BOOKLET

13th IAPR International Workshop on Graphics Recognition

GREC 2019

20-21 September 2019
Sydney, Australia

Welcome from Organizers

We are glad to welcome you to the 13th edition of the International Workshop on Graphic Recognition (GREC 2019). GREC 2019 builds on the success of the twelve previous editions in Penn State University (USA, 1995), Nancy (France, 1997), Jaipur (India, 1999), Kingston (Canada, 2001), Barcelona (Spain, 2003), Hong Kong (China, 2005), Curitiba (Brazil, 2007), La Rochelle (France, 2009), Seoul (Korea, 2011), Lehigh (USA, 2013), Nancy (France, 2015) and Kyoto (Japan, 2017).

The GREC workshops provide an excellent opportunity for researchers and practitioners at all levels of experience to meet colleagues and to share new ideas and knowledge about graphics recognition methods. The workshops enjoy strong participation from researchers in both industry and academia.

Graphics Recognition is a subfield of document image analysis that deals with graphical entities in engineering drawings, comics, musical scores, sketches, maps, architectural plans, mathematical notation, tables, diagrams, etc.

The aim of this workshop is to maintain a very high level of interaction and creative discussions between participants, maintaining a « workshop » spirit, and not being tempted by a « mini-conference » model.

The workshop comprise several sessions dedicated to specific topics related to graphics in document analysis and graphic recognition. For each session, there will be an invited presentation describing the state of the art and stating the open questions for the session's topic, followed by a number of short presentations that will contribute by proposing solutions to some of the questions or presenting results of the speaker's work. Each session will be concluded by a panel discussion.

For this 13th edition of GREC, the authors had the opportunity to submit short or long paper depending of the maturity of their research. We finally selected 23 papers from 8 different countries, 17 long papers and 6 short papers. Each submission was reviewed by two expert reviewers. We would like to take this opportunity to thank the program committee members and sub-reviewers for their meticulous reviewing efforts.

For this edition, we aimed to highlight three topics related to graphic recognition in order to organize special sessions on : Music Scores analysis and recognition, Comics Analysis and Understanding, Sketch Recognition and Understanding. However, only one paper on sketch analysis and one paper on music score have been submitted for this edition. So the program will only include a special session dedicated to Comics Analysis and Understanding.

Finally, we have included in this edition of GREC, a round-table discussion with industrial and academic participants in order to discuss about the future of Graphic Recognition.

We hope you will enjoy the workshop and we are looking forward to welcome you to the 13th edition of the International Workshop on Graphic Recognition in September in Sydney.

The GREC 2019 General Chair
Jean-Christophe BURIE

Organizing Committee

Program Chairs

Gunther Drevin – North-West University, South-Africa
Partha Pratim Roy – Indian Institute of Technology, Roorkee, India
K.C. Santosh – University of South Dakota, USA

Program Committee

Eric Anquetil (INSA Rennes, France)
Jorge Calvo Zaragoza (University of Alicante, Spain)
Mickaël Coustaty (Université de la Rochelle, France)
Tiny Du Toit (North-West University, South Africa)
Alicia Fornes (Universitat Autònoma de Barcelona, Spain)
Nina Hirata (University of Sao Paulo, Brazil)
Pierre Héroux (Université de Rouen, France)
Bart Lamiroy (Université de Lorraine, France)
Christoph Langenhan (Technical University of Munich, Germany)
Josep Lladós (Universitat Autònoma de Barcelona, Spain)
Umapada Pal (Indian Statistical Institute, Kolkata, India)
Partha Pratim Roy (Indian Institute of Technology, India)
Oriol Ramos-Terrades (Universitat Autònoma de Barcelona, Spain)
Romain Raveaux (Université de Tours, France)
Christophe Rigaud (Université de la Rochelle, France)
Richard Zanibbi (Rochester Institute of Technology, USA)

Table of Contents

This booklet includes the short papers presented during the GREC Workshop. The long papers will be available in the ICDAR proceedings

Multi-class semantic segmentation of comics: A U-Net based approach

Jochen Laubrock[1] and David Dubray[2]
Department of Psychology, University of Potsdam, Potsdam, Germany
Email: [1]laubrock@uni-potsdam.de, [2]ddubray@uni-potsdam.de

***Abstract*— Dubray and Laubrock [3] describe a U-Net based fully convolutional neural network model for speech balloon segmentation. Here we extend their model to multiple classes, including panels and captions. Multi-class semantic segmentation is achieved by using k ouptut channels for k classes, and arranging the ground truth maps accordingly. First results are very promising: the different classes can be separated rather well, including difficult cases such as elements delineated by illusory contours. We achieve state-of-the-art performance on the Graphic Narrative Corpus (GNC) [4] validation set, with F1-scores of 0.95 for speech balloons, 0.92 for captions, and 0.99 for panels. Error inspection suggests that cases that are problematic for the network are often cases that are also difficult for humans.**

***Index Terms*—Convolutional neural networks; Deconvolution; Image segmentation; Document Analysis; Comics**

I. Introduction

A recent review of comics analysis concludes that "segmenting the panels or reading the text of any comics is still challenging because of the complexity of some layouts and the diversity of the content" [2]. Several methods have been proposed for segmentation of individual elemens. Recently, Dubray and Laubrock [3] achieved state-of-the art performance in speech-balloon segmentation using a fully convolutional variant of the neural network architecture U-Net [10]. Here we extend their approach to multiple classes. We modified the U-Net to enable multi-class joint semantic segmentation of comics books pages into panels, captions, and speech balloons.

A. Related work

Deep convolutional neural networks (DCNNs) using learned feature hierarchies outperform classical engineered features in a wide variety of visual recognition and localization tasks. DCNNs features shaped by training on large photographic datasets can be re-used to describe comics pages.

For example, Laubrock and Dubray [6] successfully categorized illustrator style using DCNNs. Comics readers' attention/gaze and DCNN predictions of empirical saliency focus on similar regions of comics pages [7]. Nguyen et al. [9] describe a multi-task learning (MTL) model for comic book image analysis derived from Mask R-CNN that can segment several classes such as panels, balloons, and characters, and that also predicts balloon-character associations. We will use this MTL model as a baseline for our performance evaluation.

B. The present approach

Here we use a deep neural network to learn about the shape and appearance of panels, speech balloons, and captions, given annotated pages from the GNC. We regard this as a semantic segmentation problem, i.e., a pixel-wise classification task. In a joint architecture, we use the same encoder-decoder pathways to generate three prediction maps simultaneously, one for each target class. Thus the main difference to the approach described in [3] is in the number of output maps. Two further differences are that we use (a) more training material, and (b) more variants of data augmentation during training.

II. Methods

To save space, we will only describe the differences to [3].

A. Dataset

The training material consisted of 3,430 annotated pages, downscaled to 768×512 pixels in RGB, from more than 200 comic books of the GNC. Binary mask images were generated from ground truth annotations of panels, balloons, and captions.

B. Model architecture

We used a fully convolutional approach based on the U-Net architecture for predicting pixel-based image segmentations. The network can be divided into an encoding and a decoding branch, devoted to describing the image using a pre-trained VGG-16 network and semantically-guided re-mapping back to input space, respectively.

To achieve joint segmentation of images of width w and height h into k classes, we had the model output a tensor of $h \times w \times k$, and computed pixel-wise loss functions with respect to accordingly arranged groud truth masks. A sigmoid activation function as the last step of decoding resulted in a $768 \times 512 \times 3$ representation of floating values in (0,1), which can be considered to give the probability of each pixel to belong to a given class, independently. Each of the three layers codes for one class.

C. Loss function

Binary cross entropy loss and Dice loss were each computed over each pixel i in the set of $h \times w \times k$ pixels of binary image masks y and model predictions $\hat{y}$. As loss we defined their unweighted sum.

D. Training and augmentation

The dataset was randomly split into training and validation sets. We applied image augmentation to the training set, randomly varying hue, brightness, saturation, contrast, shifting height and widths, and flipping the image horizontally and vertically within reasonable boundaries.

TABLE I
SEGMENTATION PERFORMANCE IN PERCENT.

Method	Speech balloons			Captions			Panels		
	Recall	Precision	F_1	Recall	Precision	F_1	Recall	Precision	F_1
Nguyen et al. [9], Comic MTL, ebdthèque	74.9	92.8	82.9				77.0	73.2	75.0
Our method, Graphic Narrative Corpis	95.6	96.5	95.0	92.8	94.5	91.8	99.0	98.8	98.6

III. RESULTS

Table I lists the values for recall, precision, and F_1 score for instances of each of our three output classes. These measures were computed pixelwise. To put our results into perspective, we compare the values with results from the MTL model reported in [9]. To be fair, we note that different data sets were used in the evaluation, and the ebdthèque data set used by Nguyen et al. [9] is rather heterogenous. Thus it is expected that performance will be somewhat weaker for the ebdthèque. An evaluation of our model with other data sets is on the way. Also note that the ebdthèque and the MTL model do not currently differentiate between speech balloons and captions.

Our model achieves state-of-the art performance. Even though a direct comparison is difficult, it is clear that our model performs at least on par with previous approaches.

Figure 1 shows some qualitative results. The examples from the test set illustrate that the model can well capture different shapes of speech balloons and panels, and can also guess the shape of a caption that does not have a physical boundary, but just an imaginary continuation of the gutter. Inspection of errors on other pages suggests that the model usually does well, but is sometimes tempted to classify scene text as caption, and sometimes confuses caption and balloon.

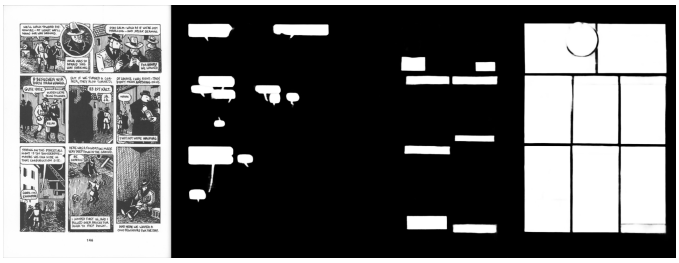


Fig. 1. Example from Maus [11] illustrating typical performance. From left to right: original, model predictions for ballons, captions, panels, respectively. Note that captions without borders can be delineated quite well (e.g., lower right panel).

IV. DISCUSSION

We obtained very good segmentation results for several classes of objects in comics by training a fully-convolutional variant of the U-Net. Segmentation of panels and speech balloons was excellent, and caption segmentation was still very good, especially when considering that the definition of what constitutes a caption varies somewhat between annotators.

Even elements that were partly defined by illusory contours were successfully segmented, and the model is flexible enough to detect the shape and direction of speech balloons' tails. Some of the remaining misclassifications appear to be relatively easily addressable by some heuristic post-processing.

A potential limitation is related to the training material. We anticipate similar problems as Dubray and Laubrock [3] for balloon and caption segmentation with material containing non-Latin script. These should be resolvable using training on the Manga109 data set [8].

REFERENCES

[1] *2nd International Workshop on coMics Analysis, Processing, and Understanding, 14th ICDAR 2017, Kyoto, Japan*, 2017. IEEE.

[2] O. Augereau, M. Iwata, and K. Kise (2017). An overview of comics research in computer science. In [1], 54–59.

[3] D. Dubray and J. Laubrock (2019). Deep CNN-based Speech Balloon Detection and Segmentation for Comic Books. *CoRR*, abs/1902.08137. 2019.

[4] A. Dunst, R. Hartel, and J. Laubrock (2017). The Graphic Narrative Corpus (GNC): Design, annotation, and analysis for the digital humanities. In [1], 15–20.

[5] I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, eds. (2019). *MultiMedia Modeling, MMM 2019 Proceedings, Part II*. Springer.

[6] J. Laubrock and D. Dubray (2019). CNN-based classification of illustrator style in graphic novels: Which features contribute most? In Kompatsiaris et al. [5], 684–695.

[7] J. Laubrock, S. Hohenstein, and M. Kümmerer (2018). Attention to comics: Cognitive processing during the reading of graphic literature. In A. Dunst, J. Laubrock, and J. Wildfeuer, editors, *Empirical Comics Research: Digital, Multimodal, and Cognitive Methods*, 239–263, New York, Routledge. ISBN 9781138737440.

[8] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa (2017). Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838.

[9] N.-V. Nguyen, C. Rigaud, and J.-C. Burie (2019). Multi-task model for comic book image analysis. In Kompatsiaris et al. [5], 637–649.

[10] O. Ronneberger, P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[11] A. Spiegelman (1997). *Maus: a survivor's tale*. Pantheon Books, New York, 1st ed edition. ISBN 0679406417.

Confidence criterion for speech balloon segmentation

Christophe Rigaud, Van Nguyen and Jean-Christophe Burie
Laboratoire L3i,
University of La Rochelle
17042 La Rochelle CEDEX 1, France
{*christophe.rigaud, nhu-van.nguyen, jean-christophe.burie*}*@univ-lr.fr*

***Abstract*—This short paper investigates how to improve the confidence of speech balloon segmentation algorithms from comic book images. It comes from the need of precise indications about the quality of automatic processing in order to accept or not each segmented regions as a valid result, according to the application and without requiring any ground truth. We discuss several applications like result quality assessment for companies and automatic ground truth creation from high confidence results to train machine learning based systems.We present some ideas to combine several domain knowledge information (e.g. shape, text, etc.) and produce an improved confidence criterion.**

I. Introduction

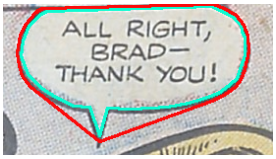
Digital comic content is produced to facilitate transport, reduce publishing cost and allow reading on screens from television to smartphones like newspapers and other documents. To get a user-friendly experience of digital comics on all mediums, it is necessary to extract, identify and adapt comic book content originally designed for paper printing [1]. Comic book image content is composed of different textual and graphical elements such as panel, balloon, text, comic character and background.

The speech balloon is a major link between graphic and textual elements. They can have various shapes (e.g., oval, rectangular) and contours styles (e.g., smooth, wavy, spiky, partial, absent). Speech balloons entirely surrounded by a black line (closed) have attracted most of the researches so far. They were initially based on region detection, segmentation and filtering rules [2], [3] and evaluated on small privates datasets. Liu *et al.* [4] proposed a clump splitting based localization method which can detect both closed and open speech balloons. They performed the evaluation on the eBDtheque [5] dataset using recall and precision metrics.

In addition, Liu *et al.* [6] and Rigaud *et al.* [7] proposed approaches making use of text contained in speech balloons for segmenting speech balloon contours at pixel level. They both used the same evaluation metric (recall and precision) and dataset (eBDtheque) with a little difference that Rigaud *et al.* computed a confidence value within their contour candidate filtering operation.

All these approaches have been tested on datasets which give indication about how well they perform and make comparison possible with new researches. However, they can not give an indicator on a single detection out of the tested dataset, except [7]. Private companies may not be satisfied by such evaluation because published datasets may not exactly reflect their private dataset characteristics. Moreover, a confidence value associated to each detected element would be helpful for deciding if it fits their requirements or requires extra processing.

Table I
EXAMPLES OF CONFIDENCE CRITERION FOR CORRECT AND WRONG BALLOON SEGMENTATION.

	
ALL RIGHT, BRAD WHANK you!	(i - . . ,
$cShape$ = 0.92% $cLex$ = 0.95%	$cShape$ = 0.22% $cLex$ = 0.00%

In this paper we propose an improvement of an existing confidence criterion for speech balloon segmentation quality.

II. Proposed method

We define the speech balloon segmentation confidence criterion as a score between zero and one encoding the confidence of each segmented region similarly to the confidence value introduced in [7]. In [7], the confidence value is based on two weighted variables such as the contained text (all connected components) alignment $cAlign$ and segmented region shape $cShape$. We propose to strengthen the first parameter initially based on alignment features (distance and position analysis of neighbouring connected components) by replacing it by the confidence value computed from at least one Optical Character Recognition (OCR) system. OCR systems take into account a lot of features to recognise characters, words and text lines from images (e.g. alignment, shape, size, spaces, contrast) which increases their results reliability. Such systems usually provide a confidence value associated to each results (classification likelihood) but in-

stead of using it , we preferred to use a readily observable quantity that correlates well with true accuracy of the recognized text as describe in [8]. This metric was originally proposed to compare OCR system output performances but in our case, we use it as an OCR independent indicator about the OCR output quality. We compute, for each OCR token, the minimum edit distance (Levenshtein distance) to its most probable lexical equivalent from a lexicon of the corresponding language (e.g. Grammalecte, WordNet). The sum of these distances d over all tokens is, therefore, a statistical measure for the OCR output, and the lexicality defined as *cLex = (1 - mean Levenshtein distance per character ratio)* is a measure for accuracy.

The second term $cShape$, as described in the original publication [7], encodes the overall convexity of the balloon outline in order to find how similar to a perfect bubble (or rectangle) the balloon candidate is. It is defined as the ratio between the Euclidean perimeter of the convex hull of the measured shape S and the Euclidean perimeter of the measured shape S as follows:

$$cShape = \frac{arcLength(hull(S))}{arcLength(S)} \tag{1}$$

The original Equation (2) from [7] becomes as follows when we replace $cAlign$ by $cLex$:

$$C = \alpha \times cLex + \beta \times cShape \tag{2}$$

The best weighting parameter values were validated as $\alpha = 0.75$ and $\beta = 0.25$ in [7] ($\alpha + \beta = 1$). However, because $cLex$ is based on really different features compared to the original $cAlign$ parameter, they both need to be re-validated according to the desired application. The main advantage of replacing the alignment-based measure by a OCR-based measure is that it is much more reliable to detect the presence of text with a high confidence inside segmented regions, thanks to the growing progress of OCR systems [9]. However, if the OCR system is not able to recognize a part of text because it is written with an "unseen" typewritten font or handwritten style, it will result in a poor confidence score even if the segmentation region is a true positive. Also, words may be recognized as others, or with minor errors and still get a good confidence score in some cases.

An example of the proposed confidence criterion is given Table I. In this table, segmented contours are represented in cyan and convex hulls in red in the first row. Wrong transcriptions are highlighted in red in OCR output in the second row. Corresponding confidences are given in the last row.

III. Conclusion

This short paper investigates how to compute a confidence criterion that can indicate speech balloon segmentation quality without requiring any ground truth. It relies on comics domain knowledge i.e. bubble-like shape and text content. It may be suitable for continuous quality control in large digitization and indexation processes or automatic ground truth generation for machine learning techniques.

In the future, we would like to investigate other features that can improve further the quality of such confidence criterion. The combination with external information like the position in the panel and the overlap with other elements could be other sources of information to aggregate.

Acknowledgment

This work is supported by Research National Agency (ANR) in the framework of 2017 LabCom program (ANR 17-LCV2-0006-01), CPER NUMERIC program funded by the Region Nouvelle Aquitaine, CDA, Charente-Maritime French Department, La Rochelle conurbation authority (CDA) and the European Union through the FEDER funding.

References

[1] O. Augereau, M. Iwata, and K. Kise, "A survey of comics research in computer science," *CoRR*, vol. abs/1804.05490, 2018. [Online]. Available: http://arxiv.org/abs/1804.05490

[2] K. Arai and H. Tolle, "Method for real time text extraction of digital manga comic," *International Journal of Image Processing (IJIP)*, vol. 4, no. 6, pp. 669–676, 2011.

[3] A. K. N. Ho, J.-C. Burie, and J.-M. Ogier, "Panel and Speech Balloon Extraction from Comic Books," *2012 10th IAPR International Workshop on Document Analysis Systems*, pp. 424–428, mar 2012.

[4] X. Liu, Y. Wang, and Z. Tang, "A clump splitting based method to localize speech balloons in comics," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, pp. 901–905.

[5] C. Guérin, C. Rigaud, A. Mercier, F. Ammar-Boudjelal, K. Bertet, A. Bouju, J. C. Burie, G. Louis, J. M. Ogier, and A. Revel, "eBDtheque: A representative database of comics," in *2013 12th International Conference on Document Analysis and Recognition*, Aug 2013, pp. 1145–1149.

[6] X. Liu, C. Li, H. Zhu, T.-T. Wong, and X. Xu, "Text-aware balloon extraction from manga," *The Visual Computer*, vol. 32, no. 4, pp. 501–511, Apr 2016. [Online]. Available: https://doi.org/10.1007/s00371-015-1084-0

[7] C. Rigaud, J.-C. Burie, and J.-M. Ogier, "Text-independent speech balloon segmentation for comics and manga," in *Graphic Recognition. Current Trends and Challenges*, B. Lamiroy and R. Dueire Lins, Eds. Cham: Springer International Publishing, 2017, pp. 133–147.

[8] C. Rigaud, J. Burie, and J. Ogier, "Segmentation-free speech text recognition for comic books," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 03, Nov 2017, pp. 29–34.

[9] T. M. Breuel, "High performance text recognition using a hybrid convolutional-lstm implementation," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 11–16.

Tree-Based Structure Recognition Evaluation for Math Expressions: Techniques and Case Study

Mahshad Mahdavi
Document and Pattern Recognition Lab
Rochester Institute of Technology
Rochester, NY, USA
mxm7832@rit.edu

Richard Zanibbi
Document and Pattern Recognition Lab
Rochester Institute of Technology
Rochester, NY, USA
rxzvcs@rit.edu

I. Introduction

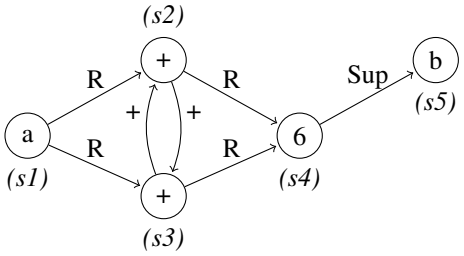
Evaluating visual structure recognition for math expressions is complex due to the interactions between input primitives, detected symbols, and their spatial relationships. Visual structure is expressed using trees describing the arrangement of symbols on writing lines, and the hierarchical spatial arrangement of these writing lines (see Figure 1(b)). LaTeX formulas represent this information along with additional formatting directives (e.g., for fonts, symbol sizes, and spacing).

Formula recognition comprises three major tasks: detecting symbols, classifying symbols, and determining spatial relationships between symbols. With the correct tools, symbols and relationships can be evaluated separately, and specific errors compiled using confusion matrices and *confusion histograms* that tabulate and count specific errors for given sub-trees in ground truth formulas [1]. As a simple illustration, if the expression '$xy+1$' is recognized as '$2a+b$,' the output can be considered as having the correct spatial relationships/structure (i.e., five symbols on one writing line), but only one symbol is shared between the two formulas ('+').

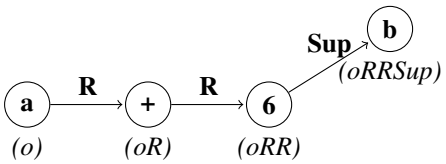
A number of state-of-the-art systems were inspired by automated image captioning, avoiding explicit symbol segmentation and producing LaTeX output [2], [3]. By representing structure only over detected symbol classes, we lose the correspondence of input primitives (e.g., handwritten strokes in Figure 1(a)) to symbols in the output. So far, LaTeX outputs have been evaluated using string-based metrics such as exact matching, string edit distances, or n-gram-based metrics such as BLEU, or by computing distances between images produced after rendering TeX formulas. Unfortunately, different LaTeX strings can represent identical formulas, or produce differences in spacing or formatting for the same underlying expression. These measures approximate rather than directly capture differences in visual structure at the level of symbols and relationships seen in Figure 1(b).

II. Visual Structure Representation (symLG) and Symbol-Level Evaluation Metrics

To allow us to evaluate and compare visual structure recognition using symbols and relationships directly, we convert LaTeX and other structure representations to Symbol Label Graphs (*symLGs*, see Figure 1(b)). In a symLG, each symbol has an identifier. Identifiers are defined by the sequence of spatial relationships from the root symbol to each symbol in the tree. The adjacency matrix for a symLG contains symbol class labels on the diagonal (for symbol self-edges), and spatial relationships between symbols in off-diagonal entries. Using this, we can then directly compare formulas based on the agreement between adjacency matrix entries, even when symbol identifiers are missing in one of the graphs [1].



(a) Stroke Label Graph (LG)



(b) Symbol Label Graph (symLG)

Fig. 1: Different representations for formula '$a+6^b$' written using five strokes. Node identifiers are shown in brackets.

To obtain symLGs, we first need to produce a uniform symbol-level structure representation, for which we have used Presentation MathML. We convert TeX formulas to MathML using `pandoc`.[1] This transformation preserves symbols and spatial relationships while removing formatting directives (e.g., `\quad`, fonts). For primitive-based output representations, such as the stroke label graph in Figure 1(a) [4], we use a simple transducer to produce MathML. In a label graph, all strokes in a symbol share the same spatial relationship with strokes in the related symbol (e.g., for 'a' and '+'), and symbol segmentation is given using bidirectional edges labeled with

[1] https://pandoc.org

TABLE I: symLG im2latex-100k results (9,378 test formulas). Shown are correct symbol/relationship locations (Detection), symbol/relationship classes (Det.+Class), formula SLT structure ignoring symbol labels, and valid structure and symbol labels (Str.+Class).

	Symbols		**Relationships**		**Formulas**	
	Det.	Det.+Class	Det.	Det.+Class	Str.	Str.+Class
IM2TEX	95.70	93.48	95.50	95.50	86.79	**83.15**

their associated symbol's class (e.g., for '+').

Once we have a MathML representation for a formula, we generate symbol identifiers using the spatial relationship sequence from the root symbol (see Figure 1(b)). Identifiers allow us to address symbols on writing lines from different structure representations. This produces a *symbolic* representation for recognition outputs, one that ignores the correspondence of output symbols to input data [1].

Symbol-Level Metrics. Once we have our symLG representation, we compute symbol-level metrics using evaluation tools from the CROHME handwritten math recognition competitions [1], [4], [5] originally designed for stroke-level evaluation (the LgEval library). LgEval metrics include formula and symbol recognition rates, along with recall and precision for detection and detection + classification of both symbols and relationships [1]. The symLG representation allows us to identify specific relationship classification errors, structure errors, and symbol classification errors (*when symbol locations/identifiers are correct*; see Section IV).

Related Work. Symbolic evaluation has been considered previously, e.g., EMERS [6] is a tree edit distance using an Euler string representation to quantify partially correct recognition for MathML trees. Symbol errors are weighted inversely proportional to their distance from the main writing line (baseline) of the expression, to decrease the impact of errors inside branches. A form of symbolic evaluation based on unlabeled trees was used in early CROHME competitions [1]. The IMEGE metric [7] is a pixel-based image distance metric, which has been used for evaluation by rendering an image from output LaTeXstrings [2].

III. Case Study

We use symLGs to evaluate the IM2TEX system by Deng et al. [2]. As shown in Table 1, our symLG metrics provide measures for correct symbol detection (i.e., symbols exist at expected spatial locations), correct symbol locations and labels, correct relationships, and structure and symbol classification accuracy at the expression level. Note that because spatial relationships determine symbol locations, a correctly *detected* relationship is also correctly classified.

For the im2latex-100k data set, we were able to convert 9,378 of the 10,355 test formulas (90.6%) from LaTeX to MathML using `pandoc`. Many failed conversions are caused by invalid syntax (e.g., missing brackets).

For the 9,378 formulas that were converted successfully to MathML, We are now able to report that the percentage of correct formulas with both correct symbols and structure is 83.15%, that 93.48% of symbols are in the proper location with their correct class, and that 95.50% of spatial relationships are correct. The metrics previously reported by the IM2TEX authors include BLEU (tok) at 58.41, BLEU (norm) at 87.73, exact image-based pixel matching of 77.46, and image-based pixel matching with a whitespace tolerance (-ws) of 79.88 [2].

Moreover, using symLGs we can provide detailed error analysis that string and image-based representations cannot capture (omitted for space). The most common error is 'missing' symbols. This happens because symbols are identified by their absolute path - therefore, errors in structure lead to errors in symbol detection and classification. Note that this also means that correctly detected symbols at the incorrect position in a symLG are identified as invalid.

IV. Conclusion

We have presented a technique that allows string and tree-based formula structure representations to be meaningfully compared at the level of recognized symbols and relationships. Further, this permits fine-grained evaluation of recognition results at the individual symbol and relationship level, as well as at the expression level, addressing limitations with the previous use of string-based and image-based metrics used to evaluate LaTeX output. In future work, we hope to use more robust methods for converting from LaTeX to MathML.

Our symLG-based metrics were used for the recent ICDAR 2019 CROHME + TFD competition [8], as they are simple to understand, and provide useful global performance metrics and automated error analyses.

References

[1] H. Mouchère, R. Zanibbi, U. Garain, and C. Viard-Gaudin, "Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014," *Int'l. J. Document Analysis and Recognition*, vol. 19, no. 2, pp. 173–189, 2016.

[2] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," *arXiv preprint arXiv:1609.04938*, 2016.

[3] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[4] R. Zanibbi, A. Pillay, H. Mouchere, C. Viard-Gaudin, and D. Blostein, "Stroke-based performance metrics for handwritten mathematical expressions," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 334–338.

[5] R. Zanibbi, H. Mouchere, and C. Viard-Gaudin, "Evaluating structural pattern recognition for handwritten math via primitive label graphs," in *Document Recognition and Retrieval XX*, vol. 8658. International Society for Optics and Photonics, 2013, p. 865817.

[6] K. Sain, A. Dasgupta, and U. Garain, "Emers: a tree matching–based performance evaluation of mathematical expression recognition systems," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 1, pp. 75–85, 2011.

[7] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "An image-based measure for evaluation of mathematical expression recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2013, pp. 682–690.

[8] M. Mahdavi, R. Zanibbi, H. Mouchère, and U. Garain, "ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection," in *Proc. ICDAR 2019*, to appear.

Graph matching over Hypothesis Graphs for the Analysis of Handwritten Arithmetic Operations

Arnaud Lods
Learn&Go, Univ Rennes, CNRS, IRISA
F-35000 Rennes
arnaud.lods@learn-and-go.com

Éric Anquetil
Univ Rennes, CNRS, IRISA
F-35000 Rennes
eric.anquetil@irisa.fr

Sébastien Macé
Learn&Go
F-35000 Rennes
sebastien.mace@learn-and-go.com

***Abstract*—This paper presents a preliminary research work for the analysis of handwritten arithmetic operations in the context of e-education. Given a mathematical exercise, an answer in the form of an arithmetical operation is expected from a child. This answer can be represented by a graph containing both expected numbers and their corresponding relationship one to another. We propose to compute several hypotheses Fuzzy Visibility Graph of symbols over a child's input. The widely used A* algorithm to compute exact graph edit distance is applied over those graphs to match the expected answer. To reduce the search complexity, simplified graphs of operands are first generated and used. Each operand is a sub-graph of the original graph, the algorithm is then applied on the pairs of matched sub-graphs. The hypothesis graph with the smallest graph edit distance with the expected graph and the matched differences can be used to produce an adapted feedback. The result of an experiment over a given example is presented.**

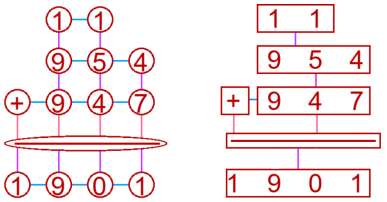
***Index Terms*—graph edit distance, A* algorithm, handwritten arithmetic operation, fuzzy visibility graph, graph analysis.**

I. Introduction

The improvement of pen-based devices over the recent years offers new ways of teaching in school. We now have the opportunity to provide interfaces with complete liberty for the children. Thus one can devise an adapted system to create personalized and extensive feedback on mistakes made without a costly human analysis. For our domain of application, learning mathematics, such problem can be represented by a graph $G_T = \{V_T, E_T\}$ where V_T, the set of vertices, represents the set of mathematical symbols and E_T, the set of edges, represents the mathematical relationships between symbols, as displayed in Fig. 1a.

The scientific problematic boils down to a problem of graph matching. Given a source graph G_S produced from the child's input and a target graph G_T which is the expected solution, we are looking for the best graph edit distance (GED) that minimizes the number of required operations to transform G_S into G_T. The community of pattern recognition has since long tackled the problematic of graph matching. A study in [1] covers most of the related works on the matter, from exact graph matching to inexact graph matching. Recent works focus on the improvement of the computation of a higher bound for classification purpose. In [2] the authors present several optimization algorithms. However, to compute the best GED, the A* algorithm [3] is a robust but costly solution. In [4] an algorithm is presented to compute the exact GED in half the time and with a lower memory consumption.

With the support from the LabCom **ScriptAndLabs** funded by the **ANR** ANR-16-LVC2-0008-01



(a) Target graph of symbols G_T (b) Target graph of operands G_TO

Fig. 1: The graph G_T generated with rules over the arithmetic operation: 954 + 947.

As a first milestone for our research work, we present in Section II the application of a naive A* algorithm on an intermediate then complete representation of the graph to reduce overall computational complexity. Then we present a first experiment in Section III. In Section IV we discuss our ongoing research work on this problematic.

II. Our system

To produce a graph that can be matched to the target graph presented earlier, we extend a system previously built in [5] called Fuzzy Visibility Graph. Fuzzy landscapes, corresponding to fuzzy areas in the space each representing an observed mathematical relationship, are learned over pairs of symbols and are used to build the graph from a set of symbols. This set of symbols is produced by two different Random Forest classifiers to segment the strokes into symbols and to classify those symbols with two set of geometrical features. The classifiers give us a probability for each class. As children are still in the learning process and can have very different and ambiguous input, instead of generating one graph G_S to match to the target graph G_T, we generate a set of hypotheses graphs $\mathcal{G} = \{G_{S1}, G_{S2}..G_{Sn}\}$ for each uncertain classification. As such the goal is to match each graph G_{Si} to the target graph G_T to find the graph with the lowest GED. Fig. 2 display an example of two hypotheses graphs made on the same input where the segmentation was ambiguous.

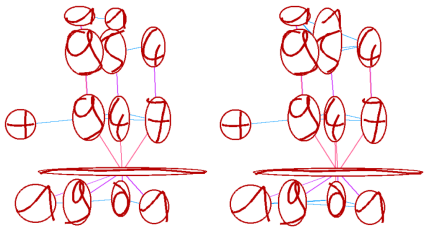


Fig. 2: The two Fuzzy Visibility Graphs of symbols hypotheses G_{S1} and G_{S2}.

As stated earlier, a simplified A* algorithm is used to match each graph G_{Si} to G_T. The costs of transformations (Eq. 1) are defined to put more emphasis on *how* the child has built his operation with a higher cost for edges deletion/insertion rather than *what* he has written with a lower cost for different vertices matching. For vertices substitution, $||l_u - l_v||$ currently refers to the Levenshtein distance on the string of symbols labels. For edges substitution, $f(r_{(u,v)} - r_{(u',v')})$ has a cost of 0 if the relationship is similar or a cost of 1 otherwise. We also impose a restriction on the deletion and insertion of vertices. Let N_T and N_S be respectively the number of vertices for the target and source graph considered. We only allow the insertion of vertices if $N_T > N_S$ and the deletion of vertices if $N_T < N_S$. In other words, as we are working with operations that are expected to contain lots of mistakes from both positioning and calculus, the goal is to force a matching, even if symbols have different labels, rather than deleting and adding new vertices with correct labels if the child made mistakes on numbers. Thus, if we observe too mane or too few vertices, we allow a number of insertion/deletion operations as to match exactly those which are missing/in excess. Once the best GED for the pairs of graphs of operands is computed, we can compute the GED on the graph of symbols for each hypothesis.

$$\begin{cases} c(u \to v) = ||l_u - l_v|| \\ c(u \to \epsilon) = c(\epsilon \to v) = \tau \\ c((u,v) \to (u',v')) = f(r_{(u,v)}, r_{(u',v')}) \\ c((u,v) \to \epsilon) = 0 \\ c(\epsilon \to (u',v')) = 2\tau \end{cases} \quad (1)$$

We now have the matched pairs of operands between each graph G_{SOi} and G_{TO}, each operand being a sub-graph of the original graph of symbols. The algorithm A* is applied to find the best GED for each pair of matched sub-graphs. We identify two types of edges: internal edges between symbols of the sub-graphs, that are matched as usual, and external edges between symbols and operands. For the latter case, the cost of edges edition is computed from a symbol vertex to an operand vertex: $\{(u, V)\}$. This allows us to avoid matching larger sub-graphs of related symbols which would in turn increase a lot the number of matches. The lowest sum of the GED and its corresponding edit path on the set of sub-graphs represent the best matching for each graph G_{Si} to G_T. The hypothesis graph with the lowest cost is kept and the resulting GED represents the analysis result with the differences on what was expected.

III. Experiment

We ran the system on the given arithmetic addition displayed in Fig. 2. Two hypotheses are generated while producing the fuzzy visibility graph (Fig. 2). These hypotheses (G_{S1}, G_{S2}) are converted to two operands graphs (G_{SO1}, G_{SO2}) using the same rules to produce G_{TO}. The two matching steps presented in Section II are applied and the expected hypothesis graph has the lowest GED with a missing relationship between the + and the horizontal bar. Otherwise if we were to select the second graph as the best hypothesis, then the edit path: Insert node, Insert relation, Insert relation on the carry-over in G_T give us the information that the carry-over was forgotten, and thus an adapted feedback can be displayed.

IV. Conclusion and perspectives

We presented the first step of a system that make use of fuzzy visibility graph representation to compute the graph edit distance from hypotheses graphs to a given target graph. The standard A* algorithm is used, and an intermediary representation is used to match sub-graph of the initial pair of graphs to reduce the computation time. Simple cost functions for the graph edition are used. On a first sample, the correct hypothesis graph is matched at a lower cost and the expected matching between symbols is found in a reasonable time.

In the future we intend to learn the cost functions on a complete dataset to take into account the probabilities of the classifiers producing the initial hypotheses graphs. Then we will be able to produce experimental validation on a corresponding dataset. Another step is to implement a more efficient algorithm to reduce computation for complex operation, and to merge the different hypotheses graphs in a single graph with vertices and hypotheses edges to avoid matching the same pairs several times. Other improvement can be made on the initial graph representation. We also intend to expand experiments on a large dataset collected from children aged 6 to 8-years-old to evaluate the effectiveness of the matching with several methods and different type of arithmetical operations. Eventually transformations resulting from the graph edit distance will be used to produce adapted feedback for the children.

Acknowledgment

With the support from the LabCom **ScriptAndLabs** founded by the **ANR** ANR-16-LVC2-0008-01

References

[1] M. Vento, "A long trip in the charming world of graphs for pattern recognition," *Pattern Recognition*, vol. 48, no. 2, pp. 291–301, 2015.

[2] K. Riesen, "Structural pattern recognition with graph edit distance," *Advances in computer vision and pattern recognition, Cham*, 2015.

[3] K. Riesen, S. Fankhauser, and H. Bunke, "Speeding up graph edit distance computation with a bipartite heuristic.," in *MLG*, pp. 21–24, 2007.

[4] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau, "An exact graph edit distance algorithm for solving pattern recognition problems," in *4th International Conference on Pattern Recognition Applications and Methods 2015*, 2015.

[5] A. Lods, E. Anquetil, and S. Macé, "Fuzzy visibility graph for structural analysis of online handwritten mathematical expressions," in *2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019.

Recognition, encoding, and transcription of early mensural handwritten music

José M. Iñesta, David Rizo, and Jorge Calvo Zaragoza
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Alicante (Spain)
Email: inesta@dlsi.ua.es

I. Introduction

Optical music recognition (OMR) systems are key tools for publication of music score collections that are currently found only on paper. The main goal of an OMR system is to convert music sheet images to a digital structured format, like XML-based ones. Under the perspective of cultural heritage preservation, many of these collections are handwritten material, and some of them can be found in early notation systems.

In particular, Spanish white mensural notation was the dominant code for writing music in Spain during the 16th to 18th centuries, producing large collections of documents yet to be made accessible to the public. In all these cases, just the scanned or photographed images of the scores are available. The OMR techniques available today are far to be ready for being applied to that kind of handwritten documents.

The system under development in the Hispamus project (named MuRET) is intended to produce both a transcribed copy of the original (*diplomatic* version) and the material for elaborating a *critical* edition, where possible mistakes and stains, blots or even holes that may affect the original material in the original work may be corrected and solved. This is not a trivial task.

Typical OMR systems such as Audiveris[1] or Photoscore[2] are mainly devised to extract the musical content from printed or manuscript score sheets in order to edit them further in music edition applications. Aruspix [4] goes a step beyond, and allows the superimposition and the collation of early music prints. However, none of them is devised for recognizing the contents of printed or manuscript scores from different approaches, allowing the manual introduction of new materials, the encoding into all current standards, and the assisted transcription into a edited version ready for preparing a critical edition.

II. Methodology

One important feature of MuRET is the possibility to render also a translated version to a modern notation score, readable by a contemporary musician, and allowing the public to enjoy and search into the digital contents of these works, either as a musicologist or as a performer.

From the technical point of view, it is important to note that the symbol recognition and processing stages of the system are designed using Machine Learning (ML) techniques. Most available systems are based on heuristic rule methods, but one of the problems of OMR is the existence of too many repertoire-dependent context rules [1]. Therefore, it is hard to extend them to recognize early music notations, that are sometimes very different from the modern one. By applying innovative ML technologies we can build models based on *training pairs*: sets of images that are presented together with ground-truth labels. This is a key point of this method, because it opens up the possibility of adapting the system, in principle, to any music notation, provided that labelled data are available for the system to learn. In fact, the proposed system is being developed working on two different collections of handwritten music scores: vocal music manuscripts, written in Spanish mensural notation, and Spanish traditional songs transcribed by hand in modern music notation by ethno-musicologists.

MuRET has been designed to fulfill these requirements, giving support to the whole process of transcription of a manuscript source in such a way the traceability of each step is maintained while allowing different OMR approaches (detailed below). In addition, as this system is conceived as a research tool, it has been equipped with features to measure the efficiency and effectiveness of the different user and machine tasks utilized.

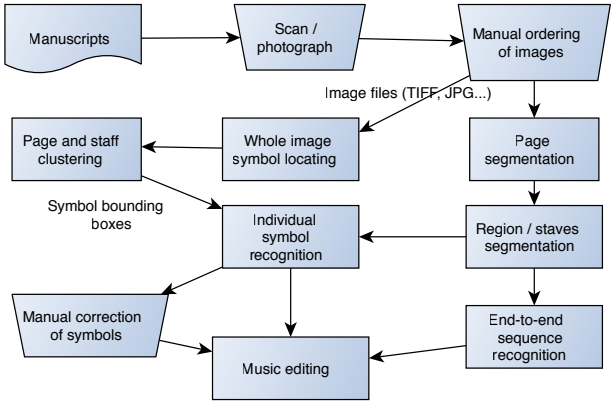


Figure 1. Allowed workflows in MuRET

[1]https://github.com/Audiveris
[2]https://www.neuratron.com/photoscore.htm

The scheme in Fig. 1 describes the different automatic and manual operations performed in the system for any of the considered approaches. The different workflows are implemented in such a way that the process can be accomplished in an arbitrary number of working sessions and users. Also, the stages include the logging of user actions for future performance assessment.

In the first stage, the system input are digitized image files of the manuscripts. The OMR operations generate two kind of sequence symbol encodings. One is named *agnostic*, which contains symbols by their graphic value and their position, without analyzing the role of the symbol in the score, *e.g.*, a ♯ symbol is tagged as a *sharp*, regardless it is a note modifier or an element of a key signature (like a D sharp key that has two of them). The other encoding is the *semantic* one, where meaningful musical information can be encoded in any standard music format. Here, the D Major key signature is encoded as an explicit D Major key code. The conversion from the agnostic encoding to a semantic one in mensural music requires a specific procedure to deal with context-dependent interpretations of graphical symbols.

Furthermore, MuRET is designed under the assumption that no OMR approach is able to achieve a perfect performance. Thus, the user intervention is required. This assessment and correction task can be accomplished at graphical symbol level (agnostic) or by editing the music content obtained directly (semantic).

We believe that it is difficult to find a single approach for OMR that can be effective in all kinds of situations. Therefore, we have implemented different approaches for this task:

User-driven symbol recognition: in this case, the user locates the symbols manually. The use of a digital pen [3] results in a more ergonomic interaction (see Fig. 2), specially for making corrections to system errors. This procedure yields a bounding box for each symbol, that is received by a classifier as input. The output is the category prediction of the symbol therein. Another classifier is then used to estimate the vertical position of the symbol within the staff. By clustering the positions of the bounding boxes on the y-axis, the number of staves can be determined, while an ordering on the x-axis allows determining the actual order in the sequence of symbols. The advantage of this approach is that the classifiers need little ground-truth data to obtain good results, but users' effort is demanding.

Holistic staff recognition: in this approach, each staff is processed holistically, that is, in just one step, without previous symbol segmentation. This can be achieved by using models like recurrent neural networks [2]. The advantage of these approaches is that the ground-truth data consists simply of pairs of staff images and their corresponding sequence of music symbols, without the need to indicate any geometrical information, but needs more training data.

From the semantic representation of the music content

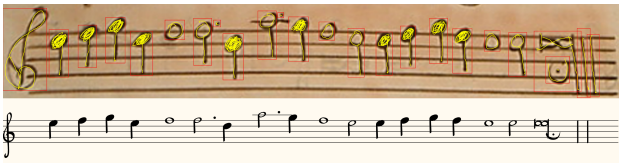


Figure 2. User-driven symbol recognition. Classifiers deal with manually located symbol bounding boxes.

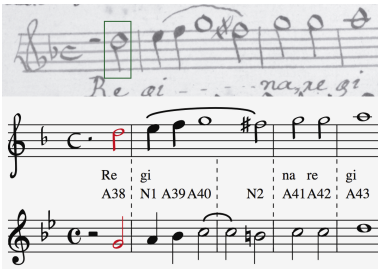


Figure 3. Spanish late white mensural transcription and translation to modern notation.

of a early music notation score, the system can make a transcription to modern music notation (see Fig. 3), task that can be implemented as a number of rules, stated by specialists.

The current goal of the works covered by the Hispamus project is not to generate final edited preprints, but to produce contents to be sent to online services or publishers. The MuRET system is being designed to export in any interchange format that will be eventually edited by the publishers to fulfill their publishing workflow.

ACKNOWLEDGEMENTS

This work is supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU.

REFERENCES

[1] Donald Byrd and Jakob Grue Simonsen. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, 44(3):169–195, 2015.

[2] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4):606–629, 2018.

[3] Jorge Calvo-Zaragoza, David Rizo, and José Manuel Iñesta. Two (note) heads are better than one: Pen-based multimodal interaction with music scores. In *17th Int. Society for Music Information Retrieval Conference*, pages 509–514, 2016.

[4] Laurent Pugin. Editing Renaissance Music: The Aruspix Project. *Internationales Jahrbuch für Editionswissenschaften*, pages 94–103, 2009.

Creating destruction animations by transferring hand-drawn styles

Takumi Kato
Graduate School Information Science and Engineering
Ritsumeikan University
Kusatsu, Shiga, Japan
is0301re@ed.ritsumei.ac.jp

Susumu Nakata
College of Information Science and Engineering
Ritsumeikan University
Kusatsu, Shiga, Japan
snakata@is.ritsumei.ac.jp

***Abstract*—Hand-drawn animation is a long-established means of artistic expression and still plays an important role, despite the spread of computer-generated animation. In addition, several studios have even retouched CG movies to make them look hand-drawn. In this study, we present a tool to support the production of destruction animations and help create hand-drawn animations. With our tool, the artist provides a sequence of hand-drawn pictures of an object being destroyed (source) and the geometry of another object (target) for which they want to create a destruction animation. Then, the tool automatically generates an animation of the target being destroyed. The key idea behind the tool is to effectively reflect the hand-drawn style of the source in the target. Two techniques are used to generate the animations: transferring the shapes of the source's cracks in the initial frame, and those of its fragments in the second and subsequent frames.**

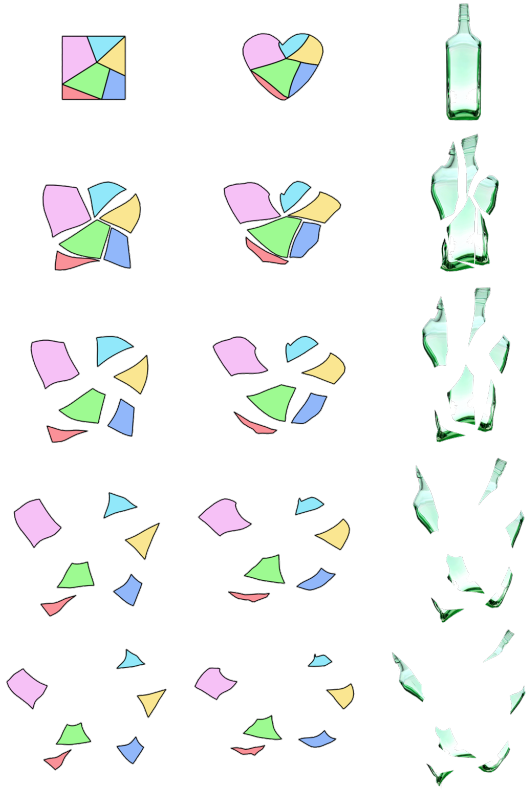
Keywords-animation; hand-drawn; drawing tool

I. INTRODUCTION

Hand-drawn animation, involving sequences of frame-by-frame drawings, is a long-established means of creating moving images. One of its distinctive features is that artists intentionally exaggerate particular movements depending on the situation [1], and that each frame contains artist-specific stroke variations that create particular impressions, different from those produced by accurate computer-generated animations. Destruction scenes involving many fragments are clear examples of such artistic expression, in the sense that even rigid fragments shift and deform as they move, due to changes in the artist's strokes. Some attempts have been made to automatically generate exaggerated animations of, e.g., rigid bodies [2].

This study aims to develop a tool that helps artists to produce destruction animations with a hand-drawn feel. Here we assume these animations satisfy the following three conditions: the object's initial shape is represented by a two-dimensional polygon; the first frame consists of a set of line segments that divide the original polygon into fragments; and each fragment deforms as it moves in the second and subsequent frames. The input is assumed to consist of a sequence of drawings of an object being destroyed (source) and the shape of another object (target) for which the artist wishes to create a destruction animation (Fig. 1).

The tool is designed to generate a destruction animation for the target by appropriately projecting the source's geometric information onto it. This is conceptually similar to an image style transfer [3], in the sense that the features of



(a) Source (b) Target 1 (c) Target 2

Figure 1. The source animation (a) is transferred to two targets, at the tops of (b) and (c), to generate destruction animations.

one image are reflected in another. In the first frame, the aim is to project the source's line segments onto the target in order to divide the target into suitable fragments. We perform this projection based on an interactive shape deformation technique for two-dimensional solid objects with meshes [4]. In the second and subsequent frames, the motion and deformation of each source fragment is projected onto the corresponding target fragment. We perform this projection based on the fragment's center of gravity and the relative positions of its vertices.

II. TRANSFERRING CRACK SHAPES

Our first task is to transfer the shapes of source's cracks in the first frame to the target. Here, we assume that the

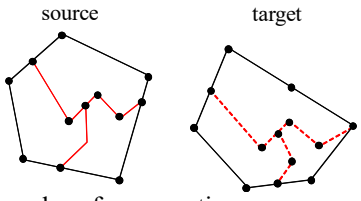


Figure 2. Examples of representing a source and a target by points and lines. The black and red lines represent the object outlines and the cracks, respectively.

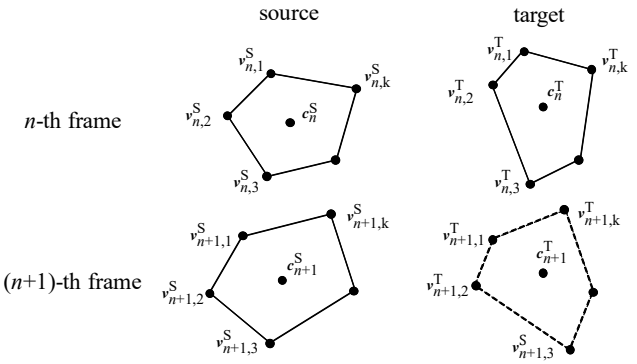


Figure 3. Examples of source and target fragments.

source and target shapes are polygons with the same number of vertices and that the source's cracks are given as line segments that divide the original shape into fragments (Fig. 2). The problem now is to transfer the cracks to the target shape, i.e., determine the crack vertices' positions within the target.

Our approach is to define the positions using a spatial projection from the source area to the target area. We calculate this projection using a shape deformation technique [4], which gives us the spatial projection between two shapes that preserves the spatial distribution as far as possible. This spatial preservation property enables us to generate target fragments whose shapes are close as possible to those of the source fragments.

III. Transferring the Fragment Shapes

Our second task is to transfer the source fragments' movements and deformations to those of the target fragments in the second and subsequent frames. More precisely, we assume that we are given a source fragment's vertices in the n-th and $(n+1)$-th frames, $\boldsymbol{v}^{\mathrm{S}}_{n,1},\ldots,\boldsymbol{v}^{\mathrm{S}}_{n,k}$ and $\boldsymbol{v}^{\mathrm{S}}_{n+1,1},\ldots,\boldsymbol{v}^{\mathrm{S}}_{n+1,k}$, and the corresponding target fragment's vertices in the n-th frame, $\boldsymbol{v}^{\mathrm{T}}_{n,1},\ldots,\boldsymbol{v}^{\mathrm{T}}_{n,k}$. The goal is to determine the target fragment's vertices in the $(n+1)$-th frame, $\boldsymbol{v}^{\mathrm{T}}_{n+1,1},\ldots,\boldsymbol{v}^{\mathrm{T}}_{n+1,k}$ (Fig. 3). We divide this task into two steps, namely determining the target fragment's centroid and the relative positions of its vertices.

The aim of the first step is to trace the source fragment's path. We determine the target fragment's centroid in the $(n+1)$-th frame by copying the source centroid's motion over to that of the target, i.e., $\boldsymbol{c}^{\mathrm{T}}_{n+1}=\boldsymbol{c}^{\mathrm{T}}_{n}+\boldsymbol{c}^{\mathrm{S}}_{n+1}-\boldsymbol{c}^{\mathrm{S}}_{n}$, where $\boldsymbol{c}^{\mathrm{S}}_{n}$, $\boldsymbol{c}^{\mathrm{S}}_{n+1}$, $\boldsymbol{c}^{\mathrm{T}}_{n}$, and $\boldsymbol{c}^{\mathrm{T}}_{n+1}$ are the centroids of the source and target fragments in the n-th and $(n+1)$-th frames, respectively.

Then, the second step aims to reflect the source fragment's deformation in the target. Here, we position each target vertex so as to preserve the corresponding source vertex's rotation and the scaling of the distance from it to the centroid. In other words, the i-th target vertex in the $(n+1)$-th frame, $\boldsymbol{v}^{\mathrm{T}}_{n+1,i}$, is given by

$$\boldsymbol{v}^{\mathrm{T}}_{n+1,i}=\boldsymbol{c}^{\mathrm{T}}_{n+1}+\frac{\left\|\boldsymbol{v}^{\mathrm{S}}_{n+1,i}-\boldsymbol{c}^{\mathrm{S}}_{n+1,i}\right\|}{\left\|\boldsymbol{v}^{\mathrm{S}}_{n,i}-\boldsymbol{c}^{\mathrm{S}}_{n,i}\right\|}\cdot R(\theta)(\boldsymbol{v}^{\mathrm{T}}_{n,i}-\boldsymbol{c}^{\mathrm{T}}_{n}),$$

where $R(\theta)$ is the rotation matrix and θ is the angle between $(\boldsymbol{v}^{\mathrm{S}}_{n,i}-\boldsymbol{c}^{\mathrm{S}}_{n})$ and $(\boldsymbol{v}^{\mathrm{S}}_{n+1,i}-\boldsymbol{c}^{\mathrm{S}}_{n+1})$.

IV. Results

Fig. 1 shows the results of creating some example destruction animations. Here, we provided the images in Fig. 1(a) as the source animation, and the two polygons at the tops of Figs. 1(b) and (c) as the target shapes. The results of transferring the cracks are shown at the tops of Figs. 1(b) and (c), with the animations produced by our tool following below. These results show that the crack shapes were generated based on the source cracks, and that the subsequent source fragment movements and deformations were successfully transferred to the target fragments.

V. Conclusion

We have created a tool that can efficiently create destruction animations in a hand-drawn style by transferring crack and fragment shapes from an existing animation. The style can be transferred while preserving the source features as far as possible by using an interactive shape deformation technique and projecting the center of gravity and the relative positions of each fragment's vertices from the source onto the target. As our results demonstrate, this enables us to create animations with hand-drawn features, such as exaggerated and variable strokes. One of the limitations of our method is that fragments are assumed to be described as a set of polygons, i.e., a sketch recognition system is required for practical use. The main idea of this work is expected to be applied not only to destructing objects but also to other effects like flame and fluid animations.

References

[1] F. Thomas, and O. Johnston, Disney Animation: The Illusion of Life. New York: Abbeville Press, 1981.

[2] M. Dvorožňák, P. Bénard, P. Barla, O. Wang, and D. Sýkora, "Example-based expressive animation of 2D rigid bodies," ACM Trans. Graph., vol. 36, Article 127, July 2017.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," The IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423, June 2016.

[4] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," ACM Trans. Graph., vol. 24, pp. 1134–1141, July 2005.