# Correspondence-free Associative Learning

Erik Jonsson
Computer Vision Laboratory
Linköping University
erijo@isy.liu.se

Michael Felsberg
Computer Vision Laboratory
Linköping University
mfe@isy.liu.se

## Abstract

*We study the problem of learning a non-parametric mapping between two continuous spaces without having access to input-output pairs for training, but rather to groups of input-output pairs, where the correspondence structure within each group is unknown and where outliers may be present. This problem is solved by transforming each space using the channel representation, and finding a linear mapping on the transformed domain. The asymptotical behavior of the method for a large number of training samples is found to be very related to the case of known correspondences. The results are evaluated on simulated data.*

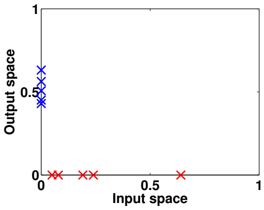

**Figure 1. One training example of a 1D problem: 5 points in each space, with unknown correspondence structure**

## 1 Introduction

Traditional supervised learning approaches [10] have mostly aimed at solving a *classification* or *regression* problem. In both cases, the starting point is almost always a number of corresponding examples of input and output data. In this paper, we consider the harder problem of learning relations between two continuous spaces without actually having access to matching training samples, but only to internally unordered sets of input/output examples (Fig. 1). The goal is to learn the transformation by looking at several such examples. We call this problem setting *correspondence-free learning*.

The most common related problem is finding a parameterized mapping (e.g. a homography) between two spaces, given only a single set of unordered points. For this problem, robust methods like RANSAC [6], [9], have been highly successful. Other related approaches are [1], [3], all looking for parameterized mappings. In [4], a minimum-work transformation between two point sets is sought instead of a parameterized mapping. All these approaches have in common that they start out from just a single set of points in each domain. As a result, the types of transformations that can be obtained are very limited. In this paper, we seek an arbitrary non-parametric transformation, but assume having access to a large number of sets of unordered points as training data. Despite an extensive literature search, the authors are not aware of any other work trying to solve this problem.

The correspondence-free learning problem is expected to be encountered frequently by self-organizing cognitive systems. A discrete example is language acquisition, where a child hears a lot of different words while observing a lot of different events in the world, having no idea which word should map to which particular aspect of its experiences. A more continuous example is in learning the *perception-action map*. A cognitive system is confronted with a large number of percepts observed simultaneously, that transform as a result of an action. Given a percept list $\mathbf{p}_t$ at time $t$ and another list $\mathbf{p}_{t+1}$ at time $t+1$ as a result of some action $a$, it is desired to learn the mapping $(\mathbf{p}_t, a) \rightarrow \mathbf{p}_{t+1}$ without necessarily knowing a priori which percepts from the two time instances that correspond. As a final example, consider the temporal credit assignment problem in reinforcement learning [12], which is the problem of attributing a reward to some previous action. Assume that an agent generates actions that produce a randomly delayed reward. By taking the set of all actions performed and all rewards given in the previous $T$ time steps, we know that some of these actions correspond to some reward, but the exact correspondence structure is unknown. Using a number of such sets as training samples, the action-reward mapping could be learned.

In this paper, we show that it is possible to perform a (virtually) non-parametric estimation of such a mapping by formulating the problem as a linear least-squares problem. The method is based on the *channel representation* [8], which is a local information representation inspired from biology. The method can also be viewed in a statistical sense, and is related to kernel density estimation [2]. The problem is solved by finding a linear transformation on the channel vectors representing the inputs and outputs. This approach was introduced in [8], but regarding only training samples consisting of single values in each domain.

In Sect. 2, the problem is defined formally, the channel representation is reviewed, and the proposed method is presented. Some theoretical properties of the method are examined in Sect. 3 and finally, the method is evaluated in a number of experiments in Sect. 4.

## 2 Correspondence-free Associative Learning

### 2.1 Notation and Problem Formulation

Consider an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. We want to find a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ given a set of training samples $\{S_t \mid t = 1 \dots T\}$. Each sample $S_t$ is a tuple $(X_t, Y_t)$, where $X_t = \{x_{t,i} \mid i = 1 \dots m\} \subset \mathcal{X}$ and $Y_t = \{y_{t,i} \mid i = 1 \dots n\} \subset \mathcal{Y}$. Furthermore, $X_t$ and $Y_t$ are divided into *inliers* and *outliers*. For each inlier $x_{t,i}$, there is an inlier $y_{t,j}$ such that $y_{t,j} = f(x_{t,i})$. The outliers are random and independent. The sets $X_t$ and $Y_t$ are unordered, such that we have no information about which $x_{t,i}$'s and $y_{t,j}$'s that correspond or which are outliers. One example of a single training sample $S_t$ for a 1D problem is shown in Fig. 1, with $X_t$ on the x-axis and $Y_t$ on the y-axis.

It will be convenient for the later discussion to assume that for all $t$, $|X_t| = m$ and $|Y_t| = n$, and that each $S_t$ contains $n_c$ corresponding $(x, y)$-pairs and $o_x, o_y$ outliers in $\mathcal{X}$ and $\mathcal{Y}$ respectively, such that $m = n_c + o_x$ and $n = n_c + o_y$. The proposed method will work even if each training sample contains a different ratio of inliers and outliers, but the theoretical analysis will be clearer this way.

The goal is now to learn the function $f$. It is non-trivial to define an objective function to minimize for this problem. Instead, we adopt a bottom-up approach: first designing a promising method, and later analyzing its theoretical properties.

### 2.2 Channel Representations

This section reviews the Channel Representation, upon which the solution method is based. In this framework, a scalar value $x$ is represented by a *channel vector*

$$\mathbf{a} = \mathrm{enc}(x) = [K(x - \xi_1), \dots, K(x - \xi_N)] \ , \qquad (1)$$

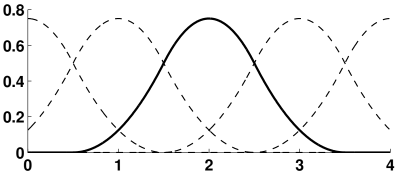

**Figure 2. Channel basis functions (second order B-splines) located uniformly on the real axis.**

where $K$ is some kernel function called the *channel basis function*, and $\xi_n$ are the *channel centers*. The basis functions are usually smooth, localized functions with compact support, scaled such that the basis functions for different channels overlap (Fig. 2). Given a channel vector $\mathbf{a}$, it is possible to reconstruct the value $x$ by a procedure called *decoding* [7], [5]. This decoding procedure should be exact, such that $\mathrm{dec}(\mathrm{enc}(x)) = x$.

By encoding several values $x_i$ and summing the corresponding channel vectors elementwise, we get a *soft histogram* of the $x_i$ values, which is like a histogram with smooth and overlapping bins. From this representation, peaks can be detected with sub-bin accuracy. In this work, $2^{\text{nd}}$ order B-spline ($B_2$) kernels and the corresponding decoding from [5] is used. This decoding essentially views the channel coefficients as a sampled continuous function and finds the maximum using $B_2$-spline interpolation, which can be done analytically in a local context since the interpolant is piecewise quadratic. An important property of this decoding is that it is invariant to a constant scaling of the channel vector, i.e. the channel vector is a *homogeneous* representation.

### 2.3 Solution Method

To solve the correspondence-free problem, we encode all inputs and outputs in each $X_t$ and $Y_t$ together, and seek a direct linear transformation in the channel domain. From each training sample $S_t$, we define

$$\bar{\mathbf{a}}_t = \sum_{i=1}^{m} \mathbf{a}_{t,i} = \sum_{i=1}^{m} \mathrm{enc}(x_{t,i}) \qquad (2)$$

$$\bar{\mathbf{u}}_t = \sum_{i=1}^{n} \mathbf{u}_{t,i} = \sum_{i=1}^{n} \mathrm{enc}(y_{t,i}) \ . \qquad (3)$$

We now want to solve

$$\min_{\mathbf{C}} \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{C}\bar{\mathbf{a}}_t - \bar{\mathbf{u}}_t\|^2 \qquad (4)$$

This can be solved[1] using standard linear least-squares methods, e.g. by forming the normal equations $\mathbf{CG} = \mathbf{H}$, with

$$\mathbf{G} = \frac{1}{T}\sum_{t=1}^{T} \bar{\mathbf{a}}_t \bar{\mathbf{a}}_t{}^{\mathrm{T}} \ , \quad \mathbf{H} = \frac{1}{T}\sum_{t=1}^{T} \bar{\mathbf{u}}_t \bar{\mathbf{a}}_t{}^{\mathrm{T}} \ . \qquad (5)$$

Ideally, we would like to have the same $\mathbf{C}$ as if the correspondences would have been known. To motivate the method, assume that there exists a perfect $\mathbf{C}$, implementing the sought mapping $f$ exactly, such that $\mathrm{enc}(y) = \mathbf{C}\,\mathrm{enc}(x)$ if $y = f(x)$. If there are no outliers in $X_t$ and $Y_t$, we would also have $\bar{\mathbf{u}}_t = \mathbf{C}\bar{\mathbf{a}}_t$ because of the linearity, which makes the method intuitively appealing. However, when no such exact $\mathbf{C}$ exists or when there are outliers, it is not as obvious how the solution to this problem relates to the solution to the ordered problem.

# 3 Asymptotical properties

In this section, we examine the asymptotical properties of the method as the number of samples drawn goes to infinity. Will the chosen $\mathbf{C}$ approach that of the corresponding ordered, outlier-free problem, or will it be biased in some way?

Each training sample $S_t$ is now viewed as a realization of a random process, where $\mathbf{a}_{t,i}, \mathbf{u}_{t,j}$ are realizations of the random variables $\mathbf{a}_i, \mathbf{u}_j$. We assume that the inliers and outliers follow the same distribution and are drawn independently, such that $\mathbf{a}_i$ and $\mathbf{a}_j$, $i \neq j$, are i.i.d, as are $\mathbf{u}_i$ and $\mathbf{u}_j$, $i \neq j$. In a similar way, we can view $\bar{\mathbf{a}}_t$ and $\bar{\mathbf{u}}_t$ as different realizations of the random vectors $\bar{\mathbf{a}}$ and $\bar{\mathbf{u}}$, where

$$\bar{\mathbf{a}} = \sum_{i=1}^{m} \mathbf{a}_i, \quad \bar{\mathbf{u}} = \sum_{i=1}^{n} \mathbf{u}_i \ . \qquad (6)$$

To summarize, a "bar" always means "sum over $i$", and dropping the index $t$ means "view as a stochastic variable".

## 3.1 The Ideal Ordered Problem

We would like to compare the behavior of the method to a hypothetical ideal setting. For the sake of presentation, assume that the first $n_c$ $x$'s and $y$'s in each $S_t$ are mutually corresponding inliers, such that $y_{t,i} = f(x_{t,i})$ for $1 \leq i \leq n_c$, and that the rest are outliers. Of course, the solution method is not allowed to take advantage of this information, since the correspondence structure is supposed to be unknown. However, it makes it easier to compare the method to the case where the correspondences are known. In this ideal case, we could minimize

$$\min_{\mathbf{C}} \frac{1}{Tn_c} \sum_{t=1}^{T} \sum_{i=1}^{n_c} \|\mathbf{C}\mathbf{a}_{t,i} - \mathbf{u}_{t,i}\|^2 \ . \qquad (7)$$

As $T \to \infty$, this expression tends towards

$$\min_{\mathbf{C}} \mathrm{E_c}[\|\mathbf{C}\mathbf{a}_i - \mathbf{u}_i\|^2] \ , \qquad (8)$$

where $\mathrm{E_c}$ means the expectation over the inlier set, i.e. for $i \leq n_c$. The normal equations of this problem are $\mathbf{C}\mathbf{G}_c = \mathbf{H}_c$, with

$$\mathbf{G}_{\mathrm{c}} = \mathrm{E_c}[\mathbf{a}_i \mathbf{a}_i{}^{\mathrm{T}}], \quad \mathbf{H}_{\mathrm{c}} = \mathrm{E_c}[\mathbf{u}_i \mathbf{a}_i{}^{\mathrm{T}}] \qquad (9)$$

analogous to (5).

## 3.2 The Correspondence-free Problem

We now return to the correspondence-free problem. The main result of this section can be summarized in the following theorem:

**Theorem 1** *Let* $\mathbf{a}_\mu = \mathrm{E}[\mathbf{a}_i], \mathbf{u}_\mu = \mathrm{E}[\mathbf{u}_i]$ *(which is independent of* $i$*). Then, as* $T \to \infty$*, the unordered problem from* (4) *is equivalent to*

$$\min_{\mathbf{C}} \mathrm{E_c}\left[\|\mathbf{C}\mathbf{a}_i - \mathbf{u}_i\|^2\right] + (m-1)\|\mathbf{C}\mathbf{a}_\mu - k\mathbf{u}_\mu\|^2 \ , \quad (10)$$

*where* $k = (mn - n_c)/(mn_c - n_c))$.

Before jumping to the proof (given in Sect. 3.3), we analyze the consequences of this theorem. First assume that $o_y = 0$. Then $n = n_c$, $k = 1$, and the unordered problem becomes equivalent to the ordered problem, but with the additional term $\|\mathbf{C}\mathbf{a}_\mu - \mathbf{u}_\mu\|^2$ included in the minimization. The larger $m$ is, the more weight this term gets. As $m \to \infty$, the problem approaches a constrained minimization problem, where the first term is minimized subject to the last term being exactly zero. But $\mathbf{C}\mathbf{a}_\mu = \mathbf{u}_\mu$ is a very natural constraint, just saying that the mean output of the method should equal the true mean of the channel encoded output training samples. Furthermore, this constraint only uses up one degree of freedom of each row of $\mathbf{C}$ and is not expected to degrade the performance much.

It is also interesting to note that the number of x-outliers $o_x$ and the number of correspondences $n_c$ enters (10) only through $m$, so increasing $o_x$ has the same effect as increasing $n_c$. However, this only holds for the asymptotical solution - the speed of convergence may be degraded. Also, keep in mind that we assumed the outliers to follow the same distribution as the inliers.

[1] In [8], [11], a positivity constraint on $\mathbf{C}$ was used as a regularization, and $\mathbf{C}$ was found using an iterative solution method. In that case, the matrices $\mathbf{G}$ and $\mathbf{H}$ appear in the iterative update. The learning problem is still completely defined by $\mathbf{G}$ and $\mathbf{H}$, so most of the discussion here applies also in that case.

Unfortunately, when $o_y > 0$ the story is different. Then $k \approx n/n_c > 1$, and suddenly the second term of (10) forces $\mathbf{Ca}_\mu$ to be larger than is motivated by the corresponding data only. There is an unbalance between the two terms of (10), which leads to undesired results.

### 3.3 Proof of Theorem 1

As $T \to \infty$, $\mathbf{G}$ and $\mathbf{H}$ from (5) tend towards

$$\mathbf{G} = \mathrm{E}[\bar{\mathbf{a}}\bar{\mathbf{a}}^{\mathrm{T}}], \quad \mathbf{H} = \mathrm{E}[\bar{\mathbf{u}}\bar{\mathbf{a}}^{\mathrm{T}}] \ . \tag{11}$$

By combining (6) and (11), expanding the products, and swapping the sum and expectation, we get

$$\mathbf{G} = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathrm{E}[\mathbf{a}_i{\mathbf{a}_j}^{\mathrm{T}}], \quad \mathbf{H} = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathrm{E}[\mathbf{u}_j{\mathbf{a}_i}^{\mathrm{T}}] \ . \tag{12}$$

The expectation $\mathrm{E}[\mathbf{a}_i\mathbf{a}_j^T]$ is independent of the actual indicies $i, j$ - what matters is only if $i = j$ or not (note that $\mathrm{E_c}[\mathbf{a}_i{\mathbf{a}_i}^{\mathrm{T}}] = \mathrm{E}[\mathbf{a}_i{\mathbf{a}_i}^{\mathrm{T}}]$, since the inliers and outliers are assumed to follow the same distribution). Thus, we can split the sum into two parts:

$$\mathbf{G} = m\mathrm{E_c}[\mathbf{a}_i{\mathbf{a}_i}^{\mathrm{T}}] + (m^2 - m)\mathrm{E}_{i\neq j}[\mathbf{a}_i{\mathbf{a}_j}^{\mathrm{T}}] \ . \tag{13}$$

$\mathbf{H}$ can be treated in a similar way. $\mathrm{E}[\mathbf{u}_j{\mathbf{a}_i}^{\mathrm{T}}]$ is only dependent on whether $\mathbf{a}_i$ and $\mathbf{u}_j$ correspond or not, and since each training sample contains $n_c$ correspondences, we can split the sum into

$$\mathbf{H} = n_c\mathrm{E_c}[\mathbf{u}_i{\mathbf{a}_i}^{\mathrm{T}}] + (mn - n_c)\mathrm{E_{nc}}[\mathbf{u}_j{\mathbf{a}_i}^{\mathrm{T}}] \ , \tag{14}$$

where $\mathrm{E_c}$ takes the expectation over corresponding inliers $\mathbf{a}_i, \mathbf{u}_i$, and $\mathrm{E_{nc}}$ takes the expectation over non-corresponding pairs - inliers as well as outliers.

Note that the first expectation terms in (13) and (14) are exactly $\mathbf{G}_\mathrm{c}$ and $\mathbf{H}_\mathrm{c}$ from (9) of the ordered problem. Furthermore, the two factors in the last terms are independent, since non-corresponding $(x, y)$-pairs are assumed to be drawn independently. We can exchange the order of the expectation and the product, which gives

$$\mathbf{G} = m\mathbf{G}_\mathrm{c} + (m^2 - m)\mathbf{a}_\mu{\mathbf{a}_\mu}^{\mathrm{T}} \tag{15}$$

$$\mathbf{H} = n_c\mathbf{H}_\mathrm{c} + (mn - n_c)\mathbf{u}_\mu{\mathbf{a}_\mu}^{\mathrm{T}} \ . \tag{16}$$

$\mathbf{C}$ only needs to be determined up to a multiplicative constant, since the channel decoding is invariant to a scaling of the channel vectors. This means that we can normalize $\mathbf{G}$ and $\mathbf{H}$ by dividing with $m$ and $n_c$ respectively. We reuse the symbols $\mathbf{G}$ and $\mathbf{H}$ and write

$$\mathbf{G} = \mathbf{G}_\mathrm{c} + (m - 1)\mathbf{a}_\mu{\mathbf{a}_\mu}^{\mathrm{T}} \tag{17}$$

$$\mathbf{H} = \mathbf{H}_\mathrm{c} + (mn - n_c)/n_c\mathbf{u}_\mu{\mathbf{a}_\mu}^{\mathrm{T}} \ . \tag{18}$$

On the other hand, the normal equations of (10) are

$$\mathbf{CG}_\mathrm{c} - \mathbf{H}_\mathrm{c} + (m - 1)(\mathbf{Ca}_\mu{\mathbf{a}_\mu}^{\mathrm{T}} - k\mathbf{u}_\mu{\mathbf{a}_\mu}^{\mathrm{T}}) = 0 \ , \tag{19}$$

which after some trivial rearranging become exactly $\mathbf{CG} = \mathbf{H}$ with $\mathbf{G}$ and $\mathbf{H}$ from (17) and (18).

## 4 Experiments

### 4.1 1D Example

In the first experiment, a one-dimensional function was learned using various numbers of inliers and outliers. The input space was encoded using 12 channels, and the output space using 8 channels. Figure 3 shows the function together with the learned approximation in two different settings. Note that the accuracy is much higher than the output channel spacing.

A number of experiments on learning speed was performed, and in all cases the results were averaged over 30 runs. In Fig. 4, the RMS approximation error is shown as a function of the number of training samples. Note that *the method converges faster when the number of simultaneously presented pairs increase*, which is explained by the fact that the total number of $(x, y)$-pairs grows faster in this case. To further illustrate this effect, Fig. 5 shows the error after 50 samples against the number of correspondences $n_c$ in each sample (no outliers). We see that the benefit of using more $(x, y)$-pairs in each training sample saturates rather quickly. If $n_c$ is very high, all vectors $\bar{\mathbf{a}}$ and $\bar{\mathbf{u}}$ have a large DC offset, and the significant part will be small in comparison, which can lead to numerical problems. This is illustrated in the right plot of Fig. 5, where white Gaussian noise with a standard deviation of $1\%$ of the mean channel magnitude has been added to the channel vectors. When $n_c$ is large, this relatively small noise term starts to destroy the significant part of the channel vectors, leading to an increased error.

We see that $o_x$ does not seem to affect the asymptotical solution, as expected. Even when $o_x$ is four times $n_c$, the method converges reasonably fast. However, when $o_y$ is large the method breaks down and leaves a remaining error.

### 4.2 2D Example

The second experiment implements a cognitive systems *perception-action* learning scenario. Suppose a cognitive system obtains a list of percepts $\mathbf{p}_1$, performs an action $a$, and observes a new list of percepts $\mathbf{p}_2$. It then wants to learn the mapping $(\mathbf{p}_1, a) \to \mathbf{p}_2$ without knowing the correspondence structure between $\mathbf{p}_1$ and $\mathbf{p}_2$. To make a simple conceptual example, we chose $\mathbf{p}_1$ to be 50 random points in an image plane and $\mathbf{p}_2$ to be the same points rotated in 3D

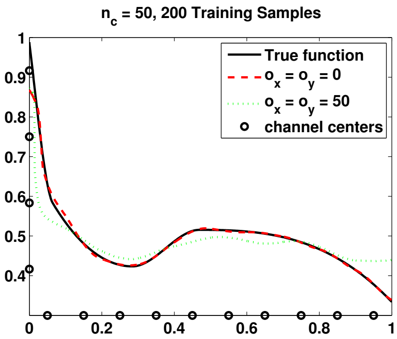

Figure 3. True 1D function and two examples of resulting approximations

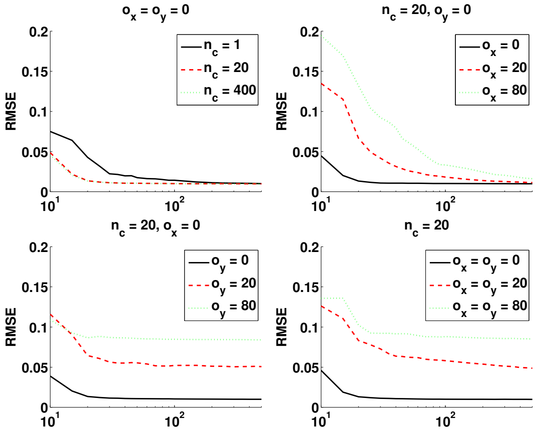

Figure 4. Approximation error under various configurations, averaged over 30 runs

space as a result of some action. In a real system, these points could be the result of some points-of-interest operator, e.g. Harris. The system was trained using 300 such $(\mathbf{p}_1, \mathbf{p}_2)$-pairs as training data, and was then used to predict the outcome of performing the same action on a novel configuration. The input space was encoded using $15 \times 15$ channels, and the output space using $10 \times 10$ channels, evenly distributed in the two spaces. The qualitative behavior is illustrated in Fig. 6.

## 5 Conclusions

In this paper, we have studied the not-so-common problem of learning mappings through training data with unknown correspondence structure. A rather simple method has been presented, which gives surprisingly good results. In the outlier- and noise-free case, the mapping is learned at least as quickly as in the known-correspondences case, regardless of the size of each group. Outliers in the input

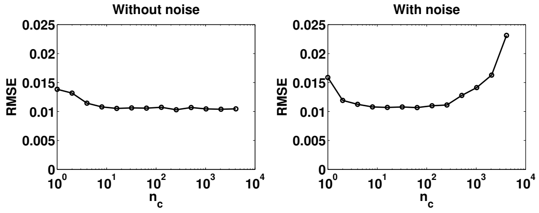

Figure 5. Approximation error after 50 samples using different $n_c$, no outliers.

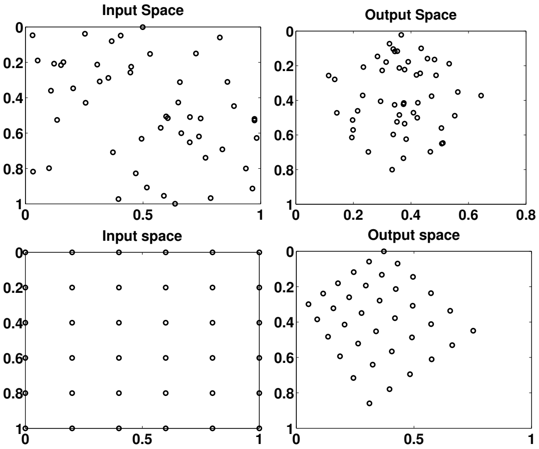

Figure 6. Top: Example of one training sample. Bottom: Operation mode example

space following the same distribution as the inliers are suppressed, but arbitrary outliers in any domain lead to remaining errors.

The method has been evaluated on artificial problems. The real applications are partly left to the imagination of the reader, but cognitive systems engineering is one tentative area of application. Since all artificial cognitive systems approaches have failed to produce anything even remotely similar to human beings in terms of learning and self-organizing capabilities, we need to look at alternative methods, structures and problems - trying to attack the problem from new angles. This paper has been an attempt in that direction.

## Acknowledgments

This work has been supported by EC Grant IST-2003-004176 COSPAL. This paper does not represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

## References

[1] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-9:698–700, 1987.

[2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. SoftPOSIT: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59:259–284, Sept. 2004.

[4] M. Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner. Many-to-many feature matching using spherical coding of directed graphs. In *Proc. 8th European Conf. on Computer Vision (ECCV)*, LNCS 3021, pages 322–335, May 2004.

[5] M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, February 2006.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] P.-E. Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, March 2004. Dissertation No. 858, ISBN 91-7373-876-X.

[8] G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.

[10] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Prentice Hall, second edition, 1999.

[11] B. Johansson. *Low Level Operations and Learning in Computer Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, December 2004. Dissertation No. 912, ISBN 91-85295-93-0.

[12] G. Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8:257–277, May 1992.