Effect of Image Linearization on Normalized Compression Distance

Jonathan Mortensen[1], Jia Jie Wu[2], Jacob Furst[3], John Rogers[3], and Daniela Raicu[3]

[1] Case Western Reserve University
[2] University California, San Diego
[3] DePaul University
Jonathan.Mortensen@case.edu,
jjw017@ucsd.edu,
jfurst@depaul.edu,
draicu@depaul.edu
http://facweb.cti.depaul.edu/research/vc/

Abstract. Normalized Information Distance, based on Kolmogorov complexity, is an emerging metric for image similarity. It is approximated by the Normalized Compression Distance (NCD) which generates the relative distance between two strings by using standard compression algorithms to compare linear strings of information. This relative distance quantifies the degree of similarity between the two objects. NCD has been shown to measure similarity effectively on information which is already a string: genomic string comparisons have created accurate phylogeny trees and NCD has also been used to classify music. Currently, to find a similarity measure using NCD for images, the images must first be linearized into a string, and then compared. To understand how linearization of a 2D image affects the similarity measure, we perform four types of linearization on a subset of the Corel image database and compare each for a variety of image transformations. Our experiment shows that different linearization techniques produce statistically significant differences in NCD for identical spatial transformations.

Key words: Kolmogorov Complexity, Image Similarity, Linearization, Normalized Compression Distance, Normalized Information Distance

1 Introduction

Image similiarity applications spann from medical image retrieval to security and intellectual property and is an important research topic in the imaging field. Recently, research has applied derivations of Kolmogorov complexity ($K(x)$) in order to measure image similarity because its they provide a universal distance measure [10]. First proposed by M. Li et al., Normalized Compression Distance (NCD) quantifies object similarity by calculating the compressed sizes of two strings and their concatenation [10].

In [10], [11], [4], normalized compression distance (NCD) has been shown to yield promising results in constructing phylogeny trees, detecting plagiarism, clustering music, and performing handwriting recognition. These results demonstrate the generality and robustness of NCD when comparing one dimensional data. The simplicity of NCD presents an automatic way of grouping related objects without the complicated task of feature selection and segmentation.

In [7] the compressed size of the concatenation of x and y is used to estimate $K(xy)$ as well as $K(yx)$. Compressors search for sequences shared between x and y in order to reduce the redundancy in the concatenated sequences. Therefore, if the result of this compression is much smaller than the compression of x and y separately, it implies that much of the information contained in x can be also used to describe y.

To approximate the NCD between two images, image linearization accompanied with a variety of compression algorithms, has been applied. Kolmogorov complexity is not computable but it can be approximated by using compression. Image compression is the notion that given a group of neighboring pixels with the same properties, it is more efficient to use a single description for the entire region. Thus, if image x is more complex than image y, the Kolmogorov complexity of x, which is approximated by the size of compressed x, will be larger than that of y. With lossy compression, a group of pixels can be replaced with an average color to create a more compact description of the region, at the expense of slight distortions. To make the best approximation of Kolmogorov complexity, the best possible compression must be used. The approximation of Kolmogorov complexity is limited by the fact that it is not possible to design the best system of image compression: whichever compression we use, there is always a possibility for improvement.

NCD determines image similarity based on the theory that the visual content of an image is encoded into its raw data. Theoretically, this makes NCD a suitable metric for Content-Based Image Retrieval (CBIR) which attempts to find similar images based on a query image's content. The application of NCD for CBIR in [7] has shown to produce statistically significant dissimilarity measures when tested against a null hypothesis of random retrieval. The NCD between images was used as a metric to search the visual content encoded in the raw data directly, thus bypassing feature selection and weighting. This approach performed well even when compared against several feature based methods. Although the approach in [7] uses a variety of real-world data sets, different compressors were used for each experiment and the method of linearization was unclear. In particular, the ability of a string to represent a 2D image is not clear. This paper expects to determine the effects of different methods of linearization on NCD.

Similarly, [13] investigates the parameters of visual similarity and verifies that NCD can be used as a predictor for image similarity as experienced by the human visual system. While NCD performs well as a model for similarities involving addition or subtraction of parts, it fails to determine similarity among objects that involve distortions involving form, material, and structure. Results also imply that when two images are very similar, this similarity may be better

approximated by other similarity measures, such as the pixel-by-pixel difference of two images. Although [13] determined that the use of different compression algorithms and transformations have varying effects on NCD, only one photograph and one type of linearization (row-by-row) was used for NCD calculations. Therefore, it is unclear whether the results are universally applicable to images of different content. We present an experiment to illustrate the effects of different methods of linearization and transformations on a database of 100 images.

Due to the compression approximation of Kolmogorov complexity, all of the above examples create a linear string from a 2D image and then find its relative similarity to other strings and therefore other images. Each string was created with a linearization technique; however, an effective formal analysis was not conducted to evaluate the impact of the linearization method. Each linearization produces a distinctly different string and it is important to understand how the 2D signal (image) is converted to a 1D string. Thus, a question is still left standing: Can a string effectively represent a 2D image? This paper explores four different linearization techniques and their impact on the similarity measure of spatially transformed images, and tests the null hypothesis that all linearizations result in the same NCD across several transformations. Section 2 describes the basis of Kolmogorov Complexity, presents the methodology behind the four linearization techniques: Row-Major, Column-Major, Hilbert-Peano, and Self-Describing Context Based Pixel Ordering (SCPO) and presents the methodology of producing the dataset; Section 3 shows results of measuring image similarity distance between an image and a spatially transformation version of an image; and Section 4 contains discussion along with comments on future work.

2 Methodology

2.1 Kolmogorov Complexity and Derivations

To understand the necessity of linearization, a short derivation of Kolmogorov complexity is helpful. Kolmogorov complexity describes the smallest number of bits used to represent an object x [12]. $K(x)$ is the length of the shortest program or string x* to produce x. Furthermore, conditional Kolmogorov complexity, $K(x|y)$, is the length of the shortest program to compute x given y. The conditional complexity of two objects begins the notion of similarity. Building upon conditional complexity, information distance, $E(x, y)$, is the $max\{K(x|y), K(y|x)\}$ and describes the absolute bit change needed to convert one object into another. $E(x, y)$ is not normalized. Normalization provides the Normalized Information Distance (NID),

$$NID(x, y) = \frac{\max\{K(y|x*), K(x|y*)\}}{\max\{K(y), K(x)\}} \ . \tag{1}$$

NID describes the theoretical conception of object similarity. It is shown by [10] that this measure is universal, in that it captures all other semi-computable normalized distance measures. However, because this measure is also upper semi-computable, the complexity of an object, $K(x)$, is approximated using modern

compression algorithms, denoted by $C(x)$. This is referred to as the Normalized Compression Distance,

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \ . \tag{2}$$

This is the standard formula applied to most applications of Kolmogorov complexity similarity measures and also the basic algorithm used in the CompLearn toolkit[2]. Because this approximation uses compression, and compression techniques currently depend on string inputs, the images must be transformed into linear strings [10]. We present four common linearization methods.

2.2 Linearization Methods

The scan-line is a standard scanning method that traverses an image line by line. There are two main scan-line methods: Row-Major and Column-Major. Row-Major concatenates pixel intensities to a string row by row, starting with the upper left pixel and then continuing across the row, and then proceeding down each row. Column-Major follows much the same, but begins in lower right pixel and continues up the first column, before moving toward the right, proceeding column by column. There are variations to this, but all are in a linear fashion.

The Hilbert-Peano curve tranverses all pixels in a quadrant of an image before it linearizes the next quadrant and as a result, this method of linearization has an inherently strong locality property [9]. The Hilbert-Peano space-filing curve guides the exploration of a two dimensional array and linearizes it into a continuous one dimensional string that contains information from all pixels in the plane while respecting neighborhood properties. As described by [9], a Hilbert-Peano curve subdivides a unit square into four sub-parts and after a finite number of well adapted iterations, every pixel is captured by the Hilbert-Peano curve. The Hilbert-Peano Curve can be approximated by self similar polygons and grows recursively following the same rotation and reflection at each vertex of the curve. The self similarity of polygons allows efficient computation of discrete versions of curves while preserving locality. Therefore, a search along a space-filling linearization will result in points that are also close in the original space. This type of linearization is simple and requires no contextual knowledge of the data set. In [5], the fixed space-filling curve approach has been shown to use much less computational resources compared to clustering and other context-based image linearizations. The entropy of a pixel-sequence obtained by Hilbert-Peano curves converges asymptotically to the two-dimensional entropy of the image, resulting in a compression scheme that is optimal with encoders that take advantage of redundant data [6].

[6] also proposes the use of a context-based space filing curve to exploit the inherent coherence in an image. Ideally, an adaptive context-based curve would find and merge multiple Hamiltonian circuits to generate a context-based Hamiltonian path that would traverse every pixel. [8] proposes a self-describing context-based pixel ordering for images that uniquely adapts to each image at

hand and is inherently reversible. SCPO uses pixel values to guide the exploration of the two-dimensional image space, in contrast to universal scans where the traversal is based solely on the pixel position [8]. SCPO incrementally explores the region by exploring sections with pixel intensities most similar to a current pixel and by maintaining a frontier of pixels that has already been explored. Neighboring pixels around the current pixel are added to the frontier and are concatenated to a string. Then, the pixel in the frontier with the closest intensity to the starting point is chosen as the next point about which to explore and is also removed from the frontier. The outcome is a one-dimensional representation of an image with enhanced autocorrelation. Empirical results in [8] show that this method of linearization, on average, improves the compression rate by 11.56% and 5.23% compared to raster-scans and Hilbert-Peano space-filling curve scans, respectively.

Fig. 1. Linearization of four images. Beginning in upper left, Row-Major, Column-Major, Hilbert, SCPO

2.3 Null Hypothesis Test

To gain an understanding of the effects of linearization on an image, a similarity test is chosen in which theoretically, linearization should not affect results. Although different methods of linearizations produce distinct strings, identical spatial transformations to the image would theoretically result in the same NCD. In this experiment, a battery of spatial transformations is applied to a large standard image library. The resulting image transformations are then compared to the original and the relative similarity distance is calculated. This process is

done for each of the described linearization techniques from Section 2.2. Our null hypothesis, Eq. 3, states that each linearization technique produces the same relative distance for each transformation. This would demonstrate that linearization can effectively represent a 2D image.

Our null hypothesis H_o states that different types of linearizations will produce the same NCD values across identical spatial transformations of images. Our alternative hypothesis states that at least one of the linearizations will produce a different NCD across identical spatial transformations. Formally, the null hypothesis and alternative hypothesis are presented as follows:

$$\mathrm{H}_o : \forall s, t \; NCD_s(x, y) = NCD_t(x, y) \; s \neq t \tag{3}$$

$$\mathrm{H}_a : \exists s, t \; NCD_s(x, y) \neq NCD_t(x, y) \; s \neq t \tag{4}$$

where s,t denote any of the 4 different linearizations, x denotes the original image and y denotes one of the 7 spatial transformations comparisons.

2.4 Dataset and Transformations

In this experiment, we selected 100 images from the Corel image database and converted the color images to grayscale by calculating a weighted sum of R, G, and B components using the following equation:

$$\mathrm{I} = 0.2989 \times \mathrm{R} + 0.5870 \times \mathrm{G} + 0.1140 \times \mathrm{B} \tag{5}$$

To measure the effect of linearizations on different spatial transformations, we employed 7 different types of transformations. The original bitmap image has 8 different versions: original, left shifted, down shifted, 90°, 180°, 270° rotation clockwise, and reflections across the x and y axis. All transformations were automated for 100 original images in Photoshop 7.0, thus creating a experimental test set of 100 originals x 8 versions = 800 images. Shifted versions were translated 35 pixels. It should be noted that the empty space created by these shifts were replaced by white (255, 255, 255) pixels in Photoshop 7.0.

Using Matlab, we linearized the 800 bitmap images into text files by extracting gray level intensities from each pixel in four different fashions. Row-by-row (from left to right) and column-by-column (from bottom to the top) linearization were performed by concatenating pixel intensities to a string. To linearize an image in a column-by-column fashion, the image is flipped 90° clockwise and then linearized row-by-row. The image is also linearized using a Hilbert-Peano space-filling curve which grows incrementally, but is always bounded by a square with finite area. The mathmatical deconstruction of the Hilbert-Peano curve can be found in [9]. For each image, a position p(j) is computed along each step in a Hilbert-Peano traversal of space. Then, our algorithm incrementally reads gray-level values from an image at these coordinates. Thus, in order for a Hilbert-Peano space-filling curve to traverse every pixel within the image, the image is resized to 128x128 pixels, potentially distorting the image. Similarly,

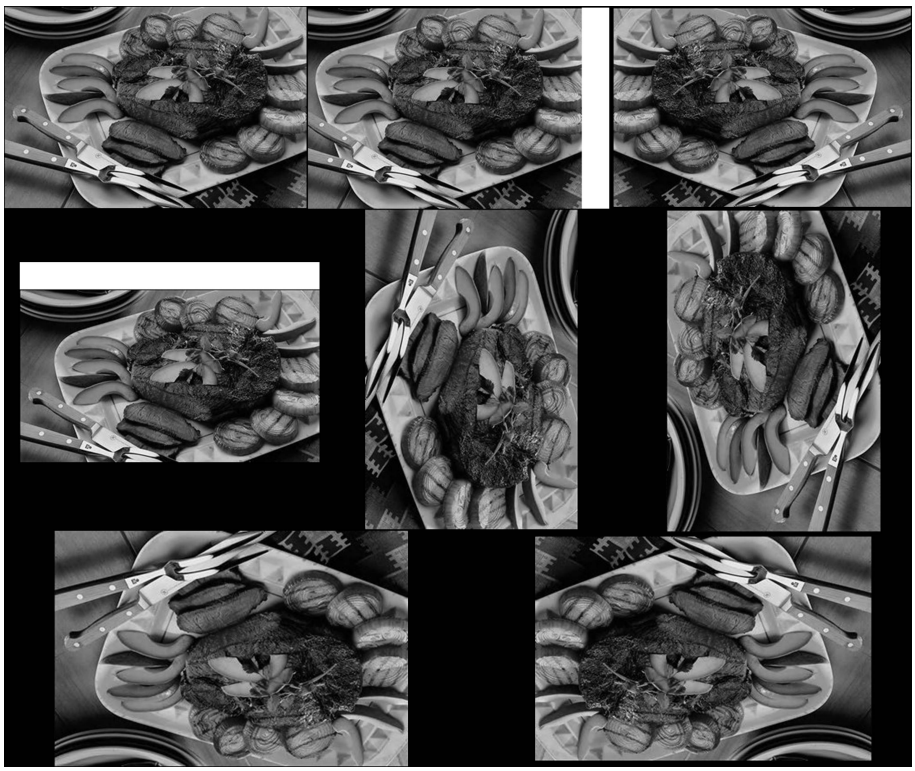

Fig. 2. Spatial Transforms of an image. From left to right: Original, Left-Shift, Ref-Y, Down-Shift, 90°, 2700°, Ref-X,180°

SCPO images are reduced to 35% of its size, preserving the aspect ratio, in order to expedite the computationally-heavy SCPO process. All distorted and reduced images were compared to images with the same distortion, removing any comparison problems. In total, 800 images x 4 linearizations = 3200 text files were compared within their respective linearization groups.

2.5 CompLearn

For this experiment, we used NCD(x,y) and a block sorting compression algorithm (bzip2) built into Complearn, a data analysis toolkit that provides relative compression ratios. Bzip2 exploits frequently recurring character sequences to compress data and detects patterns within a 900 kilobyte window [1]. Since this compression algorithm takes advantage of redundancy in an image to shrink its representation, it approximates the semicomputable Kolmogorov complexity of a string. The NCD(x,y) between two images is calculated using Complearn. Complearn takes several text files and compresses them, noting the bit length each and applies Eq. 2 to find the NCD between two images. Complearn ultimately creates a data set that reflects normalized compression distances between all permutations of two files. The smaller the difference or the closer the NCD is to 0, the more similar the text files. Likewise, the less redundancy between text files, the closer the NCD is to 1. Although a majority of the distances collected were between 0 and 1, some distances were slightly above 1. In [10], [4], and [3] this is not uncommon; when comparing files that share very little information, sometimes NCD can reach values greater than 1. Theoretically, this can be explained by the region in which $C(xy) - min\{c(x), c(y)\}$ is greater than $max\{c(x), c(y)\}$.

The NCD between JPEG compressed images was also calculated to provide a reference for the bzip2 compression algorithm. To measure the NCD between JPEGs, the Complearn toolkit was not used as JPEG compression is not integrated. Instead, Eq. 2 was determined with bc (Linux). First, two images were concatenated side by side in Matlab. Next, the concatenated image and the two originals were compressed to JPEG losslessly in Matlab with 100% quality. To find $NCD(x, y)$, bc (Linux) was used with the file bit lengths of the resulting files as inputs to Eq. 2. Comparisons between 90° and 270° rotations could not be made because two images of differing dimensions could not be concatenated.

2.6 Statistical Analysis

The NCD between each of the transformations and the original image were averaged for 100 images to find the overall effect of each spatial transformation on NCD. To determine if different linearizations produced statistically different NCD values, an analysis of variance (ANOVA) test was performed. ANOVA is a well-known statistical test that compares mean square differences between groups and within groups to determine whether groups of data are statistically different from each other.

3 Results

The results are presented in Table 1, which shows the average NCD for each linearization from the original image to each transformation. This was done for 100 different images chosen from the food subset located in the Corel Image Database. There are several interesting averages to note. Although row and column major linearization could not easily recognize the transformed image when it is rotated, these methods of linearization produced significantly lower NCDs when the original image was compared to the down shifted, left shifted image. In addition, Hilbert-Peano and SCPO linearizations produced NCDs consistently over 0.96 when comparing the original image to its shifted copies. With respect to transformations across the y and x axis, for row major linearization the mean NCD was 0.383669 when the original was compared to its transformation that was reflected across the x axis, while for column major linearization the mean NCD was 0.382232 when the original was compared to its transformation that was reflected across the y axis. For 90° and 270° rotations, Hilbert-Peano linearizations produced the lowest average NCDs at 0.949334 and 0.935854 respectively while SCPO produced average NCDs around 0.96 consistently across all transformations. For JPEG compression, nearly all NCD values are over 1, which indicating that JPEG compression found little or no redundancies between the concatenation of two images.

Table 1. Mean Normalized Compression Distance (NCD)

Transform	Row	Column	Hilbert	SCPO	JPEG
Original	0	0	0	0	1.0011
Down-Shift	0.4661	0.4811	0.9697	0.9629	1.0072
Left-Shift	0.4676	0.4594	0.9683	0.9607	1.0050
90°	0.9726	0.9725	0.9493	0.9640	
180°	0.9634	0.9698	0.9664	0.9658	1.0011
270°	0.9727	0.9726	0.9358	0.9630	
Ref-X	0.3836	0.9697	0.9638	0.9640	1.0013
Ref-Y	0.963	0.3822	0.9353	0.9625	0.9976

The ANOVA test results are shown in Table 2. The standard p-value threshold of 0.05 was used to test if the type of linearization significantly affected the NCD measure across transformations. ANOVA results show that different types of linearizations produce statistically significant differences in NCD for identical spatial transformations. Therefore, we can reject the null hypothesis that different types of linearizations will produce the same NCD values for spatial transformations. This suggests that images may not be fully expressible as a string, at least using current compression algorithms. It certainly indicates that the method of linearization does matter.

Table 2. ANOVA Normalized Compression Distance (NCD)

Transform	Grouping	Sum of Sq.	df	Mean Square	F	Sig.
Down-Shift	Between Groups	24.29344	3	8.09781	13063.1	0*
	Within Groups	0.24547	396	0.0006198		
	Total	24.5389	399			
Left-Shift	Between Groups	25.1020	3	8.36735	14643.3	0*
	Within Groups	0.226278	396	0.0005714		
	Total	25.32834	399			
90°	Between Groups	0.036090	3	0.01203	49.1997	4.78458E-27
	Within Groups	0.096828	396	0.0002445		
	Total	0.132918	399			
180°	Between Groups	0.002070	3	0.0006	7.51982	6.62437E-05
	Within Groups	0.036336	396	9.17588E-05		
	Total	0.038406	399			
270°	Between Groups	0.090856	3	0.03028	128.202	4.90716E-58
	Within Groups	0.093547	396	0.0002362		
	Total	0.184404	399			
Ref-X	Between Groups	25.42660	3	8.475535	55040.8	0*
	Within Groups	0.060978	396	0.0001539		
	Total	25.48758	399			
Ref-Y	Between Groups	24.561154	3	8.187051	27445.8	0*
	Within Groups	0.118125	396	0.0002982		
	Total	24.67928	399			

*Approximates to 0

4 Discussion and Conclusion

It is shown that linearization techniques affect the measured relative distance between two images. Nearby pixels in one linearization may not be near to the same pixel in another. Thus, when applying Kolmogorov complexity as a similarity metric, careful consideration of linearization technique must used, a topic which has not been explained or explored. Also, our data suggests that the use of multiple linearizations for one comparison pair may be effective. Using row major and column major linearization to calculate NCD may be better at capturing similar information between shifted copies. Our results are consistent with [13], which show that row-major calculations of NCD can easily capitalize on shared information among shifted copies. Additionally, row major linearization may best recognize images that share characteristics across the x axis while column major linearization may best capture likeness among images similar across the y axis.

In addition, Hilbert-Peano and SCPO linearizations produced NCDs consistently over 0.96 when comparing the original to its shifted copies, showing that these types of linearizations may not easily capture similarity among images that are strongly shifted. Statistically significant values of NCD created by Hilbert-Peano linearization show that Hilbert-Peano linearization may capture

similarity among rotated copies better than other forms of linearizations in this experiment. Although SCPO has been shown to enhance the autocorrelation and compression ratio within a single image[8], in this study it does not seem to effectively describe the similarities between two images. Other conversion methods may need to be sought out to find types of linearizations that are robust to spatial transformations.

Also, JPEG compression, a standard image compression algorithm, produces dissimilar values of NCD and is shown in this study to be ineffective for finding image similarity. Image compression algorithms may not yet approximate Kolmogorov complexity and future work for image compression algorithms may need to be done to bypass the potentially flawed linearization process. Nonetheless, because linearization affects the NCD, clearly a string does not fully or totally represent a 2D image. More consideration on this topic must be taken.

4.1 Future Work

Further investigation will include linearization's effect on similarity distance with regard to intensity transformations, such as intensity shifts, introduction of noise, and watermarking. The degree of a transformational change and its correlation to the degree of NCD change will also be measured. Additionally, compression algorithms need to be more rigorously compared, and as [13] demonstrated, there is a qualitative difference between the performance of different compression algorithms. [2] also mentions file size limitations to compression algorithms that may limit the sizes of files compared. We expect that efficiently linearizing an image would lead to greater compression ratio, which in turn would lead to more meaningful values of NCD if similarity exists between two images. We also expect that certain linearizations will generate NCDs that are more consistent with similarity measured by the human visual system and thus be more robust to spatial transformations. Furthermore, linearizing an image into a one dimensional string may not be the best method to represent an image. To accurately approximate the Normalized Information Distance between two images, other forms of image compression will need to be investigated.

Acknowledgments The research in this paper was supported by NSF award IIS-0755407.

References

1. R. Cilibrasi. *Statistical Inference Through Data Compression.* PhD thesis, Universiteit van Amsterdam, 2007.
2. R. Cilibrasi, A. L. Cruz, S. de Rooij, and M. Keijzer. CompLearn home. http://www.complearn.org/.
3. R. Cilibrasi, P. Vitanyi, and R. de Wolf. Algorithmic clustering of music. *Arxiv preprint cs.SD/0303025*, 2003.
4. R. Cilibrasi and P. M. B. Vitanyi. Clustering by compression. *IEEE Transactions on Information theory*, 51(4):15231545, 2005.

5. Yeo B.L. and Yeung M. Craver, S. Multi-Linearization data structure for image browsing. *Storage and Retrieval for Image and Video Databases VII: 26-29 January, 1999, San Jose, California*, page 155, 1998.
6. R. Dafner, D. Cohen-Or, and Y. Matias. Context-based space filling curves. In *Computer Graphics Forum*, volume 19, pages 209–218. Blackwell Publishers Ltd, 2000.
7. I. Gondra and D. R. Heisterkamp. Content-based image retrieval with the normalized information distance. *Computer Vision and Image Understanding*, 111(2):219–228, 2008.
8. A. Itani and D. Manohar. Self-Describing Context-Based pixel ordering. *Lecture notes in computer science*, pages 124–134, 2002.
9. C. H. Lamarque and F. Robert. Image analysis using space-filling curves and 1D wavelet bases. *Pattern Recognition*, 29(8):1309–1322, 1996.
10. M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50:12, 2004.
11. M. Li and R. Sleep. Melody classification using a similarity metric based on kolmogorov complexity. *Sound and Music Computing*, 2004.
12. M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag New York, Inc., 1st edition, 1993.
13. N. Tran. The normalized compression distance and image distinguishability. *Proceedings of SPIE*, 6492:64921D, 2007.