BIM: IMAGE MATCHING USING BIOLOGICAL GENE SEQUENCE ALIGNMENT

Hung-sik Kim *Hau-Wen Chang* *Haibin Liu* *Jeongkyu Lee*[+] *Dongwon Lee*

The Pennsylvania State University, USA
{hungsik, hachang, haibin, dongwon}@psu.edu

[+]University of Bridgeport, USA
jelee@bridgeport.edu

ABSTRACT

Matching two images with similar contents is one of the most fundamental tasks in image processing. Due to its importance, in recent years, many novel techniques have been proposed with great successes. Toward this effort, in this paper, we propose a radically different idea by bridging two seemingly unrelated fields – *Image Processing* and *Biology* – i.e., we propose to use the popular gene sequence alignment algorithm in Biology, BLAST, in determining the similarity between images. In this proposal, we map image features to a sequence of gene alphabets (e.g., A, C, G, and T in DNA, or 23 letters in protein) to utilize a wealth of advanced algorithms and tools in BLAST. Under the new idea, in particular, we study various images features and gene sequence generation methods that impact the accuracy and performance in matching similar images. Our proposal, termed as BLASTed Image Matching (BIM), is empirically validated using real data sets. Our work can be viewed as the "first" step toward bridging Image Processing and Biology fields in the application of the well-studied image matching problem.

Index Terms— Image Matching, CBIR, BLAST

1. INTRODUCTION

Due to its importance as one of the fundamental operations in many multimedia applications, the problem of matching similar images, to be called as the *Image Matching* problem in this paper, has been extensively studied in recent years. As a result, many successful solutions to the image matching problem have been proposed and adopted (e.g., [1, 2, 3, 4, 5]). Informally, the image matching problem can be defined as follows.

> **Image Matching.** Given a set of query images I_q and a collection of source images I_s, for each query image i_q ($\in I_q$), find all images, I_r ($\subseteq I_s$) that are "similar" to i_q.

By and large, contemporary solutions have focused on how to identify similar images accurately and efficiently by designing new algorithms, data structures, or models in a particular application or context. However, often, it is hard to apply newly-developed solutions for the image matching problem to new data sets of different scenarios, let alone using additional tools to visualize or analyze the results further. One way to approach the problem is to develop a suite of image matching algorithms and tools for "generic" usage so that the developed solutions can be used in a variety of situations by many users. Another way is to extend an existing generic solution to solve the image matching problem so that one can leverage on the development of the generic solution and its user base.

In this paper, we take the former approach and apply one of such popular and generic solutions drawn from Biology, called BLAST (*B*asic *L*ocal *A*lignment *S*earch *T*ool) [6]. The BLAST, developed in 1990, is one of the most popular (and best cited) algorithms for aligning biological sequence information such as nucleotides of DNA sequences and amino acid sequences of proteins. In its essence, given a query sequence s_q, BLAST finds top-k most similar sequences above a threshold from a database of sequences D using approximation heuristics based on Smith-Waterman algorithm [7]. Note that our focus is to define a novel similarity measure between two images that uses the popular gene sequence alignment algorithm in Biology. Other important issues in CBIR such as indexing or performance are not discussed in this paper.

Our proposal is based on the observations that: (1) Both image matching and gene sequence alignment problems can be variants of approximate pattern matching. By capturing various content-based features of images in high dimensions and converting them into one-dimensional sequences of gene alphabets, both problems can be solved as the approximate pattern matching problem; (2) The alignment results from BLAST provide a robust similarity measure of *S-score* as well as a sound reliability measure of *E-value* with a statistical guarantee. Further, BLAST is known for its fast running time and ability to handle a large amount of sequence data (e.g., up to 256 TB on 64bit OS); and (3) BLAST has a wealth of advanced algorithms (e.g., nucleotide-nucleotide, protein-protein, protein-nucleotide, and position-specific version), implementations (e.g., NCBI BLAST, FPGA-based BioBoost BLAST, WU-BLAST), and tools (e.g., KoriBLAST for visualization and Parallel BLAST for parallel processing) to leverage on.

Our main contributions are as follows: (1) To our best knowledge, this is the first attempt to solve the image matching problem using the BLAST technology. We formally introduce the image matching problem and propose a generic framework, termed as BLASTed Image Matching (BIM), to show how to apply BLAST algorithm; (2) We study diverse schemes to generate gene sequences from images including *protein/DNA* sequences, *default/interpolated*, and *space filling curve* assignment; and (3) We implement and empirically validate the proposed BIM using real data sets.

2. BLASTED IMAGE MATCHING

Figure 1 shows an overview of the proposed BIM. First, we extract a set of feature $\mathcal{F}$ from all source images I_s. The extracted features $\mathcal{F}$ are translated to a gene sequence s_s (either DNA or protein) that is feeded to BLAST. For the gene sequence generation, we propose various methods depending on how we "translate" the extracted feature values to gene alphabets. Note that 4 letters (i.e., A, C, G, and T) are used in DNA sequences, while 23 letters (i.e., A, R, N, B, D, C, Q, Z, E, G, H, I, L, K, M, F, P, S, T, W, Y, V, and X) are used in protein sequences. Then, the query image i_q is also translated to a gene sequence s_q similarly. Finally, BLAST is used for matching the query sequence s_q with the source sequence s_s in database D, and the associated images with the matched sequence are returned.

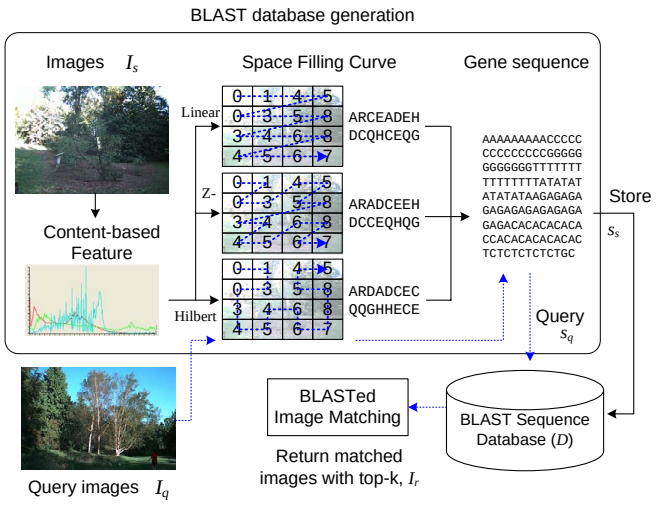


Fig. 1. Overview of the BLASTed Image Matching.

2.1. Image Features for BIM

The first step of BIM is to extract feature values that characterize image data. For the most appropriate features for /bim, we select 3 groups of features as follows:

- Histogram-based feature ($\mathcal{F}_H$): To capture the color distribution of an image, we compute histograms of various color domains (i.e., RGB, YCbCr, and HSV) as well as a gray-scale with 23 bins. The values of each component in color domains are accumulated to obtain a Y- or H-histogram.
- Texture-based feature ($\mathcal{F}_T$): To capture the repetitive patterns of an image, i.e., texture, Law's texture energy measurement is used. The five 1-D convolution kernels (length of 5) stand for Level(L), Edge(E), Spot(S), Wave(W), and Ripple(R) components presented by L5=[1,4,6,4,1,], E5=[-1,-2,0,2,1], S5=[-1,0,2,0,-1], W5=[-1,2,0,-2,1], and R5=[1,-4,6,-4,1]. The 25 2-D convolution kernels are made by the combination of 1-D convolution kernels. For example, E5×E5 detects the edge information in 2-D images.
- Block-based feature ($\mathcal{F}_B$): To describe the color layout of an image, we divide an image into 32 × 32 pixels blocks. Then, for each block we compute an average of feature values computed in $\mathcal{F}_H$ and $\mathcal{F}_T$, i.e., H_{avg}, Y_{avg}, and E_{avg} values.

2.2. Default vs. Interpolated BIM

The core idea of BIM is to represent contents of an image, i.e., features, as a gene sequence to utilize BLAST. To do that, we propose a conversion table that translates image feature values into pre-assigned DNA or protein gene alphabets. For instance, Table 1 shows the conversion table from feature values to protein gene alphabets. The first column of Table 1 shows the quantized values (i.e., q-value) from 0 to 22 for 23 bins (i.e., for 23 protein alphabets). For instance, Y component of values 0–255 and H component of values 0–359 are projected to q-values of 23 bins. The number of pixels are accumulated in each bin with corresponding q-values. Table 1 also shows pre-assigned *default* and *interpolated* DNA & protein letters for corresponding q-values.

With the *default* BIM scheme, in the case of the translation of $\mathcal{F}_H$, each bin number is translated into corresponding gene alphabets

q-value	default		interpolated		q-value	default		interpolated	
	DNA	Pro	DNA	Pro		DNA	Pro	DNA	Pro
0	AAC	A	AAAA	AA	12	ATT	L	CTTA	QQ
1	CCT	R	CAAA	AR	13	ATG	K	CTGA	QZ
2	CAG	N	GAAA	RR	14	CAC	M	CTCA	ZZ
3	AAG	B	TAAA	RN	15	ACT	F	CTCC	ZE
4	ACC	D	TCAA	NN	16	CCC	P	GTCC	EE
5	AAT	C	TGAA	NB	17	CGC	S	GACC	EG
6	CCG	Q	TTAA	BB	18	CGG	T	GGCC	GG
7	GAG	Z	CTAA	BD	19	CTG	W	GGAC	GH
8	ACG	E	CGAA	DD	20	GAC	Y	GGAT	HH
9	AGC	G	CCAA	DC	21	CTC	V	GGTT	HI
10	AGG	H	CCCA	CC	22	CTT	X	GCTT	I I
11	AGT	I	CCTA	CQ					

Table 1. A conversion table using protein alphabets.

using the conversion table, and then the alphabets are "repeated" the same number of times with the value of bin. For example, $bin[1] = 5$ can be converted into RRRRR. In the *interpolated* BIM scheme, on the other hand, gene alphabets are "interpolated" to capture the characteristic of adjacency. The hypothesis is that two adjacent values are closer (or more similar) than non-adjacent ones. In other words, two colors of bin[2] and bin[3] are more similar than those of bin[2] and bin[4]. In order to capture the characteristic of adjacency, we assign a similar pattern of gene alphabets to the adjacent numbers as shown in the columns of interpolated in Table 1. For example, q-values of 0, 1, and 2 are assigned to AA, AR, RR, so that the similarity between AA and AR is higher than that of AA and RR. That is because 0 and 1 is closer than 0 and 2. We also keep the similarity between 1 and 2, i.e., between AR and RR.

2.3. Space Filling Curves

Both default and interpolated BIM schemes can work with all three categories of features mentioned in Section 2.1. However, due to the nature of block-based feature $\mathcal{F}_B$ that characterizes the color layout of an image based on blocks, they are not readily applicable to either default or interpolated schemes. To address this problem, we exploit the *space filling curve* to re-order feature values by converting 2-D space into 1-D sequence.

- *Linear ordering* is a naive way to convert 2-D feature values into 1-D sequence from left to right, top to down fashion.
- *Z-ordering* [8] is known as a good locality-preserving space filling curve algorithm. To cover 2-D space with 1-D sequence, it starts from top-left of an image, and moves like Z-shape.
- *Hilbert curve* [9] is another space filling curve that is known to preserve a locality better than Z-ordering.

"Space Filling Curve" in Figure 1 shows the illustrations of three ordering methods. After all values of $\mathcal{F}_B$ are re-ordered using one of ordering schemes, each value is then translated into gene letter(s) by either default or interpolated scheme.

3. EXPERIMENTAL VALIDATION

To validate our proposals, we present several experimental results for BIM with different data sets of images and various combinations of image features and gene generation schemes.

Set-up. Gene sequences were generated from images on a desktop (Intel Core 2 Duo 1.8GHz, 2GB RAM, Windows XP Home) while

Corel (**C**)			Washington (**W**)		
ID	Category	N	ID	Category	N
C1	Architecture	132	W1	Arborgreens	47
C2	Beach	54	W2	CampusInFall	48
C3	Caverns	51	W3	CannonBeach	48
C4	Dinosaur	40	W4	Cherries	55
C5	DolphinWhale	61	W5	Football	48
C6	Food	121	W6	Greenland	255
C7	Mountain	87	W7	Iran	49
C8	Space	42	W8	LeaflessTrees	48
C9	Sunset	78	W9	Sanjuans	48
C10	ValleyFire	124	W10	SwissMountains	30
S=79, P=790 CL=95% ($\pm$ 10)			S=68, P=676 CL=95% ($\pm$ 11)		

Table 2. Statistics of two data sets (N: number of images, S: sample size, P: population size, and CL: confidence level).

the sequence matching by WU-BLAST 2.0[1] was done on a IBM Z60t (Intel Pentium-M 1.6GHz, 1.5GB RAM, Ubuntu 7.10). Table 2 summarizes two real-world data sets used in the experiments: `Corel`[2] and `Washington`[3]. As the evaluation metric, we mainly used the average precision with the top-k query model – i.e., how many returned images of top-k are similar to a query image? In Table 2, we used 10% of random samples as queries (denoted as S) against the entire population (denoted as P), achieving 95% confidence levels with 10-11 confidence intervals. If a returned image belongs to the same category of the query image, we consider it to be a "match." Since such an evaluation is based on a very loose ground truth, we also conduct more realistic annotation-based evaluation.

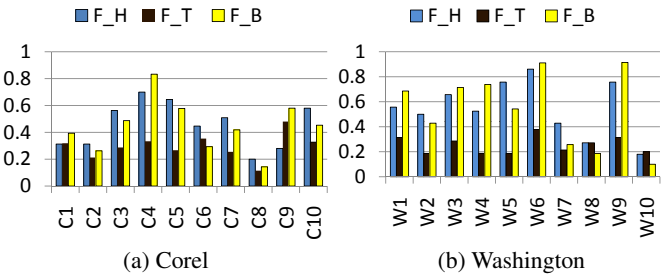
Comparison among Image Features. For the evaluation, we perform top-k result per query for each data set, i.e., C and W, using the representative features of $\mathcal{F}_H$, $\mathcal{F}_T$, and $\mathcal{F}_B$. For $\mathcal{F}_H$, H and Y are selected since they are known as the best color components to represent image characteristics in CBIR among various color domains [5]. For $\mathcal{F}_T$, we compute Law's edge texture energy (E component), and translate a texture histogram to gene letters. For $\mathcal{F}_B$, we test three space filling curves that localize image contents in different ways with H, Y, and E components. All H, Y, and E are translated with both *default* and *interpolated* gene letters. Figure 2 shows the precision among different features with *default* H in $\mathcal{F}_H$, *default* E in $\mathcal{F}_T$, and *default* H in $\mathcal{F}_B$ (linear-order). Although the results are diverse depending on data sets and categories therein, we obverse that most of categories have similar texture information since $\mathcal{F}_T$ provides low accuracy. However, among three different selected features, $\mathcal{F}_B$ (linear-order) outperforms the others, because BIM with $\mathcal{F}_B$ preserves the 2-D localized contents well during which a 2-D image is translated to a 1-D gene sequence.

Annotation Based Evaluation. Note that precision results of Figure 2 are based on the loose ground truth of whether two images belong to the same category or not. Since the categories of two data sets are formed in a subjective manner, therefore, we noticed that precision results of Figure 2 tend to be underestimates of true precision. For instance, Figure 3 shows many returned images that do not belong to the same category as the query image (thus a non-match) but human experts view them as a match. To mitigate this impact, we conduct a second set of more objective evaluation. All images in the Washington data set contain a list of annotations, which provide a good hint on the contents of images. Therefore, an alternative way of precision can be measure based on how annotations of returned images match with those of the query image. That is, for a query image i_q in the category of W7 with annotations "cloud, lake, sand, mountain", if an image i_s in different category of W6 with annotations "lake, green, mountain" is returned, we treat this as a match. That is, while the former evaluation yields a *lower-bound* of true precision, this annotation based evaluation yields a *upper-bound* of it. Figure 4(a) shows the precision comparison between the former and the annotation based evaluations. We notice that, across all 10 categories of the Washington data set, on average, the precision has improved by 0.35 with *default* H $\mathcal{F}_B$ (linear-order). Figure 4(b) shows different precision results by annotation based evaluation for 9 different feature selections. Consistent with Figure 4(a), regardless of the selected features, precision stays around 0.8 on average under the annotation based evaluation method. Since we showed both the lower-bound and upper-bound of true precision, in the remaining experiments, we only show the lower-bound (i.e., worst case), measuring precisions only using category information of images.

[1]http://www.advbiocomp.com/blast/obsolete/

[2]http://wang.ist.psu.edu/docs/related.shtml

[3]http://www.cs.washington.edu/research/imagedatabase/



(a) Corel (b) Washington

Fig. 2. Top-10 precision among three features, *default* H $\mathcal{F}_H$, *default* E $\mathcal{F}_T$, and *default* H $\mathcal{F}_B$ (linear-order) using protein letters.

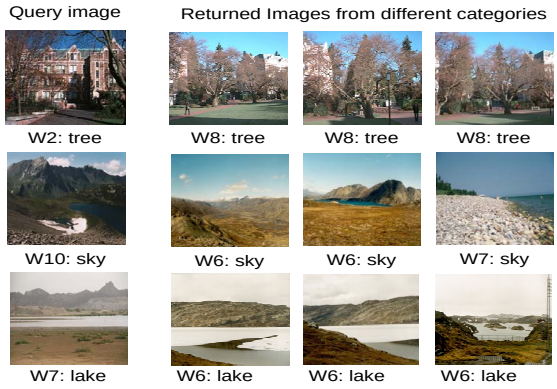
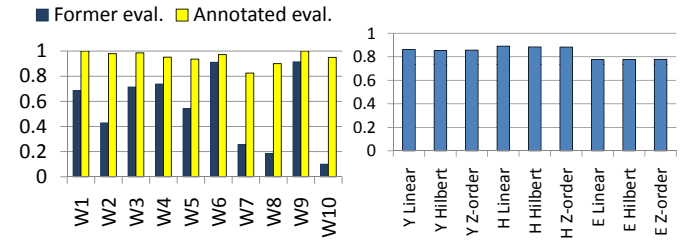


Fig. 3. Examples of a query image and returned images with similar contents and annotations, but from different categories.

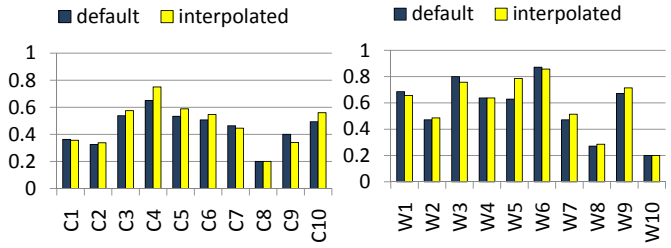
Comparison between Default vs. Interpolated. Next, we compare the performance of default and interpolated BIMs on both data sets. The selected features are *default* H $\mathcal{F}_H$ and *interpolated* H $\mathcal{F}_H$. From Figure 5 of top-10 precision results, we observe that both default and interpolated approaches for gene sequence generation work similarly for matching similar images. However, because the interpolated approach can capture the adjacent similarity better, it tends to provide better precision across many categories.

Comparison among Space Filling Curves. Since space filling curve BIM is meaningful only with $\mathcal{F}_B$, we test three different orders, $\mathcal{F}_B$ (linear, Hilbert, and z-order), with *default* Y and *default* H, respectively. As shown in Figure 6, overall the precisions of three



(a) Former eval. vs. Annotated eval. (b) Different features

Fig. 4. Top-10 precision in the Washington data set: (a) *default* H $\mathcal{F}_B$ (linear-order) (b) various features using protein letters.

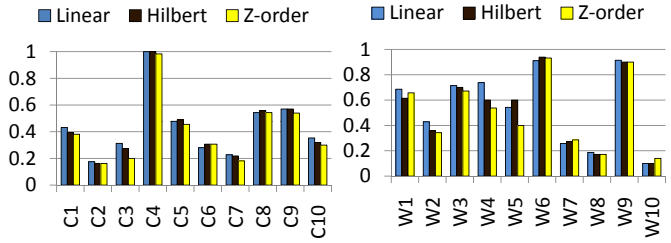


(a) Corel (b) Washington

Fig. 5. Top-10 precision of default vs. interpolated BIM using *default* H $\mathcal{F}_H$ and *interpolated* H $\mathcal{F}_H$ using protein letters.

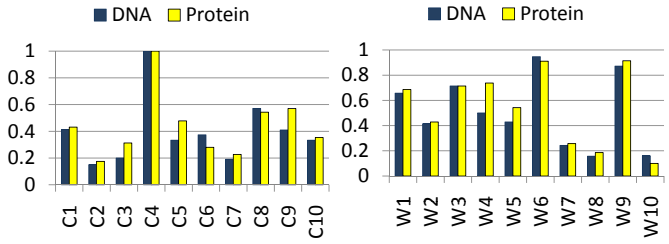
space filling methods are similar. However, because of the unique characteristics of each category, some methods outperform the others. For example, in the category $W4$, "cherries", in Figure 6(b), it is obvious that linear-ordering outperforms the others. The images in $W4$ contain cherry trees in the entire image area with other trees. Therefore, most images do not contain solid localized contents. Thus, the localization of solid contents by Z-order or Hilbert-order does not help the results much, even downgrading sometimes. Therefore, linear-order performs the best in $W4$. However, in the category W6, "greenland", many images contain "sky" as a major localized background. Thus, Hilbert- or Z-order space filling curves can capture the similarity between localized contents better, resulting in a good sub-string match in BLAST.

Comparison between DNA vs. Protein Letters. Finally, we compare the performance of DNA and protein based BIMs. The selected features are *default* Y $\mathcal{F}_B$ (linear-order) and *default* H $\mathcal{F}_B$ (linear-order), respectively. As shown in Figure 7, there is no significant difference between DNA and protein based BIMs. However, due to the



(a) Corel (b) Washington

Fig. 6. Top-10 precision of space filling curve BIM with *default* Y $\mathcal{F}_B$ in C and *default* H $\mathcal{F}_B$ in W using protein letters.



(a) Corel (b) Washington

Fig. 7. Top-10 precision of DNA vs. Protein BIM with *default* Y $\mathcal{F}_B$ (linear-order) in C and *default* H $\mathcal{F}_B$ (linear-order) in W.

fine granularity of protein letters, overall, protein based BIM slightly outperforms DNA based BIM.

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel solution, BLASTed Image Matching (BIM), to the images matching problem by bridging two seemingly unrelated fields – Image Processing and Biology. By capturing three types of images features and transforming them into gene sequences, the similarity of two images was measured by means of the similarity of two corresponding gene sequences. Utilizing the popular and mature BLAST algorithm in Biology in measuring the similarity between genes, we demonstrated the possibility to use 1-D gene sequences to capture multi-dimensional image features. As a preliminary work, more challenges are ahead. First, due to the nature of the gene sequence alignment process, we expect that near-duplicate image identification problem to be more suitable for BIM than the image matching problem is. Such a possibility will be explored. Second, another strength of using BLAST is to exploit its high-performance infrastructure. Therefore, performance issues like indexing or parallel processing will be studied.

5. REFERENCES

[1] H. Lu, B. Ooi, and K. Tan, "Efficient image retrieval by color contents," in *Int'l Conf. on Applications of Databases*, Vadstena, Sweden, June 1994, pp. 95–108.

[2] J.R. Smith and S.-F. Chang, "Automated binary texture feature sets for image retrieval," in *IEEE Int. Conf. Acoust, Speech and Signal Proc.*, Atlanta, GA, 1996.

[3] P. Howarth and S. M. Rüger, "Evaluation of texture features for content-based image retrieval," in *ACM Int'l Conf. on Image and Video Retrieval (CIVR)*, July 2004, pp. 326–334.

[4] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 20, 2008.

[6] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," *J. Mol. Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[7] T. Smith and M. Waterman, "Identification of common molecular subsequences," *J. Mol. Biology*, vol. 147, pp. 195–197, 1981.

[8] G. M. Morton, "A computer oriented geodetic data base; and a new technique in file sequencing," Technical report, IBM Ltd., Ottawa, Canada, 1966.

[9] A. R. Butz, "Alternative algorithm for hilbert's space-filling curve," *IEEE Trans. Comput.*, vol. 20, no. 4, pp. 424–426, 1971.