2, \cdots, h)$

Energy function for minimization: $E^{(h)}[F_1^{(h)}, F_2^{(h)}, \cdots, F_h^{(h)}] = \sum_{i=1}^{N} \{\Lambda^{(h)}(x_{(i)}, y_{(i)})\}^2$

MVHBF(Multi-Valued Hyper-Basis Function Network):
A function approximation network that combines unsupervised and supervised learning for multi-valued function approximation.



Learning of linear weights

Linear system for learning linear weights: $\tilde{\mathbf{K}}^{(h)}\tilde{\mathbf{r}}^{(h)} + \tilde{\mathbf{z}}^{(h)} = 0$ (hM)-dim.

$$\tilde{\mathbf{K}}^{(h)} = \begin{bmatrix} \mathbf{D}_1^T\mathbf{D}_1 & \mathbf{D}_1^T\mathbf{D}_2 & \cdots & \mathbf{D}_1^T\mathbf{D}_h \\ \mathbf{D}_2^T\mathbf{D}_1 & \mathbf{D}_2^T\mathbf{D}_2 & \cdots & \mathbf{D}_2^T\mathbf{D}_h \\ \cdots & \cdots & \ddots & \vdots \\ \mathbf{D}_h^T\mathbf{D}_1 & \mathbf{D}_h^T\mathbf{D}_2 & \cdots & \mathbf{D}_h^T\mathbf{D}_h \end{bmatrix},$$

$$\tilde{\mathbf{r}}^{(h)} = (\tilde{r}_{1,1}, \cdots, \tilde{r}_{1,M}, \tilde{r}_{2,1}, \cdots, \tilde{r}_{2,M}, \tilde{r}_{h,1}, \cdots, \tilde{r}_{h,M})^T,$$

$$\tilde{\mathbf{z}}^{(h)} = ((\mathbf{z}^{(h)})^T\mathbf{D}_1, (\mathbf{z}^{(h)})^T\mathbf{D}_2, \cdots, (\mathbf{z}^{(h)})^T\mathbf{D}_h)^T,$$

$$(\mathbf{D}_k)_{ij} = (y_{(i)})^{k-1} K(\mathbf{x}_{(i)}, \mathbf{t}_{(j)}),$$

$$\mathbf{z}^{(h)} = (\{y_{(1)}\}^h, \{y_{(2)}\}^h, \cdots, \{y_{(N)}\}^h)^T$$

EM algorithm [Dempster 1977] for learning of centers and covariances of Gaussian basis functions

EM algorithm for Gaussian Mixture Density Estimation [Ghahramani & Jordan 1994]

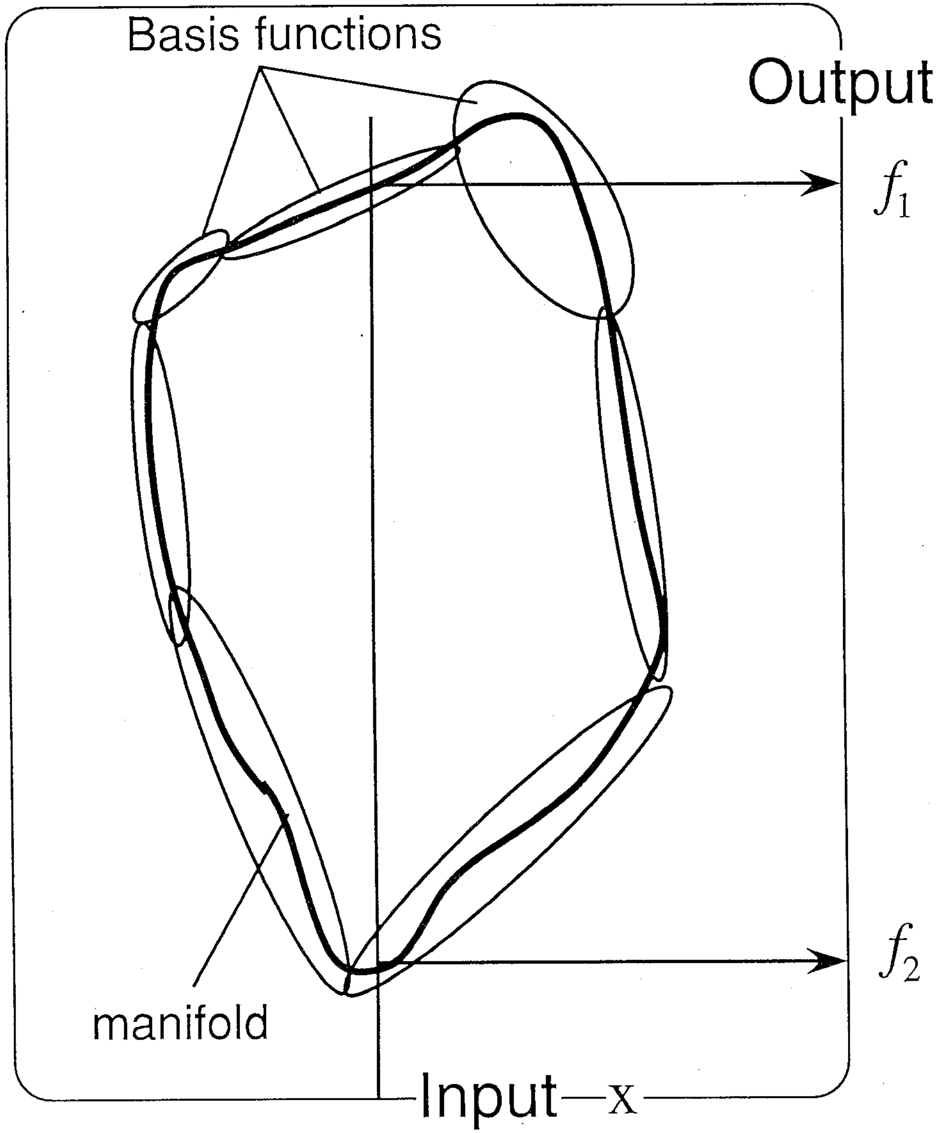
E-step (Expectation): Computation of contribution of the j-th basis function to i-th data

$$g_{ij} = \frac{1}{\sqrt{\det \Sigma_{(j)}}} \exp\left\{ -\frac{1}{2} (\mathbf{x}_{(i)} - \mathbf{t}_{(j)})^T \Sigma_{(j)}^{-1} (\mathbf{x}_{(i)} - \mathbf{t}_{(j)}) \right\} \qquad h_{ij} = \frac{g_{ij}}{\sum_{j=1}^{M} g_{ij}}$$

M-step (Maximization): Re-estimation of center and covariance

$$\mathbf{t}'_{(j)} = \frac{\sum_{i=1}^{N} h_{ij} x_{(i)}}{\sum_{i=1}^{N} h_{ij}} \qquad \Sigma'_{(j)} = \frac{\sum_{i=1}^{N} h_{ij} (\mathbf{x}_{(i)} - \mathbf{t}'_{(j)})(\mathbf{x}_{(i)} - \mathbf{t}'_{(j)})^T}{\sum_{i=1}^{N} h_{ij}}$$

Density estimation in combined input-output space for function approximation [Ghahramani & Jordan 1994]



Input-Output Space R^2

The approximation power is limited because it is basically a piecewise linear approximation.

It is difficult to evaluate the input-output mappings for multi-valued functions.

(1) It is computationally burdensome to compute multi-fold integrals for regression in high dimensions.

(2) It is computationally burdensome to find local maxima for maximum likelihood estimation.

Not suitable for real-time implementation.

Comparison between conventional unsupervised learning techniques (e.g. density estimation) and MVHBF in regard of function approximation

	unsupervised learning (density estimation)	multi-valued func. approx. (MVHBF)
learning methods	nonlinear regression (input+output)	linear regression(output) + nonlinear regression(input)
evaluation of mapping	multi-fold integral for regression / finding local maxima for MLE	one-shot, feedforward computation (+numerical methods)
approximation power	low (piecewise linear)	high (smoothness)
suitable implementation	computer program / recurrent network	feedforward network + numerical methods
realizable mapping	general relation	*h* -valued mapping

Applications of MVRN and MVHBF

1. Computer vision / Sensor fusion : locating multiple objects from multiple sensors.

   An application of 2-valued, 2-vector-output MVHBF.

2. Inverse kinematics: Learning of inverse kinematic mapping from hand posture to link parameters for a 3-link planar manipulator.

   It is possible to represent singularity and change of multiplicity h of inverse mapping.

Learning in Computer Vision: Levels of Utility of the Learning Methods

Levels	Models	Solution	Examples
0	available	analytical solution	unnecessary
1	available	numerial methods	unnecessary
2	available	*learning*	artificial examples
3	unavailable	*learning*	controlled examples
4	unavailable	*learning*	uncontrolled real examples

Levels 0, 1 --- conventional approaches of computer vision

Level 2 --- neural programming

Level 3 --- recent results in computer vision
[Murase & Nayer] [Poggio et.al.] [Weng et.al.]

Level 4 --- unsupervised learning, MVHBF

Complementary nature of the two approaches

Computational Vision (CV)

Physical (photometric, geometric) constraint

Image representation /Feature detection

Uniqueness, ambiguity

Unification of constraints

Problems

Ill-posed problem

Noise/Quantization/Uncertainty

Ill-defined problems (non-rigid, unknown structure,etc)

Real-time applications

Segmentation

Non-existence of analytical solutions

Computational Learning (CL)

(Learning from examples)

Regularization theory

Function approximation
Neural network

Statistical estimation methods

Statistical pattern recognition (clustering techniques)

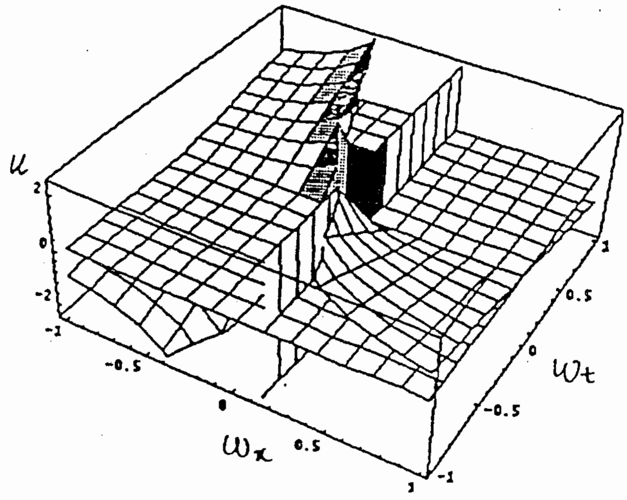
Optimization techniques

Problems

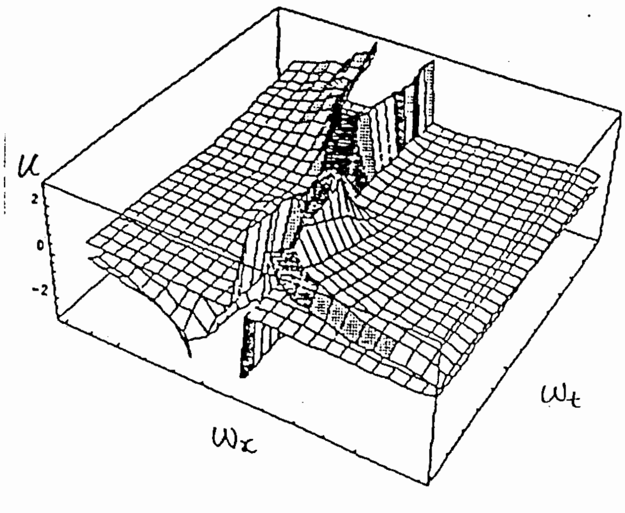
Feature detection (scale, invariance)

Integration of learning modules

Ambiguous relations in learning



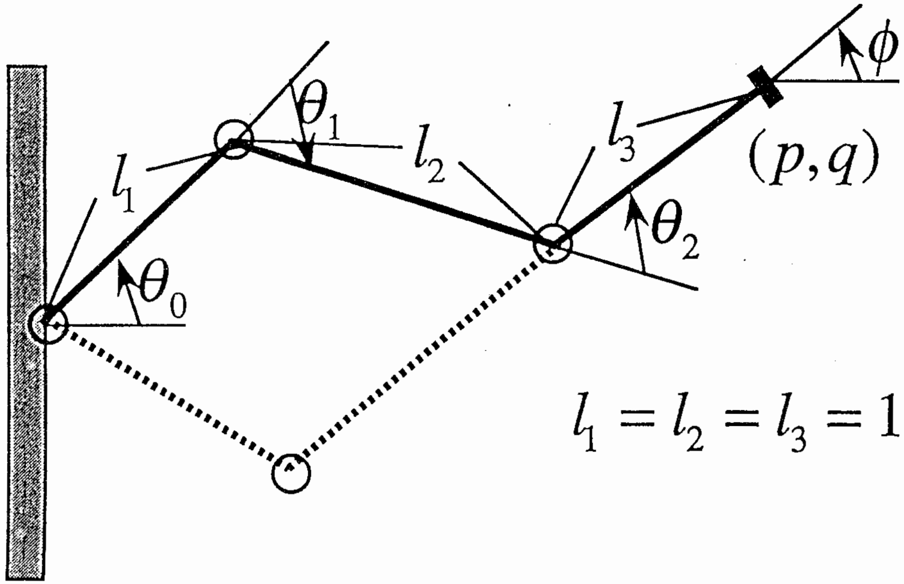
(a)



(b)

Figure 4: The result of learning a module of motion transparency
(a) A section of input-output mapping using analytical solution, (b) A section of input-output mapping using two-valued MVRBF

Inverse kinematics of 3-link planar manipulator



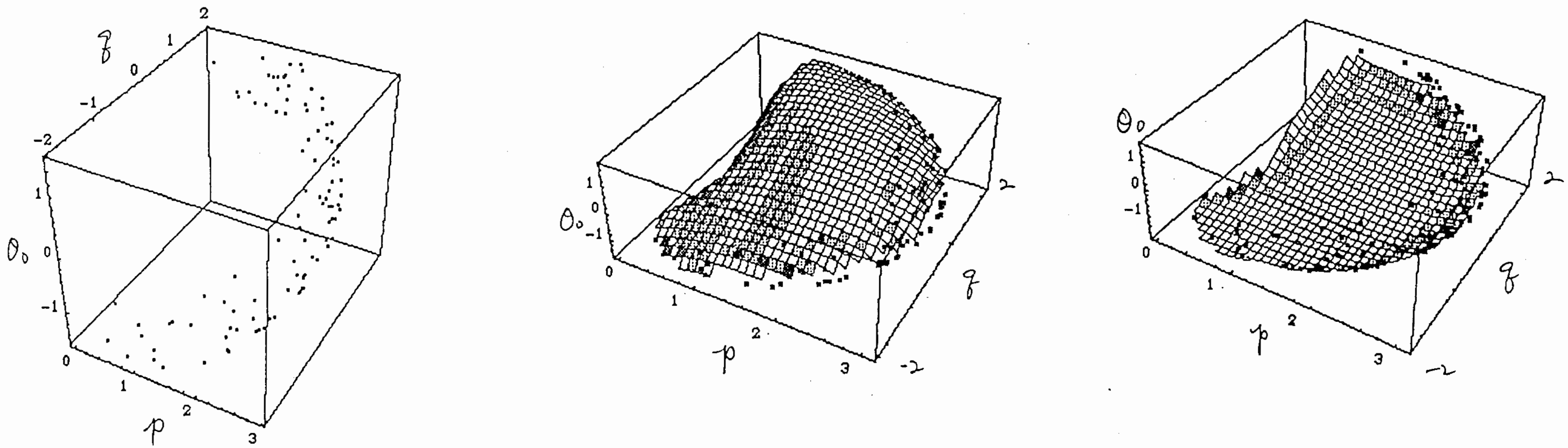
$$l_1 = l_2 = l_3 = 1$$

Forward mapping:

$$(\theta_0, \theta_1, \theta_2) \mapsto (p, q, \phi)$$

Inverse mapping:

$$(p, q, \phi) \mapsto (\theta_0, \theta_1, \theta_2)$$



例題：ガウス関数状の空間特性を持つ５個の光受容器の出力から２輝点の２次元位置を推定する

Example: Two-dimensional localization of two bright points from Gaussian photo receptors.

輝点：　２個，等輝度　Two bright points, equal brightness.
光受容器：　５個　Five photo receptors.
光受容器の点広がり関数：　等方ガウス関数 (標準偏差=2.0)
PSF of photo receptors is a uniform Gaussian with s.d.=2.0.

この逆問題には，解析解は存在しない！
There is no analytical solution for this inverse problem!

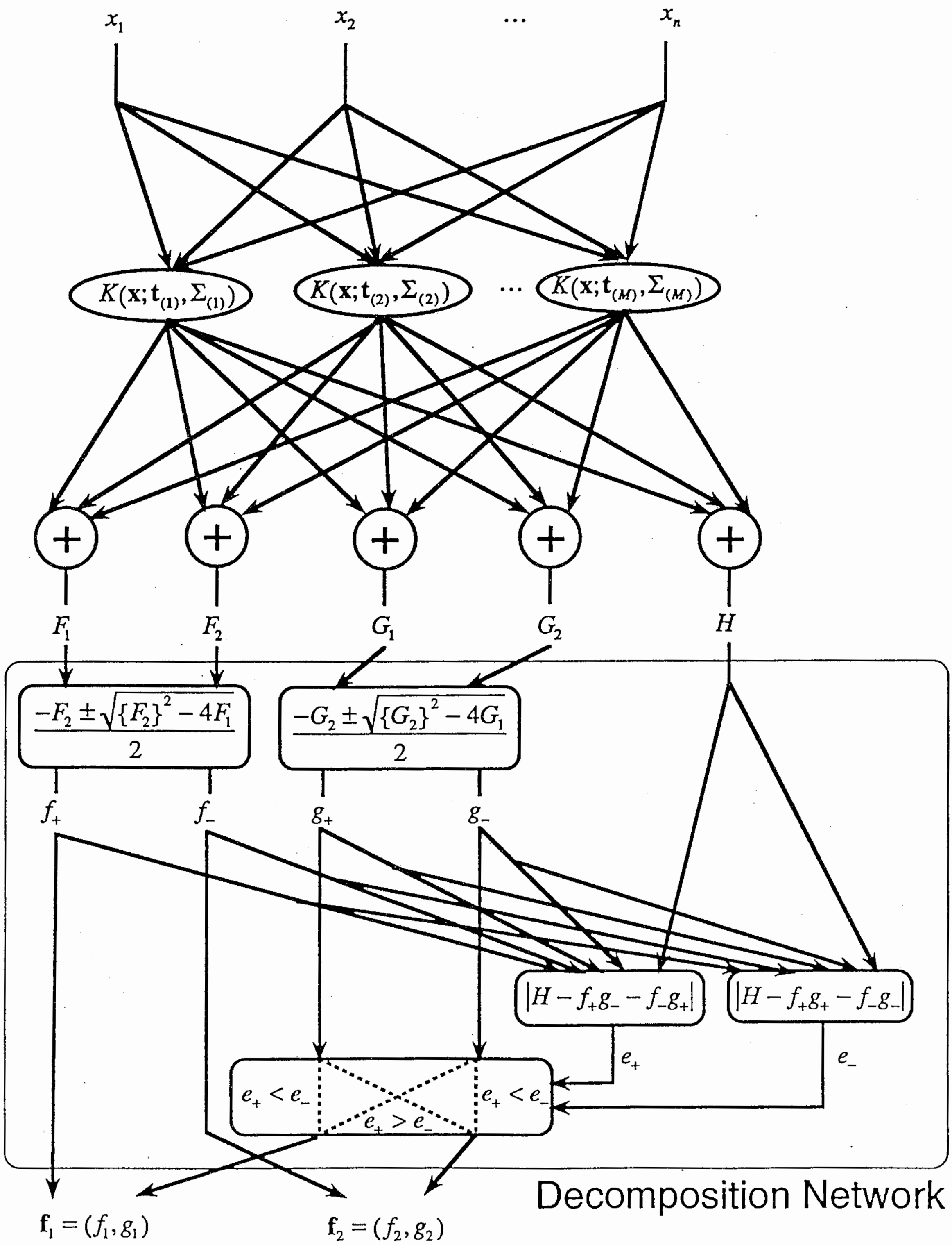
学習のための教師データ：　x ∈[0.0,3.0]，　y ∈[-1.0,1.0]
に一様ランダムに10,000個生成.
The number of training data was 10,000.
ネットワークの基底関数：一般共分散のガウス関数２４３個
Basis functions: 243 Gaussian functions with arbitrary covariances.
学習アルゴリズム：　線形回帰法＋ＥＭアルゴリズム
Learning algorithm: Linear regression + EM algorithm.

線形回帰法でネットワークの重みを学習
Learning of weight parameters with linear regression.
ＥＭアルゴリズムで基底関数の位置と形状を学習(初期位置は一様ランダム)
Learning of centers and covariances of basis functions with EM.

各光受容器には，距離に応じて異なる重みを付けて加算された２輝点の輝度情報が観測される.

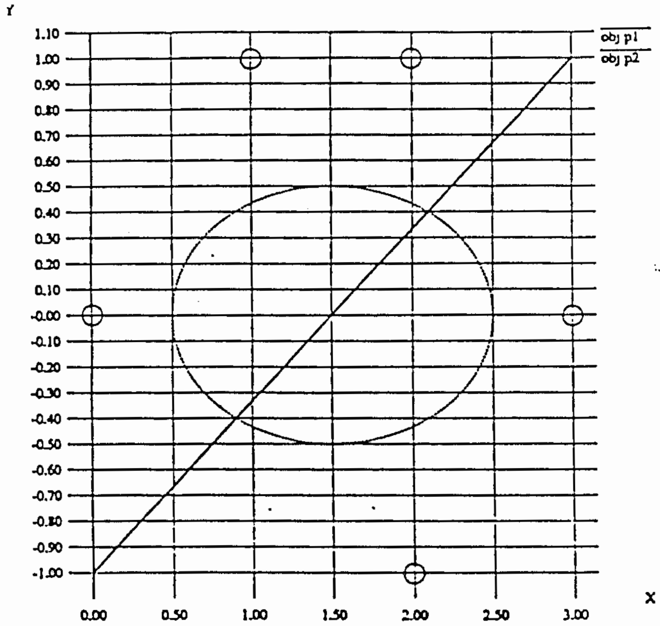
これは一種のセンサフュージョン問題である.
This is a sensor fusion problem.

評価のためのデータ生成例：Examples for evaluation.
輝点１：　(0.0,-1.0)から(3.0,1.0)まで等速度直線運動
Obj #1: Translational constant motion.
輝点２：　円軌道を半時計廻りに一周　Obj #2: Circular orbital motion.
結果として高精度の学習結果が得られた. Accurate results are obtained.
問題点：基底関数の初期値の与え方　Problem: Initial values of basis functions.

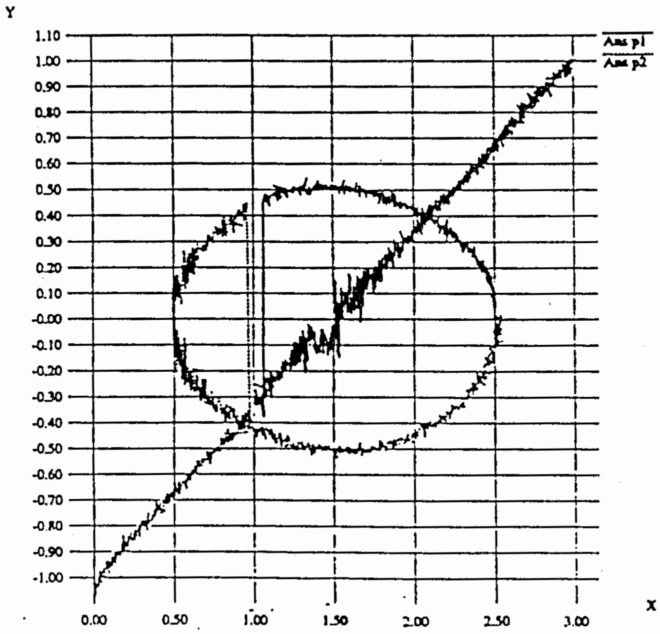


2-Valued 2-Vector-output MVHBF

[redacted]

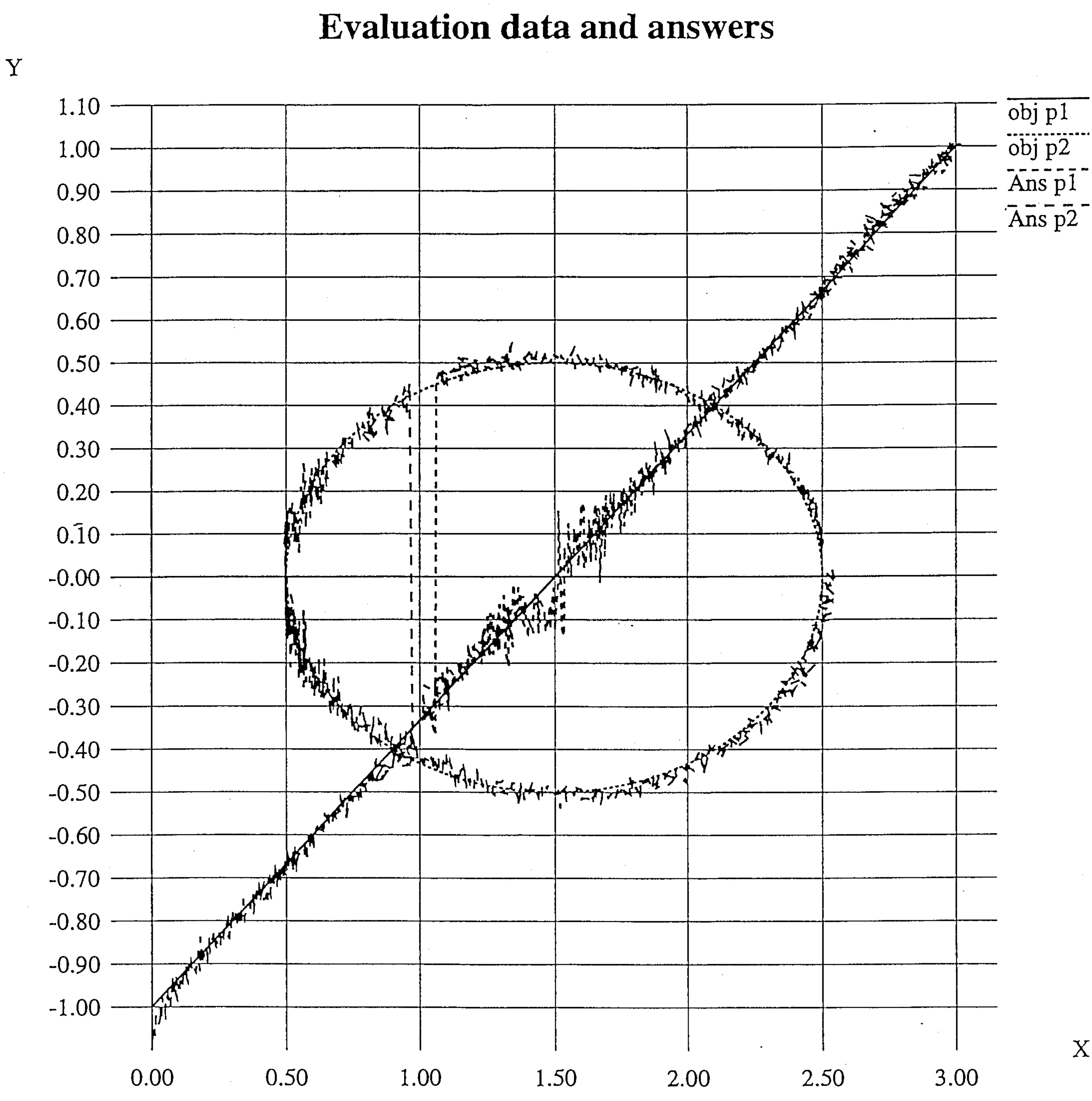


(a)



(b)

Figure 6: Localization problem for two bright points from five photo receptors (a) Arrangements of photo receptors and orbits of two bright points for evaluation, (b) Evaluation results of the learning network.


Evaluation data and answers
Y
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
-0.40
-0.50
-0.60
-0.70
-0.80
-0.90
-1.00
0.00
0.50
1.00
1.50
2.00
2.50
3.00
X
obj p1
obj p2
Ans p1
Ans p2

Conclusions

1. MVHBF that learns multi-valued, vector-output mappings was proposed.

2. applications to inverse problems including computer vision, inverse kinematics, e.t.c. are discussed.

3. Simulation results are presented for multiple objects localization and learning of invese kinematics problem.

Future directions

1. Improvement of learning methods for more robust and accutate approximation.

2. Automatic determination of the global multiplicity h of the mapping.

3. Theoretical foundations and algorithms of the learning of point-wise multiplicity changes.