Visual Search Engine for Handwritten and Typeset Math in Lecture Videos and LaTeX Notes

Kenny Davila
Center for Unified Biometrics and Sensors
University at Buffalo
Buffalo, NY 14260
Email: kennydav@buffalo.edu

Richard Zanibbi
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
Email: rlaz@cs.rit.edu

***Abstract*—To fill a gap in online educational tools, we are working to support search in lecture videos using formulas from lecture notes and vice versa. We use an existing system to convert single-shot lecture videos to keyframe images that capture whiteboard contents along with the times they appear. We train classifiers for handwritten symbols using the CROHME dataset, and for LaTeX symbols using generated images. Symbols detected in video keyframes and LaTeX formula images are indexed using Line-of-Sight graphs. For search, we lookup pairs of symbols that can 'see' each other, and connected pairs are merged to identify the largest match within each indexed image. We rank matches using symbol class probabilities and angles between symbol pairs. We demonstrate how our method effectively locates formulas between typeset and handwritten images using a set of linear algebra lectures. By combining our search engine (*Tangent-V*) with temporal keyframe metadata, we are able to navigate to where a query formula in LaTeX is first handwritten in a lecture video. Our system is available as open-source. For other domains, only the OCR modules require updating.**

I. Introduction

Despite the continuous growth in production of lecture videos, there has been limited attention to the problem of effective indexing and retrieval for graphics such as math in videos. Standard video search engines rely mostly upon text-based search of manual annotations, which are rare because of the effort needed to produce them.

To address this situation, we have created the visual search engine illustrated in Figure 1. Here we use a LaTeX formula image to locate where the formula first appears on the whiteboard in a matrix algebra lecture. Notice that this is a cross-modal search: a *typeset* image is used to query images of *handwritten* whiteboard contents. The system we use to extract keyframes from video records metadata indicating when each connected component on a keyframe appears [1], allowing us to navigate to where the formula starts to be written using a single click. Our search engine also supports the reverse search, from handwritten query images selected from whiteboard keyframes to LaTeX formula images from course notes (see Figure 2).

We avoid indexing recognized formulas, as recognition rates for isolated handwritten formulas are relatively low [2]. Instead, we support image-based search of graphics in typeset notes and handwriting within lecture videos using a new retrieval model. This model is a generalization of our previous method for math formula retrieval [3] in three key ways: 1) math is represented in images rather than symbolic representations such as LaTeX code, and these images may contain other content (e.g., text and diagrams), 2) we use a non-hierarchical structure representation (Line-of-Sight (LOS) graphs), and 3) we use no language models apart from that represented in symbol recognizers, allowing extension to other graphic types.

Our search engine is described in Sections III-V. Given a binary image, handwritten or typeset symbol recognition is applied. Detected symbols are then used to construct a Line-of-Sight graph, with edges between pairs of symbols that 'see' one another along a straight line. Candidate symbol labels and angles between LOS symbol pairs (edges) are used to construct an inverted index from symbol pairs to formula images. Formula image search is performed using a two-layer model. In the first layer, candidate images are identified using the LOS inverted index. In the second layer, matched symbol pairs are aggregated to identify the largest matching sub-graphs in each image. We present different metrics for ranking matching sub-graphs using symbol class confidences and the angles between matched symbol pairs in Section V.

In our experiments (Sections VI-VII), we use LaTeX formula images and handwritten formulas in video keyframes to demonstrate that the proposed model is effective for both within-modal and cross-modal search. A summary and opportunities for future work are provided in Section VIII. Source code for our system (*Tangent-V*) and data used in our experiments are available.[1]

II. Related Work

Our search engine is similar in spirit to word-spotting techniques, which avoid recognizing individual handwritten characters to improve recall (e.g., [4]). We avoid direct recognition of math because it is another hard problem: state-of-the-art methods have achieved somewhat low recognition rates (67.65%) for handwritten math with stroke data available [2].

More broadly, our approach is related to Content-Based Image Retrieval (CBIR). Many CBIR approaches are derived from the Bags-of-Visual Words (BoVW) model [5], with candidates selected by matching shared local descriptors like

[1] https://www.cs.rit.edu/~dprl/Software.html

978-1-5386-5875-8/18/$31.00 ©2018 IEEE
DOI 10.1109/ICFHR-2018.2018.00018

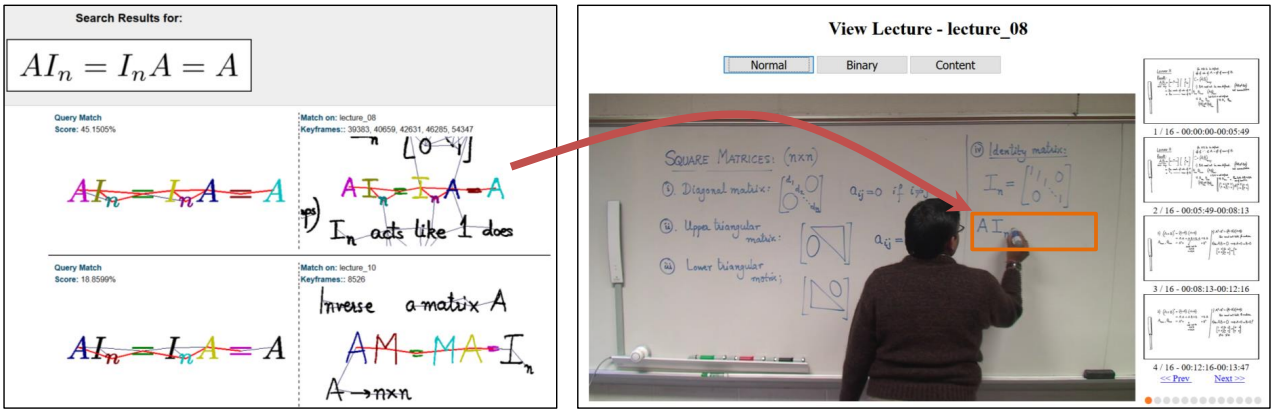


Fig. 1. Finding a LaTeX Formula within a Lecture Video. When the user clicks on the first hit in the search interface at left, the lecture video navigates to where the leftmost 'A' is first drawn (we have advanced the video). Extracted whiteboard keyframes [1] that we use for video indexing are shown at right.

SIFT [6]. It is possible to learn local features or fixed-length vector representations for retrieval of images using different techniques including Deep Learning [7]. In math, the spatial position of a symbol affects its meaning, and there are multiple CBIR techniques that consider spatial constraints. Spectral matching can be used at first, and spatial verification techniques like RANSAC [8] can be applied later. It is also possible to identify affine transformations that align multiple images of a single object, and then select the most likely transformation as the match [9], [10], [11]. Additional robustness to elastic deformation can be achieved by applying topological verifications [12]. The final number of candidates to verify spatially is reduced if the index encodes some spatial information to ensure that initial candidates have some spatial consistency with the query [13].

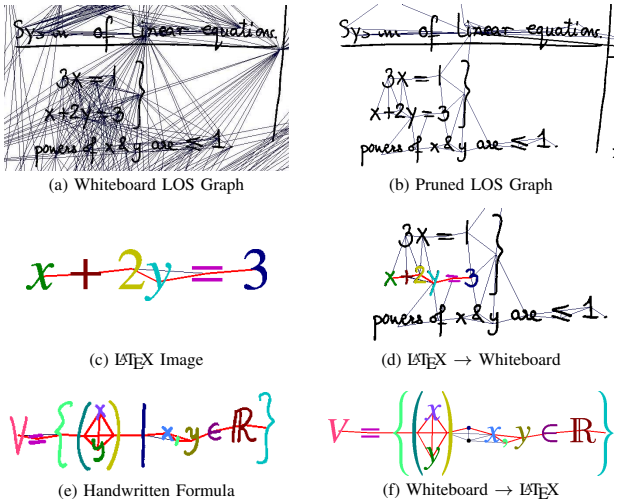
Our work also falls within the field of Mathematical Information Retrieval (MIR) [14]. Math expressions are hierarchical, and thus hard to represent or match using text-based representations. We distinguish two MIR modalities: *Symbolic* and *Image-based*. Symbols and structure are known in symbolic formula representations (e.g., LaTeX, MathML), while they are initially unknown for formula images. Math retrieval tasks held at conferences have supported the improvement of symbolic MIR systems (e.g., [15]). However, few approaches have been proposed for image-based MIR, and none have used standard benchmarks for evaluation. In the work of Marinai et al.[16], isolated mathematical symbol images are indexed and retrieved using a BoVW approach. Zanibbi and Yu [17] used dynamic time warping over pixels projections to search for typeset formula images in rendered PDF documents using handwritten queries. Chatbri et al. [18] use connected component matching to cast votes for query matches in images.

In our work, we apply an open-source math symbol recognition system [19] to obtain candidate labels for Handwritten and LaTeX-rendered math symbols, and use angles between matched symbol pairs to match visual structure. An initial topology encoding is stored in the index, and an additional spatial verification step is applied during retrieval.

III. Character Recognition

Our visual search engine retrieves formulas based on pairs of symbols and their identities, as detected within binary images. In this Section we describe how we train our classifiers, and construct symbol class probability vectors for use in indexing and retrieval.

Classifiers. We have adapted our open-source handwritten math symbol classifier [19] to work with binary images. The classifiers are constructed using Random Forests that return probabilities sorted in decreasing order of class likelihood. Classes are selected from the top of each list using a minimum cumulative probability (80%), taking at most n class labels ($n = 10$). The classifier is shape descriptor-based, including line crossings, 2D fuzzy histograms, 2D orientation histograms, and more general trace features [19]. We adapted



(a) Whiteboard LOS Graph (b) Pruned LOS Graph

(c) LaTeX Image (d) LaTeX → Whiteboard

(e) Handwritten Formula (f) Whiteboard → LaTeX

Fig. 2. Visual Formula Search using Line-of-Sight (LOS) Graphs. A typeset query and a handwritten query cropped from a whiteboard image are shown.

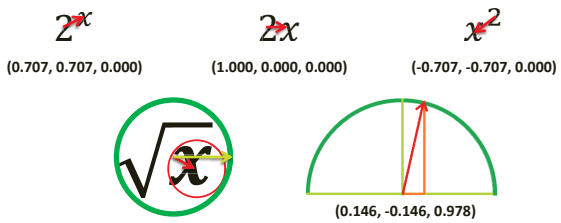


Fig. 3. 3D Symbol Displacement Vectors. **Bottom:** overlapping symbols have the child's center projected onto a sphere around the parent symbol.

this method for off-line data by using connected component contours as the input traces.

Symbol Classes. Our lecture videos have associated LaTeX notes; the LaTeX formulas define the symbol classes expected in the corresponding video. We train a specialized symbol classifier with no more than 50 classes per lecture. Remaining symbols are added to a rejection class ("Junk") for shapes unlikely to belong to math expressions in that lecture.

Training. We train one handwritten and one typeset classifier per video. To train handwritten symbol classifiers, we use expressions from the CROHME 2016 handwritten formula dataset [2]. To improve recognition results for matrices, we expand the dataset by adding the matrix recognition subtask test set. To reduce the index size and improve retrieval performance, we group classes with similar shapes (e.g., 'x' and 'X'), reducing the original 101 classes in CROHME to at most 91 classes.

We train one typeset symbol classifier per video (up to 50 classes, without a reject class). For each of our 91 CROHME symbol classes, we generate 1000 LaTeX images, using small random aspect ratio distortions ($\pm 10\%$) for characters drawn in different fonts, for a total of 91,000 training symbols.

Segmentation. For our first prototype, we used a simple segmentation technique. Initially all connected component are classified. We then try to join pairs of components into single symbols (e.g., '=', 'i'). We use a k-nearest neighbour graph ($k = 2$) to identify candidate merges. Two components are joined if their combined classification confidence is higher than that for each individual component.

IV. Visual Content Extraction and Indexing

In this work we assume that lecture videos are recorded from a single, stationary camera (single-shot). These videos are preprocessed to extract handwritten whiteboard contents. We have used our open-source lecture summarization system [1] to create binary image keyframes containing the extracted whiteboard contents in a video. Formulas are extracted from LaTeX course notes using LaTeXML[2] and rendered as binary images.

Displacement Vectors. We define the relative position of two symbol centers using a 3D unit vector $\langle d_x, d_y, d_z \rangle$. Most displacements can be modeled by just $\langle d_x, d_y \rangle$. However, this does not represent when one symbol partially or completely contains the other (see Figure 3). We add a third dimension d_z to capture overlap. d_z is non-zero when the symbol centers are at a distance smaller than that from the center of parent symbol u to its enscribing circle. We normalize all displacement vectors to make them unit vectors.

Graph Construction and Indexing. We construct a Line-of-Sight (LOS) graph over symbols (see Figure 2a), and then prune edges between symbols more than twice the median distance apart (see Figure 2b). This pruning reduces both the index size and retrieval times. Displacement vectors for edges (see Figure 3) are indexed by their lexicographically sorted symbols. For example, the symbol displacement vectors for $b \rightarrow a$ and $a \rightarrow b$ are both stored at (a, b), reversing the vector for $b \rightarrow a$. Each edge has a unique identifie to allow aggregating matches for symbols with multiple OCR hypotheses. Each symbol pair for an edge is stored using all possible class combinations. For example, displacement vectors and class confidences for two symbols with hypothesis lists $\langle b, 6 \rangle$ and $\langle w \rangle$ are stored in postings for (b, w) and $(6, w)$.

V. Visual Retrieval Model

Our system uses a two-stage retrieval model. The first stage quickly finds candidate matches in the inverted index, and the second stage finds the largest query subgraphs in candidates, ranking hits based on symbol confidences and angles between symbol pairs.

Stage 1: Lookup. Query LOS edges are retrieved by lexicographically sorted symbol pairs (e.g., (a, b), $(0, a)$). For binary image queries, OCR results are converted to a set of $(parent, child)$ symbol alternatives. Symbol pairs with matching labels, consistent spatial alignment, and similar relative sizes are returned. Different symbol hypotheses for each candidate edge are aggregated using the unique edge identifiers stored in the posting lists. For example, after applying OCR to 'x^2' we obtain one LOS edge with parent and child symbol hypotheses $\langle x, X \rangle$ and $\langle 2 \rangle$. We retrieve postings for both $(2, x)$ and $(2, X)$ in the index with similar size ratios and spatial arrangement. We aggregate matches for candidate edges with an entry in both posting lists.

Stage 2: Structural Alignment. Next we combine individual LOS edge matches on a candidate into a subgraph aligned with the query (see Figure 2). First, connected LOS subgraphs are computed and then joined if they are spatially consistent with the query. Finally, a greedy selection for the highest scoring alignment is produced (scoring functions are defined below). Due to space restrictions, we provide just a brief summary here (see [20] for details).

The initial connected subgraph matches are obtained by iteratively merging edge sets sharing at least one symbol that also preserve a one-to-one query/candidate symbol mapping. Then, connected subgraph matches on the candidate are merged if they do not share symbols and there is a path of length ≤ 4 connecting them.

A drawback of using paths to connect matches is that the compound match may be spatially inconsistent with the query layout. We restrict match growing using a spatial distortion

[2]https://dlmf.nist.gov/LaTeXML

cost. This cost is based on an estimated translation and scale ratio between aligned query and candidate nodes. After projecting the candidate into the query space, we compute the average displacement for corresponding query and candidate nodes (symbols). If this cost exceeds a threshold, matched nodes sharing a path will not be joined. Next, subgraph matches on a candidate are pruned by greedily keeping at most the best scoring match per candidate symbol.

The LOS-graph refining step described in Section IV sometimes eliminates relevant connections from the graph. This might lead to queries matching disjoint subgraphs from a candidate. In order to recover from such errors, we greedily merge disconnected matches which might not have a path connecting them, but that preserve a strict one-to-one query/candidate symbol mapping as well as a low spatial distortion cost.

Note that in the case of lecture videos, it is possible to find identical matches for the same query across contiguous keyframes representing the same content from the whiteboard. In such cases, the last step is to group matches across contiguous keyframes of the same video that have an area overlap above 50%. For rendered LaTeX, matches should be grouped for images associated with the same LaTeX string.

Ranking. A candidate subgraph match is scored based on corresponding symbol pairs from the query and a candidate formula (M). A matched symbol pair is represented by (Q, C), where $Q = ((\Omega_{q_1}, q_1), (\Omega_{q_2}, q_2))$ and $C = ((\Omega_{c_1}, c_1), (\Omega_{c_2}, c_2))$. Ω_{q_1} is a set of possible symbol identities for q_1. The corresponding three-dimensional displacement vectors between query and candidate symbol pairs are the unit vectors $\vec{q}$ and $\vec{c}$. The conditional probability of symbol class ω given visual features for query symbol q_1 is denoted by $p(\omega|q_1)$. We consider two ways to combine conditional probabilities for matching labels, using their minimum $m = \wedge$ or their product $m = \prod$, as defined by f:

$$f(\omega, q, c) = \begin{cases} p(\omega|q)\ p(\omega|c) & \text{if } m = \prod \\ min(\ p(\omega|q),\ p(\omega|c)\ ) & \text{otherwise} \end{cases} \quad (1)$$

Optionally, a pair of elements will only be matched if their size ratio is similar as defined by $s_r(Q, C) \geq 0.5$:

$$s_p(u, v) = \frac{diagonal\,(u)}{diagonal\,(v)} \quad (2)$$

$$s_r(Q, C) = \frac{min\,(s_p\,(q_1, q_2)\,, s_p\,(c_1, c_2))}{max\,(s_p\,(q_1, q_2)\,, s_p\,(c_1, c_2))} \quad (3)$$

We consider two match scoring functions. First, function $\alpha(M)$ simply adds the product of symbol confidences and angular differences, summing over symbol classes shared between matched query and candidate symbols. We restrict angular differences between $\pm 45°$.

$$\alpha_\Omega(Q, C) = \sum_{\substack{\omega_i \in \Omega_{q_1} \cap \Omega_{c_1} \\ \omega_j \in \Omega_{q_2} \cap \Omega_{c_2}}} f(\omega_i, q_1, c_1)\ f(\omega_j, q_2, c_2) \quad (4)$$

$$s_\angle(Q, C) = \begin{cases} \frac{\vec{q}\cdot\vec{c} - cos(45°)}{1 - cos(45°)}, & \text{if } \vec{q}\cdot\vec{c} \geq cos(45°) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\alpha(M) = \sum_{(Q,C)\in M} \alpha_\Omega(Q, C)\ s_\angle(Q, C) \quad (6)$$

Our second scoring function $h(M)$ generalizes the Maximum Subtree Similarity (MSS [21]). MSS is the harmonic mean of symbol and relationship recall in expression trees. The harmonic mean prefers joint optimization of the two quantities: for a given sum of symbol and relationship recall scores, the highest value is obtained when the scores are equal. Here, we are using undirected graphs rather than trees, and need to account for symbol probabilities and angles.

Let M contain the corresponding query and candidate symbol pairs in match, and $|N_Q|$ and $|E_Q|$ represent the number of nodes and edges in the query Line-of-Sight graph. Now we can define the *Line-of-Sight Similarity* (h) by:

$$R_\Omega(M) = \frac{1}{|N_Q|} \sum_{(q,c)\in M} \sum_{\omega \in \Omega_q \cap \Omega_c} f(\omega, q, c) \quad (7)$$

$$R_\angle(M) = \frac{1}{|E_Q|} \sum_{(Q,C)\in M} s_\angle(Q, C) \quad (8)$$

$$h(M) = 2\frac{R_\Omega(M) \cdot R_\angle(M)}{R_\Omega(M) + R_\angle(M)} \quad (9)$$

where R_Ω is probabilistically-weighted symbol recall, and $R_\angle$ is the cosine similarity-weighted edge recall (with a tolerance of $\pm 45°$).

VI. Experimental Design

To test the effectiveness of the proposed method, we used selected queries to evaluate retrieval within the set of LaTeX formulas in the course notes, within the extracted keyframes of the videos, and between these two collections. In our experiments, the goal was to recover the query formula exactly within an image, possibly with additional elements (e.g., additional symbols in a larger LaTeX formula, or additional elements in a video keyframe image).

Data. We use the publicly available AccessMath dataset.[3] AccessMath contains 20 lecture videos recorded using a still camera in the classroom. The resolution of each lecture video is 1080p. A total of 13 out of 20 lecture videos are accompanied by lecture notes in LaTeX format. We experiment on this subset of videos with notes describing most expressions found in each video.

Test Queries. We pseudo-randomly choose 20 unique queries for evaluation based on the following criteria. First, each query is randomly selected from the rendered LaTeX expressions from the lecture notes. Second, the query must also appear on at least one key-frame of the corresponding lecture video with no more than 1 symbol of difference.

[3]https://www.cs.rit.edu/~dprl/Software.html#accessmath

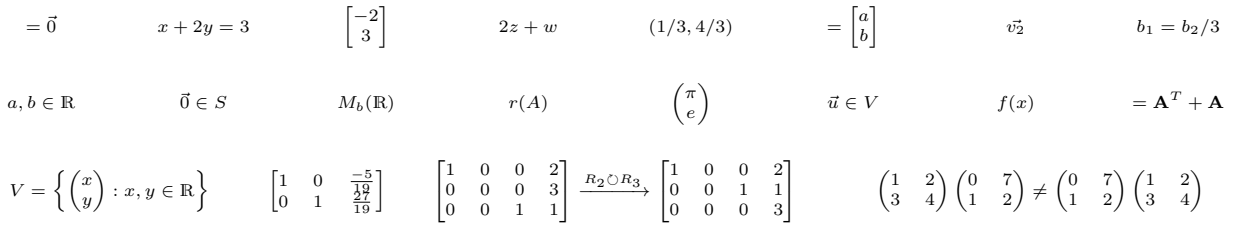


Fig. 4. Twenty Test Queries used in our Experiment. For space, only LaTeX versions are shown.

Third, the query must have at least 3 symbols. Fourth, the query is not a sub-expression of any other query previously selected. Finally, the sampled query does not completely contain any other query previously selected. The 20 rendered LaTeX expressions selected are used as typeset LaTeX queries and their corresponding handwritten versions extracted from the binary key-frame images. The final typeset queries from rendered LaTeX expressions are shown in Figure 4.

Indexing. After lecture video preprocessing, a total of 219 binary image keyframes binary are indexed from 13 videos, producing 2185 symbol pair index entries, and 788,780 pair instances in total. A total of 1471 rendered LaTeX formulas from 937 unique LaTeX strings are indexed on a parallel index for the lecture notes, which has 2,058 symbol pairs index entries, with 152,712 pair instances in total. The lecture video index takes 146 MB on disk while the lecture notes index requires 28.8 MB.

Retrieval Metrics. We evaluate retrieval performance using Recall@10 and Mean Reciprocal Rank (MRR). In this experiment, a target match is an identical match of the the query, possibly with additional unmatched elements. For query q in the evaluation set Q, we define r as the position in the ranked results where the query is found. The Reciprocal Rank (RR) of q is defined as follows:

$$RR = \begin{cases} \frac{1}{r} & if\ 1 \leq r \leq r_{max} \\ 0 & otherwise \end{cases}$$

where $r_{max} = 10$ as we only consider the top-10 matches for each query. The Mean Reciprocal Rank (MRR) as the mean of RR for all queries in the evaluation set Q. Recall@10 is the percentage of evaluation queries where the target match is found within the top-10 results.

VII. Results and Discussion

Search Modalities. Evaluation results are shown in Table I. Typeset images are very regular and both classification of symbols and LOS-graph structures are consistent for identical sub-expressions. On the other hand, handwritten images have a higher number of classification errors and inconsistency in graph structures. However, it was noticed that despite the existing errors, the system was able to find most targets as long as at least one small portion of the query was matched within the target image.

The highest Recall and MRR metrics are obtained by the same-modality search (see Table I), and as expected cross-modality search results is less accurate. Again, this is mostly due to the differences in representation consistency across modalities. In particular, the hardest modality is using handwritten content to match rendered LaTeX, as confirmed by this modality having the lowest MRR values in Table I. Handwritten queries have more irregularities and lower top label confidences, and they are used to match very regular images where such errors are rare, and despite these difficulties, our model is capable of finding most targets within the top-10 results for all conditions of this modality.

Symbol Recognition. All the handwritten content in the lecture videos used comes from a single writer. However, the math symbol classifier training set comes from a set of different writers, achieving a recognition rate of 88.85% in the Isolated Symbol Recognition task from the CROHME 2016 competition [2]. In a practical setting, we would expect that collections of lecture videos for a single course will have handwritten content produced by a single writer most of the time. The symbol classifier is expected to make some mistakes, however, we would expect such mistakes to be consistent for similar shapes given a single handwriting style, making it possible to match queries within writer even in the presence of multiple classification errors.

Ranking Metrics. On average, the proposed $h(M)$ consistently finds the targets at better ranks than $\alpha(M)$. We observe that a common mistake made by the $\alpha(M)$ metric is overweighting partial matches with very consistent pairs (high confidence labels and cosine similarities) over larger exact matches containing more inconsistent pairs (lower confidence labels and cosine similarities). This behavior can have a strong effect, especially for cross-modal search. On the other hand, $h(M)$ can be more balanced by preferring matches maximizing both node recall (based on label confidence) and edge recall (based on cosine similarity). This metric tends to prefer more spatially consistent matches than $\alpha(M)$.

In most cases, using the minimum of paired label confidences provided better MRR results than using the product. The only exception is found for h vs $h_\wedge$ in LaTeX to Whiteboard matching, where the difference can be explained by two queries having bad matches with good spatial alignment being ranked higher than targets having lower spatial consistency.

Using the size ratio filter $s_r(Q, C)$ also seems to help in most cases as long as a tolerant threshold is used. Extremely bad matches can be filtered very early in the processing when this filter is applied. However, cross-modality search is badly hurt if the selected threshold is too strict since size ratios

TABLE I

FORMULA IMAGE SEARCH RESULTS. RANKING METRIC α USES THE PRODUCT OF SYMBOL AND ANGULAR CONFIDENCES, WHILE h USES THE HARMONIC MEAN. $\wedge$ AND s REPRESENT USING MINIMUM SYMBOL CONFIDENCES RATHER THAN PRODUCTS, AND USING A SIZE DIFFERENCE THRESHOLD, RESPECTIVELY. $MRR@10$ IS THE MEAN RECIPROCAL RANK WITH QUERIES APPEARING AT RANK 11 OR LOWER SCORED AS 0.

	Recall@10						MRR@10					
	α	$\alpha_\wedge$	$\alpha_{\wedge s}$	h	$h_\wedge$	$h_{\wedge s}$	α	$\alpha_\wedge$	$\alpha_{\wedge s}$	h	$h_\wedge$	$h_{\wedge s}$
LaTeX	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**	0.98	**1.00**	**1.00**
Whiteboard	0.95	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.93	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
LaTeX $\rightarrow$ Wtbd	0.95	0.95	0.90	0.95	**1.00**	0.95	0.66	0.69	0.71	**0.89**	0.84	0.86
Wtbd $\rightarrow$ LaTeX	0.80	0.85	0.85	**0.90**	**0.90**	**0.90**	0.63	0.71	0.74	0.74	0.78	**0.84**

for LaTeX are very consistent, but handwriting has more size variations (even for a single writer).

Execution Times. Tangent-V is implemented in Python, and evaluates queries using a single thread. Using a laptop with an Intel core i7-4710 HQ processor and 16 GB of RAM, it takes average times of 1.0s and 1.3s to process same-modality LaTeX queries using $\alpha\,(M)$ and $h\,(M)$ respectively. For within-modality whiteboard queries, it takes average times of 34.1s and 41.5 s $\alpha\,(M)$ and $h\,(M)$ respectively. Using a faster programming language (e.g., C/C++) and early pruning for bad candidates could be used to speed up our approach.

VIII. CONCLUSION

We have presented a new search engine to bridge between formulas in whiteboard images extracted from lecture videos and formula images from typeset course notes (see Figure 1). Not unlike how word spotting techniques avoid challenges in recognizing handwritten text, we avoid recognizing structure in handwritten formulas to obtain strong search results using multiple symbol class hypotheses and angles between symbol pairs. Our system is available as open source, and could be adapted to other graphic types simply by retraining the symbol recognizers.

There are a number of directions for future work. Currently we do not index individual symbols. Can unsupervised symbol feature similarities rather than known a priori symbol classes be used? How can we accelerate search? How should we use vector-format images such as PDF, when exact symbol identities and classes are known, but are unknown in binary images? Finally, we might provide the math-aware operations such as variable unification, wildcard matching, and symmetry for commutative operators (e.g., 'a+b' matching 'b+a') found in symbolic math search engines [14].

Acknowledgments. This material is based upon work supported by the National Science Foundation (USA) under Grant No. HCC-1218801.

REFERENCES

[1] K. Davila and R. Zanibbi, "Whiteboard video summarization via spatio-temporal conflict minimization," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 355–362.

[2] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR 2016 CROHME: Competition on recognition of online handwritten mathematical expressions," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 607–612.

[3] K. Davila and R. Zanibbi, "Layout and semantics: Combining representations for mathematical formula search," *SIGIR*, pp. 1165–1168, 2017.

[4] S. Sudholt and G. A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *ICFHR*, Oct 2016, pp. 277–282.

[5] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*. IEEE, 2003, pp. 1470–1477.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*. IEEE, 2007, pp. 1–8.

[9] Y. Avrithis and G. Tolias, "Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval," *International Journal of Computer Vision*, vol. 107, no. 1, pp. 1–19, 2014.

[10] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *CVPR*, June 2015, pp. 5153–5161.

[11] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, 2010.

[12] W. Zhang and C.-W. Ngo, "Topological spatial verification for instance search," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1236–1247, Aug 2015.

[13] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *CVPR*. IEEE, 2011, pp. 809–816.

[14] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *IJDAR*, vol. 15, no. 4, pp. 331–357, 2012.

[15] R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topić, and K. Davila, "NTCIR-12 MathIR Task Overview," in *Proc. NTCIR-12*, 2016, pp. 299–308.

[16] S. Marinai, B. Miotti, and G. Soda, "Using earth mover's distance in the bag-of-visual-words model for mathematical symbol retrieval," in *International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2011, pp. 1309–1313.

[17] R. Zanibbi and L. Yu, "Math spotting: Retrieving math in technical documents using handwritten query images," in *ICDAR*. IEEE, 2011, pp. 446–451.

[18] H. Chatbri, P. Kwan, and K. Kameyama, "An application-independent and segmentation-free approach for spotting queries in document images," in *ICPR*. IEEE, 2014, pp. 2891–2896.

[19] K. Davila, S. Ludi, and R. Zanibbi, "Using off-line features and synthetic data for on-line handwritten math symbol recognition," in *ICFHR*. IEEE, 2014, pp. 323–328.

[20] K. Davila, "Symbolic and visual retrieval of mathematical notation using formula graph symbol pair matching and structural alignment," Ph.D. dissertation, Rochester Institute of Technology, 2017.

[21] R. Zanibbi, K. Davila, A. Kane, and F. Tompa, "Multi-stage math formula search: Using appearance-based similarity metrics at scale," *SIGIR*, pp. 145–154, 2016.