Matching probability measures on the line under translation

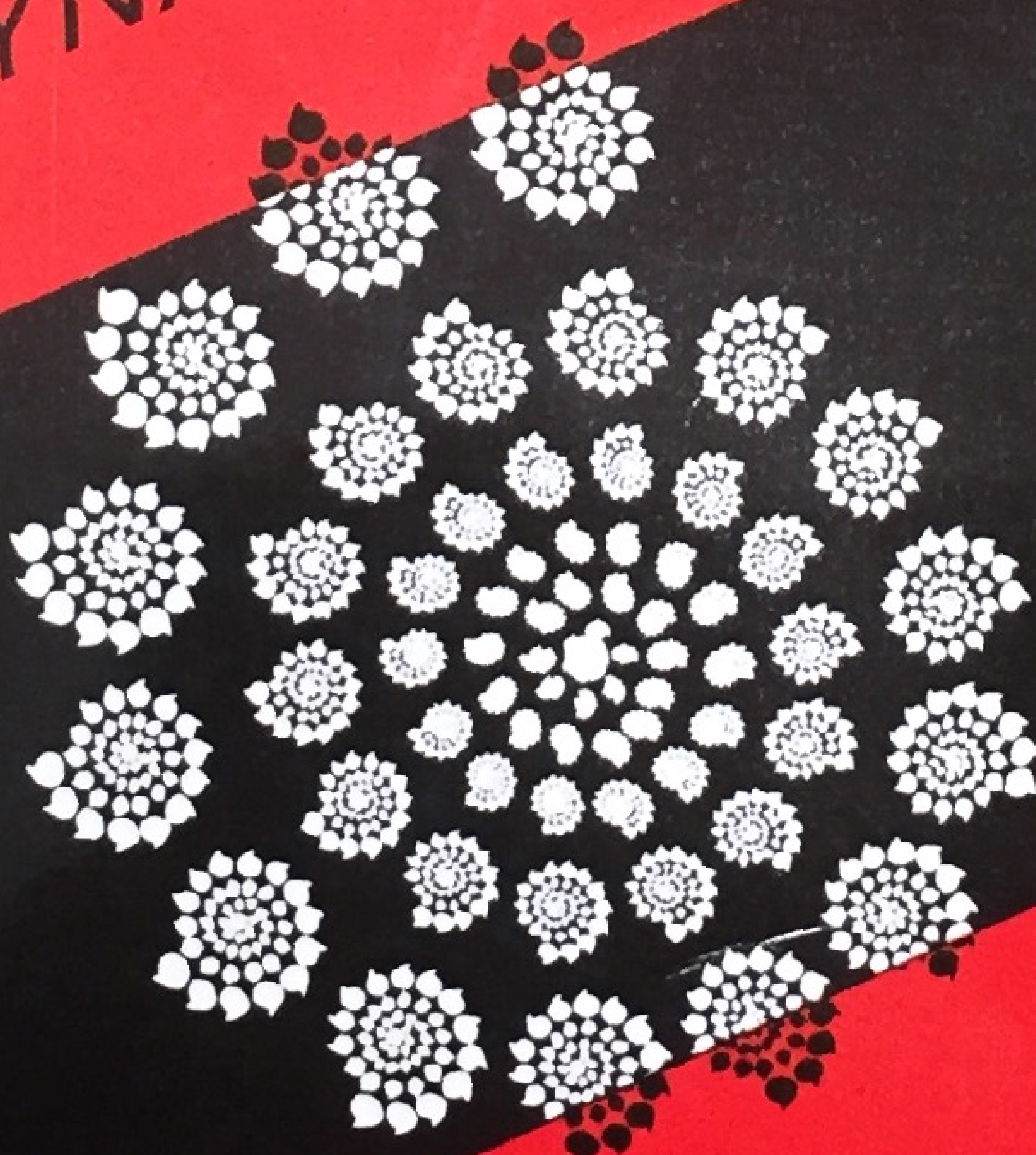
Article · January 1995

3 authors, including:

Francisco Barahona
IBM

Carlos Cabrelli
University of Buenos Aires and CONICET (ARGENTINA)

RANDOM & COMPUTATIONAL DYNAMICS



VOLUME 1 NUMBER 1 1992-93

RANDOM & COMPUTATIONAL DYNAMICS, 1(1), 117-129 (1992-93)

COMPUTING THE HUTCHINSON DISTANCE BY NETWORK FLOWS

Francisco Barahona
IBM Watson Research Center
P.O.Box 218
Yorkton Heights
NY 10598

Carlos A. Cabrelli*
Department of Applied Mathematics
University of Waterloo,
Waterloo, ON, Canada N2L 3G1

Ursula M. Molter*
Department of Applied Mathematics
University of Waterloo
Waterloo, ON, Canada N2L 3G1

Abstract

We show, that the Hutchinson distance between discrete probability measures on spaces of finite dimension can be computed as a network flow problem, and then be solved in polynomial time by efficient combinatorial methods. In the one dimensional case a linear time method had been obtained earlier by Brandt et al. in [12]. This new setting allows to compute this metric for higher dimensions. We also show that the one dimensional formula has its counterpart in the continuous case.

*Permanent address: Department of Mathematics, Universidad de Buenos Aires, Pab.I Cdad.Universitaria, (1428) Cap.Fed., ARGENTINA.

1 Introduction

The problem of comparing two digital images has been widely studied, and generally requires a considerable amount of computation, cf. [17], and references therein. If we are only interested in differences of the geometry or structure of the images, the problem can be reduced to a comparison between two black and white images where several metrics that perform such matching satisfactorily are known. In particular the Hausdorff metric has proven to be very good in the analysis of shape. A very recent algorithm has been developed for the computation of this metric and it works in linear time on the number of pixels [21].

In the case of comparing gray level images, when comparing textures for example, we have to consider more complex models. Measures have been used extensively to describe colour and gray level pictures. The way in which a picture can be associated with a measure is in the sense that it represents the intensity of light reflected by each region of the picture; e.g. regions including black portions and dark objects will be assigned less measure, opposite to that for brilliant objects and white parts. Sometimes it is convenient to normalize this measure in order to perform a better comparison.

Self-similar sets, e.g. sets that have some kind of scale invariant properties, or sets with non integer dimensions, have been studied for many years. It was Mandelbrot [18] however, who called these sets 'fractals' and pointed out the relevance of them to many fields. Fractal geometry, [16], [9], is now very often used in computer science, specially in the analysis, synthesis and representation of digital images.

Measures represent a very useful model in representing images using methods from fractal theory. Fractal measures, that started as the study of measures supported on fractals, are showing to be a more complete model for the description of certain objects and the modelling of certain phenomena. In digital image processing, fractal geometry is used to model shape, and fractal measures to render textures.

Further development and some very new research is showing that the concept of measures on fractals as well as the concept of multifractality (related with the sets of points of certain mass density from a fractal measure) represent a very promising aproach (cf [14, 23]).

On the other side, fractal methods provide a very powerful tool to generate measures. These measures are obtained as a limit process, iterating certain Markov operator in a suitable space of measures.

Barnsley et al. [6], have shown that the ergodic properties of such process lead to a surprising algorithm to generate images. The fractal model is formed by a set of contractive maps and some asociated probabilities (iterated function systems). The successive aplication of these maps, chosen according to its probabilities, generate a measure that represents the image. For each set of maps $\{w_i\}$ and probabilities p_i, $i = 1, 2, \ldots, N$, $\sum_{i=1}^{N} p_i = 1$, there exists a unique compact set $\mathcal{A}$, invariant under the "parallel action" $\bigcup_{i=1}^{N} w_i(\mathcal{A}) = \mathcal{A}$; as well as a unique invariant measure μ with support on $\mathcal{A}$. One of the most important advantages of this representation is the very high data compression rate that can be obtained.

In [10],[11] and [5] it is shown that the Markov chains obtained using this probabilistic algorithm, can be mixed in order to enrich the class of images that can be represented through this model.

An interesting problem asociated with fractal measures is the "inverse problem of fractals", where an IFS (iterated function system) has to be determined in order that the measure it defines approximates a given target image (see for example [8],[15], [24]).

One of the most remarkable results in that direction is given by the collage theorem: If a set or measure can be tiled with copies of itself to an arbitrary accuracy, then it is close in metric to the attractor or invariant measure of the IFS which produces the tiling. (See [4] for the precise mathematical formulation.)

Since most of the results obtained for IFS's are consequence of the contractivity of the maps, the election of suitable metrics plays a very important role in the setting of the model. In the case of digital images, the selected metric has to take into account what is meant by 'visually close'. Different metrics will give different notions of closeness.

The Hutchinson distance provides us with a tool that not only takes into account the 'visual closeness' of two pictures, but also allows formalization of the theory of IFS for measures. Some of the advantages of this metric have been pointed out in [22].

In the computational setting (for the direct, as well as for the inverse problem) a numerical calculation of the distance is needed in order to estimate the error. Hence, a fast algorithm to calculate this distance is an essential tool. For the one-dimensional case the Hutchinson distance admits a very simple representation [12] from which a linear time algorithm is derived. This representation is no longer true in dimension 2 or more.

Let us remark here, that although the 2 dimensional case is the most important for images, there are applications such as fractal interpolation where higher dimensional IFS are considered ([3],[7],[19]).

In [22] Stark proposed to use neural networks in order to obtain an approximation to the Hutchinson distance. In this paper, we are presenting an efficient combinatorial approach to compute the *exact* value of this distance.

The object of this article is to show that the problem of computing the Hutchinson distance in any dimension can be set as a network flow problem and then be solved in polynomial time with efficient combinatorial methods.

We include a result that extends the one dimensional formula given in [12] to the continuous case and show that this formula does not hold in general for dimensions bigger than 1.

2 The Hutchinson Distance

Since in this paper we are concerned with the computational aspect of this problem, let us first consider the discrete k-dimensional case. Therefore, given an integer $N > 1$, we consider the metric space $X = \{1, \dots, N\}^k$ with $d(x, y) = \sum_{i=1}^{k} |x_i - y_i|$, for all $x = (x_1, \dots, x_k)$, $y = (y_1, \dots, y_k) \in X$. For the special case $k = 2$, each $x = (x_1, x_2) \in X$ represents a pixel of the image. If

$$\mathcal{M}(X) = \{\mu : X \longrightarrow [0,1] \ : \ \sum_{x \in X} \mu(x) = 1\}$$

is the set of discrete probability measures, then, for $\mu, \nu \in \mathcal{M}(X)$ the Hutchinson distance, $d_H(\mu, \nu)$ is defined as:

$$d_H(\mu, \nu) = \sup_{f \in C} \left(\sum_{x \in X} f(x)\mu(x) - \sum_{x \in X} f(x)\nu(x) \right) \tag{1}$$

where $C = \{f : X \to \mathbb{R} \ : \ |f(x) - f(y)| \le d(x, y) \quad x, y \in X\}$ (see [16]).

The set of constraints in (1) which is of the order of $(N^k)^2$ can be reduced to the following set of equations:

$$|f(x + e_j) - f(x)| \le 1\,, \quad 1 \le j \le k, \quad x \in X \text{ and } x + e_j \in X, \tag{2}$$

where e_j is the k-dimensional vector that is 0 in each coordinate except for the coordinate j where it is 1. This leaves us with $k(N-1)N^{k-1}$ constraints.

This model is nothing but the discretization of the continuous case: Let (X, d) be a complete metric space, and $\mathcal{M}(X)$ the set of probability measures on X (e.g. $\mathcal{M}(X)$ is the set of Borel regular measures having bounded support and unit mass). We define for all μ, ν in $\mathcal{M}(X)$

$$d_H(\mu, \nu) = \sup_{f \in Lip_1} \left\{ \int_X f d\mu - \int_X f d\nu \right\} \tag{3}$$

where

$$Lip_1 = \{f : X \longrightarrow \mathbb{R} \ : \ Lip(f) \leq 1\} \ ,$$

and

$$Lip(f) = \sup_{x \neq y} \frac{|f(x) - f(y)|}{d(x,y)} \ .$$

Let us point out, that due to the fact that

$$f \in Lip_1 \Longleftrightarrow f + c \in Lip_1,$$

$$\text{and} \int_X f d\mu - \int_X f d\nu = \int_X (f+c) d\mu - \int_X (f+c) d\nu \ , \ c \in \mathbb{R}, \qquad (4)$$

the supremum in equation (3), can be taken over a subset of Lip_1.

3 One-Dimensional Hutchinson distance

In the one-dimensional case $X = \{1, \ldots, N\}$ and equation (1) becomes:

$$d_H(\mu,\nu) = \sup_f \left\{ \sum_{i=1}^{N} f_i(\mu_i - \nu_i) \ : \ |f_i - f_{i+1}| \leq 1 \ \ i = 1, \ldots, N-1 \ , f_1 = 0 \right\},$$

with $\mu = (\mu_1, \ldots, \mu_N)$, $\nu = (\nu_1, \ldots, \nu_N)$ such that $\Sigma_{i=1}^N \mu_i = \Sigma_{i=1}^N \nu_i = 1$. The additional condition $f_1 = 0$ comes from (4).

A very simple formula for this equation has been derived [12]:

$$d_H(\mu,\nu) = \sum_{\ell=1}^{N-1} |S_\ell|$$

$$\text{where} \quad S_\ell = \sum_{i=1}^{\ell} (\mu_i - \nu_i).$$

From this formula one can see that in order to determine the Hutchinson distance between μ and ν we need only compute S_ℓ for $\ell = 1, \ldots, N-1$.

This formula can be generalized to the continuous case. If μ and ν are two Borel regular measures on $[0,1]$, we can think of S_ℓ of equation (3) above as being the distribution function F_η of the signed measure $\eta = \mu - \nu$. That is $F_\eta(x) = \eta([0,x]) = \mu([0,x]) - \nu([0,x])$. Formula (3) can then be written as an integral.

Let us therefore state the following theorem which we will prove in the appendix.

Theorem 3.1 *If $X = [0,1]$ and $\mathcal{M}(X)$ is the set of all Borel regular measures on X, then the Hutchinson distance d_H between two measures μ and ν in $\mathcal{M}(X)$ as defined in (3) becomes:*

$$d_H(\mu,\nu) = \int_0^1 |F_\eta(x)|dx$$

$$\text{where } \eta = \mu - \nu \quad \text{and } F_\eta(x) = \eta([0,x]).$$

Unfortunately this is no longer true, neither in the continuous nor in the discrete case for higher dimentions as is shown in the appendix. We have been able, however to reformulate the problem as a network flow problem, which can then be solved (in any dimension) in polynomial time.

For the convenience of the reader, we will concentrate on the two-dimensional case.

4 Network Flows

Network flow problems are linear programs like

$$\begin{array}{c} \text{minimize } cx \\ \text{subject to} \\ Ax = b\ , \quad x \geq 0 \end{array} \tag{5}$$

where each column of the matrix A has exactly one coefficient equal to 1, exactly one equal to -1 and all other coefficients equal to 0. It follows that the components of the vector b should sum up to zero. A reference for linear programming is the book of Chvátal [13] and a survey of network flow algorithms has been published by Ahuja et al. in [1] and [2].

One can associate a graph to the matrix A as follows: To every row i we associate a node. If a column has a -1 in row k and a 1 in row ℓ we associate an arc from node k to node ℓ, the value of the variable corresponding to this column is called the flow from k to ℓ. The value b_i associated with row i is the *net flow into i*. Problem (5) consists of finding a flow of minimum cost. For example the graph in Figure 1 corresponds to the matrix A below.

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

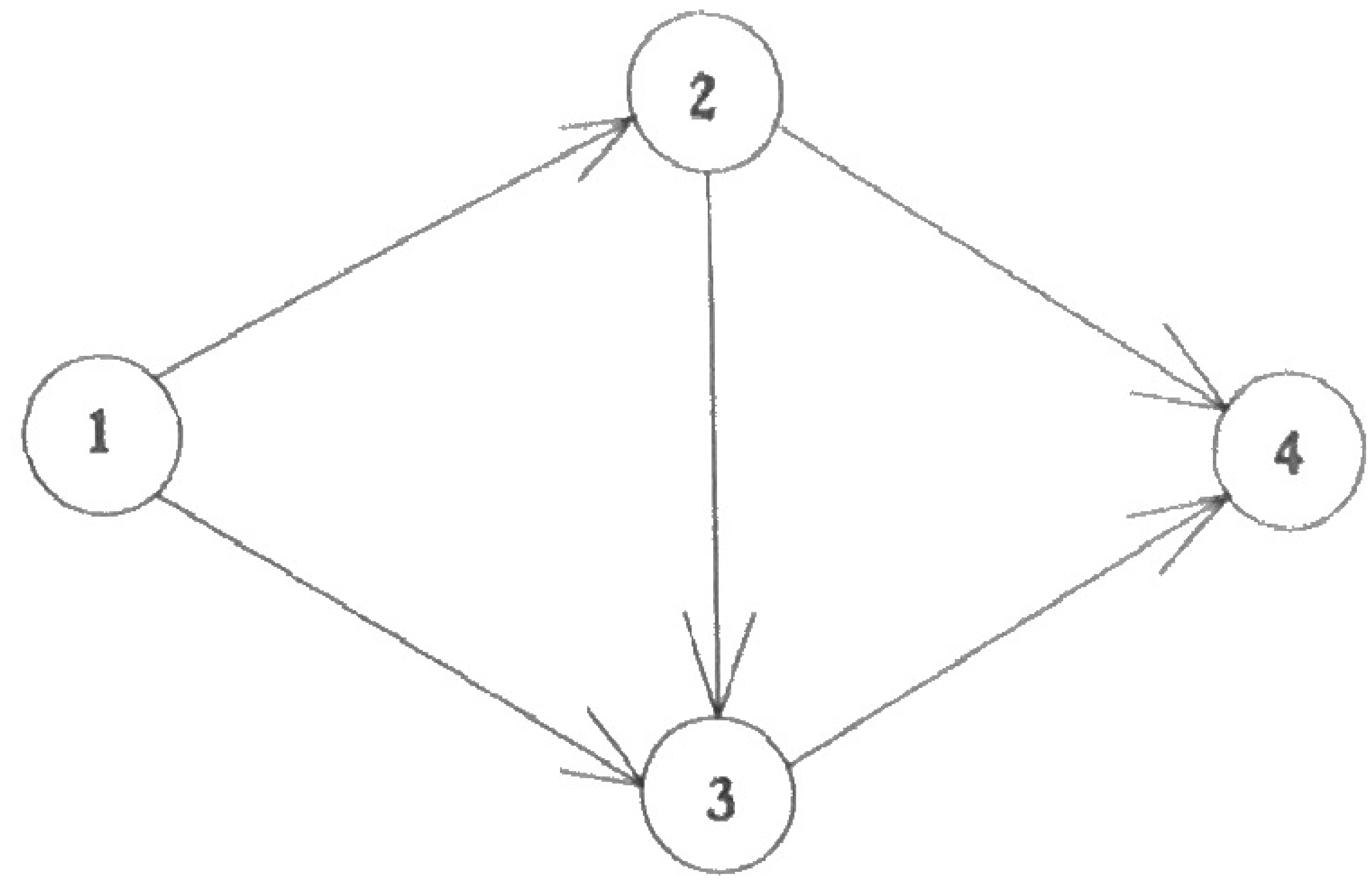


Figure 1

The dual problem is always of great significance: The dual problem of (5) is

$$\begin{gathered}\text{maximize } b^t y\\ \text{subject to}\\ A^t y \leq c^t\end{gathered} \tag{6}$$

So (6) is a linear program where every inequality has a 1 and a -1 coefficient, and the components of the objective function sum up to zero.

In the last few years, we have seen a growth in the variety of available network flow algorithms, for instance the algorithm of Orlin [20] requires in the worst case $O((p \log p)(m + p \log p))$ operations, if A is a $p \times m$ matrix. From the empirical point of view, the *network simplex method* has shown a very good performance.

5 Reduction to a network flow problem

The two-dimensional problem given by Equation 1 can be formulated as:

$$\underset{f}{\text{maximize}} \left\{ \Sigma\, f_{ij}(\mu_{ij} - \nu_{ij}) : \begin{array}{ll} |f_{ij} - f_{i+1,j}| \leq 1, & {\scriptstyle i=1,\ldots,N-1;j=1,\ldots,N;} \\ |f_{ij} - f_{i,j+1}| \leq 1, & {\scriptstyle i=1,\ldots,N;j=1,\ldots,N-1;} \end{array} \right\}.$$

This can also be written as

$$\begin{gathered}\text{maximize } \Sigma\, f_{ij}(\mu_{ij} - \nu_{ij})\\ \text{subject to}\\ -1 \leq f_{ij} - f_{i+1,j} \leq 1, i = 1, \ldots, N-1; j = 1, \ldots, N;\\ -1 \leq f_{ij} - f_{i,j+1} \leq 1, i = 1, \ldots, N; j = 1, \ldots, N-1.\end{gathered} \tag{7}$$

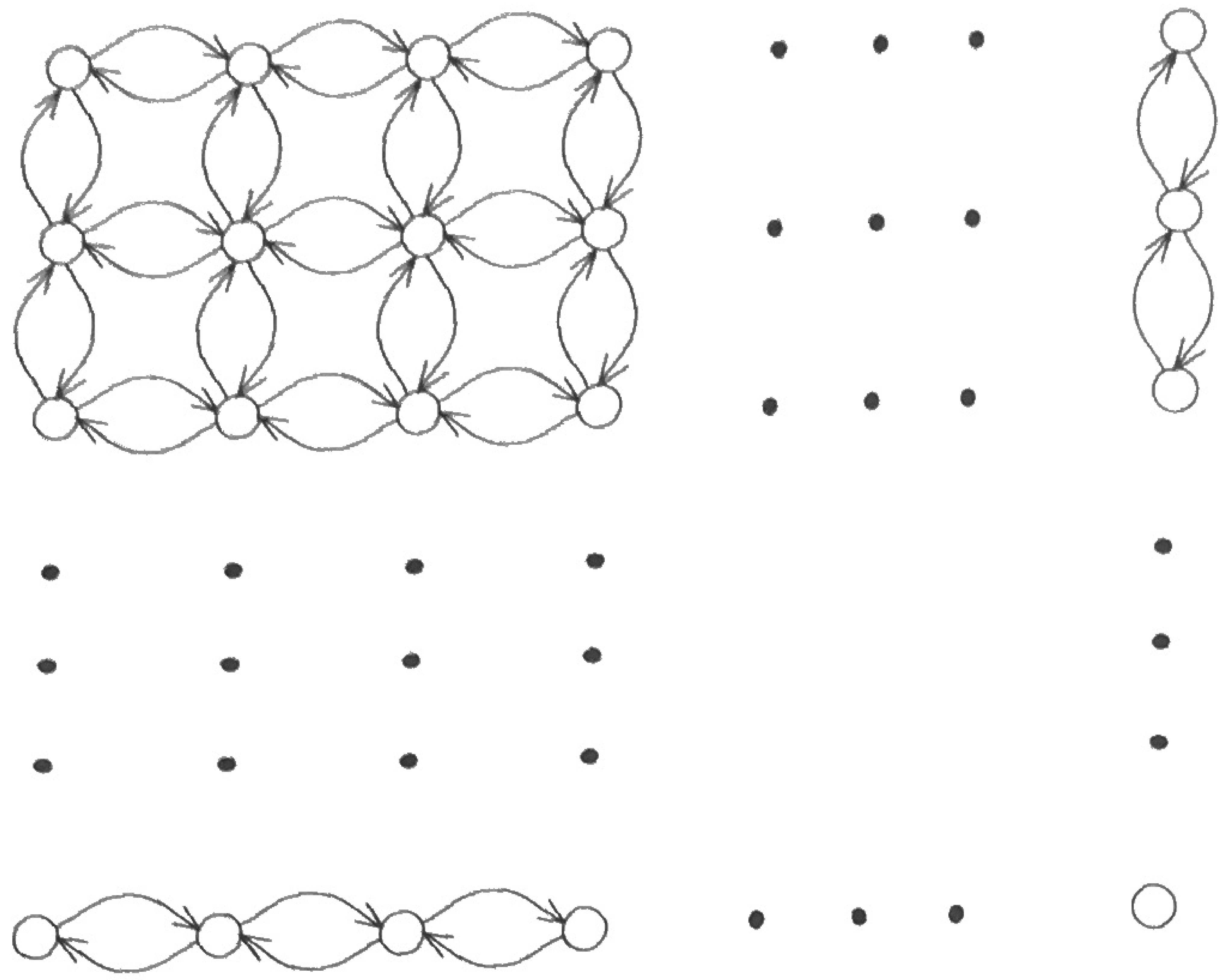


Figure 2

Figure 3

This is a linear program whose inequalities have a 1 and a -1 coefficient, and the sum of the coefficients in the objective function is zero, i.e., this is the dual of a network flow problem. The associated graph is the grid in Figure 2. This is a graph with N^2 nodes and $2N^2 - 2N$ arcs and therefore, as mentioned in the last section the problem can be solved in $O((N^2 \log N)^2)$ operations.
Note that this analysis is independent of the dimension, so the three-dimensional case can be handled in an analogous way.

For the one dimensional case the associated graph is in Figure 3. A network flow problem for this type of graphs can be solved in linear time, so this gives an alternative linear time algorithm for the one dimensional case.

If one uses the network simplex method for solving the two dimensional case, one has to compute the so-called basic solutions of linear programming. Choosing a basis can be seen as choosing a one dimensional approximation to the two dimensional problem. When the simplex method finds an optimal basis, it would have found a "right" one dimensional approximation.

6 Conclusions

We showed in this paper, that the problem of calculating the Hutchinson distance between two probability measures can be seen as a network flow problem. Therefore all the tools of that field are available in order to develop algorithms, which taking into account the structure of this particular problem, may result in even better computation time.

This result becomes particullarly relevant, when it is clear that the formula found in [12] can not be used for dimension 2 or more, i.e. the most important case for image processing.

Here we presented an application of the computation of this distance in order to compare digital images, but there are many other applications in which it is desirable to compare measures using this distance, hence this algorithm will provide the tool for a fast computation.

7 Acknowledgements

Two of us (CAC and UMM) wish to acknowledge support from the CONICET (Argentina) and from the Department of Applied Mathematics of the University of Waterloo. We would like to thank Adrian Lewis for many comments and suggestions.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network flows. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Handbooks in Operations Research and Management Science, Volume 1, Optimization*. North Holland, 1989.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Some recent advances in network flows. *SIAM Review*, 33(2):175–219, June 1991.

[3] Michael F. Barnsley. Fractal functions and interpolation. *Constructive Approximation*, 2:303–329, 1986.

[4] Michael F. Barnsley. *Fractals everywhere*. Academic Press, Inc., San Diego, CA, 1988.

[5] Michael F. Barnsley, Mark A. Berger, and H. Meté Somer. Mixing markov chains and their images. *Probability in the Engineering and Informational Sciences*, 2:387–414, 1988.

[6] Michael F. Barnsley and Steven Demko. Iterated function systems and the global construction of fractals. *Proc. Roy. Soc. Lond.*, A399:243–275, 1985.

[7] Michael F. Barnsley, John Elton, D. P. Hardin, and Peter R. Massoupoust. Hidden variable fractal interpolation functions. *SIAM Journal of Mathematical Analysis*, 20(5):1218–1242, 1989.

[8] Michael F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. *Proceedings of the National Academy of Sciences*, 83:1975–1977, 1985.

[9] Michael F. Barnsley and A. D. Sloan. A better way to compress images. *BYTE Magazine*, January issue:215–223, 1988.

[10] Mark A. Berger. Images generated by orbits of 2-d Markov chains. *CHANCE*, 2(2):18–28, 1989.

[11] Mark A. Berger and H. Meté Somer. Random walks generated by affine mappings. *Journal of Theoretical Probability*, 1:239–254, 1988.

[12] Jonathan Brandt, Carlos A. Cabrelli, and Ursula M. Molter. An algorithm for the computation of the Hutchinson distance. *Information Processing Letters*, 40:113–117, 1991.

[13] V. Chátal. *Linear Programming*. Freeman, New York, 1983.

[14] Kenneth J. Falconer. *Fractal Geometry, Mathematical Foundations and Applications*. John Wiley & Sons, 1990.

[15] C. Handy and G. Mantica. Inverse problems in fractal construction: Moment method solution. *Physica D*, 43:17–36, 1990.

[16] J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713–747, 1981.

[17] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ 07632, 1989.

[18] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York, 1977.

[19] Peter R. Massoupoust. Fractal surfaces. *Journal of Mathematical Analysis and Applications*, 151:275–290, 1990.

[20] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Proc. 20th ACM Symp. on the Theory of Comp.*, pages 377–387, 1988.

[21] Ron Schonkweiler. An image algorithm for computing the Hausdorff distance efficiently in linear time. *Information Processing Letters*, 30:87 89, 1989.

[22] J. Stark. A neural network to compute the hutchinson metric in fractal image processing. *IEEE, Transactions on Neural Networks*, 2(1):156–158, 1991.

[23] Tamás Vicsek. *Fractal Growth Phenomena*. World Scientific Publishing Co. Pte. Ltd., 1989.

[24] Edward R. Vrscay. Iterated function systems: theory, applications and the inverse problem. In *Proceedings of the NATO Advanced Study Institute on Fractal Geometry and Analysis.* Kluwer, July 1989.

A Appendix

We will now prove the theorem of section 3.

Theorem 1 *If $X = [0,1]$ and $\mathcal{M}(X)$ is the set of all Borel regular measures on $[0,1]$, then the Hutchinson distance d_H between two measures μ and ν in $\mathcal{M}(X)$ becomes*

$$d_H(\mu,\nu) = \int_0^1 |F_\eta(x)|dx$$

$$\textit{where}\quad \eta = \mu - \nu \quad \textit{and}\quad F_\eta(x) = \eta([0,x]).$$

Proof: If we denote by λ the Lebesgue measure on $[0,1]$, let us consider the set
$\mathcal{B} = \{(x,y) \in [0,1]\times[0,1], : x < y\}$ with the product measure $\varrho = \lambda \otimes \eta$.

Now, if $f \in Lip_1$ then f is continuous on $[0,1]$ and differentiable a.e.. On the other hand, if we consider the function:

$$g(x) = \begin{cases} f'(x) & \text{if it exists} \\ 0 & \text{elsewhere} \end{cases},$$

then g is λ-integrable on $[0,1]$, and $h(x,y) = g(x)$ is ϱ-integrable on $\mathcal{B}$. Using Fubini, we obtain:

$$\int_{[0,1]} f\,d\eta = f(1)\eta([0,1]) - \int_{[0,1]} f'(x)F_\eta(x)\,dx. \tag{8}$$

Since $\eta = \mu - \nu, \eta([0,1]) = 0$ and hence

$$\int_{[0,1]} f\,d\eta = -\int_{[0,1]} f'(x)F_\eta(x)\,dx. \tag{9}$$

Then

$$d_H(\mu,\nu) = \sup_{f\in Lip_1} \int_X f\,d\eta \leq \sup_{f\in Lip_1} \left|\int_X f\,d\eta\right| \leq \sup_{f\in Lip_1} \int_X |f'(x)||F_\eta(x)|\,dx.$$

But since $|f'(x)| \leq 1$ a.e., $d_H(\mu,\nu)$ is therefore bounded as

$$d_H(\mu,\nu) \leq \int_X |F_\eta(x)|\,dx = \|F_\eta\|_1. \tag{10}$$

To complete the proof, we now exhibit a function $f^0 \in Lip_1$ such that $\int_X f^0\,d\eta = \|F_\eta\|_1$.

Consider first $g(x) = -sg(F_\eta(x)) = \begin{cases} 1 & \text{if } F_\eta(x) > 0 \\ -1 & \text{if } F_\eta(x) < 0 \\ 0 & \text{if } F_\eta(x) = 0 \end{cases}$.

Then $|g|$ is integrable on X and

$f^0 = \int_0^x g(t)\,dt - \int_0^1 g(t)\,dt$ is in Lip_1, i.e., if $0 \leq x \leq y \leq 1$

$$|f^0(x) - f^0(y)| = \left|\int_x^y g(t)\,dt\right| \leq \int_x^y |g(t)|\,dt \leq \int_x^y dt = y - x = |y - x|.$$

Now, by equation (9)

$$\int_X f\,d\eta = -\int_X f'(x)F_\eta(x)\,dx = \int_X |F_\eta(x)|\,dx = \|F_\eta\|_1.$$

Therefore we obtain the desired result:

$$d_H(\mu,\nu) = \|F_\eta\|_1 = \|F_\mu - F_\nu\|_1.$$

□

Remark: It is interesting to remark here, that this result is not generalizable to higher dimensions: Therefore consider (X,d) to be the $[0,1]^2$ with the euclidean metric. Let S_r be the horizontal segment of the unit square at height r, i.e. $S_r = \{(x,r) : 0 \leq x \leq 1\}$; $0 < r < 1$. Consider also measures μ and $\nu, \mu \neq \nu$ in $\mathcal{M}(X)$ and such that the support of μ and ν are included in S_r. If $d_H(\mu, \nu)$ denotes the Hutchinson distance in $\mathcal{M}(X)$, then if $\eta = \mu - \nu$ and $F_\eta = F_\mu - F_\nu$ is the distribution function of η,

$$\int_X |F_\eta(x, y)| \, dxdy = (1 - r) d_H^1(\mu, \nu) < d_H(\mu, \nu). \tag{11}$$

(Here $d_H^1(\mu, \nu)$ represents the 1-dimensional Hutchinson distance between μ and ν considered as 1-dimensional measures on S_r).