6/93
7/93 NASA Technical Memorandum 103981

176444
p. 26

Neural Networks Application to Divergence-Based Passive Ranging

Yair Barniv

December 1992

Quick Release – This Technical Memorandum is an unedited report. It is being released in this format to quickly provide the research community with important information.

National Aeronautics and
Space Administration

NASA Technical Memorandum 103981

Neural Networks Application to Divergence-Based Passive Ranging

Yair Barniv, Ames Research Center, Moffett Field, California

December 1992

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

CONTENTS

SUMMARY

The purpose of this report is to summarize the state of knowledge and outline the planned work in divergence-based/neural-networks approach to the problem of passive ranging derived from optical flow. Work in this and closely related areas is reviewed in order to provide the necessary background for further developments. New ideas about devising a monocular passive-ranging system are then introduced. It is shown that image-plan divergence is independent of image-plan location with respect to the focus of expansion and of camera maneuvers because it directly measures object's expansion which, in turn, is related to the time-to-collision. Thus, a divergence-based method has the potential of providing a reliable range complementing other monocular passive-ranging methods which encounter difficulties in image areas close to the focus of expansion. Image-plan divergence can be thought of as some spatial/temporal pattern. A neural network realization has been chosen for this task because neural networks have generally performed well in various other pattern recognition applications. The main goal of this work is to teach a neural network to derive divergence from the imagery.

ACRONYMS USED IN TEXT

BP	- Back-Propagation
FOE	- Focus of Expansion
FOV	- Field of View
FLIR	- Forward-Looking Infra-Red
FT	- Fourier Transform
INU	- Inertial Navigation Unit
LOS	- Line of Sight
NNet	- Neural Network
OF	- Optical Flow
PCA	- Principal-Component-Analysis
PSF	- Point-Spread-Function
SNR	- Signal-to-Noise Ratio
TBD	- Track Before Detect
TTC	- Time To Collision
3-D	- 3-Dimensional

1 INTRODUCTION

Passive ranging is an area of considerable interest for applications such as obstacle avoidance for rotorcraft nap-of-the-earth navigation and spacecraft landing. Two main passive-ranging methods can potentially be employed for this purpose; one based on motion and the resulting image-plane optical flow (OF), and the other based on stationary stereo. Both methods can be thought of as special cases of a more general triangulation method known in the literature as "bearing-only" or "direction-of-arrival" (e.g. [1, 2, 3, 4]). Although this paper concentrates on monocular OF-based ranging, it is believed that most of the ideas also apply to stereo—either stationary or combined with motion (see [5, 6]).

The motion of an imaging sensor causes each imaged point of the scene to correspondingly describe a time trajectory on the image plane. The trajectories of all imaged points are reminiscent of a flow (e.g., of liquid) which may explain the term "optical flow." The OF thus consists of the sequence of angular (or projectional) measurements collected from different locations along the vehicle's flight trajectory with respect to all points in the field of view (FOV). A forward-looking imaging sensor, such as a TV camera or a FLIR, is typically used to record the optical flow. The various methods of extracting depth information from the OF can be categorized on several levels of properties. These levels are outlined in the following with the purpose of providing a familiar context for the present work.

1. Passive ranging methods can be divided into three distinct classes: object-based, feature-based, and field-based. Object-based approaches make use of assumptions which are pertinent to objects, e.g., that all points of an object share the same depth and gray level, and that an object is enclosed by some contour. Their main drawback is the need to define or identify objects between successive frames. Feature-based approaches take advantage of "interest points" in the scene such as edges, corners, or any abrupt change in gray levels. No association is necessary between interest points and actual objects. Field-based approaches regard the scene as a continuum, that is, they base ranging on local information alone. They, thus, do not require any reference to, or identification of objects.

2. Categorization can be made based on the assumptions inherent to the method—irrespective, or in addition, to whether they are feature-, object- or field-based. Assumptions, explicit or implicit, are always made with every passive ranging method. All methods rely on the basic underlying assumption that the scene and its illumination sources are temporally constant (see [7]). Object-based methods necessarily assume gray-level or/and texture constancy within each object as well as that all points belonging to the same object share the same range.

3. Passive-ranging methods can be divided into classes according to whether they work recursively or in batch, i.e., they work in a filtering or smoothing mode. Calculating

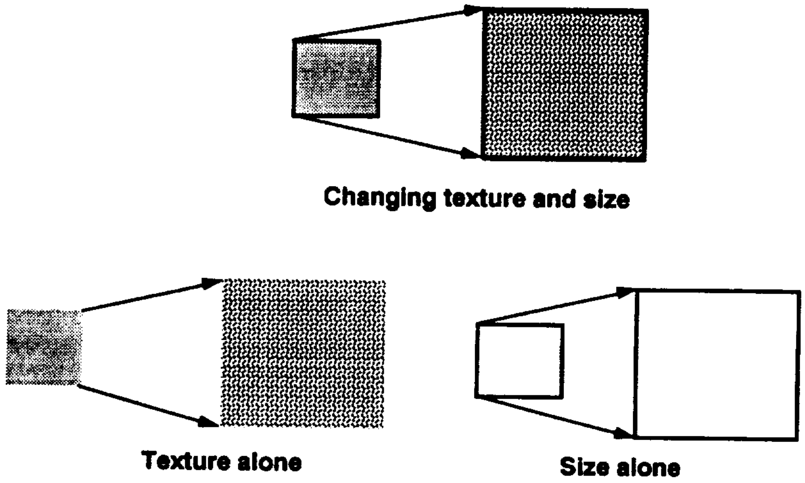
optical flow is akin to trajectory estimation for every point in the scene. As such, trajectory estimation can be performed using either a detect-then-track method, e.g., Kalman filtering—such as in [8, 9], or using a track-before-detect (TBD) method, e.g., Dynamic Programming—such as in [10, 11, 12]. The probability-of-detection divided by the probability-of-false-alarm is directly related to the signal-to-noise (SNR) ratio of the data. In the case of detect-then-track methods the SNR is that of the first measurement (which oftentimes is obtained from the first and second images), whereas with TBD methods, the SNR is that of the *combined* imagery set (which is typically in the order of 15 images). Another way to say the same is that detect-then-track methods use a very short integration time for the first detection, whereas TBD methods use a much longer integration time for detection (and tracking). When the data are noisy, as is always the case, the TBD approach is obviously advantageous.

4. Lastly, categorization can be made based on the type of information each particular method makes use of. The OF at any given point consists of two kinds of motion: a translational motion and a local divergence or expansion. Most methods make use of the motion away from the focus of expansion (FOE) of the features, objects, or some local surroundings—thus ignoring the other part of the OF information. In this work I will discuss methods of extracting depth information from the *divergence* of the OF; this can be done on a field or object basis. The divergence, which is by definition a local, or field operation, is independent of image-plane location and vehicle's maneuvers because it measures the local expansion at any given point. Usually, for points far from the FOE, the translational motion is much more pronounced than the divergence, so that ignoring the divergence does not incur too much of a loss. However, in areas close to the FOE, the opposite is true; the translational speeds in the image plane approach zero, and the divergence information is left as the only source to be utilized for the derivation of depth.

To summarize the subject of categorization, let us refer to some familiar passive ranging methods. The work of Sridhar, Phatak, and Cheng in [8, 9] and Sridhar, Suorsa and Hussien in [13] is on a feature-based method; it tracks "points of interest" through spatial correlation and Kalman filtering of their image-plane trajectories. It makes all the standard assumptions already mentioned above, it is recursive, or a detect-then-track method, and it uses the translational part of the OF information alone—specifically it *depends* on the approximation that, for a short while, the divergence is zero.

The works of Watson and Ahumada [14], Lowell and Wechsler [15], Barniv [16, 17], and Kendall and Jacobi [18] is in the area of "Velocity Filtering" or 3-D Fourier-Transform (FT) filtering. It is a field-based method, it makes standard assumptions, it is a TBD method, i.e., it uses all the imagery information in batch to feed a bank of 3-D filters, and it only uses the translational OF information—although without any implicit or explicit assumption about the divergence information.

The works of Menon and Sridhar [19], and Chatterji, Menon, and Sridhar [20, 21, 22] are closely related to the initial work of Horn and Schunck in [7]. They make use of the "Correspondence hypothesis," meaning that, between two temporally-close images, each can be expressed in terms of a Taylor-series approximation of the other which leads to an expression for the depth. This is a field-based method which makes all the standard assumptions. In its current form it is a detect-then-track method which initializes on the first pair of images, and it makes no use of the OF divergence information.


Changing texture and size

Texture alone

Size alone

Figure 1. Texture and size cues.

The works of Longuet-Higgins and Prazdny [23], Prazdny [24, 25], Koenderink [26], Koenderink and van Doorn [27, 28], and Nelson and Aloimonos [29] mainly make use of the divergence information in the OF. They are field-based, use standard assumptions, and they are detect-then-track methods. An interesting extension to these works was recently reported by Ringach and Baram in [30]. Although it is field-based, it explicitly assumes, in addition to the standard assumptions, that the scene is composed of objects and derives the *average*, or global, divergence for all objects without the need to actually delineate or identify them in any way. It is, in some sense, a TBD method, and it exclusively uses only the divergence information of the OF. The local- and global-divergence methods are intended for different kinds of objects as exemplified in figure 1. The local-divergence method is intended for textured objects with no well-defined edges, whereas the global-divergence method is intended for objects with little or no texture but having well-defined edges.

With the above background, it is quite clear that, currently, no single method can do the job of passive ranging satisfactorily. However, I will summarize what, in my opinion, are the preferred attributes of a general candidate—keeping in mind that it is quite legitimate to combine more than one approach in a complementary way:
(1) For the simple reason that most scenes contain natural (diffused) *and* man-made objects, it should probably be a combination of field- feature-, and object-based methods.
(2) It should assume, in addition to the standard assumptions, as much as possible about

the scenery—akin to taking into account a priori information.
(3) It should be a TBD algorithm.
(4) It should use all the available OF information, i.e., translation *and* divergence information.
For limited or specialized applications, one may still do well with algorithms that do not satisfy all of the above specifications.

My approach in this work is based on the above general guidelines. It consists of adapting and extending the above-mentioned divergence-based algorithms so that they can help other algorithms (such as those reported in [8, 9]) deal with the problematic FOE area where translational motions are very small. Divergence, which is by definition a local attribute, is reliable in areas of appreciable texture. In areas of little texture, the global divergence can provide a robust performance for objects which exhibit distinct edges. Since such objects (as man-made) tend to have poor texture and vice versa, these two divergence methods appear to be complementary. Thus this paper outlines the current work on developing and combining the above two ideas. The (local) divergence, which is a field-based method, is to be calculated by a Neural Network (NNet); the NNet will be trained to derive divergence by examples. The global divergence, which can be thought of as being an object-based method, is derived by another NNet which emulates a biological model of cells found in the vision system of monkeys [31].

The organization of this report is as follows. Section 2 contains a general introduction to explain the relationship between divergence and optical flow; it summarizes results pertaining to utilizing the (local) divergence for depth derivation. Section 3 summarizes results pertaining to global divergence. The idea of deriving divergence using a NNet is introduced in section 4, along with the suggested method of combining the local and global divergence algorithms. Section 5 serves to summarize this paper.

I would like to acknowledge Dr. Bassam Hussien for his help throughout this work and Drs. Dallas Denery and Banavar Sridhar for their constructive criticism of this report.

2 THE RELATION BETWEEN DIVERGENCE AND OPTICAL FLOW

Basic equations for the divergence in the image plane are derived in this section. This derivation is based on prior work described in [23] to [30].

It is convenient to think for a moment of imaging the outside scene onto a spherical surface because such projections are identical irrespective of the camera-axis direction. In fact, with such geometry, the camera axis is *defined* to coincide with the line-of-sight (LOS)

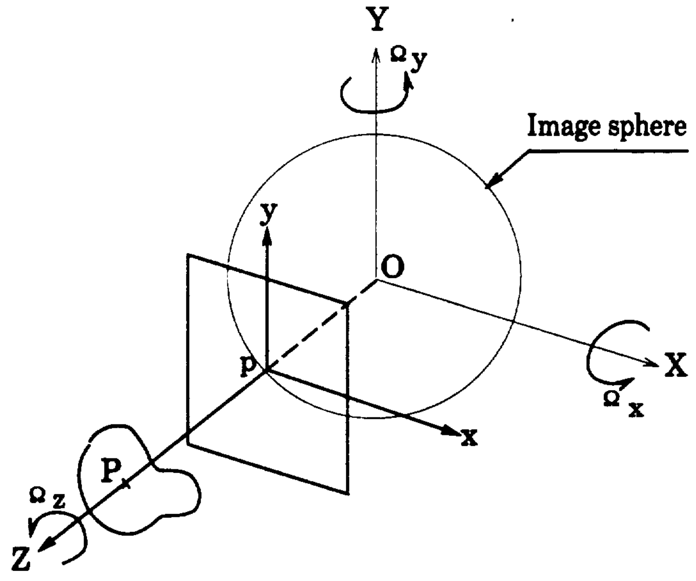


Figure 2. The geometry of projection onto the image plane.

from the center of the sphere to any imaged point, as seen in figure 2. Another motivation for regarding the image plane as a sphere is that this geometry is similar to that of imaging the world by a lens onto a spherical retina, e.g., in the human eye. Let us define the coordinate system of the spherical camera to have its origin at the sphere's center and its Z axis to pass through the imaged point P of some object. Consider the projection of P onto the sphere at point p. At that point define the origin of an (x, y) plane tangent to the sphere which is called *local projective image plane* (image plane, for short); this image plane approximates the sphere at the point of tangency, p. Let us assume that P is found on a smooth surface described by some function $z = f(x, y)$ so that its gradient $\nabla z = (z_x, z_y)$ exists. The distance of any point on that surface from the sphere's center can then be approximated in the neighborhood of P by

$$z \approx z_0 + \nabla z \cdot (x, y) \tag{1}$$

The relative motion of the camera with respect to the scene is defined by its translational velocity $\mathbf{V} \triangleq (V_X, V_Y, V_Z)$ and rotational velocity $\boldsymbol{\Omega} \triangleq (\Omega_x, \Omega_y, \Omega_z)$. It is convenient to normalize $\mathbf{V}$ by z_0 and denote $(V_x, V_y, V_z) \triangleq (V_X, V_Y, V_Z)/z_0$.

The motion of the camera causes the stationary point P and its surroundings to describe what is called a retinal velocity field around p on the image plane. Denoting the retinal velocity vector at p by $\mathrm{v}(p) \triangleq (u, v)\mid_p$ (to correspond with the (x, y) axes) and their partial derivatives by u_x, u_y, v_x, v_y, the following equations hold (see [23]).

$$u = -V_x - \Omega_y\,, \qquad v = -V_y + \Omega_x$$

$$u_x = V_z + V_x z_x\,, \qquad v_y = V_z + V_y z_y$$

$$u_y = \Omega_z + V_x z_y \,, \qquad v_x = -\Omega_z + V_y z_x \tag{2}$$

Using the above equations, the divergence at p (call it div(p)) can be expressed as

$$\text{div(p)} = \nabla \cdot \mathbf{v}(p) = u_x + v_y = 2V_z + \nabla z \cdot (V_x, V_y) \tag{3}$$

To interpret the above equation, suppose that the camera only moves in the Z direction. In that case $V_x = V_y = 0$ and $\nabla \cdot \mathbf{v}(p) = 2V_z = 2V_Z/z_0$, that is, div(p) is twice the reciprocal of the time-to-collision (TTC) of P with the camera's center. Because of this interpretation, div(p) was termed "immediacy" in [26] and other papers, that is, it measures the immediacy of an imminent collision. In the opposite case, when $(V_x, V_y) \neq (0,0)$ and $V_z = 0$, there can still be relative depth changes between the camera and the patch because it is generally slanted. In other words, div(p) will still have the same interpretation as before, except that the imminent collision is going to be with some point on the plane which is tangent to the patch at P and not with P itself. To clarify this point, let us think of a helicopter that approaches landing and is currently pointed at some marker P on the airstrip. If it continues flying on the initial LOS towards that marker ($V_x = V_y = 0$), it will collide with it at time $1/V_z$. If it reduces its forward speed to zero ($V_z = 0$) but develops a downward motion V_y (in the direction of $-Y$) it will collide with the runway down below at time $|1/V_y|$. In this case the runway is the plane tangent to the surface of the "object" at the initial point P. We thus see that both terms of the immediacy have a valid physical interpretation. Notice that the rotational velocities do not even appear in div(p). This is a very important (and well-known) observation because it says that *the TTC information is wholly contained in the imagery; no additional information is needed* (such as from the Inertial Navigation Unit (INU)). However, with INU measurements available, the translational component of the image-plane motion can also be used as an additional source of depth information.

Nelson and Aloimonos describe in [29] a straightforward mechanism for evaluating the divergence from a sequence of images. I have no intention of describing the details of this algorithm here except to point out that (a) it is fairly complicated, (b) it is highly non-linear, and (c) it requires massive amounts of calculations. In terms of performance, it appears to produce expected results and provide a hazard map for a robot arm that moves a camera through a three-dimensional environment containing various obstacles. A candidate trainable NNet is suggested in section 4 as a replacement for this algorithm.

3 THE GLOBAL DIVERGENCE

In this section, I summarize theory and results obtained with an algorithm that was developed by Ringach and Baram in [30]. The basic approach is to deal with the *average* divergence over the area of each object instead of with the divergence at each pixel. The average divergence,

$\chi(R)$, for an object whose projection onto the image plane is R—assuming for the moment that its boundary ∂R is well defined—can be written as

$$\chi(R) \triangleq \frac{1}{A(R)} \int_R \mathrm{div(p)}\, ds = \frac{1}{A(R)} \int_{\partial R} \nabla \cdot \mathbf{v}(p)\, ds = \frac{1}{A(R)} \int_{\partial R} \mathbf{v}(p) \cdot \mathbf{n}\, dl \tag{4}$$

where $A(R)$ is the object area, dl is the elemental length along ∂R, and the equality is based on the divergence theorem. In other words, the average divergence equals the line integral of the normal component of the velocity vector at the edge along the edge of each object. This entails the advantage of having to integrate **v** instead of having to differentiate it. Also, the line integral can easily be shown (see [30]) to have an intuitive interpretation, that is,

$$\chi(R) = \frac{1}{A(R)} \frac{dA(R)}{dt} \tag{5}$$

i.e., the immediacy equals the temporal rate of change of the normalized object area.

Another important innovation in Ringach and Baram's work is in the way the line integral expression for the immediacy is evaluated. They have shown in [30] that a diffusion process, initialized by the normal velocities at all image pixels, would converge to the immediacy values for all closed objects. The procedure can be described by the following steps: (a) calculate the velocities normal to the local edge for all pixels, (b) load a blank image of the same size with positive and negative velocity values in front and behind each pixel respectively according to the direction of the normal velocity vector at that pixel, and (c) start a diffusion process from these initial conditions and run it to convergence.

Ringach and Baram's work was largely motivated by advances in the understanding of visual processing in humans and primates. For example, experiments with humans suggest the existence of divergence (looming) detectors in the human visual system [32, 33, 34] as well as *vorticity* detectors [34, 35, 36]. This is why they naturally used a NNet to perform the operations required by the above equations in agreement with the assumed biological models.

A block diagram of the global divergence algorithm is shown in figure 3. The stream of images I(x,y,t) enters a block called Motion Layer (or image) where the motion-vector components normal to the local edge are calculated for all points in the scene. In the Segmentation layer, detection is performed at each pixel for *moving edges* having an edge amplitude above some threshold and speed above some other threshold (see fig. 4). No attempt is made to actually delineate or define objects. Also, in this layer, positive and negative "charges" are deposited in the front and rear vicinity of each detected pixel as shown in figure 5. In the diffusion layer a NNet of locally connected neurons is used to perform the diffusion. For an ideally closed object, the diffusion process settles on some final value which is, of course, common to the whole interior of the object and represents the desired immediacy. This algorithm can be considered to be TBD because all the operations

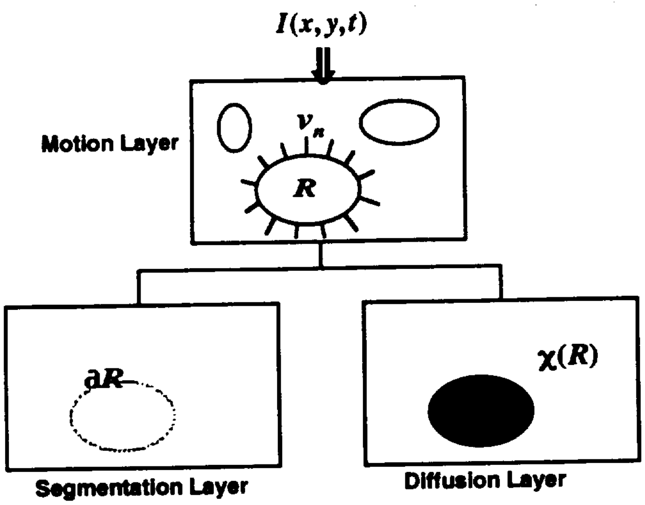


Figure 3. Block diagram of the global divergence algorithm.

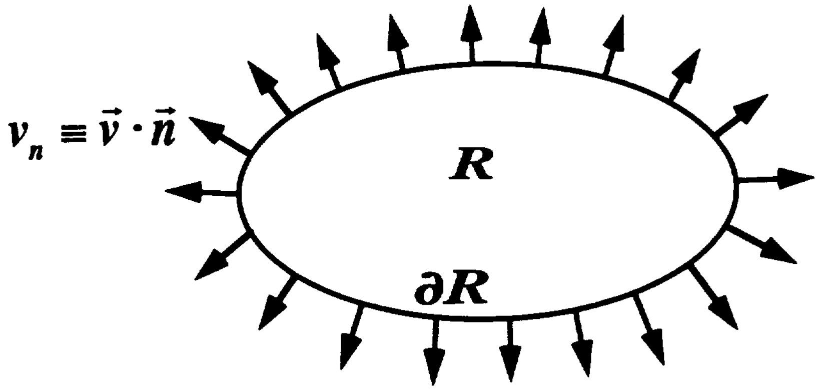


Figure 4. Finding normal velocities.

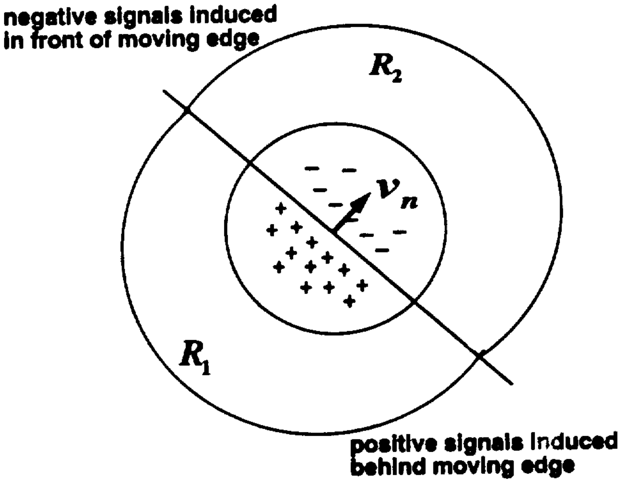

negative signals induced in front of moving edge
R_2
v_n
R_1
positive signals induced behind moving edge

Figure 5. Initializing the diffusion process.

involved evolve with time. In particular, if the diffusion process is not allowed to converge between frames, its value can be made to evolve and converge over as many frames as desired.

4 DIVERGENCE DERIVATION USING A TRAINABLE NEURAL NETWORK

In this section I explain the new idea of training an NNet to derive divergence for textured objects. The motivation for using an NNet for this task comes from the works quoted above. It is seen from equation (3) that div(p) = $u_x + v_y$, which is just a summation of the spatial partial derivatives of the image-plane velocity-vector components. This means that the time and spatial first derivatives are calculable from the imagery, and no other information is necessary. The minimum imagery size that would contain this information consists of two consecutive images of the 2×2 neighborhood of the subject pixel.

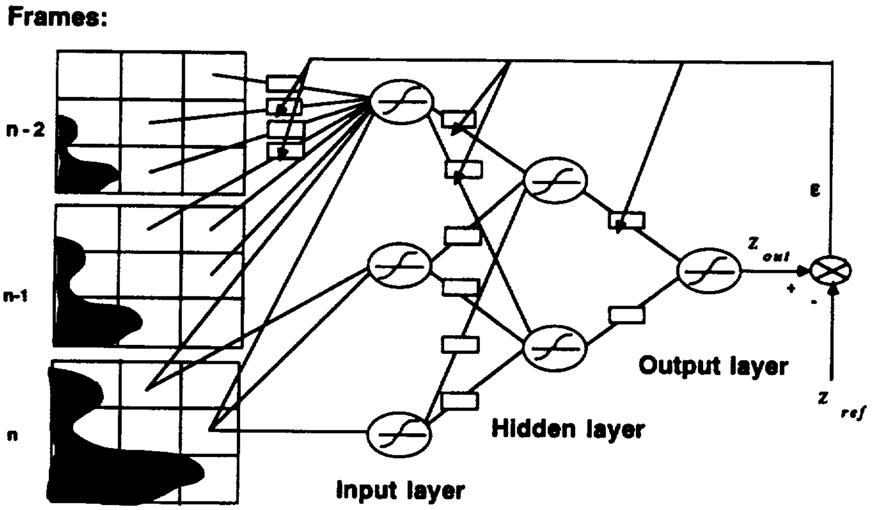


Figure 6. Trainable NNet for divergence estimation.

In figure 6 we see an example of a somewhat larger data source for an NNet—that is, 3 frames of size 3×3 pixels. Thus, the NNet receives 27 inputs of gray level, and its job is to learn to convert these to depth or, given the vehicle's speed, to TTC. A standard NNet with a single hidden layer with a still-undetermined number of neurons is shown. The output layer consists of a single neuron whose output is the predicted depth. All neurons are shown to have a sigmoidal non-linearity at their output although this is of little consequence for the output neuron. Notice that this algorithm can also be considered TBD—especially if we eventually will use many more frames than just the three shown.

A few thousand samples are used in the learning stage, where each sample consists of 27 continuous gray-level values and a single "ground-truth" TTC value, which serves for reference. After each presentation of a sample, the network's predicted TTC is compared with the reference TTC. The error is used to correct all the NNet weights using Gradient Descent towards the local minimum. The standard back-propagation (BP) technique is nominally used to update the weights (e.g. see [37, 38]). The BP performance will be compared with those of other learning techniques. Once the NNet has converged satisfactorily in the learning phase, it is tested with another independent set of samples that have not been used in training. If the average error on that never-seen-before set is low, i.e., comparable with the error in the training phase, then one can declare the NNet to perform well.

Although the basic idea is simple, the issues associated with teaching this NNet (or, for that matter, *any* NNet) what it is supposed to learn are not so. Some of these issues are listed in the following. Readers familiar with NNets applications will notice that the issues of interest are quite generic but the details are specific to our particular problem.

4.1 Input-Data Dimensionality

The crucial issue is the form and dimensionality of the input data. Under this general subject come several subtopics:

1. **Size of raw input data**
   A nominal input vector of size 27 has been chosen as shown in figure 6. However, the necessary or most effective size to be used is still an open parameter. Its value will be determined by several factors, such as the spatial/temporal power spectrum of the scene, the Point-Spread-Function (PSF) of the camera optics, and the required angular coverage (the FOV) around the FOE.

2. **Gray-levels scaling**
   A little thought makes it clear that we do not want the absolute gray-level values of a sample to affect the learned divergence. Just imagine the same scene under two different illuminations. We certainly do not want the NNet to learn anything relating to illumination since it is irrelevant to the derivation of divergence. Now the question is how to normalize the gray levels: by dynamic range, by the average, by some other statistics, etc. The first method was chosen arbitrarily for lack of any better reason to choose another.

3. **Preprocessing**
   The question is whether to use the raw gray-levels, to use their spatial or temporal differences of some order, or, in general, to use another form of representation altogether.

The key consideration is that preprocessing should only preserve the relevant information. By doing so, preprocessing should result in reduced input-vector dimensionality. We know, for example, that image-plane rotation- and/or translation-invariant transformations should discard irrelevant information while preserving the divergence information. In general, one can think of an image in a sequence of images as (approximately) being derived from an affine transformation performed on any of the others. This means that there are, at most, six transformation parameters (four in the multiplying 2×2 matrix and two to account for translation) that relate one image to another. Out of these six, only two are related to expansion—lateral and vertical—in the image plane. Preprocessing may thus take the form of a known translation-invariant transformation, such as a Fourier Transform (FT), and/or a rotation-invariant transformation, such as Cartesian-to-Polar Transformation followed by an FT.

4. **Data compression**

Reducing dimensionality is akin to data compression. In principle, most types of natural data of any source are highly redundant. In our case we know that adjacent pixels—either spatially or temporally—are highly correlated. One straightforward approach to imagery compression could be through using the standard *principal component analysis* (PCA) (e.g. see [39]). This is performed by evaluating the average covariance matrix of the sample vectors (e.g., of size 27×27), and projecting all raw vectors onto the n (to be determined) largest eigenvectors of this matrix. The value of n is chosen such that this process transforms the 27-long raw vectors into shorter ones (of length n) while preserving the essential information. Notice that the standard FT is just another form of PCA where the components are the FT values for all frequencies. Keeping only the n "relevant" components—which, most probably, will amount to some form of high-pass-filtering—will result in a similar compression ratio as with the basic PCA.

Another method to be considered for data compression is to use a standard NNet (in front of the main one) with 27 inputs, 27 outputs, and n hidden-layer elements. The idea is to teach the NNet an effective compression by training it to minimize the mean-square error between inputs and outputs. The compression ratio is $27/n$, and the goal is to find the minimum n that still yields an acceptable average error. It has been shown (e.g.[40, 41]) that an NNet with a single hidden layer essentially carries out a principal component analysis. The advantage of doing this analysis with an NNet as opposed to PCA is that the NNet can learn and adapt to new examples on-line in a robust way.

Yet another method of image compression to be considered is through the use of fractal techniques. Such techniques can especially be effective in reducing texture information by factors as high as 10,000, e.g., see [42, 43, 44]. A compression factor of such magnitude may enable one to work with much larger fractions of the imagery—say, 20×20 instead of 3×3 pixels—and, as a result, achieve much better accuracies.

5. **Coding**
   Using a continuous gray-level input (after preprocessing) does not involve any coding. However, coding techniques have proven essential in many other applications of NNets. This issue will thus be left open at this point.

4.2 The Hidden Layers

The second main issue concerns the hidden layer or layers. Under this subject let us examine the following subtopics.

1. **Number of hidden layers**
   In principle, an NNet having a single hidden layer of some unspecified size is capable of performing any desired general mapping [45]. However, there might be advantages in using more than a single hidden layer in terms of internal data representation and overall number of weights. One can think, for example, of a multilayer NNet designed to perform some pre-assigned data transformations (or mappings). The first hidden layer may be assigned to perform a principal component analysis. The second hidden layer may be assigned to perform translation- and rotation-invariant transformations, i.e., map the imagery into a new space in which only expansion information is represented. An output layer would then be responsible for the final mapping into TTC.

2. **Number of hidden-layer units**
   A very important question, when using one or several hidden layers, is how many units are required in each layer. There is no formulative answer to this question, but there are some clues in the form of Widrow's and Cover's bounds as in [37] and [46] respectively. Accordingly, we know that, for good generalization, the number of samples in the training set should be around 10 times the total number of weights in all layers. Obviously, the choice of hidden-layer neuron number is quite flexible (which is one of the manifestations of NNet's robustness in general). In practice, these numbers can be determined as part of the learning process, where the NNet has the option of adding or discarding superfluous weights and/or neurons based on their effect on the average error. One way of doing that is to add penalty terms to the cost-function such that they push all weights towards zero. Those weights that give in and decay towards zero get discarded, so that the NNet essentially learns its own size.

3. **Total automation of learning**
   Another approach to the choice of the NNet size is to automate the whole learning process, i.e., start from a small net, having a single hidden layer, and let it grow as needed in both number of layers and number of neurons per layer. The advantage in such an approach is that we do not force any of *our* notions about data representation

on the NNet. In other words, we do not impose constraints which might not be advantageous. For example, it may turn out that a single-hidden-layer NNet can learn a valid data transformation *and* compression, where we would be otherwise tempted to divide these tasks between two hidden layers. The problem with such an approach is that, to convince ourselves, we are still very curious (and rightfully so) to understand the physical interpretation of whatever internal representation the NNet did find, and that might turn out to be a difficult task.

4.3 Data Preparation

The third main subject is concerned with the preparation of the training/testing samples and the method of training. These are discussed under the following items.

1. **Number of training samples**
   The first question to ask is how many samples do we need for training. Considering that the dimensionality of the input space is nominally 27, and we use, say, 256 discrete gray-levels, then the potential number of valid data points is 256^{27} (which is more than the number of electrons in the known universe). It is thus clear that we can only afford to sample the input space in a very sparse way—to say the least. A somewhat different approach is to ask what is the total information content conveyed by a single sample—since we know that every sample by itself contains the divergence information in it. The information content of a sample is the number of bits required to specify it, that is,

   $$I = 27 \log_2 256 = 216 \text{ bits} \tag{6}$$

   which is not such a large number. This kind of argument means that, if we only knew how, we could have taught the NNet all it needs to learn based on just a single sample.

   The above considerations may help in developing some intuitive guidelines regarding the necessary number of training samples, but do not necessarily lead to any rigorous method of estimating this number. It at least says that, in principle, a very sparse sampling should suffice in conveying the essential information. In practice, I intend to experiment in order to develop that intuition—and, hopefully, also gain some additional unexpected insights.

2. **Specifying the samples**
   The samples to be used must be specified by the following items:

   (a) Field of view.

   (b) Image-plane velocities (affected by the FOV). These velocities must be small enough so that the same general area will appear in the 3×3 pixels window during the three frames that constitute the input data.

(c) Scaling of TTC in terms of Frames-to-Collision. This seems to be an effective normalized measure because it combines the actual vehicle's velocity, the depth, and the inter-frame time.

(d) Power-spectrum of texture, or graininess. The texture for simulation should be chosen such that it is small compared to the 3 × 3 pixels, or any other chosen window. Again, this texture is normalized. In practice, one can always reduce the actual image-plane texture to the chosen normalized texture by pre-averaging the image-plane pixels as necessary. For example, if the actual texture has a typical spatial frequency of one cycle per 10 pixels, one would average every 10 × 10 image-plane pixel to produce a single effective pixel; these spatially "scaled" pixels will serve as inputs to the NNet.

3. **Scenario simulation for generation of NNet training samples**

We initially intended to train the NNet with real imagery. There were, however, several reasons why that proved to be intractable. First, the imagery we have does not come with complete ground-truth ranges, so that points to where interesting as-textured features usually lacked range information. Second, even if some good samples do exist in that imagery, their number and generality are inadequate for training an NNet. Third, with a given imagery set, one has only little control on the parameters of interest such as texture fineness, dynamic range of the gray-levels, distance from the FOE, and others. Therefore, a scenario simulation has been developed to generate the required samples as specified above. The scenario consists of a vehicle (helicopter) flying on any pre-chosen trajectory (including maneuvers) and imaging a slanted vertical wall. The wall is slanted by any desired angle (in the range 0 to 90°) with respect to the LOS so that its left side is closer to the vehicle than its right side. The wall is painted with texture which is generated by filtering white Gaussian noise through a prescribed Gaussian-shaped PSF. There are a few points of interest regarding this simulation.

The first problem encountered was that the far side (right) of the wall looked darker than the close side. Since, in reality, one would expect a wall like that to appear uniformly illuminated, this artifact was corrected to obtain uniform gray levels on the image plane. Another artifact was that the image-plane projection of the simulated slanted uniformly-textured wall exhibited a non-uniform texture—its fineness was proportional to the depth. This phenomenon raised the question of what would be expected from a natural scene. Using the analogy of fractals, I argue that the natural-scene texture (as defined by any texture measures) should nominally be independent of depth—trees at a large distance would, in principle, present as fine a texture as the leaves of a single tree at a very close range. Thus, the expected texture in the image plane is nominally constant; the simulation was adjusted accordingly. Figure 7 shows a sequence of three frames resulting from the simulation of a straight and level flight towards the center of a 45^0-slanted wall. The distances to the wall are 150, 125, and 100 m. The left side of the wall is closer than the right side, which explains why

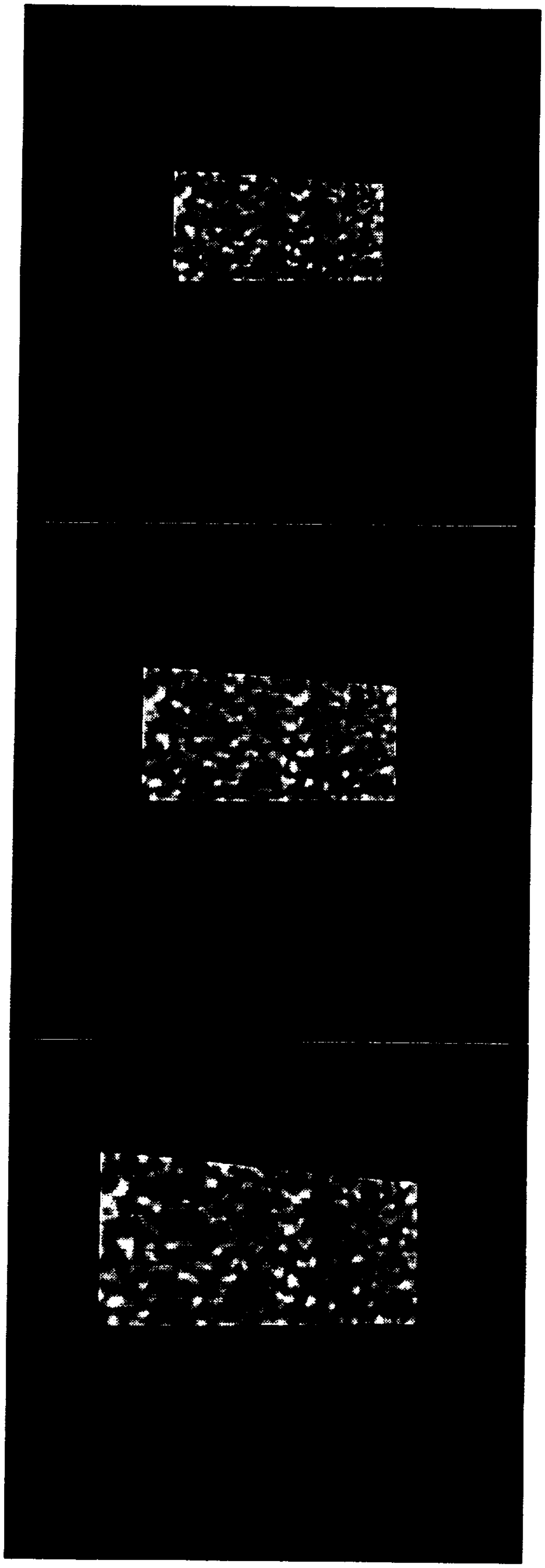


Figure 7. Three consecutive frames of the simulated textured wall.

its projection is wider on the left. The texture equalization in these figures has been turned off to better convey the sense of perspective.

Using a sequence like that, the wall is systematically sampled every other pixel in both directions. The $3 \times 3 \times 3$ spatial-temporal window around each such pixel constitutes a single NNet sample. The known frames-to-collision for the pixel serves as the reference associated with that sample for training the NNet.

4. **Specifying the training procedure**
Once we have produced a data set containing, say, 10,000 samples, there are a few alternatives available concerning the training procedure. The first question has to do with the order of sample presentation to the NNet. We first randomize the order of the samples in the data set so that they will not appear to be in any geometrically-related order (since the wall is scanned by rows). The data set is then divided into two distinct parts: one for training and one for testing, say half and half. Training can either be done by running the training set in the same order cyclically as many times as necessary for error convergence, or picking up samples at random from the training set until convergence is achieved. We noticed that the second method produces a "noisy" error curve resembling the behavior of simulated-annealing learning. This is why we prefer to use this method; it is less likely to get stuck in a local minimum.

   Another subject of consideration is whether to train a multi-hidden-layer network as a single NNet or train each hidden layer separately. As pointed out earlier, separate training can only be performed when each layer is assigned with a physically interpretable task. For example, if a hidden layer is expected to perform data compression, it is trained to do just that.

   An additional question is whether to train the NNet in increasing stages of difficulty or mix up all cases irrespective of difficulty. Cases of points which are very close to the FOE should be considered easy because the lateral translation would be very small. The same is true regarding the vehicle's maneuvers. Easy cases would be those derived from a rectilinear flight path. Intuitively, one would expect the easy-to-difficult learning approach to have higher chances of converging into the absolute error's minimum as opposed to falling into some local minimum; this is why we chose to use this approach.

5. **Specifying the testing procedure**
Testing is relatively simple. The trained NNet is presented with a few thousand samples that it has not seen before in the training phase. The requirement is that the average error on this set falls in the same order of magnitude as the average error obtained on the training set.

4.4 Integrating the Local and Global Divergence Methods

My goal is to use the local and global divergence methods to complement each other. In textured areas the local divergence should take the lead whereas in structured areas the global divergence should lead. There are several conceivable ways of combining the two algorithms. The simplest way is to run both of them in parallel, and switch in the result which is more robust—to be determined by some measure which has not yet been developed. A more desirable approach is to integrate the two methods at some earlier stage so that the switching between them will take place in a more natural way. At the moment I will leave this problem open because it might turn out to be more involved than just integrating the two divergence methods. As I have pointed out earlier, these two methods are potentially advantageous in the image area around the FOE. Thus integration should actually be done on a larger scale, that is, it should include the two divergence methods along with other methods which are advantageous in areas far from the FOE.

5 CONCLUDING REMARKS

In this paper I presented my current views and approach to monocular depth derivation. In particular, two divergence-based depth estimation methods were suggested (local and global) as promising to be effective in the image area close to the FOE. A general feed-forward multilayer neural network was suggested for the realization of this approach. In addition to relevant background and some basic theory, I described the parts of the work which have already been completed as well as my overall plans for the future work.

REFERENCES

[1] Wegner, L. H: On the Accuracy Analysis of Airborne Techniques for Passively Locating Electromagnetic Emitters. Report R-722-PR AD 729 767, NTIS ASTIA D.C., Rand Corp, 1971.

[2] Poirot, J. L.; and McWilliams, G. V.: Application of Linear Statistical Models to Radar Location Techniques. *IEEE Trans. on Aerospace and Electronic Systems*, vol. 10, no. 6, November 1974, pp. 830–834.

[3] Torrieri, D. J.: Statistical Theory of Passive Location Systems. *IEEE Trans. on Aerospace and Electronic Systems*, vol. 20, no. 2, March 1984, pp. 183–198.

[4] Gavish, M.; and Fogel, E.: Effect of Bias on Bearing-Only Target Location. *IEEE Trans. on Aerospace and Electronic Systems*, vol. 26, no. 1, January 1990, pp. 22–25.

[5] Sridhar, B.; and Suorsa, R. E.: Comparison of Stereo and Motion Sensors in Passive Ranging Systems. In *American Control Conference*, San Diego, CA, May 1990.

[6] Barniv, Y.: Error Analysis of Combined Optical-Flow and Stereo Passive Ranging. *IEEE Trans. on Aerospace and Electronic Systems*, to be published, October 1992.

[7] Horn, B. K. P.; and Schunck, B. G.: Determining Optical Flow. *Artificial Intelligence*, vol. 17, no. 3, August 1981, pp. 185–203.

[8] Sridhar, B.; and Phatak, A. V.: Simulation and Analysis of Image-Based Navigation System for Rotorcraft Low-Altitude Flight. In *Proceedings of the AHS Meeting on Automation Application for Rotorcraft*, Atlanta, GA, April 1988.

[9] Sridhar, B.; Cheng, V. H .L.; and Phatak, A. V.: Kalman Filter Based Range Estimation for Autonomous Navigation Using Imaging Sensors. In *Proceedings of the 11th Symposium on Automatic Control in Aerospace*, Tsukuba, Japan, July 1989.

[10] Barniv, Y.: Dynamic Programming Solution for Detecting Dim Moving Targets. *IEEE Trans. on Aerospace and Electronic Systems*, vol. 21, no. 3, March 1985, pp. 144–156.

[11] Barniv, Y.; and Kella, O.: Dynamic Programming Solution for Detecting Dim Moving Targets Part II: Analysis. *IEEE Trans. on Aerospace and Electronic Systems*, vol. 23, no. 6, November 1987, pp. 776–788.

[12] Barniv, Y. (Ch. 4) (Yaakov Bar-Shalom, Editor). *Multitarget-Multisensor Tracking: Advanced Applications.* Artech House, 1990.

[13] Sridhar, B.; Suorsa, R. E.; and Hussien, B.: Passive Range Estimation for Rotorcraft Low Altitude Flight. *J. Machine Vision and Applications*, 1991.

[14] Watson, B. W.; and Ahumada, A. J., Jr.: Model of Human Visual-Motion Sensing. *J. Optical Society of America*, vol. 2, no. 2, February 1985, pp. 322–341.

[15] Lowell, J.; and Wechsler, H.: Derivation of Optical Flow Using a Spatiotemporal-Frequency Approach. *Computer Vision, Graphics, and Image Processing*, vol. 38, 1987, pp. 29–65.

[16] Barniv, Y.: Velocity Filtering Applied to Optical Flow Calculations. TM-102802, NASA Ames Research Center, Moffett Field, CA, August 1990.

[17] Barniv, Y.: Velocity Filtering Applied to Optical Flow Calculations. *IEEE Trans. on Aerospace and Electronic Systems*, to be published, October 1992.

[18] Kendall, W. B.; and Jacobi, W. J.: Passive Electro-Optical Sensor Processing for Helicopter Obstacle Avoidance. Contractor Report NAS2-12774, NASA Ames Research Center, Moffett Field, CA, August 1989.

[19] Menon, P. K.; and Sridhar, B.: Passive Navigation Using Image Irradiance Tracking. In *AIAA Guidance, Navigation and Control Conference*, Boston, MA, August 1989.

[20] Menon, P. K.; Chatterji, G. B.; and Sridhar, B.: Vision-Based Optimal Obstacle Avoidance Guidance for Rotorcraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, August 1991.

[21] Chatterji, G. B.; Menon, P. K.; and Sridhar, B.: Passive Obstacle Location for Rotorcraft Guidance. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, August 1991.

[22] Chatterji, G. B.; Menon, P. K.; and Sridhar, B.: A Fast Algorithm for Image-Based Ranging. In *Proceedings of the SPIE International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, Orlando, FL, April 1991.

[23] Longuet-Higgins, H. C.; and Prazdny, K.: The Interpretation of a Moving Retinal Image. *Proc. R. Soc., London B*, vol. 208, 1980, pp. 385–397.

[24] Prazdny, K.: Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer. *Computer Vision, Graphics, and Image Processing*, vol. 17, 1981, pp. 238–248.

[25] Prazdny, K.: Egomotion and Relative Depth Map from Optical Flow. *Biological Cybernetics*, vol. 36, 1980, pp. 87–102.

[26] Koenderink, J.: Optic Flow. *Vision Research*, vol. 26, no. 1, 1986, pp. 161–180.

[27] Koenderink, J. J.; and van Doorn, A. J.: Invariant Properties of the Motion Parallax Field Due to the Movement of Rigid Bodies Relative to an Observer. *Optica Acta*, vol. 22, no. 9, 1975, pp. 773–791.

[28] Koenderink, J. J.; and van Doorn, A. J.: Local Structure of Movement Parallax of the Plane. *J. Optical Society of America*, vol. 66, no. 7, July 1976, pp. 717–723.

[29] Nelson, R. C.; and Aloimonos, J.: Obstacle Avoidance Using Flow Field Divergence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, 1989, pp. 1102–1106.

[30] Ringach, D. L.; and Baram, Y.: A Diffusion Mechanism for Obstacle Detection from Size-Change Information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, no. 5, 1992, pp. 6–7.

[31] Fukada, Y.; Tanaka, K.; and Saito, H.: Analysis of Motion of the Visual Field by Direction, Expansion/Contraction, and Rotation Cells Clustered in the Dorsal Part of the Medial Superior Temporal Area of the Macaque Monkey. *J. Neurophysiology*, vol. 62, no. 3, September 1989, pp. 624–641.

[32] Reagan, D.; and Beverley, K. I.: Looming Detectors in the Human Visual Pathway. *Vision Research*, vol. 18, 1978, pp. 415–421.

[33] Reagan, D.; and Beverley, K. I.: Visual Responses to Changing Size and Sideways Motion for Different Directions of Motion in Depth: Linearization of Visual Responses. *J. Optical Society of America*, vol. 70, no. 11, November 1980, pp. 1289–1297.

[34] Hershenson, M.: Visual System Responds to Rotational and Size-Change Components of Complex Proximal Motion Patterns. *Perception and Psychophysics*, vol. 42, no. 1, 1987, pp. 60–64.

[35] Cavanagh, P.; and Favreau, O. E.: Motion Aftereffect: A Global Mechanism for the Perception of Rotation. *Perception and Psychophysics*, vol. 9, 1980, pp. 175–182.

[36] Reagan, D.; and Beverley, K. I.: Visual Responses to Vorticity and the Neural Analysis of Optic Flow. *J. Optical Society of America*, vol. 2, no. 2, February 1985.

[37] Widrow, Bernard; and Stearns, Samuel D.: *Adaptive Signal Processing.* Prentice-Hall, Englewood Cliffs, N.J, 1985.

[38] Rumelhart, D. E.; and McClelland, J. L.: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition I & II.* MIT Press, Cambridge MA, 1986.

[39] Strang, G.: *Linear Algebra and its Applications.* Academic Press, 1976.

[40] Linsker, R.: Self-Organization in a Perceptual Network. *IEEE Computer*, vol. 21, March 1988, pp. 105–117.

[41] Baldi, P.; and Hornik, K.: Neural Networks and Principal Component Analysis. *Neural Networks*, vol. 2, 1989, pp. 53–58.

[42] Barnsley, M. F.; and Sloan, A. D.: A Better Way to Compress Images. *BYTE*, pp. 215–222, January 1988.

[43] Barnsley, M. F.: *Fractals Everywhere.* Academic Press, 1988.

[44] Keller, J. M.; Chen, S.; and Crownover, R. M.: Texture Description and Segmentation Through Fractal Geometry. *Computer Vision, Graphics, and Image Processing*, vol. 45, 1989, pp. 150–166.

[45] Minsky, M. L.; and Papert, S.: *Perceptrons.* MIT, Cambridge, 1969.

[46] Cover, T. M.: Geometrical and Statistical Properties of Linear Threshold Devices. Report SEL-64-052, TR 6107-1, Stanford University, Stanford, CA, May 1964.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1992	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE

Neural Networks Application to Divergence-Based Passive Ranging

5. FUNDING NUMBERS

505-64-52

6. AUTHOR(S)

Yair Barniv

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-92198

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-103981

11. SUPPLEMENTARY NOTES

Point of Contact: Yair Barniv, Ames Research Center, MS 210-9, Moffett Field, CA 94035-1000
(415) 604-5451

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited
Subject Category – 04

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this report is to summarize the state of knowledge and outline the planned work in divergence-based/neural-networks approach to the problem of passive ranging derived from optical flow. Work in this and closely related areas is reviewed in order to provide the necessary background for further developments. New ideas about devising a monocular passive-ranging system are then introduced. It is shown that image-plan divergence is independent of image-plan location with respect to the focus of expansion and of camera maneuvers because it directly measures object's expansion which, in turn, is related to the time-to-collision. Thus, a divergence-based method has the potential of providing a reliable range complementing other monocular passive-ranging methods which encounter difficulties in image areas close to the focus of expansion. Image-plan divergence can be thought of as some spatial/temporal pattern. A neural network realization has been chosen for this task because neural networks have generally performed well in various other pattern recognition applications. The main goal of this work is to teach a neural network to derive divergence from the imagery.

14. SUBJECT TERMS

Divergence, Neural networks, Passive ranging

15. NUMBER OF PAGES

26

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT