Motion Estimation Using Statistical Learning Theory

Harry Wechsler, *Fellow*, *IEEE*, Zoran Duric, *Senior Member*, *IEEE*,
Fayin Li, and Vladimir Cherkassky, *Senior Member*, *IEEE*

Abstract—This paper describes a novel application of Statistical Learning Theory (SLT) to single motion estimation and tracking. The problem of motion estimation can be related to statistical model selection, where the goal is to select one (correct) motion model from several possible motion models, given finite noisy samples. SLT, also known as Vapnik-Chervonenkis (VC), theory provides analytic generalization bounds for model selection, which have been used successfully for practical model selection. This paper describes a successful application of an SLT-based model selection approach to the challenging problem of estimating optimal motion models from small data sets of image measurements (flow). We present results of experiments on both synthetic and real image sequences for motion interpolation and extrapolation; these results demonstrate the feasibility and strength of our approach. Our experimental results show that for motion estimation applications, SLT-based model selection compares favorably against alternative model selection methods, such as the Akaike's fpe, Schwartz' criterion (sc), Generalized Cross-Validation (gcv), and Shibata's Model Selector (sms). The paper also shows how to address the aperture problem using SLT-based model selection for penalized linear (ridge regression) formulation.

Index Terms—Aperture problem, complexity control, condition number, image flow, model selection, motion estimation, robust learning, statistical learning theory, tracking, visual motion.

1 Introduction

Learning plays a fundamental role in facilitating "the balance between internal representations and external regularities." As "versatility <generalization> and scalability are desirable attributes in most vision systems," the "only solution is to incorporate learning capabilities within the vision system" [22]. Many challenging problems in computer vision could thus benefit from using predictive learning, when the goal is to come up with "good" models based on available (training) data under fairly general (flexible) assumptions. A good model is expected to provide accurate predictions for future (test) data. Toward that end, this paper describes a novel application of Statistical Learning Theory (SLT) to motion estimation. SLT provides the mathematical and conceptual framework needed for estimating dependencies from finite training data, it enables a better understanding of issues responsible for generalization, and it facilitates the development of better (more rigorous) learning algorithms. SLT provides analytical generalization bounds for model selection, which relate unknown prediction risk (generalization performance) and known quantities such as the number of training samples, empirical error, and a measure of model complexity called the Vapnik-Chervonenkis (VC)-dimension.

In the predictive learning framework [29], [30], [9], obtaining a good model with finite training data requires the specification of admissible models (or approximating functions), e.g., the regression estimator, an inductive principle for combining admissible models with available data, and an optimization ("learning") procedure for estimating the parameters of the admissible models. The inductive principle is responsible for model selection and it directly affects the generalization ability in terms of prediction risk, i.e., the performance on unseen/future data. Conversely, a learning method is a constructive implementation of an inductive principle, i.e., an optimization or parameter estimation procedure, for a given set of approximating functions. Model selection corresponds to one seeking an optimal model for a well-defined predictive learning problem formulation, i.e., for a given set of admissible models and the loss function that provides a measure of generalization performance. In this setting, model selection amounts to *model complexity control* [9], [10].

In computer vision applications, model selection using predictive learning expands on the classical statistical framework and defines a novel framework, that of robust learning. Robustness is related to both accuracy and functionality, which have been succinctly defined in the context of computer vision as "how close the captured motion corresponds to the actual motion performed by the subject" and "the fewer assumptions a system imposes on its operational conditions, the more robust it is considered to be. Many systems are based on knowing the initial state of their systems and/or a well-defined model fitted (offline) to the current subject. In a real life scenario, we may expect a system to be capable of autonomy and run on its own, i.e., adapt to the current situation. Related to this is the problem of how to recover from failure. A number of systems are based on incremental updates or searching around a predicted value. Many of these fail due to bad predictions and are not able to recover" (see Moeslund and Granum [21]). Poggio and Shelton [23] further address robust learning in computer vision, when they make a strong case

- *H. Wechsler, Z. Duric, and F. Li are with the Department of Computer Science, George Mason University, Fairfax, VA 22030-4444. E-mail: {wechsler, zduric, fli}@cs.gmu.edu.*
- *V. Cherkassky is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: cherkass@ece.umn.edu.*

Manuscript received 14 May 2002; revised 20 Feb. 2003; accepted 6 Oct. 2003. Recommended for acceptance by L. Vincent.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 116542.

that "the problem of learning is arguably at the very core of the problem of intelligence, both biological and artificial. Since seeing is intelligence, learning is also becoming a key to the study of artificial and biological vision. Vision systems that learn and adapt represent one of the most important directions in computer vision research. It may be the only way to develop vision systems that are robust and easy to use in many different tasks."

This paper describes an application of an SLT-based model selection to the challenging problem of estimating optimal motion models from small sets of image measurements (flow). We present results of experiments on both synthetic and real image sequences for both motion interpolation and extrapolation; these results demonstrate the feasibility and strengths of our approach. In addition, our results showed that our approach compares favorably against alternative model selection methods regarding the confidence they offer on model selection for motion estimation. Our experimental results show that for motion estimation applications, SLT-based model selection compares favorably against alternative model selection methods, such as the Akaike's Final Prediction Error (*fpe*), Schwartz' criterion (*sc*), Generalized Cross-Validation (*gcv*), and Shibata's Model Selector (*sms*). The paper also shows how to address the aperture problem using SLT-based model selection for penalized linear (ridge regression) formulation.

2 Model Selection for Regression

This section briefly reviews model selection in predictive learning, and reviews classical and VC-based analytic methods for model selection. A learning method is an algorithm that estimates an unknown *mapping* (dependency) between system's inputs and outputs, from the available data, i.e., *known* (input, output) samples. Once such a dependency has been estimated, it can be used for prediction of system outputs from the input values. The usual goal of learning is the prediction accuracy, also known as generalization. A generic learning system [13], [9], [30] consists of:

- a *Generator* of random input vectors $\mathbf{x}$, drawn from a fixed (unknown) probability distribution $P(\mathbf{x})$;
- a *System* (or teacher) which returns an output value y for every input vector $\mathbf{x}$ according to the fixed conditional distribution $P(y|\mathbf{x})$, which is also unknown;
- a *Learning Machine*, which is capable of implementing a set of approximating functions $f(\mathbf{x}, \omega)$, $\omega \in \Omega$, where Ω is a set of parameters of an arbitrary nature.

The goal of learning is to select a function (from this set), which approximates best the System's response. Many problems in computer vision can be formally stated as the problem of real-valued function estimation from noisy samples. This problem is known in statistics as (nonlinear) regression. In the regression formulation, the goal of learning is to estimate an unknown (target) function $g(\mathbf{x})$ in the relationship: $y = g(\mathbf{x}) + \varepsilon$, where the random error ε (noise) is zero mean, $\mathbf{x}$ is a d-dimensional vector and y is a scalar output. A learning method (or estimation procedure) selects the "best" model $f(\mathbf{x}, \omega_0)$ from a set of (parameterized) approximating functions (or possible models) $f(\mathbf{x}, \omega)$ specified a priori, where the quality of an approximation is measured by the loss or discrepancy measure $L(y, f(\mathbf{x}, \omega))$. A common loss function for regression is the squared error. Thus, learning is the problem of finding the function $f(x, \omega_0)$ (regressor) that minimizes the prediction risk functional

$$R(\omega) = \int (y - f(\mathbf{x}, \omega))^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

using only the training data $(\mathbf{x}_i, y_i),\ i = 1, \ldots, n$, generated according to some (unknown) joint probability density function (pdf) $p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x})$. Prediction risk functional measures the accuracy of the learning method's predictions of the unknown target function $g(\mathbf{x})$.

The standard formulation of the learning problem (as defined above) amounts to function estimation, i.e., selecting the "best" function from a set of admissible functions $f(\mathbf{x}, \omega)$. Here, the "best" function (model) is the one minimizing the prediction risk. The problem is ill-posed since the prediction risk functional is unknown (by definition). Most learning methods implement the idea known as "empirical risk minimization" (ERM), which is choosing the model minimizing the empirical risk, or the average loss for the training data:

$$R_{emp}(\omega) = \frac{1}{n} \sum_{k=1}^{n} (y_k - f(\mathbf{x}_k, \omega))^2. \quad (1)$$

The ERM approach is only appropriate under parametric settings, i.e., when the parametric form of unknown dependency is known. Under such a (parametric) approach the unknown dependency is assumed to belong to a narrow class of functions (specified by a given parametric form). In most practical applications, parametric assumptions do not hold true, and the unknown dependency is estimated in a wide class of possible models of varying complexity. Since the goal of learning is to obtain a model providing minimal prediction risk, it is achieved by choosing a model of optimal complexity corresponding to smallest prediction (generalization) error for future data. Existing provisions for model complexity control include [24], [9]: penalization (regularization), weight decay (in neural networks), parameter (weight) initialization (in neural network training), and various greedy procedures (also known as constructive, growing, or pruning methods).

Classical methods for model selection are based on asymptotic results for linear models. Recent approaches [3], [14], [18] based on approximation theory extend classical rate-of-convergence results to nonlinear models (such as multilayer perceptrons); they are, however, still based on asymptotic assumptions. Nonasymptotic (guaranteed) bounds on the prediction risk for finite-sample settings have been proposed in VC-theory [29]. We also point out that all approximation theory results are aimed at deriving accurate estimates of risk since the goal of (prediction) risk estimation is equivalent to complexity control when the number of samples is large (i.e., asymptotic case). However, there is a subtle but important difference between the goal of accurate estimation of prediction risk and using those estimates for model complexity control with *finite samples*. That is, a model selection criterion can provide poor estimates of prediction risk, yet the *differences* between its risk estimates (for models of different complexity) may yield accurate model selection [10].

There are two general approaches for estimating prediction risk for regression problems with finite data: analytical and data-driven. Analytical methods use analytic estimates of

the prediction risk as a function of the empirical risk (training error) penalized (adjusted) by some measure of model complexity. Once an accurate estimate of the prediction risk is found, it can be used for model selection by choosing the model complexity that minimizes the estimated prediction risk. In the statistical literature, various analytic prediction risk estimates have been proposed for model selection (for linear regression). These estimates take the form of:

$$R_{est} = r\left(\frac{d}{n}\right)\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2}$$

where r is a monotonically increasing function of the ratio of model complexity d (the number of degrees of freedom) and the training sample size n. r is often called a *penalization factor* because it inflates the average residual sum of squares for increasingly complex models.

SLT provides analytic upper bounds on the prediction risk that can be used for model selection [29], [30]. To make practical use of such bounds for model selection, practical values for theoretical constants involved have to be chosen [9], [10]; this results in the penalization factor, $r(p, n)$, called the Vapnik's measure (vm):

$$r(p,n) = \left(1 - \sqrt{p - p\ln p + \frac{\ln n}{2n}}\right)_{+}^{-1}, \tag{3}$$

where $p = h/n$, h denotes the VC-dimension of a model and $(\cdot)_+ = 0$, for $x < 0$. In SLT, $R_{est}(\omega)$ is obtained by substiting $r(p, n)$ for $r(d/n)$ in (2). For linear estimators with m degrees of freedom, the VC-dimension is $h = m + 1$. For a given training data set, selection of the "best" model from several parametric models corresponds to choosing the model providing minimum bound on prediction risk (2), with a penalization factor given by (3).

3 Motion Estimation

The estimation of motion from image sequences is "a difficult problem that involves pooling noisy measurements to make reliable estimates;" furthermore, motion estimation "assumes some model of the image variation within a region" [6]. Early attempts at robust (optical flow) motion estimation involved least-squares regression followed by outlier detection and rejection, and then reestimation of the motion for the remaining pixels [4] using robust statistical techniques (M-estimators) known for their low breakdown point to compute a dominant motion while down weighting the outliers.

Much of computer vision, including motion, registration, segmentation, and stereo, calls for optimal estimation using linear and nonlinear (penalized) regression. In motion analysis, regression models are often used in two different contexts, i.e., interpolation or extrapolation, as explained next. The regression problem involved in motion estimation has a "training" set of examples, i.e., input vectors x_i along with corresponding targets y_i, which put in correspondence similar image locations, drawn from two consecutive frames sampled at time t and $t + 1$, respectively, or several consecutive frames from a video sequence. Using the training set, one seeks to learn how to model the dependency of the targets ("dependent variables") on the inputs ("independent variables"). The objective is to make accurate predictions on image points available but not included in the training set, or on image points not yet seen from future frames; this corresponds to *interpolation* and *extrapolation*, respectively.

We consider a short "real image sequence" and a long "synthetic image sequence" both involving approximately constant motion for the purpose of motion estimation. Ground truth is available only for the synthetic image sequences. The real image sequence is inherently noisy and no ground truth is available. In the case of small image displacements, we generate data $\{x_i, y_i\}$ for regression using normal flow computation (see Section 3.1); for large displacements, i.e., the synthetic image sequence, the image correspondences and ground truth are given. Model fitting for each of a set of admissible models is done using the LS estimation (see Section 3.2). Section 3.3 details how to choose the optimal model using VC-based complexity control.

Note that the goal here is not to suggest novel motion analysis models, but rather to choose, based on finite and noisy data, an optimal model that would yield minimum error for future inputs (i.e., minimum prediction risk). Under motion analysis setting, we use model selection criteria in order to select "correct" motion model from several possible motion models, where the parametric motion models are known to contain the true motion model. This setting is much simpler than the general problem of model selection [9], where the set of possible models may not contain the true model.

3.1 Normal Flow

Normal flow computation [2], [12] provides the training data needed for model selection and model fitting, i.e., parameter estimation. Let $\vec{i}$ and $\vec{j}$ be the unit vectors in the x and y directions, respectively; $\delta\vec{r} = \vec{i}\delta x + \vec{j}\delta y$ is the projected displacement field at the point $\vec{r} = x\vec{i} + y\vec{j}$. If we choose a unit direction vector $\vec{n}_r = n_x\vec{i} + n_y\vec{j}$ at the image point $\vec{r}$ and call it the normal direction, then the normal displacement field at $\vec{r}$ is $\delta\vec{r}_n = (\delta\vec{r}\cdot\vec{n}_r)\vec{n}_r = (n_x\delta x + n_y\delta y)\vec{n}_r$. The normal direction $\vec{n}_r$ can be chosen in various ways; the usual choice is the direction of the image intensity gradient $\vec{n}_r = \nabla I/\|\nabla I\|$. Note that the normal displacement field along an edge is orthogonal to the edge direction. Thus, if at time t we observe an edge element at position $\vec{r}$, the apparent position of that edge element at time $t + \Delta t$ will be $\vec{r} + \Delta t\delta\vec{r}_n$. This is a consequence of the well-known *aperture problem* (see Section 6). We base our method of estimating the normal displacement field on this observation.

For an image frame (say collected at time t), we find edges using an implementation of the Canny edge detector. For each edge element, say at $\vec{r}$, we resample the image locally to obtain a small window with its rows parallel to the image gradient direction $\vec{n}_r = \nabla I/\|\nabla I\|$. For the next image frame (collected at time $t_0 + \Delta t$), we create a larger window, typically twice as large as the maximum expected value of the magnitude of the normal displacement field. We then slide the first (smaller) window along the second (larger) window and compute the difference between the image intensities. The zero of the resulting function is at distance u_n from the origin of the second window; note that the image gradient in the second window at the positions close to u_n must be positive. Our estimate of the normal displacement field is then $-u_n$, and we call it the normal flow. A real color image sequence used in our experiments is shown in Fig. 1. The corresponding normal flow is shown in Fig. 2.

Fig. 1. Frames 2, 4, 6, 8, 10, and 12 from a 13-frame sequence of a moving arm.

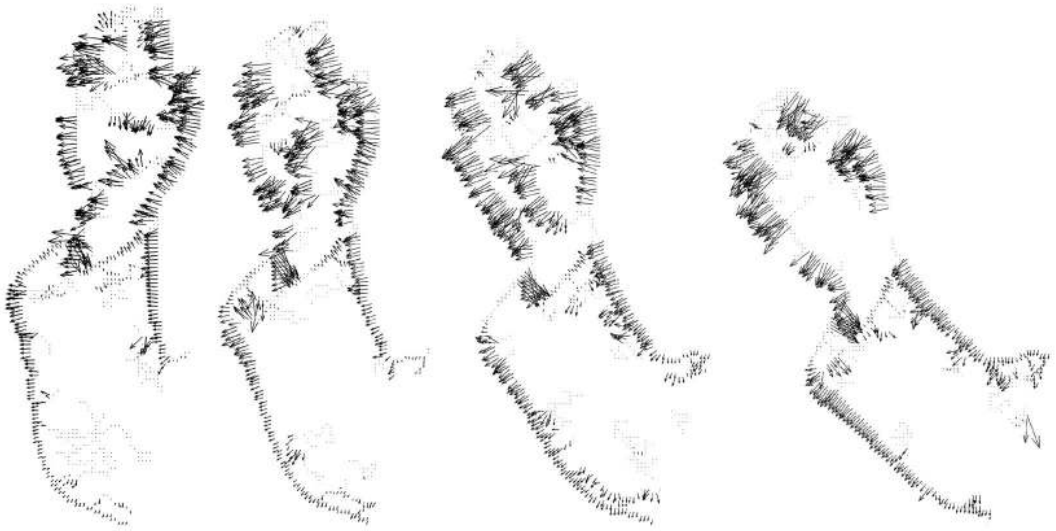

Fig. 2. Normal flow computed from pairs of frames 2-3, 4-5, 7-8, and 10-11 of the moving arm sequence.

3.2 Parameter Estimation

A hierarchy of parametric flow models has been developed in the past starting from pure translation, image plane rotation, 2D affine, and 2D homography (8-parameter flow, also known as quadratic flow). We will consider all those models here. Eight-parameter flow corresponds to the instantaneous projected image motion field generated by a moving plane. Other models used here can be obtained by setting some of the eight parameters to zero. In the 8-parameter model, coordinates of a point (x, y) in the first frame will move to (x', y') in the next frame:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} w_1 \\ w_4 \end{pmatrix} + \begin{pmatrix} w_2 & w_3 \\ w_5 & w_6 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \begin{pmatrix} w_7 \\ w_8 \end{pmatrix}. \tag{4}$$

Equation (4) relates corresponding points in successive image frames. To obtain the displacement $\vec{u}(x, y) = (\delta x \ \delta y)^T$ of (x, y) we subtract $(x \ y)^T$ from both sides of (4). The left-hand side of (4) is replaced by $(\delta x \ \delta y)^T$ and on the right-hand side w_2 and w_6 get replaced by $w_2^1 = w_2 - 1$ and $w_6^1 = w_6 - 1$. We obtain

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \begin{pmatrix} w_1 \\ w_4 \end{pmatrix} + \begin{pmatrix} w_2^1 & w_3 \\ w_5 & w_6^1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \begin{pmatrix} w_7 \\ w_8 \end{pmatrix}. \tag{5}$$

The normal displacement field at (x, y) is given by

$$\begin{aligned} u_n(x, y) = \delta\vec{r}_n \cdot \vec{n}_r = n_x\delta x + n_y\delta y = w_1 n_x + w_2^1 x n_x \\ + w_3 y n_x + w_7 x^2 n_x + w_8 x y n_x + w_4 n_y + w_5 x n_y \\ + w_6^1 y n_y + w_7 x y n_y + w_8 y^2 n_y = \mathbf{w} \cdot \mathbf{p}, \end{aligned}$$

where $\vec{n}_r = n_x\vec{i} + n_y\vec{j}$ is the gradient direction,

$$\mathbf{p} = (n_x \ \ x n_x \ \ y n_x \ \ n_y \ \ x n_y \ \ y n_y \ \ x^2 n_x + x y n_y \ \ x y n_x + y^2 n_y)^T,$$

and $\mathbf{w} = (w_1 \ \ w_2^1 \ \ w_3 \ \ w_4 \ \ w_5 \ \ w_6^1 \ \ w_7 \ \ w_8)^T$ is the vector of parameters.

We use the method described in Section 3.1 to compute the normal flow. For each edge point $\vec{r}_i$, we have one normal flow value $u_{n,i}$, that we use as an estimate of the normal displacement at the point, a vector $\mathbf{p}_i$ computed from (x_i, y_i) and $\vec{n}_{r,i} = n_{x,i}\vec{i} + n_{y,i}\vec{j}$, and an approximate equation $\mathbf{w} \cdot \mathbf{p}_i \approx u_{n,i}$. Let the number of edge points be $N \geq 8$. We need to find a solution of $P\mathbf{w} - \mathbf{b} = \mathbf{e}$, where $\mathbf{b}$ is an N-element vector with elements $u_{n,i}$, P is an $N \times 8$ parameter matrix with rows $\mathbf{p}_i$, and $\mathbf{e}$ is an N-element error vector. We seek the model $\mathbf{w}$ that minimizes $\|\mathbf{e}\| = \|\mathbf{b} - P\mathbf{w}\|$; the solution satisfies the system $P^T P\mathbf{w} = P^T\mathbf{b}$ and corresponds to the linear least squares (LS) solution. Model fitting for large image displacements sampled from the synthetic image sequence is also done using the LS as described above.

3.3 Model Selection

Training data consists of normal flow or point displacements. Affine and quadratic models are responsible for data generation. The motion estimation (learning) problem corresponds to choosing the best motion model from a given set of possible motions using the observed (training) data. The goal is to choose a model that will yield the lowest error at the image points not used in training. In this section, we combine the SLT regression (see Section 2) and motion estimation techniques described in the preceding sections to choose a flow model that has the best predictive performance.

We use the square loss function—i.e., the squared difference $R_{emp}(\mathbf{w}_m) = \frac{1}{n}\sum_{i=1}^{N}(u_{n,i} - u_{n,i}^m)^2$ between the computed normal flow $u_{n,i}$ and the predicted normal flow $u_{n,i}^m = \mathbf{w}_m \cdot \mathbf{p}_i$, where $\mathbf{w}_m$ corresponds to the estimated

model. The task of model selection corresponds to choosing the best predictive model from a given set of linear parametric models, using a small set of noisy training data. We use VC-generalization bounds (see (2) and (3)). The VC-dimension h of a linear model is given by the number of degrees of freedom (DoF) of the model plus one. We choose from the following five models that we obtain by setting various elements of $\mathbf{w}$ in (5) to zero:

- [M_1:] *pure translation*, $w_2^1 = w_3 = w_5 = w_6^1 = w_7 = w_8 = 0$, 2 DoF, $h = 3$.
- [M_2:] *translation, shear, and rotation*, $w_2^1 = w_6^1 = w_7 = w_8 = 0$, 4 DoF, $h = 5$.
- [M_3:] *translation and scaling*, $w_3 = w_5 = w_7 = w_8 = 0$, 4 DoF, $h = 5$.
- [M_4:] *6-parameter affine*, $w_7 = w_8 = 0$, 6 DoF, $h = 7$.
- [M_5:] *full affine, quadratic flow*, 8 DoF, $h = 9$.

The next two sections present the results of experiments using both synthetic and real image sequences for motion interpolation and extrapolation. These results demonstrate the feasibility and the strengths of our approach. The experiments assume single (spatially and temporally coherent) motions for all data sets. The same parameter values apply to all frames of each data set.

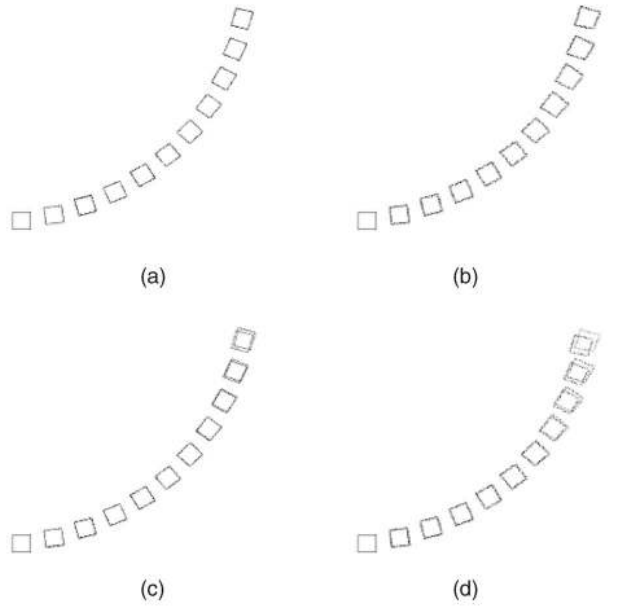

Fig. 3. Motion interpolation results for a noise-free (a) and a noisy sequence (b). Motion extrapolation results for a noise-free (c) and a noisy sequence (d). The pixels corresponding to the ground truth (M_4) and the average estimated positions are shown.

4 Experimental Results for Synthetic Image Sequences

We generated a synthetic "first moving square" image sequence consisting of 11 frames (see Figs. 3a and 3c); the square consists of 128 pixels. A corresponding "noisy" sequence was created by adding Gaussian noise with mean 0 and variance 0.5 (see Figs. 3b and 3d). The ground truth corresponds to the affine transformation model $M4$ (see (4) and Section 3.3):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 2.699308 \\ -2.4648887 \end{pmatrix} + \begin{pmatrix} 0.991562 & 0.129631 \\ -0.129631 & 0.991562 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

The other motion models considered are: M_1, *pure translation*, $w_2 = w_6 = 1$, $w_3 = w_5 = w_7 = w_8 = 0$; M_2, *translation and scaling*, $w_3 = w_5 = w_7 = w_8 = 0$; M_3, *translation, shear, and rotation*, $w_2 = w_6 = 1$, $w_7 = w_8 = 0$; and M_5, *quadratic model*.

The parameters w_i have qualitative interpretations in terms of image motion. For example, w_1 and w_4 represent horizontal and vertical translation, respectively. Additionally, it is possible to express divergence (isotropic expansion), curl (rotation about the viewing direction), and deformation (squashing or stretching) as combinations of the w_is, while the parameters w_7 and w_8 roughly represent the yaw and pitch deformations in the image plane. (Similar qualitative interpretation for the affine optical flow model is possible in terms of rotation, translation, scale, and skew.) Black et al. [5] point out that, for small regions of human body images such as eyes and fingers, the quadratic model may not be necessary and the motion of these regions can be approximated by the simpler affine model defined earlier in which the terms w_7 and w_8 are zero.

For interpolation experiments, we randomly subsample $n = 32$ or 64 pixel correspondences (out of 128) from 10 successive pairs of frames $(< i, i+1 >,\ i = 1 \ldots 10)$, and estimate the parameters for each of the motion models using LS (see Section 3.2). Note that estimation is done independently for the x and y coordinates. As a consequence, the VC-dim h for M_1, M_2, M_3, M_4, and M_5 is now 2, 3, 3, 4, and 5, respectively. The bound on risk (see (2) and (3)) for each model is derived using the LS error and the penalty vm (3). The corresponding (to each model) interpolating total error (for all frame pairs) is calculated using all the 128 points and the ground truth information. Note that the total error is an estimate of (unknown) prediction risk in the VC-theoretical formulation. The experiment is repeated 100 times with different random realizations of training data, for both nonnoisy and noisy image sequences. The average interpolated sequences are shown in Fig. 3. The box plots (see Fig. 4) summarizing the empirical risk, the bound on prediction risk, and the total error across the whole sequence, for $n = 32$, show that the bound on prediction risk can be used for model selection for both non-noisy and noisy sequences. Note that the bound on prediction risk is a better predictor than the empirical risk. The empirical risk, the bound on prediction risk and (total) interpolation error for M_3 are too large to be displayed. Ground truth M_4 is consistently found as the optimal motion model; its interpolation error is minimum. The quadratic model M_5 is a very close runner-up to M_4. Similar results were obtained for experiments with $n = 64$.

For extrapolation ("tracking") experiments, we randomly subsample $n = 32$ or 64 pixel correspondences (out of a stack of 5×128 correspondences) from the first five pairs of frames $(< i, i+1 >,\ i = 1 \ldots 5)$, and estimate the parameters for each of the models using LS (see Section 3.2). Note that the data available for extrapolation is much less than the data available for the interpolation experiment. Here again, estimation is done independently for the x and y coordinates. As a consequence, the VC-dim h for M_1, M_2, M_3, M_4, and M_5 is now 2, 3, 3, 4, and 5, respectively. The bound on risk for each model is derived using the LS error and the penalty vm (3). The corresponding extrapolating

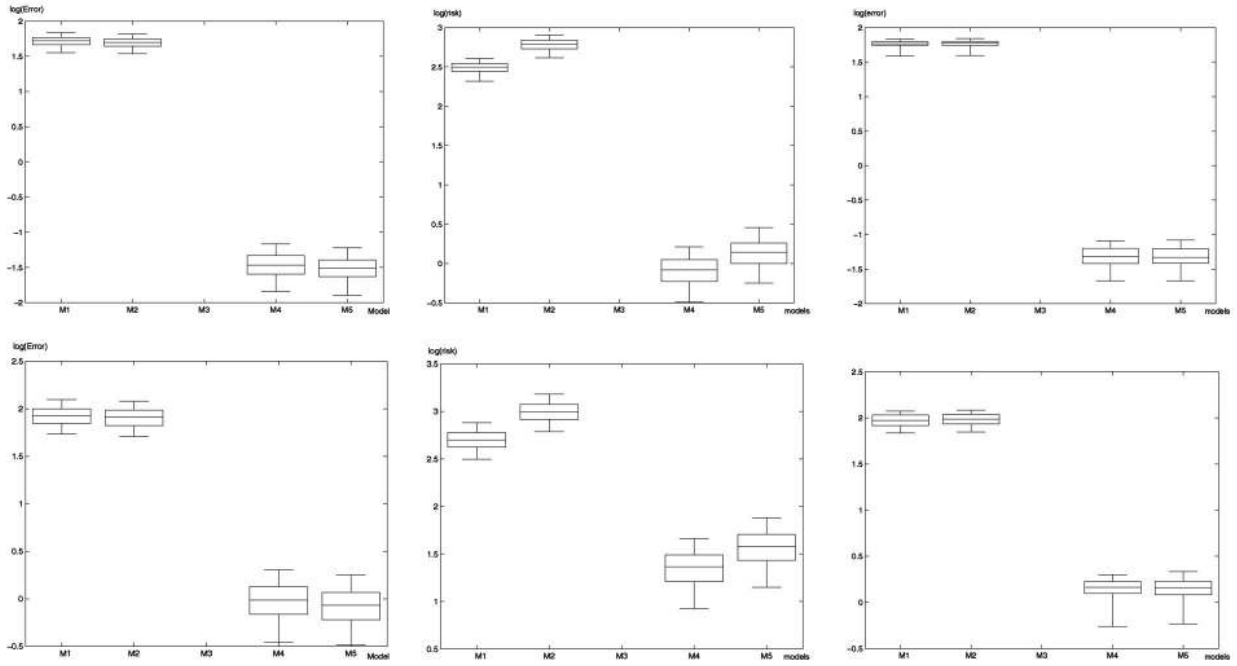

Fig. 4. Summary of interpolation results for the "first moving square" sequence, $n = 32$. The ground truth motion model is M_4. Left to right: empirical risk, bound on prediction risk, and interpolation error. Top row: nonnoisy sequence. Bottom row: noisy sequence.

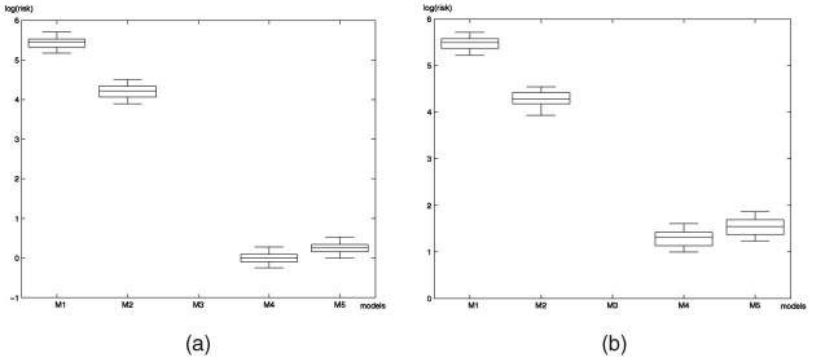

Fig. 5. Bounds on prediction risk for the "first moving square" sequence, $n = 32$. The ground truth motion model is M_4. (a) Nonnoisy sequence. (b) Noisy sequence.

error for the remaining frames is calculated using the ground truth data starting from image frame 6. The experiment is repeated 300 times with different random realizations of training data for nonnoisy and noisy image sequences. The average extrapolated sequences are shown in Fig. 3. Fig. 5 shows the bound on prediction risk for each motion model, while Tables 1 and 2 show the median extrapolation error for each of the remaining five image frames for all models. It can be seen that the bound on prediction risk yields good model selection. The bound on prediction risk for M_3 is too large to be displayed. Ground truth M_4 is found as the optimal model for motion tracking and its extrapolated error is minimum. The quadratic model M_5 is again a very close runner-up to the ground truth. Similar results were obtained for experiments with $n = 64$. It can be seen from Tables 1 and 2 that the second ranked model, M_5, can keep track with the optimal model M_4 only for the first two extrapolated frames (6 and 7).

In the next experiment, we generate a "first moving square" synthetic image sequence consisting of 11 frames. To create the corresponding noisy sequence, we add Gaussian noise (the mean 0 and the variance 0.5). The motion is a quadratic transformation—M_5, with the center of the square as a reference point—given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 4.386585 \\ 4.983113 \end{pmatrix} + \begin{pmatrix} 0.965926 & 0.258819 \\ -0.258819 & 0.965926 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \begin{pmatrix} 0.002140 \\ 0.006435 \end{pmatrix}.$$

For interpolation experiments, we randomly subsample $n = 32$ or 64 pixel correspondences (out of 128) from 10 successive pairs of frames $(< i, i+1 >,\ i = 1 \ldots 10)$, and estimate the parameters of the motion models using LS (see Section 3.2). The bound on prediction risk for each model is derived using the LS error and the penalty vm (3). The corresponding (to each model) interpolating error (for all pair wise frames) is calculated using all the 128 points and the ground truth information. The experiment is repeated 100 times with different random realizations of training data. The box plots summarizing the empirical risk, bound on prediction risk, and total error for the sequence

TABLE 1
Median Extrapolation Error for the Nonnoisy Sequence, $n = 32$, the Ground Truth is M_4

	M_1	M_2	M_3	M_4	M_5
frame 7	*515.73*	*48.17*	*2909.6*	*0.42*	*0.50*
frame 8	*2765.1*	*455.4*	*2394.2*	*0.91*	*1.72*
frame 9	*7991.3*	*2263.1*	*12025.9*	*2.03*	*6.28*
frame 10	*17722.9*	*8095.3*	*15165.1*	*4.53*	*20.43*
frame 11	*33531.4*	*23908.2*	*37564.4*	*8.98*	*59.50*

TABLE 2
Median Extrapolation Error for the Noisy Sequence, $n = 32$, the Ground Truth is M_4

	M_1	M_2	M_3	M_4	M_5
frame 7	*511.7*	*45.8*	*2757.04*	*1.42*	*1.88*
frame 8	*2714.4*	*405.9*	*2346.5*	*3.91*	*7.82*
frame 9	*7891.4*	*1973.7*	*11247.0*	*8.71*	*27.51*
frame 10	*17412.7*	*7201.0*	*14881.6*	*17.42*	*86.18*
frame 11	*32985.7*	*21482.4*	*35146.5*	*33.89*	*230.54*

(see Fig. 6) show that the bound on prediction risk can be used for model selection in motion estimation for both nonnoisy and noisy image sequences. The empirical risk, bound on prediction risk, and interpolation error for M_3 are too large to be displayed. The ground truth model, M_5, is consistently found as the optimal motion model; its interpolation error is minimum. Similar results to those shown in Fig. 6 were obtained for $n = 64$.

For extrapolation ("tracking") purposes we randomly subsample $n = 32$ or 64 pixel correspondences (out of a stack of 5×128 correspondences) from $(< i, i+1 >,\ i = 1 \ldots 5)$, and estimate the parameters for each of the models using LS (see Section 3.2). Note that the data available for extrapolation is much less than the data available for the interpolation experiment. The corresponding extrapolation error for the remaining frames is calculated using the ground truth data starting from frame 6. The experiment is repeated 300 times with different random realizations of training data. Fig. 7 shows the bound on prediction risk for each motion model. Tables 3 and 4 show the median extrapolation error for each of the remaining five image frames for all models. It can be seen that the bound on prediction risk yields good model selection. The bound on prediction risk for M_3 is too large to be displayed. The ground truth M_5 is found as the optimal model for motion tracking and its extrapolation error is minimum. Similar results were obtained for experiments with $n = 64$.

As could be expected, for synthetic image sequences, both the bound on prediction risk and the interpolation and extrapolation errors, were consistently decreasing as we increased the number of sample points from 16 to 128 (see Fig. 8). Note that the bound on prediction risk for the ground truth model is consistently lower than the bound on prediction risk for the next best model. Similar results for both interpolation and extrapolation were obtained if the

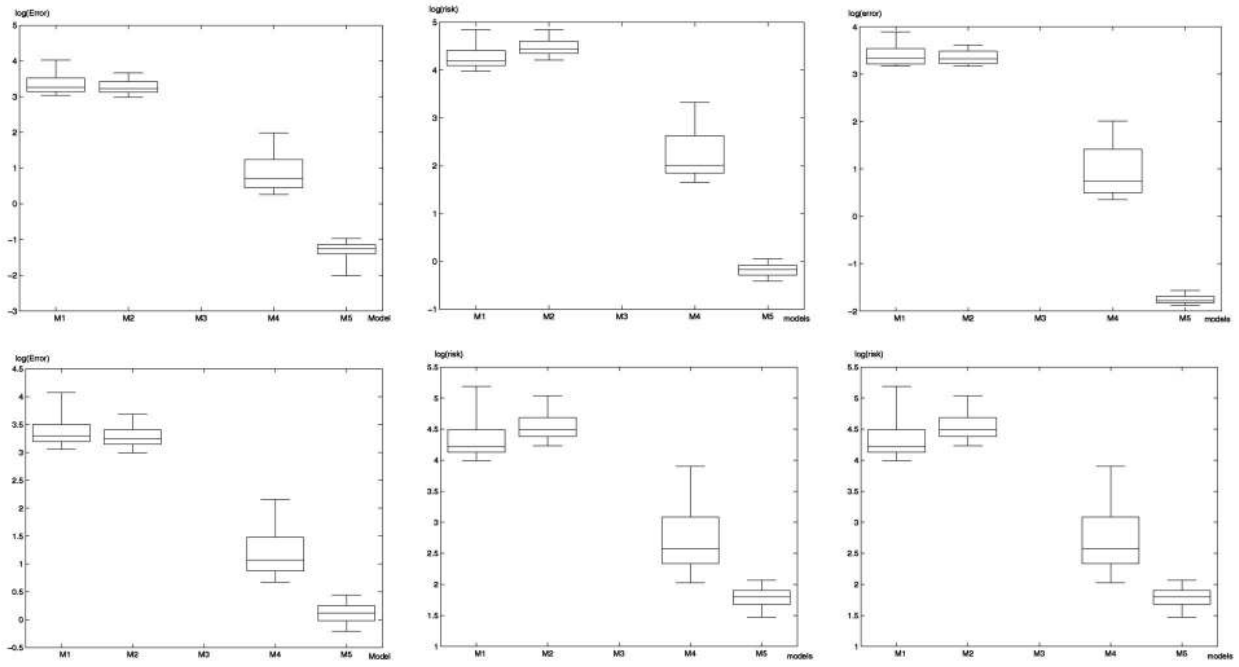

Fig. 6. Summary of interpolation results for the "first moving square" sequence, $n = 32$. The ground truth motion model is M_5. Left to right: empirical risk, bound on prediction risk, and interpolation error. Top row: nonnoisy sequence. Bottom row: noisy sequence.

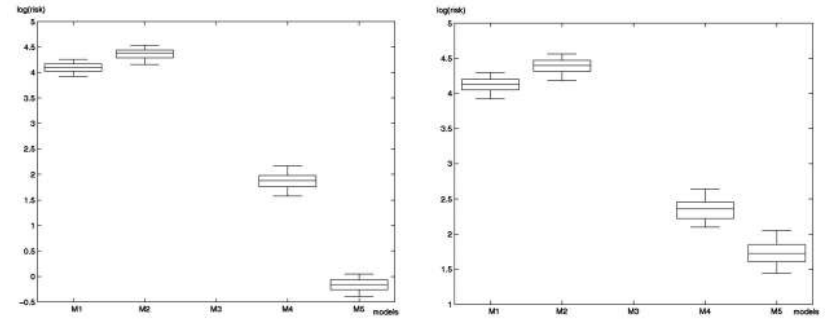

Fig. 7. Bounds on prediction risk for the "first moving square" sequence, $n = 32$. The ground truth motion model is M_5. (a) Nonnoisy sequence. (b) Noisy sequence.

samples were selected randomly across all the training frames, which we label as the "whole" interpolation or extrapolation, rather than the "standard" interpolation using random sampling from successive pairs of frames and then pooling the data together.

4.1 Comparative Analysis of Model Selection Criteria

Various analytic prediction risk estimates have been proposed in the statistical literature for model selection (2) using the observed empirical risk and penalization factor r. We performed a comparative analysis of model selection criteria for different classical forms of penalization factor $r(p) = r\left(\frac{d}{n}\right)$ (2), which have been proposed in the statistical literature. These penalization factors are listed below and compared to the Vapnik measure *vm* (3).

- Final prediction error (*fpe*) [1]: $r(p) = (1+p)(1-p)^{-1}$.
- Schwartz's criterion (*sc*) [27]: $r(p, n) = 1 + \frac{\ln n}{2} p(1-p)^{-1}$.
- Generalized cross-validation (*gcv*) [11]: $r(p) = (1-p)^{-1}$.
- Shibata's model selector (*sms*) [26]: $r(p) = 1 + 2p$.

Note that, in this experiment, all model selection criteria require two regressions. All these classical approaches are motivated by asymptotic arguments for linear models and, therefore, apply well for large training sets. In fact, for large n, prediction estimates provided by *fpe*, *gcv*, and *sms* are asymptotically equivalent. The penalization factor r inflates the average residual sum of squares for increasingly complex models. Note in particular that the *fpe* risk follows from general Akaike Information Criterion (AIC) when the noise variance is estimated via empirical risk for each chosen model complexity [9]. AIC is derived under a very restrictive setting such as asymptotic linear models and known noise model. AIC does not indicate how to estimate the noise model, which is required for model selection and is assumed to be known.

Our experimental results show that the Vapnik measure compares favorably against alternative model selection methods, regarding the confidence they offer in model selection for motion estimation (see Table 5). The results shown in the table are for standard interpolation, whole interpolation, and standard extrapolation. The entries show the percentage of time that each model selection criteria is accurate in its predictions. Please note that Table 5 is about choosing the right model rather than measuring the prediction error. Recall that the goal for VC-theory is to select the model providing the best prediction accuracy rather than selecting the correct model. Hence, it is possible that the VC-based model selection approach may select the "wrong" model providing the best generalization (prediction) accuracy [9], [10]. For example, Cherkassky and Mulier [9, pp. 28-29] provide experimental evidence from an experiment where a simpler linear decision rule, which does not match the underlying class distributions, often performs better than the ground truth quadratic decision rule.

TABLE 3
Median Extrapolation Error for the Nonnoisy Sequence, $n = 32$, the Ground Truth is M_5

	M_1	M_2	M_3	M_4	M_5
frame 7	28.58	28.85	383.2	2.70	0.19
frame 8	115.7	114.9	492.7	13.71	0.45
frame 9	263.5	258.3	549.0	39.49	0.72
frame 10	474.2	461.9	673.2	88.27	1.30
frame 11	759.8	735.3	858.7	172.7	2.39

TABLE 4
Median Extrapolation Error for the Noisy Sequence, $n = 32$, the Ground Truth is M_5

	M_1	M_2	M_3	M_4	M_5
frame 7	29.71	30.11	393.8	4.31	1.27
frame 8	120.9	119.9	513.6	19.55	2.94
frame 9	280.9	277.9	577.2	56.18	5.37
frame 10	526.4	516.2	731.3	131.8	9.47
frame 11	889.8	869.1	1003.0	286.2	20.36

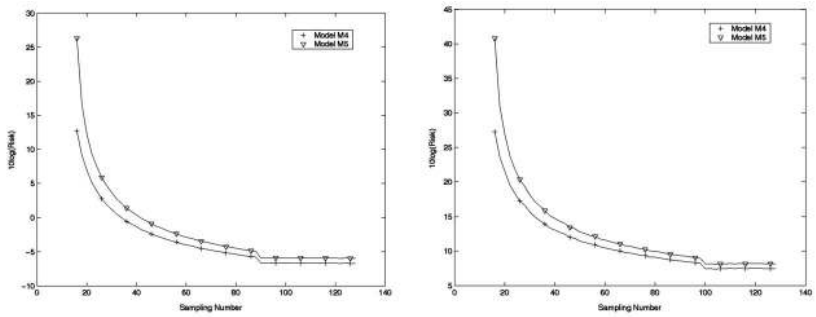

Fig. 8. Bounds on prediction risk as a function of number of samples for a nonnoisy (a) and a noisy sequence (b). The ground truth models are M_4 and M_5.

TABLE 5
Comparison of Model Selection Criteria on the "First Moving Square" Synthetic Image Sequence

Sequence	*Experiments*	*Samples*	*Criteria*				
			vm	*fpe*	*gcv*	*sc*	*sms*
Non-noisy	*Interpolation*	*32*	*92.7%*	*70.8%*	*74.5%*	*67.1%*	*66.5%*
		64	*90.7%*	*66.3%*	*68.3%*	*65.7%*	*59.4%*
	Extrapolation	*32*	*100%*	*90.2%*	*94.3%*	*84.2%*	*80.2%*
		64	*100%*	*85.0%*	*90.0%*	*80.0%*	*84.3%*
	Whole	*32*	*98.8%*	*84.8%*	*86.6%*	*86.0%*	*81.0%*
	Interpolation	*64*	*99.6%*	*85.0%*	*86.2%*	*87.0%*	*82.0%*
Noisy	*Interpolation*	*32*	*92.9%*	*65.5%*	*71.5%*	*62.7%*	*60.0%*
		64	*93.0%*	*50.9%*	*54.2%*	*52.0%*	*46.0%*
	Extrapolation	*32*	*100%*	*76.8%*	*86.4%*	*74.0%*	*74.1%*
		64	*100%*	*80.0%*	*80.3%*	*68.5%*	*70.1%*
	Whole	*32*	*99.5%*	*85.3%*	*87.5%*	*79.2%*	*76.1%*
	Interpolation	*64*	*99.0%*	*86.2%*	*84.8%*	*86.0%*	*86.2%*

5 Experimental Results for Real Image Sequences

Training data comes from eleven frames drawn from a real image sequence of a moving arm and the corresponding normal flow (see Figs. 1 and 2). We report the results obtained for both interpolation and extrapolation (using (5) and in Section 3.3). The ground truth is not known and the images are inherently noisy. In the interpolation experiment, we randomly subsampled 25 percent of image flow values (out of approximately 400 points); the experiment is repeated 100 times with different random realizations of training data. Training data is drawn from two frame pairs: $(i, i+1)$ and $(i+2, i+3)$; interpolation is performed for frames $(i+1, i+2)$. Fig 9 summarizes the prediction risk and interpolation error for the whole experiment. The prediction risk ranks the motion models so that its optimal choice, M_2, yields the minimum total interpolation error.

In the extrapolation experiment we randomly subsampled 10 percent of image flow values (out of approximately 400 points); the experiment is repeated 100 times with different random realizations of training data. Training data was drawn from the first five pairs of frames: $(i, i+1)$ starting with $i = 1$. Extrapolation is performed for the remaining five frames of the sequence. Fig. 10 summarizes the prediction risk and extrapolation error for the whole experiment. The prediction risk ranks

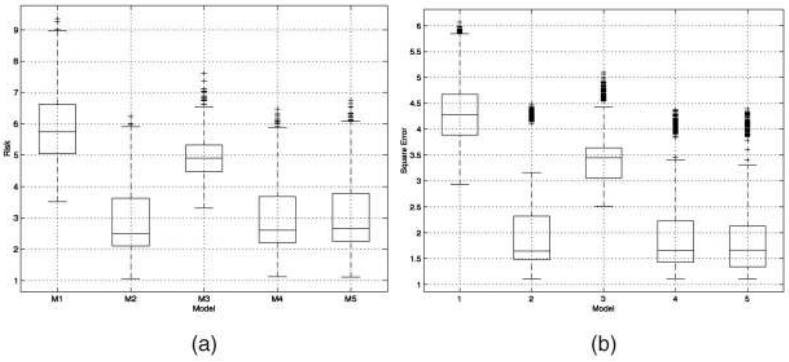

Fig. 9. Interpolation results for the moving arm sequence. (a) Bounds on prediction risk. (b) Average square error.

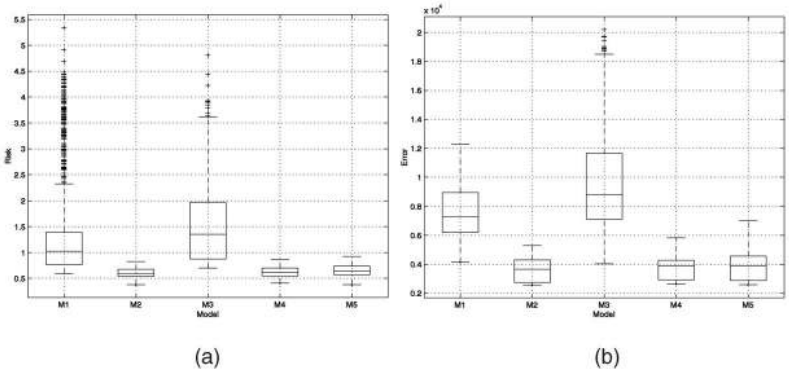

Fig. 10. Extrapolation results for the moving arm sequence. (a) Prediction risk. (b) Square error.

the motion models so that its optimal choice, M_2, yields the minimum total extrapolation error.

To further illustrate the workings of the extrapolation experiment, Fig. 11 shows predicted position of hand contours for each of the motion models. The starting hand contour, the predicted hand contour, and the actual hand contour after two frames, are displayed using pluses, triangles, and dots, respectively. It can be seen that the optimal motion model, M_2, predicts the hand contour closest to the actual hand contour position. Figs. 9 and 10 show that a simple model, M_2, explains the data as well as more complex models. Since the hand moves in a plane, M_5 is probably the true model.

6 Condition Number and the Aperture Problem

In Section 3.2, we showed how to estimate the flow parameters $\mathbf{w}$ by solving the LS problem $\min \|P\mathbf{w} - \mathbf{b}\|$. Condition number $\kappa_2(P)$ is computed as the ratio of the largest and the smallest singular values of P:

$$\kappa_2(P) = \frac{\sigma_{\max}(P)}{\sigma_{\min}(P)}. \tag{6}$$

The sensitivity in estimating $\mathbf{w}$ is roughly proportional to

$$\varepsilon(\kappa_2(P) + \rho_{LS}\kappa_2^2(P)), \tag{7}$$

where ρ_{LS} is the magnitude of the residual of the LS solution and $\varepsilon = \|\Delta\mathbf{b}\|/\|\mathbf{b}\|$ is the relative error in $\mathbf{b}$. The normal flow is computed with subpixel accuracy [12] and $\varepsilon = O(0.01)$. Since, in the examples presented here, $\rho_{LS} = O(1)$, it can be seen that condition numbers greater than 10 are undesirable. A large condition number typically corresponds to either inappropriate *scaling* of columns of P or to the *aperture problem*.

Scaling of columns P is handled as follows: When the columns of P have different scales the problem can be fixed by scaling the columns before solving the LS problem. The original problem is replaced by $\min\{\|(PG)\mathbf{y} - \mathbf{b}\|\}$. G is chosen to be a diagonal matrix whose elements are $\|P(:,i)\|^{-1}$, where $P(:,i)$ is the ith column of P. If the matrix PG is well conditioned, $\mathbf{y}$ is estimated using the LS method and $\mathbf{w} = G\mathbf{y}$ is computed. In the experiments with a moving forearm (see Figs. 1 and 2 and Section 5), typical values of $\kappa_2(P)$ are in the range of 1 to 2 for M_1, in the range of 30 to 40 for M_2 and M_3, in the range of 40 to 50 for M_4, and in the range of 3,000 to 4,000 for M_5. Scaling brings all these condition numbers below 5, i.e., $\kappa_2(PG) < 5$. This scaling procedure has been implemented and used for the experiments using real image sequences (see Section 5).

When the condition number after scaling, $\kappa_2(P)$, is still too large, it can be said that the data is not appropriate for the parameter estimation due to the aperture problem [28]. Note that the aperture problem refers to the fact that the flow cannot be estimated from the given normal flow due to the inappropriate distribution of feature points. This distribution is reflected in the data matrix P. In this case, the solution is not provided by standard LS. The problem has to be solved by minimizing the penalized risk functional

$$R_{pen}(\mathbf{y}) = \frac{1}{n}(\|(PG)\mathbf{y} - \mathbf{b}\|^2 + \mathbf{y}^T\Phi\mathbf{y}), \tag{8}$$

where Φ is a symmetric and nonnegative definite penalty matrix [9]. A reasonable choice of the penalty term is the

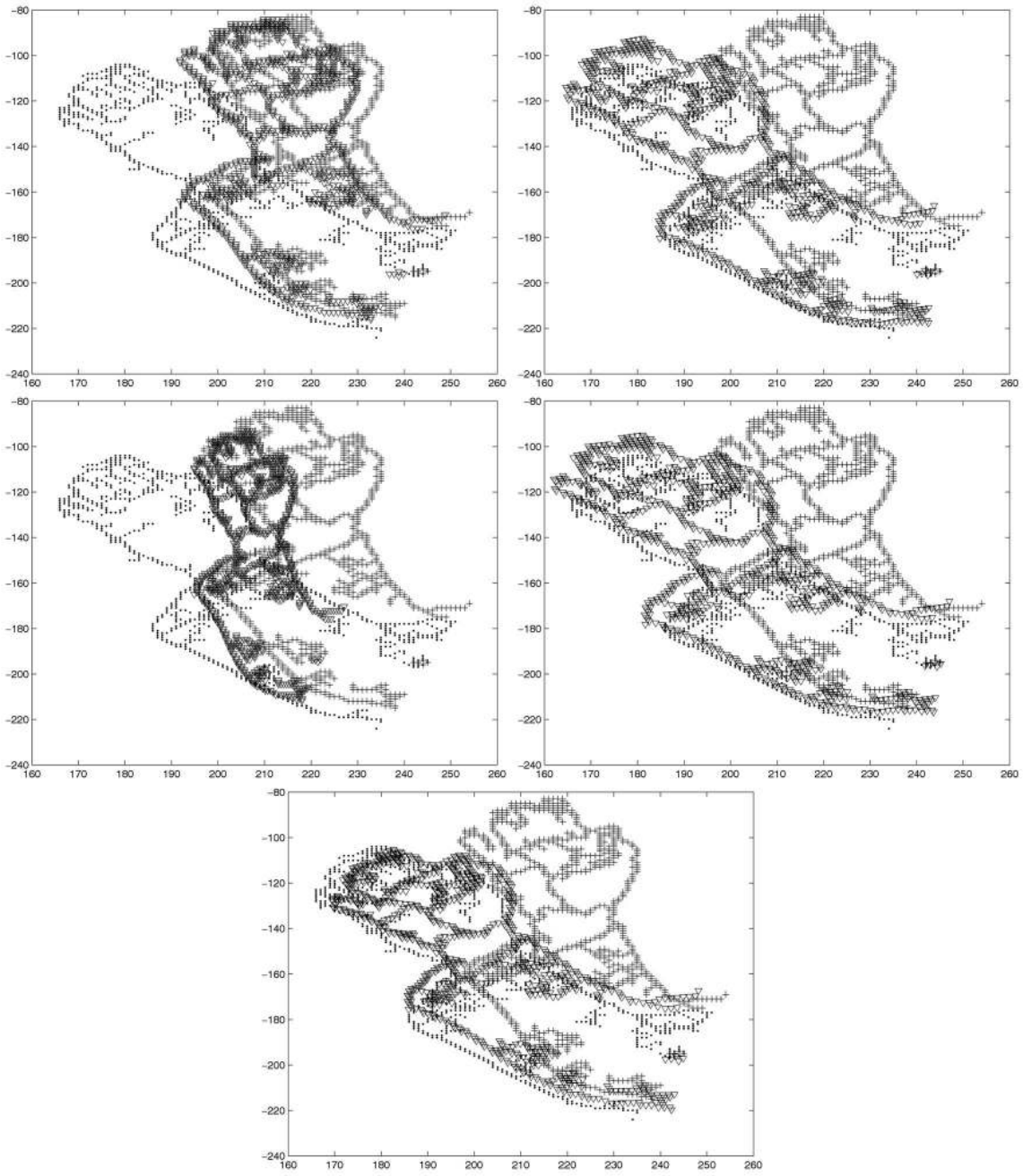

Fig. 11. Tracking results for the real image sequence. Left to right and top to bottom are shown the extrapolation results for M_1, M_2, M_3, M_4, and M_5. Each figure shows starting hand contour, predicted hand contour after two frames, and actual hand contour after two frames.

ridge regression penalty function $\Phi = \lambda I$, where I is an identity matrix [15]. Solving the following modified least squares problem minimizes $R_{pen}(\mathbf{y})$ (8):

- Create the modified data matrices

$$U = \begin{pmatrix} PG \\ \sqrt{\lambda} I \end{pmatrix}, \; v = \begin{pmatrix} \mathbf{p} \\ \mathbf{0} \end{pmatrix},$$

  where $\mathbf{0}$ is a column vector of zeroes.
- Minimize the empirical risk functional $R_{emp} = \frac{1}{n}\|U\mathbf{y} - \mathbf{v}\|$. The minimization is done by solving for y by LS method. Finally, compute $\mathbf{w} = G\mathbf{y}$.
- Compute the effective DoF for the penalized problem as $DoF = \sum_{i=1}^{m} \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$, where σ_i are the singular values of PG.

λ is chosen to make $\kappa_2(P)$ small. For illustration purposes, experiments with $\kappa_2(U) = 10$ and $\kappa_2(U) = 100$ were performed for the example shown in Fig. 12. The data comes from a small region along the arm shown in Figs. 1 and 2. In this example, the models M_1, M_2, and M_3 have small values of the condition numbers of the PG matrices. The condition numbers of data matrices PG for both M_4 and M_5, however, are very large ($> 10^{16}$) and their effective ranks are 5 and 7, respectively. The estimated empirical risk for models $M_1 - M_5$ are $(0.552, 0.015, 0.3, 0.0139617, 0.0131344)$ and the corresponding prediction risks are

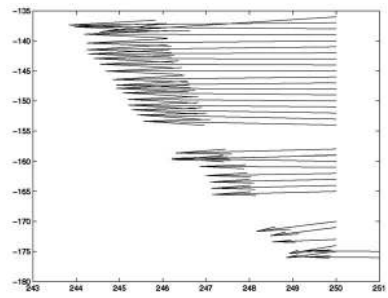

Fig. 12. Normal flow vectors from a small region along the arm in Fig. 2.

$(1.364, 0.049, 0.978, 0.059, 0.072)$. Note that, based on (2) and (3), M_2 would be chosen. However, if there were more feature points, either M_4 or M_5 could have been chosen since they have smaller empirical risk values than the other models; since they are poorly conditioned, however, penalized solutions have to be used. (Note that the penalization factor is heavy for small numbers of feature points.) The relevant results are as follows. In the case of the affine model M_4, the penalization coefficients $\lambda = (0.017, 0.17)$ result in condition numbers $(100, 10)$, empirical risk values $(0.0139622, 0.0190552)$, effective DoF values $(4.9969, 4.7258)$, and prediction risk values $(0.052, 0.068)$. The effective DoF are a crude estimate of the VC-dimension. It can be seen that the larger the penalization term, the more bias is introduced. In the case of the quadratic model (M_5), the penalization coefficients $\lambda = (0.020, 0.203)$ result in condition numbers $(100, 10)$, empirical risk values $(0.0131595, 0.022909)$, effective DoF values $(6.1726, 5.2896)$, and prediction risk values $(0.057, 0.088)$. Note that, for small penalization factors, the empirical risk goes up slightly, but the prediction risk goes down due to the lowered effective DoF.

Note that the bound on the prediction risk will be very high for higher order models. The model choices are biased toward lower order models rather than zero velocity motion. Zero velocity motion results only from LS parameter estimation errors. Lack of data, which is characteristic of the aperture problem, biases the choice involved towards selecting lower order models.

7 Conclusions

This paper describes a novel application of Statistical Learning Theory to optimal model selection, with applications to single motion estimation and tracking from small data sets of image measurements (flow). This is accomplished without using restrictive assumptions such as asymptotic settings and/or Gaussian noise. The experimental results, using both synthetic and real image sequences, demonstrate the feasibility and strengths of our approach for motion model selection using SLT. Our experimental results also show that our approach compares favorably against alternative model selection methods regarding the confidence they offer on motion estimation. The paper also shows how to address the aperture problem using SLT-based model selection under penalized regression formulation.

In practical computer vision applications, one is likely to encounter two modifications of the basic formulation for model selection and motion estimation used in this paper. Namely, the type of motion can change (at some unknown time moments)—this is known as temporal partitioning problem. Also, different portions of an image may undergo different types of motion—this is known as spatial partitioning problem. Here, the nature of the learning problem changes since the primary goal of learning/estimation becomes to partition the data into two subsets: inlier samples that will be used for estimating motion parameters and outlier samples that will be ignored or downplayed. We are presently working on those problems using methodological aspects of SLT, in general, and robust Support Vector Machines (SVM) regression, in particular.

Another aspect relevant to many CV applications—i.e., motion analysis—and presently under investigation is the need to identify/estimate several motions from a given data set. In this case, the goal of robust learning is to estimate different models for each appropriately chosen subset of the original data set. The resulting problem of *robust multiple model estimation* is intrinsically more difficult than the standard problem of single model estimation since the former involves simultaneous partitioning of the original data into several subsets and estimating a model (structure) for each subset.

Acknowledgments

The work of V. Cherkassky was supported, in part, by the US National Science Foundation grant ECS-0099906.

References

[1] H. Akaike, "Statistical Predictor Information," *Ann. Inst. of Statistical Math.,* vol. 22, pp. 203-217, 1970.
[2] Y. Aloimonos and Z. Duric, "Estimating Heading Direction Using Normal Flow," *Int'l J. Computer Vision,* vol. 13, pp. 33-56, 1994.
[3] A. Barron, "Universal Approximation Bounds for Superposition of a Sigmoid Function," *IEEE Trans. Information Theory,* vol. 39, pp. 930-945, 1993.
[4] M.J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding,* vol. 63, pp. 75-104, 1996.
[5] M.J. Black, Y. Yacoob, and S.X. Ju, "Recognizing Human Motion Using Parametrized Models of Optical Flow," *Motion-Based Recognition,* S. Mubarak and R. Jain, eds., pp. 245-269, Kluwer, 1997.
[6] M.J. Black, D.J. Fleet, and Y. Yacoob, "Robustly Estimating Changes in Image Appearance," *Computer Vision and Image Understanding,* vol. 78, pp. 8-31, 2000.
[7] K. Bubna and C.V. Stewart, "Model Selection and Surface Merging in Reconstruction Algorithms," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 895-902, 1998.
[8] K. Bubna and C.V. Stewart, "Model Selection Techniques and Merging Rules for Range Data Segmentation Algorithms," *Computer Vision and Image Understanding,* vol. 80, pp. 215-245, 2000.
[9] V. Cherkassky and F. Mulier, *Learning from Data.* Wiley, 1998.
[10] V. Cherkassky, X. Shao, F. Mulier, and V. Vapnik, "Model Selection for Regression Using VC-Generalization Bounds," *IEEE Trans. Neural Networks,* vol. 10, pp. 1075-1089, 1999.
[11] P. Craven and G. Wahba, "Smoothing Noisy Data with Spline Functions," *Numerische Math.,* vol. 31, pp. 377-403, 1979.
[12] Z. Duric, F. Li, Y. Sun, and H. Wechsler, "Using Normal Flow for Detection and Tracking of Limbs in Color Images," *Proc. Int'l Conf. Pattern Recognition,* 2002.
[13] J.H. Friedman, "An Overview of Predictive Learning and Function Approximation," *From Statistics to Neural Networks: Theory and Pattern Recognition Applications,* V. Cherkassky, J.H. Friedman, and H. Wechsler, eds., NATO ASI Series F, vol. 136, Springer, 1994.
[14] F. Girosi, "Regularization Theory, Radial Basis Functions and Networks," *From Statistics to Neural Networks: Theory and Pattern Recognition Applications,* V. Cherkassky, J.H. Friedman, and H. Wechsler, eds., NATO ASI Series F, v. 136, Springer, 1994.
[15] G.H. Golub and C.F. Van Loan, *Matrix Computation,* third ed. John Hopkins Univ. Press, 1996.

[16] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust Statistics: An Approach Based on Influence Functions.* Wiley, 1986.
[17] P.J. Huber, *Robust Statistics.* Wiley, 1981.
[18] W. Lee, P. Bartlett, and R. Williamson, "Efficient Agnostic Learning in Neural Networks with Bounded Fan-In," *IEEE Trans. Information Theory,* vol. 42, pp. 2118-2132, 1996.
[19] P. Meer, D. Mintz, and A. Rosenfeld, "Robust Regression Methods for Computer Vision: A Review," *Int'l J. Computer Vision,* vol. 6, pp. 59-70, 1991.
[20] P. Meer, C.V. Stewart, and D.E. Tyler, "Robust Computer Vision: An Interdisciplinary Challenge," *Computer Vision and Image Understanding,* vol. 78, pp. 1-7, 2000.
[21] T.M. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding,* vol. 81, pp. 231-268, 2001.
[22] S. Nayar and T. Poggio, *Early Visual Learning,* S. Nayar and T. Poggio, eds., Oxford Univ. Press, 1996.
[23] T. Poggio and C.R. Shelton, "Machine Learning, Machine Vision, and the Brain," *AI Magazine,* vol. 20, no. 3, pp. 37-55, 1999.
[24] B.D. Ripley, *Pattern Recognition and Neural Networks.* Cambridge Univ. Press, 1996.
[25] P.H.S. Torr, "An Assessment of Information Criteria for Model Selection," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 47-53, 1997.
[26] R. Shibata, "An Optimal Selection of Regression Variables," *Biometrika,* vol. 68, pp. 45-54, 1981.
[27] G. Schwartz, "Estimating the Dimension of a Model," *Ann. Statistics,* vol. 6, pp. 461-464, 1978.
[28] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision.* Prentice Hall, 1998.
[29] V.N. Vapnik, *Statistical Learning Theory.* Wiley, 1998.
[30] V.N. Vapnik, *The Nature of Statistical Learning Theory,* second ed. Springer Verlag, 1999.

Harry Wechsler received the PhD degree in computer science from the University of California, Irvine, in 1975. He is presently a professor of computer science and the director for the Center of Distributed and Intelligent Computation at George Mason University (GMU) http://cs.gmu.edu/~wechsler/DIC_Center/. His research, in the field of intelligent systems, has been in the areas of perception: automatic target recognition, computer vision, signal and image processing; machine intelligence: data mining, neural networks, pattern recognition, and statistical learning theory; evolutionary computation: animats and swarm intelligence, and genetic algorithms; biometrics: face recognition, performance evaluation, and surveillance; and human-computer intelligent interaction: hand gesture recognition, smart interfaces, and video tracking and interpretation of human activities. He was the director for the NATO Advanced Study Institutes (ASI) on "Active Perception and Robot Vision" (Maratea, Italy, 1989), "From Statistics to Neural Networks" (Les Arcs, France, 1993), and "Face Recognition: From Theory to Applications" (Stirling, UK, 1997). He served as cochair for the International Conference on Pattern Recognition held in Vienna, Austria, in 1996. He has authored more than 200 scientific papers and the book *Computational Vision* (Academic Press, 1990). He was the editor for *Neural Networks for Perception*, vols. 1 and 2, (Academic Press, 1990), and the principal coeditor for *Face Recognition: From Theory to Applications* (Springer-Verlag, 1998). He was elected as an IEEE fellow in 1992 and as an IAPR (International Association of Pattern Recognition) fellow in 1998.

Zoran Duric received the PhD degree in computer science from the University of Maryland at College Park in 1995. He is an associate professor of computer science at George Mason University. From 1982 to 1989, he was a member of the research staff in the Division for Vision and Robotics, Energoinvest Institute for Control and Computer Science, Sarajevo. He was also affiliated with the Electrical Engineering Department of the University of Sarajevo. From 1995 to 1997, he was an assistant research scientist at the Machine Learning and Inference Laboratory at George Mason University and at the Center for Automation Research at the University of Maryland. From 1996 to 1997, he was also a visiting assistant professor at the Computer Science Department of George Mason University. He joined the faculty of George Mason University in the Fall of 1997 as an assistant professor of computer science. His research interests include computer vision, human-computer interaction, motion analysis, machine learning in vision, and information hiding in images. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Fayin Li received the BS degree in electrical engineering from Huazhong University of Science and Technology, China, in 1996, and the MS degree in computer science from the Institute of Automation, Chinese Academy of Sciences, China, in 1999. He is currently working toward the PhD degree at George Mason University, Fairfax, VA. His research interests include automatic gesture recognition, face recognition, video tracking and surveillance, image processing, pattern recognition, and model selection.

Vladimir Cherkassky received the PhD degree in electrical engineering from the University of Texas at Austin in 1985. He is a professor of electrical and computer engineering at the University of Minnesota. His current research is on methods for predictive learning from data, and he has coauthored a monograph *Learning From Data* published by Wiley in 1998. He has served on editorial boards of several journals including *IEEE Transactions on Neural Networks*, *Neural Networks*, and *Neural Processing Letters*. He served on the program committee of major international conferences on Artificial Neural Networks. He was Director of NATO Advanced Study Institute (ASI) From Statistics to Neural Networks: Theory and Pattern Recognition Applications held in France, in 1993. He presented numerous tutorials and invited lectures on neural network and statistical methods for learning from data. He is a senior member of the IEEE and a member of the IEEE Computer Society.