7 (4 bits), so a table with $2^{4\times 2}$ entries must be constructed. For each possible input combination, we used the 35 fuzzy rules shown in Fig. 2 to obtain its output Z (4 bits) in advance, and then saved Z in the corresponding entry. With the help of the table, which needs only 128 bytes of memory, fuzzy reasoning can be implemented easily.

Compared with the full-search VQ, the FP-VQ may cause distortion of image quality. Tables VIII and IX show the image quality for both full-search VQ and FP-VQ in terms of four different measures: 1) average absolute error per pixel (ABE), 2) average mean square error (MSE) per pixel, and 3) average signal-to-noise ratio (SNR), and 4) average peak signal-to-noise ratio (PSNR). Obviously, FP-VQ causes only very little distortion (slightly higher MAE and lower PSNR) but achieves better compression efficiency (shown in Tables II–VII) and needs less computational complexity (fewer SAD operations). Fig. 4 shows the original image and three reconstructed images for FP-VQ with codebook sizes of 128, 256, and 512 codevectors, respectively.

Since the FP-VQ encodes each index on-line without scanning the whole index map in advance nor storing the map afterwards, the memory required is small. Besides, it needs fewer computations and achieves more efficient coding than VLC. Hence, the FP-VQ is very suitable for real-time applications and hardware implementation. Its VLSI architecture is currently under development.

IV. Conclusions

In this paper, an efficient FP-VQ algorithm is proposed. It performs better than other VQ algorithms, such as SOC-VQ, LCIC-VQ, STC-VQ, and DST-VQ, in terms of computational complexity and coding efficiency. As our experimental outcomes show, FP-VQ is both efficient and robust not only for various test images, but also for codebook sizes and codebook initialization methods. Since its computation complexity is low and its memory size is small, it is a good real-time coder for image vector quantization.

Acknowledgment

The author would like to thank C.-Y. Hu and J.-K. Huang for their helps in the implementation of the proposed algorithm.

References

[1] K. Sayood, *Introduction to Data Compression*. San Mateo, CA: Morgan Kaufmann, 2000.

[2] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[3] R. M. Gray, "Vector quantization," *IEEE Acoust, Speech, Signal Processing Mag.*, pp. 4–29, Apr. 1984.

[4] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 957–969, 1988.

[5] Y. Linde, A. H. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.

[6] J. Foster, R. M. Gray, and M. O. Dunham, "Finite-state vector quantization for waveform coding," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 3, pp. 348–359, 1985.

[7] C. H. Hsieh, K. C. Chuang, and J. S. Shue, "Image compression using finite-state vector quantization with derailment compensation," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 3, no. 5, pp. 341–349, 1993.

[8] N. M. Nasrabadi, C. Y. Choo, and Y. Feng, "Dynamic finite-state vector quantization," *IEEE Trans. Commun.*, vol. 42, pp. 2145–2154, 1994.

[9] N. M. Nasrabadi and Y. Feng, "Image compress using Address vector quantization," *IEEE Trans. Commun.*, vol. 38, pp. 2166–2173, 1990.

[10] T. Lookabaugh, V. Riskin, P. A. Chou, and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. Commun.*, vol. 41, no. 1, pp. 186–199, 1993.

[11] C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search order coding," *IEEE Trans. Image Processing*, vol. 41, no. 2, pp. 327–331, 1995.

[12] Y. C. Hu and C. C. Chang, "Low complexity index-compressed vector quantization for image compression," *IEEE Trans. Consumer Electron.*, vol. 45, no. 1, pp. 219–224, 1999.

[13] M. H. Shen, S. C. Tsai, and M. D. Shieh, "A lossless index coding algorithm and VLSI design for vector quantization," in *Proc. 1st IEEE Asia Pacific Conf. ASICs*, 1999, pp. 198–201.

[14] C.-Y. Hu and P.-Y. Chen, "An efficient on-line lossless vector-quantized index coder," in *Int. Conf. Fundamentals of Electronics Communications and Computer Sciences*, 2002.

[15] L. A. Zadeh, "Fuzzy sets," *Inf. Contr.*, vol. 8, pp. 338–353, 1965.

[16] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, pp. 1585–1588, 1974.

[17] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I & Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–435, 1990.

[18] J. M. Jou and P.-Y. Chen, "A hybrid adaptive fuzzy system using error backpropagation learning," in *CFSA/IFIS/SOFT'95 on Fuzzy Theory and Applications*, Taipei, Taiwan, R.O.C., 1995, pp. 537–542.

[19] T. Nozawa, M. Konda, M. Fujibayashi, M. Imai, K. Kotani, S. Sugawa, and T. Ohmi, "A parallel vector-quantization processor eliminating redundant calculations for real-time motion picture compression," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1744–1751, Nov. 2000.

Fuzzy Fractals and Fuzzy Turbulence

Kausik Kumar Majumdar

***Abstract*—In this paper, we have defined and discussed fuzzy fractals from image generation point of view. We have also proposed a fuzzy system modeling of a two dimensional turbulence just as a chaotic occurrence of fuzzy vortices in a two dimensional dynamic fluid.**

***Index Terms*—Fuzzy differential inclusion (FDI) relations, fuzzy dynamical systems (FDS), fuzzy fractals, fuzzy turbulence, iterated fuzzy sets systems (IFZS).**

I. Introduction

The contractive iterated function systems on R^2 have been extensively studied by Barnsley [1]. Since these are dissipative systems they must have attractors. The attractors of such systems turn out to be fractal subsets of R^2 or at least these subsets are regarded as fractals in a broad sense. Barnsley has shown that with just a few (often not more than four) simple contractive iterated functions a very complicated shaped image can be generated [1]. This has been fuzzified by Cabrelli *et al.* [3]. They have treated any grey level image as a fuzzy subset of R^2, where the membership value at each point (pixel) is a normalized grey level pixel intensity value at that point (pixel). They have shown how an IFZS can generate a grey level image. In Section II of this paper we have simplified and extended this method. We have defined fuzzy fractals (which Cabrelli *et al.* did not do) in a very broad sense. Crisp fractals (e.g., those generated by Barnsley's systems) turn out to be special cases of fuzzy fractals.

In Section III we have treated turbulence as a chaotic occurrence of vortices in a dynamic fluid [2]. Then we have presented a simple two

Manuscript received September 26, 2002; revised February 26, 2003. This paper was recommended by Associate Editor I. Bloch.

The author is with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700108, India (e-mail: kausik_r@isical.ac.in).

Digital Object Identifier 10.1109/TSMCB.2003.811517

1083-4419/04$20.00 © 2004 IEEE

dimensional model of a turbulence, which satisfies this notion. In this model we have treated the generation of vortices as fuzzy systems. We have presented the algorithm and the simulation results. For the complete MATLAB program, see [20]. In the entire simulation process all the parameters have been kept uncontrolled so that it becomes very unlikely that any two turbulence generated by this program, even with the same argument, turn out to be identical. This is an essential feature of turbulence, e.g., it is very unlikely that any two coils of smoke emanating out of the same chimney at different time will ever be identical.

II. Fuzzy Fractals and Image Generation

Mandelbrot defined *fractal* as a special class of subsets of a complete metric space for which the Hausdorff–Besicovitch (or just Hausdorff) dimension strictly exceeds the topological dimension [12]. Barnsley has shown that, fractal sets can be generated as attractors of a class of randomized, contractive or hyperbolic, iterated function system (IFS) [1]. In other words a given set of randomized, hyperbolic, IFS is able to generate a particular image as a fractal set. But such a system is a discrete dynamical system with a fractal attractor. The initial mathematical background for this work was earlier prepared by Hutchinson [8]. Barnsley further extended this background by proposing and proving a number of important mathematical results [1]. A fractal attractor is called a *strange attractor*, where the underlying dynamical system may or may not be chaotic [9]. Of course there is no universal agreement yet over this point [13].

Definition 2.1: Let (X, d) be a complete metric space. $f: X \rightarrow X$ is a *hyperbolic iterated function system* if there exist N contraction mappings $\{w_i\}_{i=1}^N$, $w_i: X \rightarrow X$ for $i \in \{1, \ldots, N\}$, s.t., $f(D) = \cup_i^N w_i(D)$ for $D \subseteq X$. When each w_i is an affine transformation then f is called an *Hutchinson operator*.

Definition 2.2: Let (X, d) be a complete metric space. $f: X \rightarrow X$ is a *randomized hyperbolic iterated function system* if $f = (w_1, \ldots, w_N, p_1, \ldots, p_N)$. Each $w_i: X \rightarrow X$ is a contraction mapping. Each p_i is the probability of occurring w_i. Thus $\Sigma_{i=1}^N p_i = 1$. And for any $D \subseteq X$, $f(D) = \cup_{i=1}^N w_i(D) f$, under iterations, has been called *chaos game* in [1].

Barnsley has shown how a simple randomized, hyperbolic IFS, often consisting of just three to four such simple contractive functions, can generate extremely complicated arbitrary shaped images [1]. This method has been extended, with suitable modifications, by Cabrelli *et al.* to generate, analyze and/or approximate images as the fractal attractors of IFZS [3].

Images with grey or color levels admit a natural representation in terms of fuzzy sets. In this regard our method incorporates the technique of the discrete dynamical system of IFS. Let f be an IFS as defined in Definition 2.2. Since each w_i is contractive [i.e., $d(w_i(x), w_i(y)) \leq sd(x, y), s \in (0, 1)$] f is also contractive or dissipative. So f must have an attractor $A \subseteq X$, i.e., $\lim_{n \rightarrow \infty} f^n(X) = A$, where $f^n = f \, o \, f^{n-1}$. Under each iteration of f, the possibility (for a detailed discussion on *possibility* refer to [19]) of occurrence of w_i is p_i. For a given set of possibilities $\{p_i\}$ there exists a unique invariant fuzzy membership measure μ with support A. The novelty of our approach may be summarized in terms of the following two key points:

1) The entire setting is provided by a subclass $(\Im, \delta)$ of the class $F(X)$ of fuzzy subsets of X, where $\Im$ is the collection of all the nonempty compact fuzzy subsets of a compact metric space (X, d) and δ is the metric defined on $\Im$ (the collection of all nonempty compact fuzzy subsets of X). All images are considered as fuzzy sets. This leads to two possible interpretations:
   a) in image representation the value of the fuzzy membership function at a point (pixel) $x \in X$, within the image will be interpreted as the normalized (values lying in [0, 1] only) grey level intensity associated with that point (pixel);
   b) in pattern recognition, the membership value of x when lies in (0, 1] indicates the possibility that the point x is in the foreground of an image.

Associated with each contraction map w_i, $i = 1, \ldots, N$, is a grey level map $\varphi_i: I \rightarrow I$, where $I = [0, 1]$ is the grey level domain. The collection of maps $\{w_{i,} \varphi_i\}$ is used to define an operator $T: U^n \rightarrow U^n$ (U^n is the collection of normal, uppersemicontinuous members of $\Im$), which is contractive with respect to the metric δ on U^n. This metric is induced by the Hausdorff distance defined on the collection of the nonempty compact subsets of X. Starting with an arbitrary initial fuzzy set $\mathbf{u}_0 \in U^n$, the sequence $\mathbf{u}_n \in U^n$ produced by the iteration $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ converge in the δ metric to a unique invariant fuzzy set $\mathbf{v} \in U^n$, where $T(\mathbf{v}) = \mathbf{v}$.

Definition 2.3: $f: \Im \rightarrow \Im$ will be called *iterated fuzzy sets systems* (IFZS) if $f = (w_1, \ldots, w_N, \mu(w_1), \ldots, \mu(w_N))$. Each w_i: $\Im \rightarrow \Im$ is a contraction mapping. Each $\mu(w_i)$ is the possibility or fuzzy membership value [19] associated with w_i. For any $\mathbf{u} \in U^n$

$$f([\mathbf{u}]^\alpha) = \bigcup_{i=1}^N w_i([\mathbf{u}]^\alpha) \tag{2.1}$$

where $[\mathbf{u}]^\alpha$ is the α-level set of $\mathbf{u}$ for $0 \leq \alpha \leq 1$.

Definition 2.4: The compact set X will be called the *base space* of the IFZS.

If there exists a fuzzy set $\mathbf{v} \in U^n$ such that, $f^n([\mathbf{v}]^\alpha) \subseteq [\mathbf{v}]^\alpha$ for $0 \leq \alpha \leq 1$ and for all positive integral n, then $\mathbf{v}$ is clearly the fuzzy nonwandering set or invariant set of the discrete fuzzy dynamical system $(\Im, f)$ or just f. When $f^n([\mathbf{v}]^\alpha) = [\mathbf{v}]^\alpha$ holds for $0 \leq \alpha \leq 1$ and for all positive integral n, $\mathbf{v}$ will be the unique invariant (fuzzy) set of f under iterations [8]. Then clearly $\mathbf{v}$ is an (fuzzy) attractor of the IFZS f.

Definition 2.5: A fuzzy subset A of a complete metric space (X, d) is called a *fuzzy fractal* if A is also a fractal subset of X according to Mandelbrot's definition.

Clearly $\mathbf{v}$ is the fuzzy attractor of the image generating operator T in (2). In addition if $\mathbf{v}$ is a fractal subset of X (which is more often than not the case) $\mathbf{v}$ is a grey level fuzzy fractal image generated by T just like Barnsley's chaos games generating crisp fractal images.

In case of image generation, $\mu(w_i)$ will depend on a normalized grey level pixel intensity value. We could also take $\mu(w_i) = p_{i,}$ where p_i is a probability measure. That is, $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. When p_i is a probability measure instead of a possibility measure the underlying IFZS reduces to Barnsley's chaos game and the resulting fractal set generated is a crisp fractal. So crisp fractal is only a special case of fuzzy fractal.

A. Images as Fuzzy Sets

A black and white digitized image is a pixel matrix $\{p_{ij}\}$, where p_{ij} is the (i, j) (both are nonnegative integers) coordinate point (pixel) in R^2. Associated with each pixel p_{ij} is a nonnegative grey level or brightness value t_{ij}. We assume a normalized measure for grey levels, i.e., $0 \leq t_{ij} \leq 1$ (0 = black: the background, 1 = white: the foreground).

Definition 2.6: The function h: $\{p_{ij}\} \rightarrow$ [0, 1] defined by the grey levels distribution of the image is called the *image function*.

The digitized image is fully described by its image function h. This is also the situation in the more theoretical case where grey level are distributed continuously on the base space X. At this point one can see that, an image as described by an image function is nothing but a fuzzy set $\mathbf{u}: X \rightarrow$ [0, 1].

It is usual to classify the pixels according to their grey levels in the following way. For each $\alpha \in$ (0, 1], we consider the set $\{x \in$

$X|\mathbf{u}(x) \geq \alpha\} = [\mathbf{u}]^\alpha$, i.e., the set of all pixels whose grey levels exceed the threshold value α is the α-level set (α-cut) of $\mathbf{u}$. For $\alpha \in$ (0, 1], $[\mathbf{u}]^\alpha$ represents a thresholding of the grey level distribution at α. Clearly there is a one to one correspondence between the image function h as a fuzzy set and the α-cuts of $\mathbf{u}$.

Since X is a compact metric space $(\Im,\, \delta)$ is also a compact metric space [6]. In particular it contains all the α-level sets $[\mathbf{u}]^\alpha$, $0 \leq \alpha \leq 1$, of all $\mathbf{u} \in U^n$. δ is the Hausdorff metric [6, 8]. The metric space $(U^n,\, \delta)$ is complete [6]. f is the IFZS of Definition 2.3, which being a contraction map (time-evolution law of a dissipative dynamical system) must have an attractor $[\mathbf{v}]^0 = A \subseteq X$, where $\mathbf{v} \in U^n$. In all practical situations X is a compact subset of R^2 and A is a fractal subset of X, which is also a fuzzy subset of X.

Definition 2.7: A as described above is called a *fuzzy fractal image* generated by the IFZS f.

One very important thing to notice in this subsection is a novel interpretation of fuzzy membership value. Traditionally fuzzy membership values signify the degree of vagueness or nonspecificity [5]. But in this subsection fuzzy membership value can not be interpreted that way. Here, fuzzy membership value means "lack of uniformity" (in the normalized grey level pixel intensity values). So far we have not gone into the detail of the grey level intensity distribution of the pixels p_{ij} of the fuzzy fractal image A. In the next subsection we are going to take up this issue.

B. Determination of the Grey Level Pixel Intensity

In Barnsley's randomized IFS each component w_i of the hyperbolic map f has a probability of occurrence p_i under each iteration of f (Definition 2.2). In the definition of IFZS (Definition 2.3) p_i of hyperbolic randomized IFS is to be replaced by $\mu(w_i)$, where $\mu(w_i)$ is the possibility (in sense of Zadeh [19]) of occurrence of w_i under each iteration of f. Since value of possibility is equal to fuzzy membership value, $\mu(w_i)$ is the fuzzy membership value associated with w_i. From image generation and processing point of view we are going to give here an interpretation of $\mu(w_i)$ in terms of the grey level map $\varphi_i\colon I \to I$, where $I = [0,\, 1]$ is the grey level domain.

Definition 2.8: A function φ: [0, 1] → [0, 1] is said to be nondecreasing right continuous (n.d.r.c) if and only if i) φ is nondecreasing and ii) φ is right continuous.

The following lemma justifies this definition.

Lemma 2.1: Let φ: [0, 1] → [0, 1] and X be an infinite compact metric space, then a necessary and sufficient condition for φ o $\mathbf{u}$ to be upper-semicontinuous for all $\mathbf{u} \in U^n$ is that φ is n.d.r.c.

Proof: Reference [3].

We are now in a position to summarize properties, which should be satisfied by a set of grey level maps $\{\varphi_i\}$, $i = 1, \ldots, N$, so that they can be associated with $\mu(w_i)$ of an IFZS f.

1) φ_i: [0, 1] → [0, 1] is nondecreasing.
2) φ_i is right continuous in [0, 1).
3) $\varphi_i(0) = 0$.
4) For at least one $j \in \{1, \ldots, N\}$, $\varphi_j(1) = 1$.

Properties 1), 2), and 4) above and Lemma 2.1 together will guarantee that the associated IFZS maps U^n into itself. Property 3) is a natural assumption in the consideration of the grey level functions, which says that, if the grey level of a point (pixel) $x \in X$ is zero, then it should remain zero after being acted upon by the maps φ_i.

C. Image Generating Fuzzy Contraction Map

Let us define a contraction map $T_s\colon U^n \to U^n$ such that, $\delta(T_s(\mathbf{u}),\, T_s(\mathbf{v})) \leq s\delta(\mathbf{u},\, \mathbf{v})$ for all $\mathbf{u}, \mathbf{v} \in U^n$ and $0 \leq s < 1$. Since T_s is a fuzzy valued contraction map defined over a collection of fuzzy sets, the system of iterated T_s is a discrete dissipative fuzzy dynamical system and hence must have an attractor. This is an IFZS, where T_s consists of N contraction mappings $\{w_i\}_{i=1}^N$

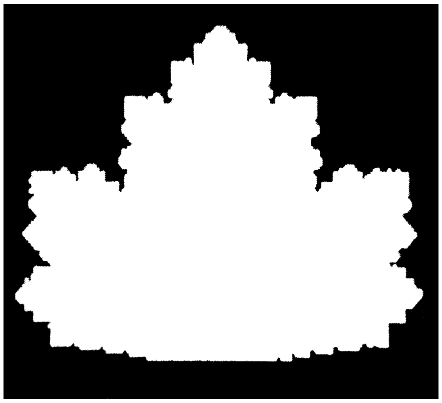

Fig. 1. The attractor of the IFZS $f = (w_1, \ldots, w_N, \mu(w_1), \ldots, \mu(w_N))$. Each $w_i\colon \Im \to \Im$ is a contraction mapping. $\mu(w_i(x)) = \varphi_i(\mathbf{u}(w_i^{-1}(x)))$ for all $x \in X$. Note that $x \in \Im$ also. (Adopted from [3].)

$$T_s(\mathbf{u}(x)) = \sup\{\varphi_1(\mathbf{u}'(w_1^{-1}(x))),\, \ldots,\, \varphi_N(\mathbf{u}'(w_N^{-1}(x)))\} \quad (2.2)$$

where

$$\begin{cases} \mathbf{u}'(B) = \sup\ \mathbf{u}(y)|y \in B \subseteq X\}, & \text{if } B \neq \emptyset \\ = 0, & \text{if } B = \emptyset. \end{cases} \quad (2.3)$$

Obviously, $\mathbf{u}'(\{x\}) = \mathbf{u}(x)$ for any $x \in X$. Since IFZS of iterated T_s must have an attractor

$$\lim_{n \to \infty} T_s^n(\mathbf{u}) = \mathbf{v}. \quad (2.4)$$

$\mathbf{v}$ is the fuzzy attractor of the IFZS, which is the generated fuzzy (fractal) image. In the next subsection we shall be generating two such fuzzy fractal images.

Equivalently, we can take $f\colon \Im \to \Im$ of Definition 2.3 as the image generating IFZS, where

$$\mu(w_i(x)) = \varphi_i(\mathbf{u}(w_i^{-1}(x))) \quad (2.5)$$

for all $x \in X$ and $\mathbf{u} \in \Im$. When $\mu(w_i(x)) \in$ [0, 1] is a constant function for $i \in \{1, \ldots, N\}$, such that, $\sum_{i=1}^N \mu(w_i) = 1$, that is, $\mu(w_i)$ is a random (probability) measure then the IFZS is simply crisp random hyperbolic IFS as described in [1] (Definition 2.2) and the fuzzy fractal attractor reduces to crisp fractal attractor, that is, the crisp fractals are only special cases of fuzzy fractals.

One important thing to note here is that in case of fuzzy fractals the determination of φ_i, for each i, is the most computationally intensive part. Compared to the crisp fractals as generated by a contractive IFS, the generation of a fuzzy fractal in terms of an IFZS is computationally more complex due to the complexity in determining the grey level pixel intensity values in terms of φ_i.

D. Examples

In the two examples considered in this subsection the base space is $X =$ [0, 1] × [0, 1]. Computer approximation of the attractors of the IFZS has been shown as normalized grey level distribution. The brightness value t_{ij} of a pixel p_{ij} representing a point $x \in X$ obeys $0 \leq t_{ij} \leq 1$, with $t_{ij} = \mathbf{v}(x)$, where $\mathbf{v}$ is as defined by the equation (2.4). $t_{ij} = 0$ if x is in the background.

Example 2.1[3]: $N = 4$.

$$w_1((x_1,\, x_2)^T) = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.04 \end{bmatrix}$$

$$w_2((x_1,\, x_2)^T) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.25 \\ 0.4 \end{bmatrix}$$

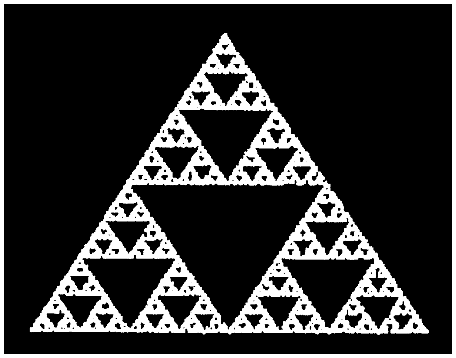

Fig. 2. Sierpinski's gasket generated as the fractal attractor of the IFZS described above. The starting point is on the gasket.

$$w_3((x_1,\, x_2)^T) = \begin{bmatrix} 0.355 & -0.355 \\ 0.355 & 0.355 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.266 \\ 0.078 \end{bmatrix}$$

$$w_4((x_1,\, x_2)^T) = \begin{bmatrix} 0.355 & 0.355 \\ -0.355 & 0.355 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.378 \\ 0.434 \end{bmatrix}.$$

To define a base space IFZS $f = (w_1,\, \ldots,\, w_N,\, \mu(w_1),\, \ldots,\, \mu(w_N))$, whose attractor $A \subset X$ is shown in Fig. 1, we consider the identity grey level intensity maps given by $\varphi_i(t) = t$ for $i \in \{1,\, \ldots,\, 4\}$. Since $\varphi_i(1) = 1$ for $i \in \{1,\, \ldots,\, 4\}, \mathbf{v} = \chi_A$, that is, $\mathbf{v}(x) = 1$ if $x \in A$ and $\mathbf{v}(x) = 0$ if $x \notin A$. The attractor $\mathbf{v}$ is shown in Fig. 1.

Example 2.2: $N = 3$.

$$w_1((x_1,\, x_2)^T) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.005 \\ 0.005 \end{bmatrix}$$

$$w_2((x_1,\, x_2)^T) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.005 \\ 0.255 \end{bmatrix}$$

$$w_3((x_1,\, x_2)^T) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.125 \\ 0.255 \end{bmatrix}.$$

To define a base space IFZS $f = (w_1,\, \ldots,\, w_N,\, \mu(w_1),\, \ldots,\, \mu(w_N))$, whose attractor $A \subset X$ is shown in Fig. 2, we consider the constant grey level intensity maps given by $\varphi_i(t) = 1/3$ for $i \in \{1,\, 2,\, 3\}$. Since $\varphi_i(1) = 1$ for $i \in \{1,\, \ldots,\, 4\}$, $\mathbf{v} = \chi_A$, that is, $\mathbf{v}(x) = 1$ if $x \in A$ and $\mathbf{v}(x) = 0$ if $x \notin A$. The attractor $\mathbf{v}$ has been shown in Fig. 2. Note that here $\varphi_i(t)$ is a probability distribution function and the generated attractor is the Sierpinski's gasket, which is a crisp fractal. Definition 2.3 and Definition 2.5 have been formulated in such a manner that the crisp fractal of Fig. 2 becomes only a special case of fuzzy fractal. This is true in general.

III. Fuzzy Turbulence

In this section we will be concerned about another novel application of fuzzy set theory based techniques in general and FDS in particular. Here we have proposed a simple model of an idealized turbulence. Turbulence is considered as one of the most complex natural phenomena, which has defied persistent efforts to reveal its mysteries by some of the most brilliant fluid dynamicists. Here by no means we claim to have solved this problem. We have made only a humble attempt to model an ideal turbulence in terms of an FDS. The model is so idealistic that it may not have any significant practical value. However our purpose in this section is not to delve deep into the mystery of turbulence but only to indicate the potentiality for application of fuzzy dynamical systems. This idealistic model at least serves this purpose.

A. Preliminary Discussion

Turbulence occurs in a moving fluid in the form of a spontaneous yet completely irregular appearance and disappearance of vortices. Despite relentless efforts of last more than hundred years, particularly by the fluid dynamicists, turbulence has not yet been completely understood. Naturally there is no mathematical model yet which can completely describe a turbulence. "Historically, many models have been proposed and many are currently in use. It is important to appreciate that there is a broad range of turbulent flows and also a broad range of questions to be addressed. Consequently it is useful and appropriate to have a broad range of models, that vary in complexity, accuracy and other attributes" [16]. An usual description of turbulence starts with the Navier–Stokes equations describing the motion of the underlying body of the fluid [16]. Here we have not considered Navier–Stokes equations. In this section starting from a simple definition of turbulence we shall try to present a mathematical model of an ideal turbulence in terms of an FDS. This may serve as yet another example to emphasize the potentiality of FDS to model various complex real life phenomena. To start with a definition of turbulence let us state.

Definition 3.1 (Brown [2]): Turbulence is chaotic occurrence of vortices in a dynamic fluid.

Devaney's definition of chaos has been stated in [4], [17]. For the sake of simplicity we shall keep our model only two dimensional. It is a common practice to use a discrete distribution of point vortices in the analysis of two and three dimensional flows of fluid [18]. In [18] an elliptic function (doubly periodic function in the complex plane) has been used to describe the distribution of vortices in a two dimensional channel flow. It is also a very common practice to describe the distribution of such vortices by probability distribution functions [16]. Here, we have described the distribution of vortices by a two dimensional chaotic function. We have outlined an extension of this method to three-dimensional case in the Conclusion.

We locate a two dimensional vortex by the center of its core. In a two dimensional cross section of a turbulent fluid to model the chaotic distribution of vortices we need some chaotic function. Let the two dimensional cross section be the xy-plane. According to our idealized model if in the xy-plane the coordinate of the center of a vortex is given by $(x(s),\, y(s))$ for some s then $x(s)$ and $y(s)$ are given by

$$x(s) = a.\lim_{n\to\infty} f^n(s) \quad \text{and} \quad y(s) = b.\lim_{n\to\infty} f^n(s) \tag{3.1}$$

where $f(s) = 4s(1-s)$, $s \in [0,\, 1]$ and coordinates of the vertices of the rectangle enclosing the cross-sectional area is given by $(0,\, 0)$, $(a,\, 0)$, $(a,\, b)$ and $(0,\, b)$. For practical computational purpose n can never be infinite. Only a sufficiently large value of n can be taken. Here it is important to remember that for every value of s there may not be a vortex with center of the core (or just center for short) at $(x(s),\, y(s))$ but every center of the core of a vortex will be given by $(x(s),\, y(s))$ for some s. It's a well known fact, that $x(s)$ and $y(s)$ as given by equation (3.1) are chaotic [4], [15], [17]. Here, we are only invoking these results for our modeling purpose. In actual computer simulation the value of n is finite but large, say $n \geq 20\,000$. Then we get the famous Feigenbaum diagram (Fig. 3).

We assume that the vortex centering $(x',\, y')$ [we shall write $(x',\, y')$ in place of $(x'(s),\, y'(s))$ whenever there is no risk of confusion] is generated by two local tiny fluid jets coming from different directions and colliding. We are not bothered about the origin of the wind jets here. Number of such collisions will be more at higher Reynolds number. After collision the resultant fluid jet will create a spiral like vortex under certain conditions. To understand these conditions let us concentrate into the Fig. 4. In Fig. 4 for convenience the coordinate system is polar $(r,\, \theta)$, where $r = ((x-x')^2 + (y-y')^2)^{1/2}$ and

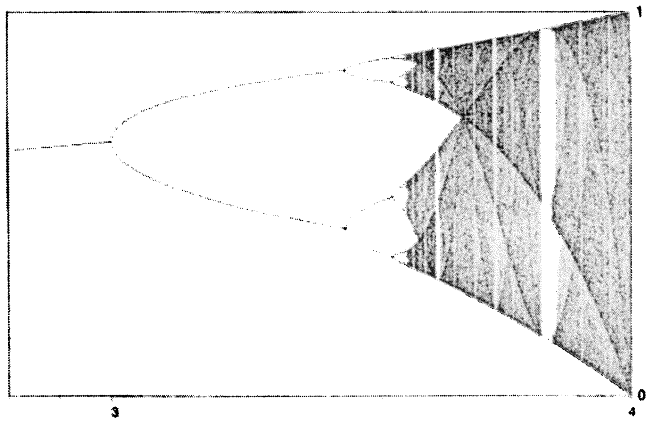

Fig. 3. Feigenbaum diagram of $x(s)$ for very large n and $a = 1$, irrespective of s, provided $s \in [0, 1]$. $f(s) = \mu s(1 - s)$, $3 \leq \mu \leq 4$. $x(s)$ is plotted along the ordinate and μ along the abscissa. For $\mu = 4$ $x(s)$ can converge on any pixel along the ordinate. Exactly the same diagram will also hold for $y(s)$. (Reproduced from [15].)

$\theta = \tan^{-1}((y - y')/(x - x'))$. On the other hand $x = x' + r.\cos\theta$ and $y = y' + r.sin\theta$.

Fig. 4 is self-explanatory. For detail explanation of radial and cross-radial component of velocity any elementary text on particle dynamics, e.g., Loney [10] may be consulted. From elementary particle dynamics we know that the expression for radial component of velocity at a point (r, θ) is dr/dt and that of the cross-radial component is $r(d\theta/dt)$ [10] with respect to some polar coordinate system. If

$$\frac{\text{Resultant radial component of velocity}}{\text{Resultant cross-radial component of velocity}} = \frac{dr/dt}{r(d\theta/dt)} = m \tag{3.2}$$

where m is a constant holds then

$$dr/d\theta = mr. \tag{3.3}$$

B. Dealing Nonlinearity With Fuzzy Quantities

In (3.3), if m is a constant then the solution of (3.3) will produce a log-spiral, which describes the shape of a vortex. But in reality m can not be constant. Instead the values of m are liable to fluctuate within some range. Note that (3.3) is a simple linear equation modeling the generation of a vortex in a dynamic fluid. But the generation of a vortex is supposed to be a nonlinear phenomenon. Then the immediate question is how is it possible to model a nonlinear phenomenon by a simple linear equation? We assume that the nonlinearity is responsible for giving rise to uncertainty in the model. Had there been no uncertainty involved in the model it would have been perfectly deterministic. In that case we could have described it with a set of linear equations. Here we take the opposite approach. We assume that the model is deterministic in the first place. So we describe it with a set of linear equations. Next, to bring it closer to the reality we try to accommodate the elements of inherent uncertainty within it. In this spirit the generation of vortex in a dynamic fluid has been described by (3.3). Our next task is to accommodate the uncertainty, inherent in the underlying nonlinearity of the system, within (3.3) (thereby bringing it closer to the real life nonlinear model). To accomplish this, we should notice that m being the ratio of two wind speeds, can not be a fixed value even for a short time. m will fluctuate within some closed bounded interval of R. In the most general case this interval is a fuzzy subset R, the membership distribution function, whose support is the interval. Without much loss of generality we may assume that this membership distribution function satisfies all the three properties (normality, upper-semicontinuity and compactness of support) to become a fuzzy number. So this fuzzy subset of R is actually a fuzzy number, say $A \in \Re$, where $\Re$ is the set of all fuzzy real numbers. Similarly, the measurement of $r(0)$, the initial value of r, is also subject to uncertainty due to measurement error [uncertainties involved in m and $r(0)$ are different, in m it is due to smoothing out the nonlinearity and in $r(0)$ it is due to measurement error]. Classically, the probability distribution of the error in measuring $r(0)$ will follow the normal distribution law. So when we are going to model this uncertainty with a fuzzy quantity we should opt for a triangular fuzzy number, say B, to describe $r(0)$, such that, $r(0) \in B \in \Re$.

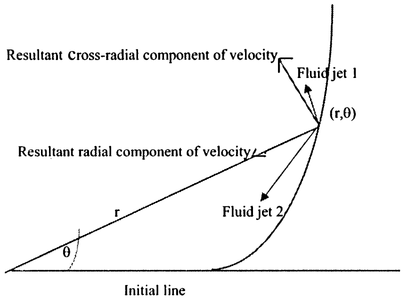
Resultant cross-radial component of velocity
Fluid jet 1
(r,θ)
Resultant radial component of velocity
r
Fluid jet 2
θ
Initial line

Fig. 4. Fluid jet 1 and fluid jet 2 are colliding at (r, θ). Their resultant radial component of velocity and resultant cross-radial component of velocity after collision have been shown. If the ratio of these two components is constant a spiral shaped vortex is generated.

TABLE I
NUMERICAL ESTIMATION OF THE CONSTANT FUZZY NUMBER A FOR A ZERO VISCOSITY FLUID IN A MODEL WITH MODERATE SHEAR

a	c	d	b
0.001	0.05	0.1	0.2

A being a fuzzy number must be increasing and continuous from right in $[a, c)$, decreasing and continuous from left in $(d, b]$, where $[a, b]$ is the support of A and $a \leq c \leq d \leq b$. Also $A([c, d]) = \{1\}$, which means that the values of highest possibility within the universe of discourse of A lies in $[c, d]$. So a trapezoidal fuzzy number $\langle a, c, d, b\rangle$ may be an approximation of A.

For the purpose of modeling, next we shall have to make some reasonable estimate for the values of a, c, d and b, which we have given in Table I. One very important thing to note that if $m \in [A]^0$ then also $m \in [-A]^0$. So $m \in [A]^0 \cup [-A]^0$. We shall take the triangular fuzzy number B as $\langle r(0) - \delta, r(0), r(0) + \delta\rangle$, where $\delta > 0$ is a crisp real number, whose value depends upon the employed measurement techniques. The more accurate the technique is the smaller is the value of δ.

In this section our aim is just to present an ideal hypothetical example to exhibit the potentiality of FDS in general and FDI in particular as a powerful tool for modeling and simulation of complex physical phenomena. Therefore we are not too bothered about the actual numerical specifications. These may be adjusted according to the outcomes of experiments to fit the model with real physical world.

So by now (3.3) is no longer an ordinary differential equation (ODE). Instead we shall have to reformulate it as a fuzzy differential inclusion (FDI) relation to make the modeling more realistic. The FDI formulation of (3.3) is described below

$$r'(\theta) \in [mr(\theta)]^\alpha, \qquad m \in [A]^\alpha,\ r(0) \in [B]^\alpha \tag{3.4}$$

where, as usual, $0 \leq \alpha \leq 1$. Similar formulation also holds for $-A$. The solution of (3.4) by Crystalline Algorithm [11] described below yields the following phase space or solution space diagram (Fig. 5), which resembles a generated vortex in a dynamic fluid. For convenience we describe step by step the solution of (3.4) below.

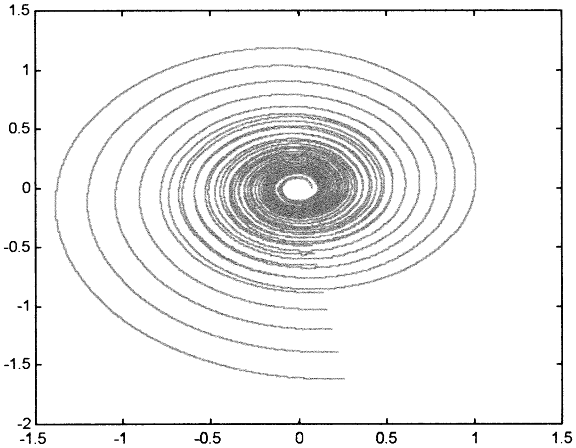

Fig. 5. The simulated diagram of a fuzzy vortex centering at (0.5, 0.5), where r_0 (in cm) belongs to a fuzzy number, with the best possible value ≤ 0.1. This diagram is the phase (solution) space of (3.4) for $\alpha = 1$. The range of θ is a random number $\leq 16\pi$. The enclosed region indicated by the spirals is the fuzzy flow representing the most possible (grade of membership 1) behavior of the system represented by (3.4). We get the spirals by putting $m \in [0.05, 0.1]$ in (3.4). If we replace A by $-A$ in (3.4) the orientation of the vortex will be reversed.

C. Simulation

Crystalline Algorithm

1) replace $\in$ by $=$ in (3.4);
2) calculate the right hand side of the equation thus obtained by Zadeh's extension principle [14];
3) fix $\alpha \in [0, 1]$;
4) take the α-level set on the right hand side, which is an n-tuple of α-cuts of fuzzy numbers for some n;
5) solve (directly or numerically) the ordinary crisp DE's only for the boundary values of the α-level set (more solutions may be derived within the region enclosed by the boundary);
6) the space enclosed by these solutions is the α-level set of the solution of the equation obtained by replacing $\in$ by $=$ in (3.4) and hence the solution of FDI (3.4) for the given α.

Fuzzy vortex is modeled by (3.4) and its chaotic occurrence within a finite rectangular cross section of area ab is given by (3.1). The simulation diagram is presented in Fig. 6, where $a = b = 1$ and $n = 20\,000$. $r(0)$ and range of θ varies from vortex to vortex. From equation (3.1) and Fig. 3 it appears that $(x(s), y(s))$ can be any pixel in [0, 1] × [0, 1]. So this model can account for all possible occurrence of vortices in [0, 1] × [0, 1]. In this sense it is the most general representation of fuzzy turbulence.

Definition 3.2: Since we get a simulated vortex in a dynamic fluid by solving the FDI (3.4), we call the vortex a *fuzzy vortex*. Chaotic occurrence of such fuzzy vortices in a dynamic fluid will be called *fuzzy turbulence*.

The MATLAB program used to simulate the turbulence in Fig. 6, has been given in [20]. Interested readers may generate more such two dimensional fuzzy turbulence using this program. This program generates turbulence in an uncontrolled manner. The probability of two turbulence generated with the same set of arguments being identical is very low, which is an essential feature of real turbulence.

The program generates a two dimensional turbulence only as a "Chaotic occurrence of fuzzy vortices." Here we have not taken care of the detail of the fluid and its motion in generating the turbulence. In this section our aim is only limited to indicating how FDS can be used to model a turbulence. A vast scope of improvement in this simulation is remaining open. For example the program can be so written to take arguments like "viscosity" and "Reynold's number," which represents the fluid and its flow to a good extent.

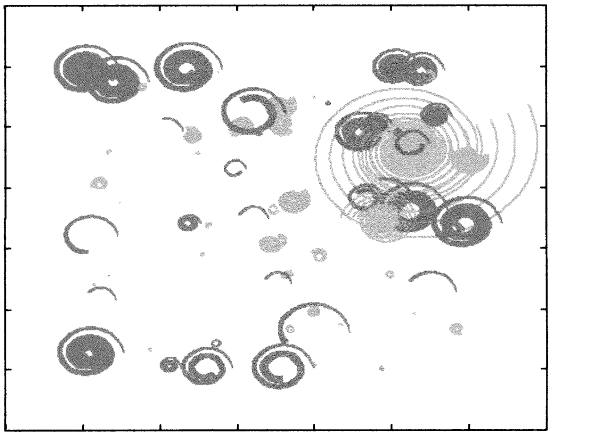

Fig. 6. Simulation of a two dimensional fuzzy turbulence in a two dimensional dynamic fluid. Light and dark vortices have different orientation.

IV. Summary and Conclusion

In this paper fuzzy fractals have been introduced. We have defined fuzzy fractals. Then we have shown that fuzzy fractals can be generated by iterated fuzzy sets systems as attractors. We have treated any grey level image as a fuzzy subset of R^2, where the membership value at each point (pixel) is a normalized grey level pixel intensity value at that point (pixel). This approach due to Cabrelli *et al.* [3] is possibilistic in nature. Where as the classical approach of Barnsley [1] was probabilistic in nature. Here we have achieved an unification of these two approaches and have shown that all fractals generated in the latter method can be generated in the former method also. It is worthy to note that there does not exist any bijective transformation between the two satisfying all the required properties [21]. Kandel observed that [22], "… impossibility implies improbability but not vice versa." This means probability implies possibility (but not vice versa). In Section II we have exactly utilized this premise and assigned each probability measure a possibility measure. As an implementation of the theory developed we have considered two examples of fuzzy fractal image generation.

Here a hypothetical mathematical model of turbulence is proposed. By the term "turbulence" we have meant chaotic occurrence of vortices in a dynamic fluid. We have kept the model two dimensional for simplicity. We have defined a two dimensional chaotic function, which gives the distribution of the vortices. It is assumed that the vortices have been created by collision of fluid jets coming from different directions under certain conditions. To model a vortex, a fuzzy differential inclusion relation has been formulated by considering the ratio of radial and cross-radial components of the resultant velocity of the colliding fluid jets at a point as a constant fuzzy number. The solution space of this FDI, obtained by the Crystalline Algorithm gives the simulation of a vortex. We call this a fuzzy vortex. Chaotic occurrence of such fuzzy vortices gives rise to fuzzy turbulence. We have presented a simulated model of a fuzzy turbulence.

Very little work has so far been undertaken in fuzzy fractal based image generation. The pioneering work in this direction is due to Cabrelli *et al.* [3]. We have introduced some modification and extension into it to make it easier to implement. However in terms of computational complexity both the methods will perform at the same level. But the whole method is for the grey level images only, where the values of intensity of red, green and blue (RGB) at each pixel are equal and measured by integral values from 0 to 255. A very challenging work for the future researchers will be to extend it to the color images also. This means that the intensity values of red, green and blue will be independent of each other at any pixel. Each point (pixel) P on the image will have three membership grades instead of usual one, namely μ_r, μ_g and μ_b, where $\mu_r(P)$ is the membership grade of P to belong to the red region,

$\mu_g(P)$ is the membership grade of P to belong to the green region and $\mu_b(P)$ is the membership grade of P to belong to the blue region. Values of $\mu_r(P)$, etc. may be determined the way described in Section II.

As far as we know fuzzy turbulence is a completely new concept, introduced only in this paper. To be more realistic this model should be three dimensional or if we also want to incorporate time into it, it will be a four dimensional space time model. In this four dimensional space the location of the center of a vortex will be denoted by $(x,\ y,\ z)$ at some instant t. The vortex will be of cylindrical shape, whose axis may make angles $\alpha,\ \beta$ and γ respectively with $x,\ y$ and z axes. Obviously, $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$. To describe the vortex at $(x,\ y,\ z)$ a local cylindrical coordinate system $(r,\ \theta,\ h)$ is necessary. Also another parameter m as described in Section III is needed. r will be expressed in terms of m and θ. So to make a more realistic model of this type of turbulence we will need the following independent parameters: $x,\ y,\ z,\ t$, any two out of $\alpha,\ \beta$ and γ and m, θ and h, i.e., nine in all. Modeling of turbulence involving all these nine independent parameters (some of them including m are fuzzy numbers) will be the subject of future research.

The author has recently proposed a model of generation of intense tropical storms over the seas in terms of fuzzy dynamical systems [23]. It has long been suspected that atmospheric turbulence plays an important role behind generation of such storms. The model of turbulence proposed here and the model of generation of storms proposed in [23] have one very common aspect, namely the "fuzzy vortex." This, at least to some extent, supports the validity of the hypothesis of generation of storms from atmospheric turbulence. In near future the author has plans to work toward a modeling of generation of storms from atmospheric turbulence.

Acknowledgment

All the simulation works in this paper have been performed using MATLAB. Reference [7] has turned out to be very useful in this regard. I would also like to acknowledge Prof. H. P. Majumdar for his appreciation and encouragement. The role of the comments of three anonymous referees in improving the paper is most sincerely acknowledged.

References

[1] M. F. Barnsley, *Fractals Everywhere*. New York: Academic, 1988.

[2] R. A. Brown, *Fluid Mechanics of the Atmosphere*. New York: Academic, 1991.

[3] C. A. Cabrelli, B. Forte, U. M. Molter, and E. R. Vrscay, "Iterated fuzzy sets systems: A new approach to the inverse problem for fractals and other sets," *J. Math. Anal. Applicat.*, vol. 171, pp. 79–100, 1992.

[4] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*. Reading, MA: Addison-Wesley, 1989.

[5] D. Dutta Majumder and K. K. Majumdar, "Complexity analysis, uncertainty management and fuzzy dynamical systems: A cybernetic approach with some case studies," Kybernetes, submitted for publication.

[6] G. A. Edgar, *Measure, Topology, and Fractal Geometry*. New York: Springer-Verlag, 1990.

[7] L. V. Fausett, *Applied Numerical Analysis Using MATLAB*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[8] J. E. Hutchinson, "Fractals and self similarity," *Indiana Univ. Math. J.*, vol. 30, pp. 713–747, 1981.

[9] T. Kapitaniak and S. R. Bishop, *The Illustrated Dictionary of Nonlinear Dynamics and Chaos*. New York: Wiley, 1999.

[10] S. L. Loney, *An Elementary Treaties on the Dynamics of a Particle and of Rigid Bodies*. Madras, India: Macmillan India, 1972.

[11] K. K. Majumdar, "One dimensional fuzzy differential inclusions," *J. Intell. Fuzzy Syst.*, vol. 13, pp. 1–5, 2003.

[12] B. B. Mandelbrot, *The Fractal Geometry of Nature*. New York: Freeman, 1982.

[13] J. Milnor, "On the concept of attractor," *Comm. Math. Phys.*, vol. 99, pp. 177–195, 1985.

[14] H. T. Nguyen, "A note on the extension principle for fuzzy sets," *J. Math. Anal. Applicat.*, vol. 64, pp. 369–380, 1978.

[15] H.-O. Peitgen, H. Jurgens, and D. Saupe, *Chaos and Fractals: New Frontiers of Science*. New York: Springer-Verlag, 1992.

[16] S. B. Pope, *Turbulent Flows*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[17] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. Boca Raton, FL: CRC, 1999.

[18] S. K. Venkatesan, "Some vortex algorithms for 2D Euler flows," in *Some Applied Problems in Fluid Mechanics*, H. P. Majumdar, Ed. Calcutta, India: Indian Statist. Inst., 1993.

[19] L. A. Zadeh, "Fuzzy sets as a basis for theory of possibility," *Fuzzy Sets Syst.*, vol. 1, pp. 3–28, 1978.

[20] [Online]. Available: http://www.isical.ac.in/~ecsu/kausik.htm

[21] T. Sudkamp, "On probability-possibility transformations," *Fuzzy Sets Syst.*, vol. 51, pp. 73–81, 1992.

[22] A. Kandel, *Fuzzy Techniques in Pattern Recognition*. New York: Wiley, 1980.

[23] K. K. Majumdar, "A mathematical model of the nascent cyclone," *IEEE Trans. Geosci. Remote Sensing*, to be published.

Obstacle Avoidance for Kinematically Redundant Manipulators Using a Dual Neural Network

Yunong Zhang and Jun Wang

***Abstract*—One important issue in the motion planning and control of kinematically redundant manipulators is the obstacle avoidance. In this paper, a recurrent neural network is developed and applied for kinematic control of redundant manipulators with obstacle avoidance capability. An improved problem formulation is proposed in the sense that the collision-avoidance requirement is represented by dynamically-updated inequality constraints. In addition, physical constraints such as joint physical limits are also incorporated directly into the formulation. Based on the improved problem formulation, a dual neural network is developed for the online solution to collision-free inverse kinematics problem. The neural network is simulated for motion control of the PA10 robot arm in the presence of point and window-shaped obstacle.**

***Index Terms*—Dual neural network, obstacle avoidance, quadratic programming, redundant manipulators.**

I. Introduction

Robot manipulators have been applied in factory automation doing repetitive and dull work, such as carrying radioactive materials and working in hazardous or cluttered environments. It is very important for a robot manipulator to avoid collisions with obstacles; otherwise, the manipulators or the object being held may result in serious damage. Unlike nonredundant manipulators that may not have the ability to avoid obstacles while completing the specified end-effector motion, redundant manipulators with extra degrees of freedom (DOF) [4] may be utilized to improve their dexterity such that they can work effectively while avoiding obstacles [5]–[16].

Many studies have been reported on the obstacle avoidance issue. For example, the pseudoinverse method and its variants [1], [5]–[7] generally yield an minimum-norm particular solution plus a homogeneous solution. A goal of obstacle avoidance can then be specified to

Manuscript received September 25, 2002; revised February 19, 2003. This work was supported by the Hong Kong Research Grants Council under Grant CUHK4165/98E. This paper was recommended by Associate Editor M. S. de Quiroz.

The authors are with the Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Digital Object Identifier 10.1109/TSMCB.2003.811519