ELVIS: Eigenvectors for Land Vehicle Image System[1]

John Hancock and Chuck Thorpe
jhancock@ri.cmu.edu, cet@ri.cmu.edu
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

ELVIS (Eigenvectors for Land Vehicle Image System) is a road-following system designed to drive the CMU Navlabs. It is based on ALVINN, the neural network road-following system built by Dean Pomerleau at CMU. ELVIS is an attempt to more fully understand ALVINN and to determine whether it is possible to design a system that can rival ALVINN using the same input and output, but without using a neural network.

Like ALVINN, ELVIS observes the road through a video camera and observes human steering response through encoders mounted on the steering column. After a few minutes of observing the human trainer, ELVIS can take control. ELVIS learns the eigenvectors of the image and steering training set via principal component analysis. These eigenvectors roughly correspond to the primary features of the image set and their correlations to steering. Road-following is then performed by projecting new images onto the previously calculated eigenspace. ELVIS architecture and experiments will be discussed as well as implications for eigenvector-based systems and how they compare with neural network-based systems.

1.0 Introduction

Researchers at CMU have been working on autonomous driving systems for nearly a decade. One of the most successful robot road-following systems is ALVINN, a simulated neural network built by Dean Pomerleau at CMU. ALVINN uses a color video camera mounted above the passenger compartment of the vehicle to watch the road. A steering wheel encoder allows ALVINN to observe human steering. After observing the road and the human steering responses for approximately two minutes, ALVINN can operate the steering wheel to follow the road on its own.

Unfortunately, why ALVINN works has remained somewhat of a mystery. ALVINN has several components which contribute to its success: careful generation of training image sets, image subsampling, color balancing, output representation, and the neural network. One model of how the neural network in ALVINN works is that the first set of weights between the input and hidden layers learns a reduced representation of the training set representing the important image features. The weighted sums at the output calculate the steering based on those image features present in a new image. ELVIS (Eigenvectors for Land Vehicle Image System) seeks to verify this model: it calculates the eigenvectors of the training set which form an explicit reduced representation of that training set. Projection of a new image onto the eigenspace produces a steering output. ELVIS also attempts to answer the question of whether the neural network itself is the key to ALVINN's success, or whether it is possible to design a system that can rival ALVINN, using the same input and output, but without using a neural network; ELVIS replaces ALVINN's neural network with an eigenvector representation that uses the same inputs and outputs as ALVINN.

In this paper we first introduce the components and structure of ELVIS. The training and processing methods that ELVIS uses to drive the vehicle are then explained. Section three describes some of the experiments we have performed with the video preprocessors used with ELVIS to improve its performance. We finish with a discussion of the merits of ALVINN and ELVIS. We explain in what types of scenarios it is possible to replace a neural-network based system like ALVINN with an eigenvector system like ELVIS.

2.0 ELVIS

2.1 Common components of ALVINN and ELVIS

Both ALVINN and ELVIS are designed to interpret video images of roads and produce steering commands.

1. For the complete version of this paper, see the online technical report on the WWW: "ftp://reports.adm.cs.cmu.edu/usr/anon/robotics/CMU-RI-TR-94-43.ps.Z" [3].

0-8186-7108-4/95 $4.00

The method each uses to calculate steering output given the same image is different, but the overall structure of the two systems is the same, and ELVIS was designed to use some of the ALVINN modules. A block diagram of ELVIS is shown below.

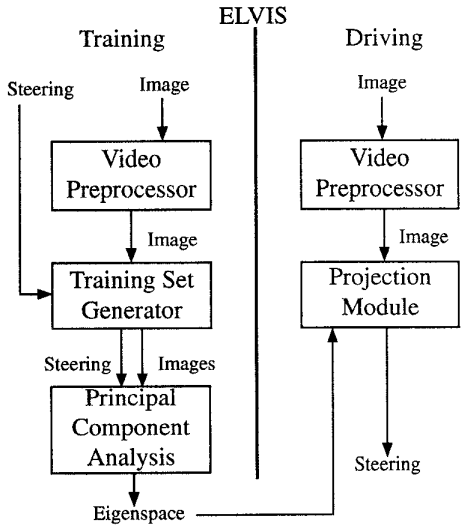

Figure 1. Block diagram of ELVIS. ELVIS replaces the ALVINN neural network with a principal component analysis in the training phase and with a projection module in the driving phase.

In the ALVINN system, the three-band RGB color images produced by the camera are preprocessed to produce a single-band image. The color transformation is performed to reduce the amount of data, and more importantly, to enhance the image features important for road-following. Each pixel is normalized to have a value between 0 and 1. The normalized value is given by:

$$v = [\alpha \times B/255] + [(1-\alpha) \times B/(R+G+B)]$$

where R,G, and B are the raw red, green, and blue values for a given pixel, and α is a weighting factor between 0 and 1. The partial normalization by intensity provides some tolerance to lighting variation within a given image, helping to filter out variations caused by shadows. Empirically, it has been determined that the blue band contains the most useful contrast for road following.

To reduce the computational expense of processing large images, the video preprocessing must reduce the dimensions of the 480 x 512 digitized camera image. The neural network in ALVINN uses a 30 x 32 input image layer, and this has been the resolution typically used for ELVIS as well. A small percentage of the pixels within each region in the original image is randomly sampled and averaged to produce the reduced image pixels.

The output for each system is a 50 element vector in which each element represents the strength of votes for a particular steering direction. During training, the correct steering direction is represented by a gaussian set of votes within the output vector, centered at the actual steering direction. This output representation has several advantages over using a single-valued output to indicate steering direction. First, the single-valued output cannot represent both the network decision and the network confidence in that decision. A single-valued output for a completely recognized scene indicating a shallow right turn might well be indistinguishable from the output for a partially recognized scene which calls for a hard right turn[8]. This is especially a problem for ALVINN which does not have an independent method of computing a confidence level. A single-valued output representation also would not allow an ALVINN hidden unit or an ELVIS eigenvector to vote for more than one steering direction[8] which is important since an eigenvector does not necessarily correspond to one feature or one type of image. Finally, the gaussian output results in a robust, distributed system so that even if one output unit fails, it is still possible to drive.

There are several pitfalls in the generation of training image sets. The first is bias: if the training used only images from left turns, the system would learn that always turning left minimizes output errors. The images selected must be balanced, including all ranges of steering positions with backgrounds or off-road areas that reasonably span the space of expected driving situations. Both ELVIS and ALVINN will do poorly if trained on a paved road, and then expected to drive on a dirt road. The other problem in training is that, in general, the human trainer drives too well so the system is never shown how to recover from minor steering errors. The solution both systems use is to create derived training images from the actual images. A geometric transform is applied to the input image to rotate or shift it slightly as if the vehicle were slightly off the desired path. The steering angle is corrected correspondingly. This provides a much broader training set for ALVINN and ELVIS. The geometric transforms for image and steering are more fully described by Pomerleau[7].

2.2 Training

During training, ELVIS learns eigenvectors of the image and steering training set. The eigenvectors are a set of basis vectors that guarantee the best linear image reconstruction, on average, given limited representational power. These eigenvectors not only represent the principal features of the image set but also these features' correlations to steering (see Figure 2 for some example eigenvectors). Given a new image we use the eigenvectors to produce the best reconstruction of the image and its fea-

tures. Since the eigenvectors also tell us how these features correlate to the steering, this allows us to compute the proper steering position. The principal eigenvectors model large, common features which we assume are useful for driving.

We represent each two-dimensional image with B color bands, R rows, and C columns, as an $n = B \times R \times C$ element one-dimensional vector. For training purposes, we add the steering vector elements to the end of the image vector. This image/steering vector combination is a training vector. The number of elements in each training vector is N = n + d where d is the number of steering units. For a monochrome 30 by 32 image with 50 output units, N is 1010. Given a set of M (typically M = 400) training vectors $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3,...\mathbf{v}_M$, the average of the training set is defined by $\mathbf{a} = (1/M)\Sigma\mathbf{v}_i$. By subtracting the average vector from each vector, we can obtain the difference vectors $\Delta_i = \mathbf{v}_i - \mathbf{a}$. Given the Δ_i we form the covariance matrix $\mathbf{C} = \mathbf{A}\mathbf{A}^T$ where $\mathbf{A} = [\Delta_1, \Delta_2, ...]$.

The covariance matrix **C** is N by N which is very large, and it is computationally expensive to find its eigenvectors. There is a way to greatly reduce the amount of computation. Since there are only M images, there can be at most M independent eigenvectors. Since M is generally much less than N, it is advantageous to find the principal components of the M by M matrix $\mathbf{U} = \mathbf{A}^T\mathbf{A}$ and then convert them into the eigenvectors of matrix **C** rather than calculate them directly from **C**. To convert the eigenvectors of **U** to eigenvectors of **C**, we simply multiply each vector by **A**: if **x** is an eigenvector of **U**, then $\mathbf{A} \bullet \mathbf{x}$ is an eigenvector of **C** [5],[10]. It is important to note that the eigenvalue of the eigenvector will be the same in either **U**-space or **C**-space, so that the vectors will be found in the same order whether we use **U** or **C**.

We then find ten to fifteen eigenvectors of the matrix **U** with the largest eigenvalues. This is done by the power method. The vector $\mathbf{U}^n \bullet \mathbf{x}$ for an arbitrary vector **x** will tend to converge towards the eigenvector of **U** with the largest eigenvalue as n becomes large. In practice, sufficient convergence occurs for n between 5 and 50, depending on the eigenvalues of the eigenvector and its nearest competitors. By orthogonalizing the matrix **U** with respect to this eigenvector, we can repeat the process to obtain the orthogonal eigenvector with the next highest eigenvalue and so on. The orthogonalized matrix $\mathbf{U}_{i+1}$ is simply $\mathbf{U}_i - \lambda\mathbf{x} \bullet \mathbf{x}^T$, where **x** is the eigenvector and λ is its eigenvalue. This orthogonalization process does not affect the other eigenvectors, but removes the dimension of the eigenspace that lies along **x**. Although the power method is not the most accurate method for calculating eigenvectors, we have found that ELVIS is not affected by its inaccuracies. The power method has the advantage that it is simple to implement, and faster than many other methods.

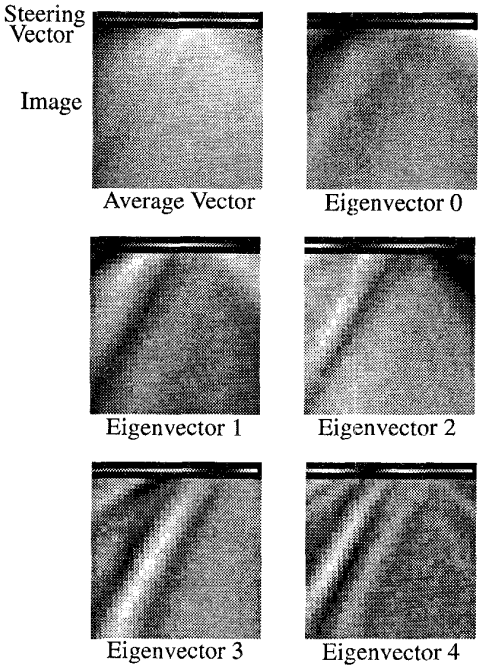

Figure 2. These eigenvectors were formed with the ALVINN color-balanced pre-processing method from a batch of images taken on Schenley Drive (a two-lane road). The bands towards the upper-left of each eigenvector represent the location of the lane markers.

2.3 Road-Following

Once ELVIS finds the principal eigenvectors, $\mathbf{e}_i$, and the average training vector, **a**, of the roadspace, it can steer on its own. To drive, ELVIS takes a new image, **x**, performs the same preprocessing as in training, and then projects it onto the eigenspace formed by the principal eigenvectors (usually ten of them). To project the image onto the eigenspace, we perform the calculation:

$$\mathbf{v} = \mathbf{a} + \sum_{i=1}^{10} ((\mathbf{x} - \text{image}(\mathbf{a})) \bullet \text{image}(\mathbf{e}_i))\,\mathbf{e}_i$$

where image(**z**) is simply the image portion of the composite vector **z**.

In this way, a composite vector, **v**, is formed which consists of both the reconstructed image and the steering vector. A single steering command is then computed from the steering subvector (the last 50 elements of **v**) by calculating the center of mass of the peak of activation surrounding the output unit with the highest activation level. This steering command is then sent to the vehicle control-

ler module. Using the center of mass of the activation rather than the most active output unit allows for sub-unit steering resolution, thus improving driving accuracy[7].

Besides knowing the proper steering direction, the system should also have a measure of the system's confidence in that calculation. A reliability estimate is important because it allows the system to disregard the new steering instruction if confidence is low, and it lets the user know that the system should be retrained if the average confidence becomes low. One simple way to measure reliability is to compute the sum-squared difference error between the image and its reconstruction. This image reconstruction error measures how closely the reconstructed image resembles the original. If the error is low, then the new image must resemble some of the images in the ELVIS training set, and so confidence should be high. If the error is high, then the image does not match closely with training set images and so the system confidence in the steering direction should be low.

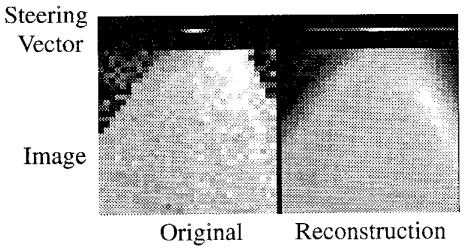

Figure 3. Original image and steering vector and their reconstructions

The reconstructed image (Figure 3) tends to be smoother than the original, generally capturing the essence of the geometry of the road while not reconstructing noise or fine details. The reconstructed steering vector also tends to be flattened. This can result in rather broad peaks. Wider peaks can cause steering errors of up to three units (out of 50), but these errors are manageable. More difficult to handle are multiple-peaked responses. Although these occur infrequently, they can lead to drastically incorrect steering vectors if ELVIS chooses the wrong peak (ELVIS computes the center of the peak with the largest amplitude). Since road-following is a continuous process that does not require sudden changes in steering (except at intersections which ALVINN and ELVIS are not equipped to handle), it would be possible to create a heuristic that would choose the peak that is closest to the present steering direction or place a threshold on the change in steering direction. This may help in avoiding the multiple peak problem. Of course, the independence of the individual results is one of the strengths of the system and removing this independence could introduce other problems.

3.0 Experiments

The video preprocessing is an important factor in ALVINN's success. ALVINN would never learn to drive if the road or road features were indistinguishable from the rest of the image. As described below, tests using simulated data with ELVIS revealed that the ALVINN video preprocessing did leave room for improvement. As an attempt to both improve ELVIS accuracy and further understand how and why ALVINN works, we measured ELVIS performance using a variety of image types and video preprocessing parameters. More details are included in the technical report, but a short summary of our findings is given below.

3.1 Simulated Data

ELVIS was first tested on synthesized road images. The simulator produced monochrome road images with appropriate steering vectors calculated by geometric transformations. Pixels were perfectly classified; road pixels were white and non-road pixels were black. By testing on perfectly classified data, we could show ELVIS concept viability and establish a baseline performance level against which we could judge the quality of various preprocessing techniques. ELVIS performance with the simulated data was good, with the standard deviation of the steering direction error being less than 0.8 steering units out of 50. After 2.5 m of travel, (the maximum distance travelled between images) this error leads to a displacement of 0.9 cm from the road center and an error of 0.41° in heading. Given real data, a perfect preprocessor would provide a clear separation between road and non-road such as that present in the simulated images.

3.2 ALVINN Color-balanced Images

Measuring ELVIS performance on image sets using the ALVINN color-balanced preprocessor, as described previously, provided the second baseline for our tests. Steering results were good, but not as accurate as with simulated data. The standard deviation of the steering direction was several times that found with simulated data, typically around 3.0 steering units out of 50 for 10 eigenvectors (an error in curvature of 10.7 km^{-1}). The ALVINN system itself performs somewhat better -- typically producing errors with a standard deviation of 2.7 units out of 50. We expected to surpass the performance of the ALVINN color-balanced scheme by providing ELVIS with more information.

3.3 Color Images

Providing color images seemed an obvious first step

to boost ELVIS performance. We hypothesized that the use of 3-band color images would improve the accuracy of ELVIS by providing ELVIS with important cues to distinguish between road and off-road pixels. Color provides humans with many obvious cues for driving. Green grass and yellow and white lane markers contrast well with black asphalt. We tested ELVIS with a variety of color combinations. First we supplied ELVIS with 3-band RGB data. However, performance declined significantly when we replaced the ALVINN pre-processed images with subsampled RGB color images. Although reconstruction of the input images themselves was quite good, the reconstruction of the steering vectors was poor. Often broad peaks or multiple peaks occurred in the reconstructed steering vectors, causing large steering errors. Apparently, the RGB values were not well correlated with the steering values.

We thought that perhaps ELVIS needed color to be provided in combinations which would better distinguish between road and non-road. Next we tried ELVIS with intensity, saturation, and hue (ISH) information as defined by Ballard and Brown[1]. ELVIS performed miserably with intensity images (as expected), but performed reasonably with saturation and hue information, though still not as well as with ALVINN pre-processed images.

The third color representation we tried was one proposed by Yuichi Ohta[6]. Ohta performed trials to derive linear color features with large discriminant power for segmenting outdoor color scenes, and found that all his test images could be segmented near-optimally if he used a transformed color space. The 3 axes of this space were intensity, (R-B)/2, and (2G-R-B)/4. Results of using ELVIS with the images in these transformed color coordinates were better than with either RGB or ISH information, although still worse than ALVINN preprocessing. As in the ISH images, results were improved slightly when intensity information was dropped altogether. Additionally, performance was only slightly worse if we dropped the third band, (2G-R-B)/4, as well. (R-B)/2 segmented the images fairly well because the road pixels tended to have more blue than red, while background pixels tended to be more reddish. This is similar to the way in which the ALVINN preprocessor and the Martin Marietta ALV road-follower function[9].

3.4 Other Experiments

Several other experiments were performed in which the vector size was altered. The digitized images are normally reduced to 30 by 32, but this coarse resolution often blurs lane markings, which provide important cues for driving along multi-lane roads. To improve ELVIS performance, especially for multi-lane roads, we tested higher resolution 60x64 images in both ALVINN color-balanced and RGB modes. Performance decreased, however, on the one-lane road and only improved slightly on the two-lane road.

Increasing the number of output steering units seemed to decrease the output accuracy. Why this is so is not entirely clear. It is possible that the greater the number of steering units, the greater the interference these output units cause in the principal component analysis in terms of finding image features to rely upon.

To test whether we could improve performance by discarding portions of the image which were not highly correlated to the output, the ELVIS images were modified so that a given portion in each image was thrown out before training and testing. For the case of deleting the bottom quarter of each image, ELVIS steering accuracy improved. As predicted, elimination of the top portion of the image led to a dramatic decrease in performance, since this is the portion of the road that a driver uses most in determining the correct steering position.

4.0 Discussion and Conclusions

ALVINN's success provides no guarantee that ELVIS will work. First, since ELVIS is calculating linear functions of the inputs, it assumes that non-linear combinations are not required. Second, the eigenvector decomposition finds the best representation of the covariances among all the data. This will give the best linear reconstruction of the data, but will not necessarily find the best mapping from inputs to outputs. For a general data set, there is no reason to assume that globally similar inputs should produce globally similar outputs. For this reason, we should expect ALVINN to perform better than ELVIS. While ELVIS principal component analysis minimizes the total error in the image reconstruction and steering vector, ALVINN directly minimizes the steering output alone. Since there are bound to be some small areas of the image which are not highly correlated with the steering output, ALVINN is able to produce better steering results.

It is possible to reformulate ELVIS so that it minimizes output error. One way is to zero those portions of the covariance matrix that represent image-image or steering-steering correlations. In this way, ELVIS learns only information that correlates steering output to image input. This has been tried, but steering results became worse rather than better. Evidently, the correlations between image pixels are important to the driving task in general, though they may not improve results on the training set. A second, and perhaps better, method would be to learn a least-squares mapping matrix of image input to steering output. This would correspond to learning a 50 by 960 matrix. This matrix might be further broken down into

eigenvector components. Finding the least-squares mapping was performed and is described in the complete version of this paper.

ALVINN has additional advantages. First, ALVINN is faster at run-time. ALVINN requires approximately 40% of the calculations of ELVIS (using 10 eigenvectors). Furthermore, since ALVINN training is an incremental rather than batch process, it is simple to train until performance is satisfactory, then stop.

In considering the use of eigenvectors for other applications in the place of a neural network, the scenario must be one in which the input and output are highly correlated. The output should depend on large features in the input, and the output representation should be smooth so that a linear solution will be adequate. An output representation such as turn radius might cause problems: near "straight ahead", the turning radius jumps from postive infinity to negative infinity. Representing such an output might require mechanisms beyond the linear calculations of ELVIS.

For some types of applications, the primary advantage of ELVIS is its simplicity and lack of pre-defined structure which does not distinguish between input and output. Since it makes no difference to ELVIS which portion of the vector is missing, the eigenvectors may function as an associative memory. Given most of the data, a reconstruction of the missing data is possible. We calculate the dot products of the available portions of the vector with the corresponding pieces of the eigenvectors. This projects the vector onto the eigenspace, giving us an approximate reconstruction of all of the information. Turk and Pentland demonstrated with their eigenface work that it was possible to recover an approximation of a person's face (and recognize it) even when a significant portion of it was occluded[10]. The concept of using eigenvectors to recover missing data could extend to the case where there was not just image information, but multiple types of distributed, highly-correlated information, stored in a single entry or vector. A traditional neural network architecture, however, would not work. The neural network approach would demand that we know a priori which (and how much) information was being provided and which was missing; input and output are more restrictive concepts.

In short, the neural network is well-suited to this task, but it is not the most critical part of ALVINN. The network does not differ greatly from the eigen calculations. Much of ALVINN's power comes from the robust representation, careful training set generation, and a good choice of video preprocessing. There is room for improvement in the video preprocessing, and it should be possible to provide a better separation of road and non-road. However, if a better segmentation is achieved, it might be more advantageous to approach the problem from a model-based method (such as in Crisman's SCARF and UNSCARF systems[2]) rather than using a neural network or eigenvector system.

5.0 Acknowledgements

Many thanks go to Dean Pomerleau and Todd Jochem for their help with the ALVINN code, both on and off the vehicle. Keith Gremban was helpful in the selection of an algorithm for computing the eigenvectors. The authors also wish to thank Shumeet Baluja, Martin Martin, and Garth Zeglin for their helpful comments. Support for this research came from: DARPA, under contracts "Peception for Outdoor Navigation" (contract number DACA76-89-C0014, monitored by the US Army Topographic Engineering Center) and "CMU Autonomous Ground Vehicle Extension" (contract number DAAE07-90-C-R059, monitored by TACOM); DOT/National Highway Traffic Safety Administration, "Run-Off-Road Counter Measures", (contract number DTNH22-93-C-07023,); NSF, "Annotated Maps for Autonomous Underwater Vehicles" (contract number BCS-9120655); and the Dept. of Transportation, "Automated Highway System."

6.0 References

[1]D. Ballard and C. Brown. *Computer Vision.* Prentice-Hall, 1982.

[2]J. Crisman. *Color Vision for the Detection of Unstructured Roads and Intersections.* Ph.D. Thesis, Electrical and Computer Engineering Department, Carnegie Mellon University, 1990.

[3]J. Hancock and C. Thorpe. ELVIS: Eigenvectors for Land Vehicle Image System. Carnegie Mellon Technical Report CMU-RI-TR-94-43, December 1994.

[4]T. Jochem and S. Baluja. *Massively Parallel, Adaptive, Color Image Processing for Autonmous Road Following.* Carnegie Mellon Technical Report CMU-RI-TR-93-10, May 1993.

[5]Murase and S. Nayar. *Parametric Eigenspace Representation for Visual Learning and Recognition.* Columbia University Technical Report CUCS-054-92.

[6]Y. Ohta. *Knowledge-Based Interpretation of Outdoor Natural Color Scenes.* Pitman, 1985.

[7]D. Pomerleau. Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving. In *Robot Learning*, J. Connell and S. Mahadevan (eds.), Kluwer Academic Publishing, 1993.

[8]C. Thorpe. Machine Learning and Human Interface for the CMU Navlab. In *Computer Vision for Space Applications Proceedings*, September, 1993.

[9]M. Turk, D. Morgenthaler, K. Gremban, and M. Marra. VITS -- A Vision System for Autonomous Land Vehicle Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, 1988.

[10]M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience,* Vol. 3:1, pp. 71-86, 1991.