A THEORETICAL STUDY ON AN ACCURATE RECONSTRUCTION OF MULTIVIEW IMAGES BASED ON THE VITERBI ALGORITHM

Belle L. Tseng & *Dimitris Anastassiou*

Center for Telecommunications Research
Department of Electrical Engineering Columbia University
530 West 120th Street CEPSR 801 New York, N.Y. 10027 USA

ABSTRACT

Utilizing the Viterbi algorithm, a forward dynamic programming technique, and a compact disparity graph representation, reconstruction of intermediate views is made possible, including those objects whose views are occluded from various viewpoints. Based on the camera configuration for capture of multiview images, and the one-to-one correspondence between a real world point and its multiview images, a number of geometric constraints are generated and taken advantaged of.

Conventionally, an object in an intermediate view can be interpolated from a left and right extreme image if that object is present in both extreme views. This is mainly performed by finding the accurate disparity for the object, followed by projection onto an intermediate frame. In our proposal, the Viterbi algorithm is used to compute a dense disparity field for every pixel on the left extreme view and consistently for those pixels on the right extreme view. Thus even objects, present in one extreme view but not the other, can be accurately projected onto an intermediate frame.

Furthermore, in the presence of ORO objects, whose views are obstructed in both extreme views but revealed in some intermediate views, a third image composed of all ORO objects is incorporated. The large set of multiview images are then reduced to a maximum of three images along with their dense disparity maps, thus accounting for all objects viewable from a range of viewpoints. Following, a simple view interpolation procedure is developed for accurate construction of all intermediate images, whether the originally captured ones or virtually added ones.

1. INTRODUCTION

Recent interests in 3D technologies start with depth perceptions from stereoscopic images, 3D moving presentations of stereoscopic video sequences, look-around capabilities with multiviewpoint pictures, and most recently, interactive applications of multiview videos. Depth perception from stereo pairs are derived from two viewpoints seen by our left and right eyes. On the other hand, depth perception from multiview images are obtained by motion parallax.[1] Because the resulting image due to motion from head movements should be smooth, the required number of viewpoint images are relative high as compared to two extreme left and right pictures. In this study, the redundancy between the views of a multiview scene is investigated with future applicability to an MPEG-2 extension for multiview video codec.

Within the MPEG-2 standardization[2], we have shown that transmission of a stereoscopic (left and right views) sequence is possible by utilizing the high profile double layer structure of temporal scalable coding[3]. The base layer can support the left-view sequence, while the enhancement layer manages the prediction of the right-view sequence. Consequently after decoding the two extreme views, we focus on an intelligent scheme to interpolate the intermediate views.

There are two distinct situations that can occur when constructing intermediate views from two extreme views. (1) All interpolated views between the left and right images can be fully and accurately (but approximately) constructed from the given two extremes. (2) It is not possible to interpolate the intermediate views due to some uncovered objects revealed and then hidden again between the two extreme left and right. Case (1) is the best situation and occurs as long as every object in the intermediate views is represented in at least one of the extreme images. Case (2) is the worse scenario but can be improved and reduced to case (1), if the obstructed-revealed-obstructed (ORO) object is represented.

If the major criteria that all intermediate views can be approximately constructed from the given coded extreme left and right images, then it can offer an MPEG-2 compatibility coding method and a non-interactive scheme to support multiple users simultaneously, which is partially achieved in this paper.

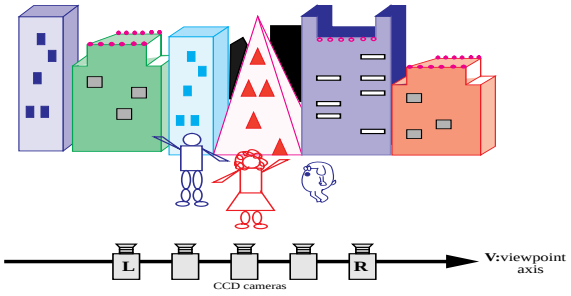

Figure 1a: Camera Configuration of Multiview Images

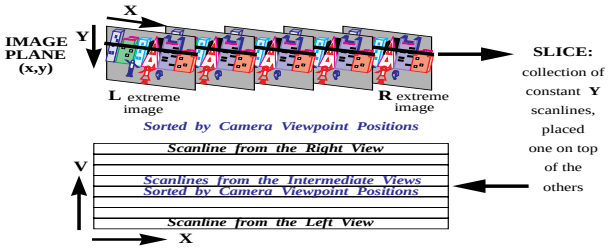

Figure 1b: SLICE Representation

2. ENCODING SYSTEM OVERVIEW

The main feature to derive from a set of multiview images is the perception of depth obtained when we move our heads to view an image of a static scene from a slightly different perspective. In the image capturing configuration, a set of multiple viewpoint images is taken simultaneously from a horizontal lineup of equi-distant cameras with parallel optical axis, positioned on the viewpoint axis v as shown in Figure 1a. The image captured by the leftmost camera is designed by **L**, and that captured by the rightmost camera by **R**. The set of images is sorted from left to right by camera viewpoint positions. Using the conventional image coordinate system **(x,y)**, we define a *slice* as a collection of constant **y** scanlines, as pictured in Figure 1b, sorted sequentially from left to right and placed one on top of the others.

Given the multiple views from the left to right extreme images, the objective is to find a way to transmit as little information as possible for the best reconstruction of all intermediate views. We propose encoding two extreme images, **L** and **R**, and two accurate dense disparity field, one for every pixel on the **L** image and the other for the pixels on the **R** image. In addition, the encoding of a third image along with its dense disparity map *may* be required for the case of **ORO** objects, as discussed in the previous sections.

Utilizing the Viterbi algorithm on the intermediate-view images, an accurate disparity field is obtained for every pixel on the two dimensional slice. To illustrate, refer to the slice pictured on Figure 2a of the scene depicted on Figure 1a. The *disparity value* of a real world point whose image shows up on pixel x_L of the **L**-view and lands on pixel x_R of the **R**-view, is defined as the difference of their pixel locations, $d = x_R - x_L$. Since every real world point casts its image on some pixel positions in the **L** and **R** views, the disparity for a pixel in any of the multiview images will be defined with respect to the two extreme views in this paper. As a result, every image pixel of one real world position acquires the same disparity value on the slice.

Because the set of disparity values for every pixel on a slice is most likely to repeat from one scanline to the next, as shown by the slope lines in Figure 2b, a more compact representation can be achieved. As a result, three classes of slope lines making up a slice are presented.

1. A disrupted/continuous slope line originating from scanline **L**. This represents a point on an object viewable from **L**, but may or may not become partially obstructed by another object with a closer depth as the viewpoint changes from **L** to **R**.
2. A disrupted/continuous slope line originating from scanline **R**. This represents a point on an object viewable from **R**, but may or may not become partially obstructed by another object with a closer depth as the viewpoint changes from **R** to **L**.
3. A continuous slope line not originating from scanline **L** nor terminating at scanline **R**. This represents a point on an object whose view is obstructed in the **L** and **R** images, but is revealed in some intermediate frames. This is what we referred to earlier as the obstructed-revealed-obstructed **ORO** object.

Consequently, three groups of disparity sets are generated. For every pixel x_L on the **L** scanline, the disparity value is recorded as $d(x_L)$; and similarly for every pixel x_R on the **R** scanline, the disparity value is recorded as $d(x_R)$. Finally for every pixel x_{ORO} from the **ORO** object, its disparity value is $d(x_{ORO})$.

With this classification, every pixel on a slice can be predicted by disparity compensation from the left scanline, right scanline, or the **ORO** object. Thus an intermediate view can be accurately reconstructed given the extreme **L** image, the extreme **R** image, an **ORO** image, and their corresponding disparity maps.

At this point, a compact disparity graph representation for a slice is introduced, and an example is shown in Figure 2c. The horizontal axis is x_L, the extreme **L** scanline, and the vertical axis is x_R, the extreme **R** scanline. For every valid pair of (x_L, x_R) corresponding to a real world point, its image luminance value is plotted on that coordinate; a similar graph representation was independently first generated by Marr and Poggio 1979.[4] In addition, for those points whose images are not present at either x_L or x_R scanlines, they are also accounted for in the graph. Because every slope line has a disparity value and a corresponding image on the x_L image and x_R image, whether revealed or obstructed, this real world point is also plotted. Note that the coordinates of every point on the graph also give us the disparity $d = x_R - x_L$ of that point; therefore, points with constant disparity, $x_R - x_L = C$, are at the same real world depth. This disparity graph representation shall prove to be useful in the disparity field estimation performed by the Viterbi algorithm.

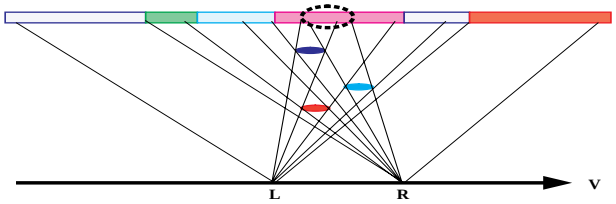

Figure 2a: Slice of Real World Scene taken from Figure 1.

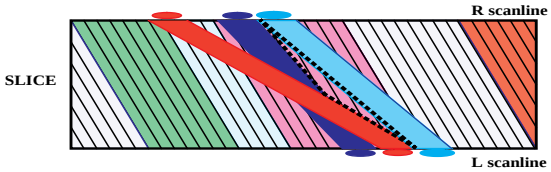

Figure 2b: SLICE Representation of Above Figure.

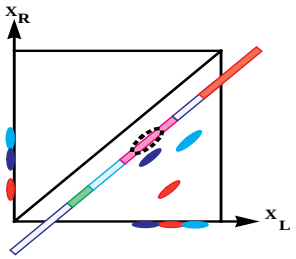

Figure 2c: Compact Disparity Graph Representation of Above Fig.

3. THEORETICAL BACKGROUND

3.1. Geometric Analysis

All analysis in this section refers to a slice, and is independently as valid for all slices. To begin, a few notations are required. We define the conventional real world coordinates as (x, y, z). The viewpoint v-axis, parallel to the x-axis, represents the axis in which the cameras are aligned on. $v = 0$ is assigned to the position of the extreme left camera, and $v = 1$ is for the position of the extreme right camera. The image coordinate system for the extreme left view is denoted by (x_L, y_L), and that for the right view by (x_R, y_R). In general the image coordinate for any intermediate view $v \in [0,1]$ is represented by (x_V, y_V).

Theorem 1: For the configuration of array cameras shown in Figure 1a, where each viewpoint camera is placed equi-distant from the next, the image of a point from the real world system shows up as a straight line in a slice.

Proof of Theorem 1: From perspective projection:

$$\frac{x_L}{f} = \frac{x}{z} \qquad \frac{x_R}{f} = \frac{x+b}{z} \qquad \frac{x_V}{f} = \frac{x+v*b}{z}$$

where b denotes the baseline distance between the L and R parallel optical axis, and f is focal length of the identical cameras. Combining and solving for the viewpoint image x_V, we obtain:

$$x_V = (1-v) * x_L + v * x_R \quad for \quad v \in [0,1],$$

a line segment with (v, x_V) slice endpoints at $(0, x_L)$ and $(1, x_R)$.

Consequently, the image line of a real world point on a slice is referred to as the *slope line* of that point. It also follows that the slope of each slope line defines the disparity value for that point, and vice versa. Furthermore, the depth of the real point is inversely proportional to the disparity value, except for a constant factor dependent on the camera focal length and the distance between cameras.

Theorem 2: Given the compact disparity graph representation for a slice, any intermediate-view scanline can be constructed by projecting the graph onto the radial axis $\theta = f(v)$ where $\theta \in [0^o, 90^o]$ for $v \in [0,1]$, and rescaling it to the original width.

Proof of Theorem 2: We shall find the correct relationship between θ and v to correctly predict the intermediate viewpoint. To project a point $X = [x_L, x_R]^T$ onto the θ-axis, the projection matrix $P = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix}$ is used, where $c = \cos\theta$ and $s = \sin\theta$.

$$x_V = \frac{\| P * X \|}{\max \| P * X \|} = \frac{c * x_L + s * x_R}{c + s}$$

$$x_V = \frac{c}{c+s} * x_L + \frac{s}{c+s} * x_R$$

From Theorem 1, $x_V = (1 - v) * x_L + v * x_R$, thus

$$v = \frac{\sin\theta}{\sin\theta + \cos\theta}$$

gives us the desired construction relationship.

Thus the compact disparity graph representation can also be used to construct intermediate views. However, because objects not represented on the disparity graph can never be predicted by projection, a "complete" disparity graph comprised of all viewpoint objects is required. In the following theorem, a test is provided to determine when the graph is complete.

Theorem 3: Given a subset of a complete disparity graph $G(x_L, x_R)$, say composed of only those points whose views are visible from the extreme L or R scanlines, the following test determines whether all views between L and R can be interpolated from $G(x_L, x_R)$, thus requiring no **ORO** objects for this slice. (Note: This is a sufficient, but not necessary condition.)

1. Segment the disparity graph $G(x_L, x_R)$ into a minimum set of continuous disparity curves.
2. Circumscribe a rectangle around each continuous stretch of disparity points.
3. If the set of constructed rectangles is connected from one end of the disparity graph to the other end, then the given sub-disparity graph $G(x_L, x_R)$ is sufficient for interpolating all viewpoint images between L and R.

Proof of Theorem 3: Since the projection of a continuous curve onto an axis also constitutes a continuous image, the projection of a continuous disparity path extending from one end of the graph to the other end also yields a continuous scanline image. Using the result from Theorem 2 and referring to the sample disparity graphs of Figure 3, the projection of a continuous disparity curve from $\theta \in [0^o, 90^o]$ is bounded by the constructed rectangle. Thus a collection of connected rectangles is a sufficient condition for continuous image interpolation of all viewpoints obtained by projection of $\theta \in [0^o, 90^o]$.

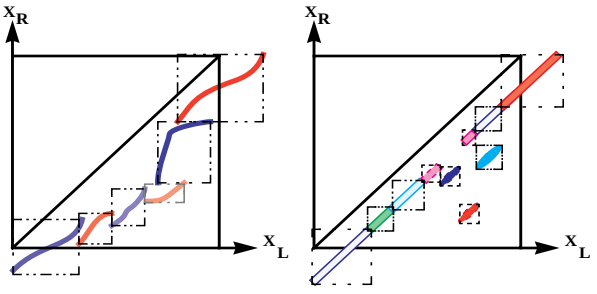

Figure 3: To Test the Completeness of Disparity Graphs for full Interpolation of all Intermediate Multiview Images between the Extreme Left & Right Views. The Left Figure shows a Complete Disparity Graphs, whereas the Right Figure demonstrates the incompleteness of the Disparity Graph for the sample scene.

3.2. Viterbi Algorithm

The Viterbi algorithm is an efficient forward dynamic programming technique introduced to find a minimum cost path from a known starting node in a graph to a given terminating node. The algorithm consists of evaluating all possible paths in the graph up to a certain level, and only the surviving remerged path at each node is retained for the next level of computation.[5]

With the objective of finding the best set of slope lines to represent a slice, the Viterbi algorithm can be simplistically incorporated to find each slope line. Minimizing some cost function, the best straight path can be found originating from every pixel on the **L** extreme scanline to some pixel on the **R** extreme scanline, and vice versa. For this situation, the nodes of the Viterbi graph are represented by the pixel values of a slice.

To improvise an improved method and combine the interdependency of neighboring slope lines, we reintroduce the compact disparity graph representation, as defined earlier. In this implementation of the Viterbi algorithm, the objective is to find the minimum cost path of the disparity graph, with formulation of the cost function discussed in the next subsection.

3.3. Constraints

In this part of the paper, we investigate the parameters and constraints for achieving the best representation for a slice. Subsequently, the Viterbi algorithm minimizes a cost function based on these constraints. The initial states taken by the Viterbi algorithm are based on conventional block matching disparity estimates. To refine the accuracy of these disparity values, certain smoothness constraints are added, a regularization consistency is checked, and a confidence measure is factored in.

Because disparity values do not overly change between neighboring pixels in either viewpoint images, spatially, or temporally, smoothness constraints are utilized to behave the final path achieved by the Viterbi algorithm. The most important smoothness requirement governs adjacent slope lines in a slice, with parameters involving the depth values of corresponding real world points. A second smoothness criterion guides the continuity of spatially intact objects, where the slope lines derived in one slice do not drastically vary among neighboring slices. A final smoothness constraint, requiring coherent disparity and motion vector fields between neighboring frames in time[6], is proposed for multiview *video* coding, but not discussed further here.

In constructing the minimum cost path for the Viterbi algorithm, a forward-backward regularization consistency check is performed at every step to insure that every slope line generated from x_L and ending at $x_L + d(x_L)$, corresponds to the same slope line generated from $x_R = d(x_L)$ and ends at $x_R - d(x_R) = x_L$.

Finally, a confidence measure is developed to enhance the accuracy of the final Viterbi path. Such a measure is assigned to every point on the disparity graph, so that the cost function can be weighted accordingly. The confidence measure we have chosen is a variance-based activity indicator. The higher the activity, the more reliable the disparity estimate, and the higher the weighting for the total cost function. Furthermore, detection of sharp edges by the measure may also indicate a possible failure of the smoothness constraints.

4. CONSTRUCTION OF INTERMEDIATE VIEWS

Upon receiving the extreme **L**, **R**, and **ORO** images, along with their corresponding disparity maps, all intermediate views between **L** and **R** can be accurately constructed. In this section, the view interpolation procedure is outlined and performed independently for each slice. Since a slice can be represented as a collection of slope lines and their respective image luminance values, we start by independently constructing the three classes of slope lines introduced in Section 2, as provided by the three received images and their disparity maps. Following, the three partial slices are combined to form the final slice construction.

To construct an intermediate view $v \in [0,1]$ between the **L** ($v = 0$) and **R** ($v = 1$) viewpoints, including views that were originally captured as well as virtually added ones, a view interpolation procedure is described for each scanline of the intermediate image. Each scanline of viewpoint v is constructed in the following steps:

1. Construct partial scanlines PS_L, PS_R, and PS_{ORO} by projecting **L**, **R**, & **ORO** images by their disparity values.
   $PS_L(x)$ consists of a luminance array $lum_L(x)$ and its corresponding disparity $d_L(x)$. It is constructed by sequentially projecting each pixel x_L of the L image scanline in ascending location order, x_L = 1, 2, ..., WIDTH, and overwriting constructions made by previous pixels.
   $PS_R(x)$ consists of a luminance array $lum_R(x)$ and its disparity $d_R(x)$. It is constructed similar to $PS_L(x)$ except in descending order, x_R = WIDTH, ..., 2, 1.
   $PS_{ORO}(x)$ consists of a luminance array $lum_{ORO}(x)$ and its corresponding disparity $d_{ORO}(x)$. It is constructed by projection of lowest disparity values first.
2. Combine the three partial scanlines, $PS_L(x)$, $PS_R(x)$, and $PS_{ORO}(x)$, into the final scanline $S(x)$ for the required view. For x = 1, 2, ..., WIDTH-1, WIDTH:
   Let $MD = \max(d_L(x), d_R(x), d_{ORO}(x))$.

$$S(x) = \begin{cases} lum_L(x) & \text{if } MD = d_L(x) \\ lum_R(x) & \text{if } MD = d_R(x) \\ lum_{ORO}(x) & \text{if } MD = d_{ORO}(x) \\ (1-v) * lum_L(x) + v * lum_R(x) & \\ \qquad \text{if } MD = d_L(x) = d_R(x) & \end{cases}$$

Theorem 4: The above view interpolation procedure of a slice is equivalent to construction by highest depth first, and results in the correct slice reconstruction.

Proof of Theorem 4: Since the disparity value of an image pixel dictates the depth of that real world point, the visibility of its corresponding luminance value from a specified viewpoint depends only on the disparity. In consistency with this observation, the described view interpolation procedure is based on pixel constructions by comparing and resolving their disparity values.

Based on this criterion, the above constructions of the three partial scanlines are acceptable. Take for instance the left partial scanline $PS_L(x)$ created from image L in ascending order. For the interpolated view v, say x_L1 projects onto $PS_L(x1)$, and similarly x_L2 projects onto $PS_L(x2)$. For $x_L1 < x_L2$, $PS_L(x1)$ is created before $PS_L(x2)$. But if the two pixels coincide, $PS_L(x1) = PS_L(x2)$, then the construction permits $PS_L(x2)$ to override. This is the desired situation because :

$$x_L1 < x_L2$$

$$\frac{x_L1 - PS_L(x1)}{v} < \frac{x_L2 - PS_L(x2)}{v}$$

$$d(x_L1) < d(x_L2).$$

As a result, the depth of the latter projected pixel, $PS_L(x2)$, is closer, and thus visible, to the viewer.

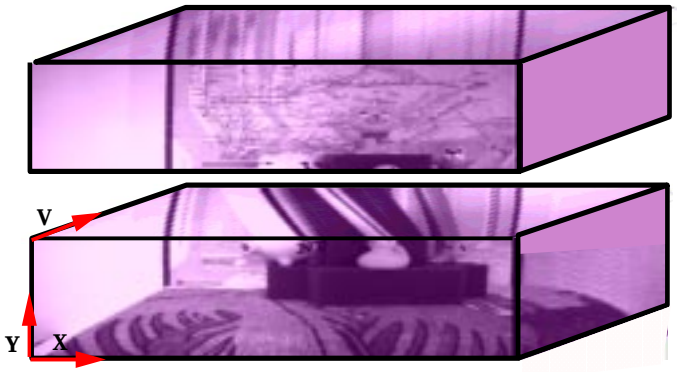

Figure 4: Top Slice 0 and Middle Slice 251 of our Multiview Image Sequence.

5. EXPERIMENTAL RESULTS

A sequence of multiview images of a static scene is captured using one CCD camera set on a robot controller and periodically displaced. The acquired views are sorted and represented as a set of slices, and the Viterbi algorithm is used to obtain the best disparity maps. The required encoding information thus consists of the **L** and **R** extreme images, a possible **ORO** image, and their respective dense disparity maps. View interpolation is then performed to construct the intermediate views.

Incorporating the proposed encoding and interpolation scheme, results indicate highly acceptable perceptual quality in the reconstruction of all multiview images between the **L** and **R** viewpoints. Furthermore, the presented formulation also provides an efficient view interpolation technique for construction of additional virtual views, so that the number of views between the extreme **L** and **R** can be scalably increased and enhanced.

Figure 4 shows the sequence of our multiview images, sequentially ordered from the extreme left to the extreme right viewpoint. In the illustrated middle slice, it can be observed that the toys which are closer to the viewer have higher disparity slopes, while background textures have lower disparity. Utilizing the Viterbi algorithm to generate the required encoding information and interpolating the intermediate views, the final multiview images prove to be of high reconstructed quality.

6. CONCLUSIONS

A theoretical framework on data compression of multiview images is presented. Based on geometric constraints of multiview image acquisition, a compact disparity graph representation and the Viterbi algorithm are adopted for our purpose. The required compressed data is reduces to the extreme left and extreme right images, along with their dense disparity maps. In addition, an obstructed-revealed-obstructed (ORO) image is created to account for objects not observable from the two extreme images, but are revealed in some intermediate views. Consequently a third ORO image and its disparity map may be required. With this set of information, a simple view interpolation procedure is developed for theoretically accurate reconstruction of all intermediate images, whether real or virtually added ones.

Because our study was illustrated for pixel-based processing, future work are being undertaken to investigate a block-based formulation, and consequently for an MPEG-2 compatible methodology. One insight is to incorporate the extreme **L** and **R** images in the two layer structure of temporal scalability of MPEG-2. Furthermore, in the MPEG-2 specification there is no limitation to the number of layers to only two. So in addition to the standalone base layer, the number of enhancement layers may be incremented for each corresponding viewpoint direction, (ie, horizontally, vertically, zoom, etc.)

7. REFERENCES

[1] I. Overington. Computer Vision: A Unified, Biologically-Inspired Approach. Elsevier, Amsterdam, 1992.

[2] MPEG Committee Draft. ISO/IEC 13818-2: Coding of Moving Pictures and Associated Audio. Mar. 1994.

[3] B. L. Tseng and D. Anastassiou. Compatible Video Coding of Stereoscopic Sequences using MPEG-2's Scalability and Interlaced Structure. International Workshop on HDTV '94. Torino, Italy, Oct. 1994.

[4] D. Marr and T. Poggio. A Computational Theory of Human Stereo Vision. Proceedings of the Royal Society, London, Vol. B204, 1979, p301-328.

[5] D. Bertsekas. Dynamic Programming:Deterministic and Stochastic Models Prentice-Hall, Englewood Cliffs, 1987.

[6] N. Nikolaidis, I. Pitas, M. G. Strintzis. Combined Evaluation of Motion and Disparity Vector Fields for Stereoscopic Sequence Coding. Proceedings of Computer Analysis of Image Patterns, Budapest, 1993.