Keyword and Image-Based Retrieval for Mathematical Expressions

Richard Zanibbi[a] and Bo Yuan[b]

[a]Department of Computer Science
[b]Department of Networking, Security, and Systems Administration
Rochester Institute of Technology, Rochester, NY, USA

ABSTRACT

Two new methods for retrieving mathematical expressions using conventional keyword search and expression images are presented. An expression-level TF-IDF (term frequency-inverse document frequency) approach is used for keyword search, where queries and indexed expressions are represented by keywords taken from LaTeX strings. TF-IDF is computed at the level of individual expressions rather than documents to increase the precision of matching. The second retrieval technique is a form of Content-Based Image Retrieval (CBIR). Expressions are segmented into connected components, and then components in the query expression and each expression in the collection are matched using contour and density features, aspect ratios, and relative positions. In an experiment using ten randomly sampled queries from a corpus of over 22,000 expressions, precision-at-k ($k = 20$) for the keyword-based approach was higher (keyword: $\mu = 84.0, \sigma = 19.0$, image-based: $\mu = 32.0, \sigma = 30.7$), but for a few of the queries better results were obtained using a combination of the two techniques.

Keywords: information retrieval, mathematical information retrieval, content-based image retrieval (CBIR), TF-IDF

1. INTRODUCTION

One can imagine situations where one would like to use a mathematical image directly for search. For example, one might wish to find similar expressions within a document being read using a .pdf viewer, or to find similar expressions within a database of technical documents such as in a conference proceedings or in the online arXiv repository*. One might also wish to retrieve math based on its structural or semantic encoding, e.g. using LaTeX, MathML or OpenMath.[1] The aim of this work is to devise retrieval methods that will be well-suited to these tasks. In this paper we consider a combination of novel keyword and image-based techniques.

Current math retrieval methods are almost entirely text-based (often using markup text, such as MathML or LaTeX, such as for Springer's recently released LaTeX-based search†). Most of these adapt vector-based term frequency-inverse document frequency (TF-IDF) methods,[2] perhaps employing parse tree paths for structured representations, e.g. MathML. In this paper we present a new TF-IDF indexing and retrieval method that operates at on individual expressions, rather than complete documents, in an effort to produce more precise retrieval results.

Image-based math retrieval is a new area of research. Previously Marinai et al.[3] devised a method for image-based retrieval of individual mathematical symbols using shape contexts. We present a new technique for retrieving *whole* expressions using images as queries, employing a simple Content-Based Image Retrieval (CBIR[4]) approach. A query image is first binarized and decomposed into connected components, after which the query components are matched with the components of expression images in the collection. Matches are currently performed using a greedy algorithm, minimizing the cost of adding a region to the match based on contour and density features, aspect ratios, and the relative positions of components. Additional penalties are computed for unmatched connected components in the query, and for candidate region components lying within the bounding box of components matched to queries in the candidate expression.

Further author information (send correspondence to R. Zanibbi):
R. Zanibbi: E-mail: rlaz@cs.rit.edu, Telephone: 1 585 475 5023
B. Yuan: E-mail: bo.yuan@rit.edu, Telephone: 1 585 475 4468

*`http://arxiv.org`
†http://www.latexsearch.com/

Our approach is an adaptation of earlier work by Yu,[5] in which images of handwritten queries were matched to expressions in document images using XY-trees[6] and contour matching.

We assume that mathematical expressions of interest are encoded as math regions in a LaTeX document (normally demarcated by dollar signs ($)). We have created a modified version of the `latex2html` utility, in order to record each math region as a separate LaTeX string and corresponding rendered image. In our preliminary experiments we have used documents from the arXiv. In essence, in this investigation we assume that mathematical expressions have been segmented correctly by the authors of the technical papers.

At the outset of this project, our expectation was that keyword-based retrieval would better represent individual symbols and relationships (e.g. superscript/upper arg, subscript/lower arg), visual matching would better represent symbol layout and symbols with similar appearance in expressions, and that their combination would yield more usable retrieval results. The retrieval methods are described in the next section, and results from a preliminary experiment are presented in Section 3. We discuss the results further and conclude in Sections 4 and 5.

2. METHODOLOGY

In this section we present two new methods for retrieving mathematical expressions. The first method adapts conventional term frequency-inverse document frequency (TF-IDF) approaches to allow indexing and retrieval at the level of individual expressions (see Section 2.1). The second approach uses a typeset image directly as a query, and performs retrieval based on image features (Section 2.2).

2.1 Keyword-Based Retrieval

Miller and Youssef first attempted to apply a classic, mature text search engine technology to solve the mathematical search problem.[2] They first substitute mathematical symbols with predefined text terms. For example, consider the expression $x^{t-2} = 1$. The LaTeX string:

```
x^{t-2} = 1
```

is translated into:

```
x BeginExponent t minus 2 EndExponent Equal 1
```

where symbols $\wedge$, - and = and parentheses are substituted with their corresponding text keywords. In this way, mathematical symbols become indexable and searchable terms in a full text search engine. The authors claim that the approach is reasonably successful, although quantitative performance measures are not provided. However, Youssef indicates later[7] that standard TF-IDF based relevance measures at the level of entire documents are inadequate for math search, as it becomes difficult for users to go through entire documents to locate a relevant formula, leading him to propose hit-summary generation methods.[7]

Recently Hijikata et al, applied Boolean search to retrieving mathematical formulas.[8] In their study, mathematical formulas are specified as MathML objects, and MathML object keywords are indexed for search. Their search methods, depth first, width first and hybrid were compared. The advantage of this approach is that MathML maintains the tree structure of a formula and when searching, the tree structure can be compared with the tree structure of a query. The disadvantage of this approach is that Boolean search does not weight the importance of terms, which has been found to be inferior to TF-IDF-based search for text documents.

In our keyword-based approach, LaTeXexpressions of formulas are converted to bags of searchable words. This approach is similar to what was first suggested by Miller and Youssef.[2] The novelty of our approach is to treat each formula as a separate document. Thus a query is matched against mathematical formulas (delimited by $ . . . $ in LaTeX) directly. When a vector space model is used to index formulas, the inverted document frequency (IDF) of a term becomes the inverted formula frequency (IFF), and is then used to calculate similarities between queries and formulas, and hence the relevance ranking of retrieved formulas. By separating individual formulas from non-mathematical text, this approach allows more targeted matching and avoids noise that may be induced by text and similar formulas. IFF helps weighting importance of different keywords in all formulas; while the term frequency (TF) weights the importance of keywords within

Table 1. Example Keyword Substitutions

Symbol	LaTeX	Type	Keywords
α	\alpha	Greek letter	alpha
$\pm$	\pm	binary operation	pm
$\sqrt{x}$	\sqrt x	unary operation	sqrt x
$\ln$	\ln	math functions	ln
$\langle$	\langle	delimiter	langle
x_a	x_a	subscript	x subscript a
x^b	x^b	superscript	x superscript b
(	(	fence	leftp
)	)	fence	rightp
[	[	fence	leftsp
]	]	fence	rightsp

Symbol	LaTeX	Type	Keywords
{	{	fence	leftcp
}	}	fence	rightcp
+	+	binary operation	plus
-	-	binary operation	minus
=	=	binary operation	equal
$\vert$	$\vert$	vertical bar	vbar
$>$	$>$	greater than	greaterthan
$<$	$<$	less than	lessthan
/	/	binary operation	division

a formula. Keywords such as “leftp" and “rightp" will have low IFF, because they appear in many formulas, but still carry some mathematical meaning when combined with other symbols. Term frequency in formulas is also important because if there are two $\sum$ symbols in a query, the matching formulas should also have two $\sum$ symbols.

In our approach, keywords for queries and document indices are produced using the following steps:

1. Insert spaces before back slashes
2. Tokenize the string, using regular expressions defined for different symbol types. For example, we isolate unary and binary operations, arrows, Greek letters, etc. embedded within tokenized ‘words’ (e.g. `'2+2'` is converted to three tokens).
3. Lookup keywords in a substitution table. Table 1 illustrates some substitutions.
4. Remove TEX commands that do not define expression symbols or their layout. For example, the commands `display` and `array` do not carry mathematical information.

Our keyword-based prototype has been implemented in Java using the well-known Lucene[9] library. Note that when applying the Lucene search engine, an ‘everything’ lexical analyzer is created. Unlike Lucene’s standard lexical analysis, the everything analyzer does not remove dots (.) or apostrophes (’s) after words, and does not filter stop words (e.g. ‘and’, ‘the’, etc.).

2.2 Image-Based Retrieval

In the section we describe a new method for using typeset images generated by `latex2html` to search amongst a database of expression images produced by our adapted `latex2html` converter. Knowing that the expressions are typeset by the same system, we use simple visual features to obtain usable retrieval results. The approach outlined in the following is an extension of earlier work in which images of handwritten queries are used to retrieve expressions in document images.[5] We are able to make a number of simplifications to this approach, due to expressions already being segmented for the most part by the authors of the LaTeX documents.

2.2.1 Visual Features

Our image representation for retrieval is simple. After binarizing an expression image, we extract the connected components. For each component, we then compute an upper and lower contour, as well as a measure of the pixel density in the vertical direction, represented as the ratio of the vertical span between the two contours that contain foreground pixels. Our shape representation is similar to that used by Rath and Manmatha for spotting handwritten words.[5,10] Connected components are represented by a vector of triples, with each representing the 1) upper contour location, 2) lower contour location, and 3) *density* (ratio) of foreground pixels in the interval containing the upper and lower contour locations (1.0 represents a solid line).

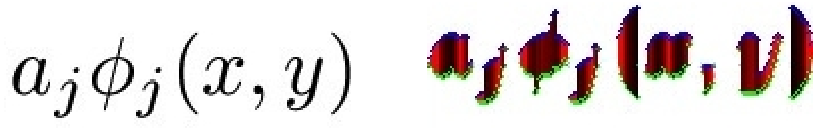

a. Expression Image b. Contours and Density Features

Figure 1. Visual Features. At right a visualization of the contour and density features extracted for the connected components in the expression at left is shown. The contour features are represented by a vector of triples, one for each column of the image, indicating the location of the upper and lower contours, and the pixel density between the upper and lower contours. Bright red indicates primarily foreground pixels between contours, while black indicates primarily background pixels between contours

The final feature representation for connected component shape is a ten-element vector. The first nine elements are a coarse histogram feature, encoding the percentage of columns for which a contour or density value 1) remains the same, 2) decreases, or 3) increases relative to the previous column, moving from left-to-right over a connected component image. The feature vector contains three of these histograms, one for each of the upper contour, lower contour, and density values. In our current implementation, in preprocessing we first average the raw contour feature vector across every two columns, to reduce sensitivity to outliers, as well as storage requirements for expressions in the collection. The tenth feature is the aspect ratio of the connected component.

Previously we decomposed expression images using X-Y cutting (similar to projection-profile cutting for recognizing typeset math[5, 11, 12]) and then matched sub-images using contour features. However, we realized that in doing so it becomes difficult to match sets of symbols/components that do not appear alone in a sub-image and/or span two or more cut regions, and so changed to our current representation.

2.2.2 Connected Component-Based Indexing and Querying

Each expression image in the search index is stored with the bounding boxes and shape features of its connected components. Retrieval is performed by finding the minimum cost correspondence of the connected components in the query and each expression image stored in the index, following by sorting expressions in the collection in increasing order of cost. Let $ssd()$ be a function returning the sum of squared differences between the elements of two vectors, and $c_f(r)$, and $a_f(r)$ be the contour summary and aspect ratio features for a connected component r. We define the visual distance $dist(r,q)$ between two connected components r and q as:

$$dist(r,q) \;=\; ssd(c_f(q),\; c_f(r)) \;+\; |a_f(q) \;-\; a_f(r)| \tag{1}$$

We found that using the absolute rather than the squared difference of the aspect ratio led to better visual matches, with the opposite being true for the contour features.

The minimum distance $dist(R,Q)$ between an expression image R and query image Q is currently estimated using a greedy algorithm. First, each connected component of the query is compared to every component in the expression collection. Then the minimum cost match between a query and index connected component is chosen as the first match. We record which query and index components are the minimum cost shape match (q_{min}, r_{min}); these are used to define an additional penalty for subsequent components added to the match, the difference in the displacement vectors between the top-left $(tl())$ and bottom-right $(br())$ hand corners of the first minimum cost match and a new candidate match $(r,\, q)$. To provide some scale invariance, the displacements are normalized by the areas of q_{min} and r_{min}.

$$ldist(q, q_{min}, r, r_{min}) \;\;=\;\; \left|\frac{tl(q)-tl(q_{min})}{area(q_{min})} - \frac{tl(r)-tl(r_{min})}{area(r_{min})}\right|^2 + \left|\frac{br(q)-br(q_{min})}{area(q_{min})} - \frac{br(r)-br(r_{min})}{area(r_{min})}\right|^2 \tag{2}$$

Intuitively, this is a layout penalty, based on the relative angle and distance between query connected component q (associated with index component r) and the best-matched query q_{min} (associated with index component r_{min}).

The cost $c(R,Q)$ for matching the components of a query Q and index expression R is defined as in the following:

$$c(R,Q) = \alpha \cdot dist(r_{min}, q_{min}) + \sum_{i=1}^{m} \min_{\substack{r_i \in R,\, q_i \in Q \\ r_i \notin \{r_{min}, r_1, \ldots, r_{i-1}\} \\ q_i \notin \{q_{min}, q_1, \ldots, q_{i-1}\}}} \alpha \cdot dist(r_i, q_i) + \beta \cdot ldist(q_i, q_{min}, r_i, r_{min}) \tag{3}$$

where $m = min(|R-1|, |Q-1|)$ (the match size) and α and β are weighting parameters for the shape and layout distances, respectively.

Once the (greedy) minimum cost match $A = \{(q_{min}, r_{min}), (q_1, r_1), \ldots, (q_m, r_m)\}$ is determined, additional costs for unmatched query components and components in the index expression image that lie in the bounding box of the match region are added, to produce the final match cost for an expression image R.

$$matchcost(R, Q) = c(R, Q) \; + \; (|Q| - |A|) \cdot (c(R, Q) + \gamma) \; + \; \gamma \cdot inside(A, R) \tag{4}$$

where $inside(A, R)$ counts the number of unmatched components in R whose center intersects the bounding box of the components in A.

Our prototype for the visual indexing and search is implemented in C++ using the OpenCV library.

3. RESULTS

Table 3 presents some preliminary results for our TEX and image-based math search prototypes. We use *precision-at-k* as our performance metric, which is simply the number of relevant expressions retrieved in the top-k results, for $k = 20$. From a corpus of 26,737 expressions extracted automatically from 50 LATEX documents downloaded from the arXiv, the 10 query expressions shown were selected at random. Expressions were extracted and indexed using a modified version of the `latex2html` utility. The image-based retrieval algorithm was run using the parameters $\alpha = 1.0, \beta = 6.0$ and $\gamma = 3.0$. These parameters were set empirically using a small subset of the corpus not included in the test set. For the combined algorithm, the top 40 matches returned by the LATEX-based retrieval were re-ranked using the image-based system.

The Lucene-based LATEX retrieval system is quite fast, returning retrieval results in well under a second on a desktop machine (Intel Core-2 Duo processor, 4GB RAM, Linux OS (Ubuntu 10.04)). On the same system, the retrieval time for the visual search increases with the number of connected components in the query: query 8 takes 25 seconds to execute, while query 10 takes 1 minute and 36 seconds. For the combined search, the image-based search is much faster after the Lucene results are produced, as only 40 expressions are ranked rather than the full expression set.

One of the authors produced relevance assessments using a weighted ranking scheme, with 0 given for irrelevant expressions, 0.5 for expressions with a small number of shared symbols or structural relationships (a 'weak' match), and 1.0 where a subexpression or expression is clearly present in a returned result. Assessing the relevance of retrieved expression was often challenging in the case of weak matches. For example, sometimes a retrieved expression may contain a symbol from the query, but differ so markedly in structure that it is unclear whether one can consider it a weak match. For the image-based retrieval, where visual features rather than symbol identity are the basis for retrieval, expressions with a completely different set of symbols but similar or identical layout would sometimes be returned. This was counted as a weak match (0.5). A possible confound here is that with only 50 documents in the collection, there may be few expressions similar to some of the queries. In Table 3 we present precision-at-k values in both weighted and unweighted form, where the unweighted precision numbers are computed by counting all 'weak' matches (0.5) as proper matches (1.0).

As seen in the table, overall the keyword (LATEX)-based retrieval outperforms both the image-based and combined matching algorithms. However, for queries 4, 6 and 7, the system combining LATEX and image-based retrieval outperforms either running alone. It appears that combining textual and visual features for expression retrieval may yet prove beneficial, but our current image-based matching algorithm is in need of improvement: for two of the queries (6 and 10), no relevant expressions were returned (in one case, due to a bug), and the other precision values are generally low with the exception of query 1.

To illustrate the behavior of the algorithms, we have provided the top 20 retrieval results for the keyword (LATEX)-based search for query 3 in Table 3, and the top 20 retrieval results for the keyword, image, and combined approaches for query 7 in Tables 4 and 5. In Table 4, connected components matched to connected components in the query are shown using red boxes.

Table 2. Precision-at-k (k=20) for 10 queries (collection: 26,737 expressions from 50 LaTeX documents). Weighted matches were assigned using values 0, 0.5 and 1.0, for no match, some shared symbols/structure, and subexpression/expression matching. Unweighted precision values include all expressions assigned weight 0.5 or 1.0 (i.e. expressions that were not labeled as irrelevant). Results are reported for the Lucene-based approach (using LaTeX strings), visual matching (using images generated from the LaTeX), and a combination where the top 40 Lucene matches are re-ranked using image-based matching.

		Weighted Matches			Unweighted Matches		
	Queries (Images)	LaTeX	Image	Combined	LaTeX	Image	Combined
1.	$d,$	100.0	100.0	95.0	100.0	100.0	95.0
2.	$L_1 \times L_2 \times L_3$	77.5	20.0	67.5	90.0	35.0	85.0
3.	$\{v \in W_0^{1,p}(D) :\ v \geq \psi \text{ a.e. in } D\}$	80.0	15.0	47.5	95.0	20.0	70.0
4.	e_{n+1}	65.0	37.5	75.0	70.0	60.0	85.0
5.	$\mathcal{U}'$	87.5	10.0	80.0	100.0	15.0	95.0
6.	$\prod_{y \in \Sigma} k(y) \to \prod_{y \in \Sigma'} k(y)$	35.0	0.0	45.0	55.0	0.0	70.0
7.	$\mathbf{x}^{\mathbf{u}_1}$	50.0	20.0	57.5	50.0	20.0	65.0
8.	$D^n = CS^{n-1}$	60.0	27.5	47.5	95.0	50.0	85.0
9.	$\Omega_{\tau a}$	87.5	10.0	77.5	100.0	20.0	100.0
10.	$\Omega = \frac{h^2}{m}(k_1^2 + k_2^2) + v_0 + v_{2k_2},$	72.5	0.0	57.5	85.0	0.0	75.0
	Mean (μ)	71.5	24.0	65.0	84.0	32.0	82.5
	Standard Deviation (σ)	19.4	29.1	16.7	19.0	30.7	12.1

4. DISCUSSION

The keyword-based search performs well. It is effective at retrieving specific symbols and spatial relationships as expected (e.g. note the matches for the subexpression '$\in W_0^{1-p}(D)$' shown in Table 3). We did not normalize our LaTeX strings (e.g. insure that subscripts always precede superscripts), so it is possible that we might improve the performance of the keyword-based search in this way.[2]

Although the precision-at-k values are low for the current image-based search, they are non-zero for most of the test queries. This indicates that our image-based method might be used to initiate searches of collections of expressions represented as LaTeX strings, or expression images (for related work using images of handwritten expressions as queries, see Yu[5]). We also hope to improve the precision of our image-based algorithm, e.g. through an alternative matching algorithm and shape representations (e.g. Hu or Zernicke moments). We currently use a table containing all expression images and their connected components. One could speed up retrieval significantly using inverted files,[13] mapping shape features to connected components in expression images, so that each unique shape vector is compared to only once. Additionally, the feature vectors could be organized into a cluster hierarchy, reducing the number of comparisons further.[13]

A user's search strategy is quite different, and thus the corresponding requirements for a retrieval algorithm, if they are attempting to retrieve a specific expression, find expressions similar to a chosen one, or simply browse through a collection of expressions images.[‡] Visual matching provides a different, but complementary method of retrieval relative to keyword-based retrieval (e.g. of LaTeX), and may be particularly well-suited to browsing. Even in the presence of strong OCR methods that can convert expression images to string encodings (e.g. the infty system[16]), matching visual features may be useful, as it permits different types of matches than those obtained through string-based matching.

[‡]see Westman[14] and Smeulders et al.[15] for discussions of information seeking behaviors associated with image search.

Table 3. Top 20 Keyword-Based (LaTeX) Matches for Query 3

Ranks 1-10	Ranks 11-20
$\{v \in W_0^{1,p}(D):\ v \geq \psi \text{ a.e. in } D\}$	$w_\delta^\epsilon(x,\omega) = \inf\{v(x):\ \triangle_p v \leq \alpha_0 + \delta \text{ in } D_\epsilon,\ v \geq 1 \text{ on } T_\epsilon \text{ and } v = 0 \text{ on } \partial D\}$
$K_\epsilon = \{v \in W_0^{1,p}(D) : v \geq 0 \text{ a. e. on } T_\epsilon\}$	$v_{\alpha_0,D}^\epsilon(x,\omega) \geq h_k^\epsilon(x,\omega) - o(1),\ \text{a. e. } x \in B_{\frac{\epsilon}{2}}(\epsilon k) \text{ and a. s. } \omega \in \Omega$
$\mathcal{F}_0(u^0) = \inf\limits_{v \in W_0^{1,p}(D)} \mathcal{F}_0(v).$	$v_{\alpha_0,D}^\epsilon(x,\omega) \geq h_k^\epsilon(x,\omega) - o(1),\ \text{a. e. } x \in B_{\frac{\epsilon}{2}}(\epsilon k) \text{ and a. s. } \omega \in \Omega$
$\{\phi \in C_0^1(D):\ \phi_- \in C_0^1(D) \text{ and } \phi \geq \psi \text{ a.e. in } D\}$	$\min\{\int_D \frac{1}{p}\lvert\nabla v\rvert^p + \frac{1}{p}\alpha_0 v_-^p - fvdx:\ \forall\ v \in W_0^{1,p}(D)\}$ $\square$
$\min\{ \int_D \frac{1}{p}\lvert\nabla u\rvert^p dx - \int_D fudx : u \in W_0^{1,p}(D),\ u \geq 0 \text{ a. e. in } T_\epsilon(\omega)\}$	$\min\{\int_D \frac{1}{p}\lvert\nabla v\rvert^p + \frac{1}{p}\alpha_0 v_-^p - fvdx:\ \forall\ v \in W_0^{1,p}(D)\}$
$\min\{\int_{\mathbf{R}^n} \frac{1}{p}\lvert\nabla v\rvert^p - fvdx:\ u \in W_0^{1,p}(D),\ u \geq 0 \text{ a. e. in } T_\epsilon(\omega)\}$	$\overline{v}_{\alpha_0+\delta,D}^\epsilon = \min(v_{\alpha_0+\delta,D}^\epsilon,\ 1)\ \textit{converges to } 0 \textit{ in } L^p(D)$
$\min\{\int_D \frac{1}{p}\lvert\nabla v\rvert^p - fvdx:\ v \in W_0^{1,p}(D) \text{ and }\ v \geq \psi^\epsilon\}$	$w^\epsilon(x,\omega) = \inf\{v(x):\ \triangle_p v \leq \alpha_0 \text{ in } D_\epsilon,\ v \geq 1 \text{ on } T_\epsilon \text{ and } v = 0 \text{ on } \partial D\}$
$\min\{\int_{\mathbf{R}^n} \frac{1}{p}\lvert\nabla v\rvert^p - fvdx:\ v \in W_0^{1,p}(D),\ v \geq \psi^\epsilon \text{ a. e. in } D$	$\begin{aligned} \int_D \lvert\nabla w^\epsilon\rvert^p dx &\leq (\int_D \lvert\nabla w^\epsilon\rvert^p)^{\frac{p-1}{p}} \cdot (\int_D \lvert\nabla h^\epsilon\rvert^p dx)^{\frac{1}{p}} \\ &+ \alpha_0 \int_D (h^\epsilon - w^\epsilon)dx \\ &\leq \frac{p-1}{p}\int_D \lvert\nabla w^\epsilon\rvert^p dx + \frac{1}{p}\int_D \lvert\nabla h^\epsilon\rvert^p dx + C\alpha_0 \end{aligned}$
$\min\{\int_D \frac{1}{p}\lvert\nabla v\rvert^p + \frac{1}{p}\alpha_0 v_-^p - fvdx:\ v \in W_0^{1,p}(D) \text{ and } v \geq \psi \text{ a. e. in } D\}$	$v_{\alpha,B_1}^\epsilon \geq \frac{\beta}{c(n,p)}\lvert x - x_0\rvert^{\frac{p}{p-1}} - \frac{\beta}{c(n,p)} > 0 \text{ in } B_1(x_0).$
$\min\{\int_D \frac{1}{p}\lvert\nabla v\rvert^p + \frac{1}{p}\alpha_0 v_-^p - fvdx:\ v \in W_0^{1,p}(D) \text{ and } v \geq \psi \text{ a. e. in } D\}.$ $\square$	$F' \geq 0, F' \in C_0^\infty\left(\left(\frac{1}{2}, \frac{3}{4}\right)\right)$

5. CONCLUSION

We have proposed two new systems for retrieving mathematical expressions using LaTeX strings and expression images, along with a system combining the approaches, where LaTeX-based retrieval is applied first, followed by re-ranking of the top 40 results by the image-based algorithm. In our preliminary experiment, image-based matching in the combined algorithm increased precision-at-k ($k = 20$) measures over that of LaTeX-based retrieval alone for 3 out of 10 test queries, but the LaTeX based technique had much higher precision over the test set ($\mu = 71.5, \sigma = 19.4$ for weighted matches, and $\mu = 84.0, \sigma = 19.0$ for unweighted matches), as seen in Table 3. Our current visual retrieval technique has low precision-at-k ($\mu = 32.0, \sigma = 30.7$ for unweighted matches), but performs well enough that it might be used to initiate a search. We plan to improve our visual and keyword-based matching techniques, and continue to explore how they may be usefully combined with one another, and with OCR methods. Improving our understanding of relevance assessment for math retrieval is another important direction for further investigation.

ACKNOWLEDGMENTS

We wish to thank George Nagy for helpful discussions. This research was partially funded by the RIT Golisano College of Computing and Information Science, the Xerox Foundation through a UAC Grant held by the first author, and the National Science Foundation under Grant No. IIS-1016815.

REFERENCES

[1] Davenport, J. and Kohlhase, M., "Unifying math ontologies: A tale of two standards," *LNCS* **5625**, 263–278 (2009).

[2] Miller, B. R. and Youssef, A., "Technical aspects of the digital library of mathematical functions," *Annals of Mathematical and Artificial Intelligence* **38**, 121–136 (2003).

[3] Marinai, S., Miotti, B., and Soda, G., "Mathematical symbol indexing using topologically ordered clusters of shape contexts," in [*ICDAR*], 1041–1045 (2009).

Table 4. Top 10 Matches for Query 7

LaTeX	Image	Combined
$\mathbf{x}^{\mathbf{u}_1}(1-\mathbf{x}^{\mathbf{v}_1-\mathbf{u}_1})$	$\mathbf{x}^{\mathbf{u}_1}$	$\mathbf{x}^{\mathbf{u}_1}$
$\mathbf{x}^{\mathbf{v}_1}(\mathbf{x}^{\mathbf{u}_1-\mathbf{v}_1}-1)$	$\mathbf{x}^{\mathbf{v}_1}$	$\mathbf{x}^{\mathbf{v}_1}$
$f_1=\mathbf{x}^{\mathbf{u}_1}-\mathbf{x}^{\mathbf{v}_1}, f_2=\mathbf{x}^{\mathbf{u}_2}-\mathbf{x}^{\mathbf{v}_2},\ldots$	$f_i \xrightarrow{w^*} 0$	$\mathbf{x}^{\mathbf{u}_1}(1-\mathbf{x}^{\mathbf{v}_1-\mathbf{u}_1})$
$\mathbf{x}^{\mathbf{u}_1}$	$f_i \xrightarrow{w^*} 0$	$\mathbf{x}^{\mathbf{u}_1-\mathbf{v}_1}-1$
$\mathbf{x}^{\mathbf{v}_1}$	$\frac{2m+3}{3}$	$\mathbf{x}^{\mathbf{v}_1}(\mathbf{x}^{\mathbf{u}_1-\mathbf{v}_1}-1)$
$1-\mathbf{x}^{\mathbf{v}_1-\mathbf{u}_1}$	$\frac{2m+3}{3}$	$a_{\mathbf{v}_2} \in \mathbb{Z}^3$
$\mathbf{x}^{\mathbf{u}_1-\mathbf{v}_1}-1$	$\frac{2m+4}{4}$	$x \in \mathbf{R}^{2m}$
$f_1=\mathbf{x}^{\mathbf{u}_1}-\mathbf{x}^{\mathbf{v}_1}$	$\frac{2m+3}{3}$	$a_{\mathbf{u}_2} \in \mathbb{Z}^3$
$V=\begin{bmatrix}\mathbf{1}_g & 0\\ C & \mathbf{1}_g\end{bmatrix}, \quad {}^tC=C, \quad C\in\mathbf{M}_g(\mathbb{Z}).$	$\frac{2m+4}{4}$	$\mathbf{R}^{2m-1}$
$U=\begin{bmatrix}\mathbf{1}_g & B\\ 0 & \mathbf{1}_g\end{bmatrix}, \quad {}^tB=B, \quad B\in\mathbf{M}_g(\mathbb{Z}).$	$\frac{2m+3}{3}$	$A.{}^tD-B.{}^tC=\mathbf{1}_g$

[4] Datta, R., Joshi, D., Li, J., and Wang, J. Z., "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys* **40**(2), 1–60 (2008).

[5] Li, Y., *Image-Based Math Retrieval Using Handwritten Queries*, Master's thesis, Rochester Institute of Technology, Rochester, NY (May 2010).

[6] Nagy, G. and Seth, S., "Hierarchical representation of optically scanned documents," in [*Proc. Seventh Int'l Conf. Pattern Recognition*], 347–349 (1984).

[7] Youssef, A. S., "Methods of relevance ranking and hit-content generation in math search," in [*Proc. Calculemus '07*], 393–406, Springer-Verlag, Berlin, Heidelberg (2007).

[8] Hijikata, Y., Hashimoto, H., and Nishida, S., [*Human Interface and the Management of Information. Designing Information Environments*], vol. 5617 of *Lecture Notes in Computer Science*, ch. Search Mathematical Formulas by Mathematical Formulas, 404–411, Springer (2009).

[9] McCandless, M., Hatcher, E., and Gospodnetic, O., [*Lucene in Action, Second Edition*], Manning Publications, Stamford, CT (2010).

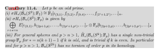
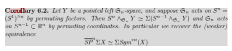
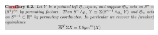
[10] Rath, T. and Manmatha, R., "Word spotting for historical documents," *International Journal on Document Analysis and Recognition* **9**, 139–152 (2007).

[11] Okamoto, N. and Miao, B., "Recognition of mathematical expressions by using the layout structures of symbols," in [*ICDAR*], **1**, 242–250 (1991).

[12] Twaakyondo, H. M. and Okamoto, M., "Structure analysis and recognition of mathematical expressions," in [*ICDAR*], **1** (1995).

[13] Salton, G. and McGill, M. J., [*Introduction to Modern Information Retrieval*], McGraw-Hill, Inc., New York, NY, USA (1983).

Table 5. Top 11-20 Matches for Query 7

LaTeX	Image	Combined
$(\mathbf{x}^{\mathbf{u}_1}, \mathbf{x}^{\mathbf{v}_1})$	$\boxed{m} \boxed{\in} \boxed{\mathbb{N}}$	$\begin{bmatrix} \boxed{A} & \boxed{B} \\ \boxed{C} & D \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & {}^tA \end{bmatrix} = \begin{bmatrix} \mathbf{1}_g & B^tA \\ CA^{-1} & D^tA \end{bmatrix} \in \Gamma_g^1(2).$
$S = R[(\mathbf{x}^{\mathbf{u}_1-\mathbf{v}_1})^{\pm 1}, (\mathbf{x}^{\mathbf{u}_2-\mathbf{v}_2})^{\pm 1}, \ldots],$	$\boxed{A} \boxed{\Rightarrow} \boxed{x}$	$\boxed{\mathbf{I}} - \boxed{AA^*} = \mathbf{I} - H_M(H_E^* H_E)^{-1} H_M^* \succ 0.$
$g_1^-, g_1^+ \in C_0^\infty(\mathbf{R})$	$\boxed{\mathbf{x}}^{\boxed{\mathbf{u}_1}}(1 - \mathbf{x}^{\mathbf{v}_1 - \mathbf{u}_1})$	$1 - \boxed{\mathbf{x}}^{\boxed{\mathbf{v}}_1 - \boxed{\mathbf{u}}_1}$
$\frac{d}{dt}q^- = 2\mathrm{Re}\left\langle \gamma^u, \frac{d}{dt}\gamma^u \right\rangle_{\mathbf{C}^{n^u}} - 2\mathrm{Re}\left\langle \gamma^s, \frac{d}{dt}\gamma^s \right\rangle_{<tex2html_comment_mark>45\mathbf{C}^{n^s}} \geq \delta^- t^{-1} q^+$	$\boxed{\mathbf{x}}^{\boxed{\mathbf{u}_1} - \mathbf{v}_1} - 1$	$(\boxed{\mathbf{x}}^{\boxed{\mathbf{u}}_1}, \mathbf{x}^{\boxed{\mathbf{v}}_1})$
$P = \begin{bmatrix} A & B \\ 0 & {}^tA^{-1} \end{bmatrix}, \quad A.{}^tB = B.{}^tA, \quad A \in \mathrm{GL}_g(\mathbb{Z}), \quad B \in \mathbf{M}_g(\mathbb{Z}).$	$\frac{\boxed{5m}\boxed{+}\boxed{2}}{2}$	$g_{\boxed{1}} \boxed{\in} C_0^{\boxed{\infty}}(\mathbf{R})$
${}^tA.D - {}^tC.B = \mathbf{1}_g$	$\frac{\boxed{5m}\boxed{+}\boxed{2}}{2}$	$\sup_{x,\xi \boxed{\in} \mathbf{R}^{\boxed{n}}, t \geq t_0} \lvert a_t(x,\xi) \rvert \leq \nu_0,$
$A.{}^tD - B.{}^tC = \mathbf{1}_g$	$\boxed{X_\bullet} \boxed{\Rightarrow} X$	$\boxed{f}_1 = \boxed{\mathbf{x}}^{\boxed{\mathbf{u}}_1} - \mathbf{x}^{\mathbf{v}_1}$
$U = U'^2 = \begin{bmatrix} \mathbf{1}_3 & 2B \\ 0 & \mathbf{1}_3 \end{bmatrix},$	 Corollary 11.4. Let p be an odd prime. [illegible]	$\boxed{f}_1 = \boxed{\mathbf{x}}^{\boxed{\mathbf{u}}_1} - \mathbf{x}^{\mathbf{v}_1}, f_2 = \mathbf{x}^{\mathbf{u}_2} - \mathbf{x}^{\mathbf{v}_2}, \ldots$
$g_1, \tilde{g}_1, g_2, \tilde{g}_2 \in C_0^\infty(\mathbf{R}^n)$	 Corollary 6.2. Let Y be a pointed left $\mathfrak{S}_n$-space, and suppose $\mathfrak{S}_n$ acts on $S^n = (S^1)^{\wedge n}$ by permuting factors. Then $S^n \wedge_{\mathfrak{S}_n} Y \simeq \Sigma(S^{n-1} \wedge_{\mathfrak{S}_n} Y)$ and $\mathfrak{S}_n$ acts on $S^{n-1} \subset \mathbb{R}^n$ by permuting coordinates. In particular we recover the (weaker) equivalence $\overline{SP}^n \Sigma X \simeq \Sigma Sym^{*n}(X)$	$\boxed{U} = U^{\boxed{\prime 2}} = \begin{bmatrix} \mathbf{1}_3 & 2B \\ 0 & \mathbf{1}_3 \end{bmatrix},$
$g_1 \in C_0^\infty(\mathbf{R})$	 Corollary 6.2. [illegible]	$\boxed{\log}\boxed{\det}\left(\mathbf{I}_n \boxed{+} (H_M^*,\ H_E^*) \begin{pmatrix} \mathbf{I}_{n_M} & A \\ A^* & \mathbf{I}_{n_E} \end{pmatrix}^{-1} \begin{pmatrix} H_M \\ H_E \end{pmatrix} K_X \right) - \log\det(\mathbf{I} + H_E K_X H_E^*).$

[14] Westman, S., [*Information Retrieval: Searching in the 21st Century*], ch. Image Users' Needs and Searching Behaviour, 63–83, Wiley, West Sussex, UK (2009).

[15] Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R., "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**, 1349 –1380 (Dec 2000).

[16] Suzuki, M., Tamari, F., Fukuda, R., Uchida, S., and Kanahori, T., "Infty: An integrated OCR system for mathematical documents," in [*Proc. Document Engineering*], 95–104, ACM, Grenoble, France (2003).