Intermodal Image-Based Recognition of Planar Kinematic Mechanisms

Matthew Eicholtz, Levent Burak Kara
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA USA
meicholt@andrew.cmu.edu, lkara@cmu.edu

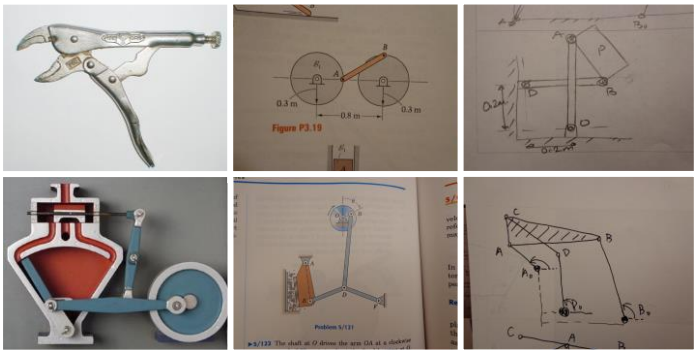
***Abstract*—We present a data-driven exploratory study to investigate whether trained object detectors generalize well to test images from a different modality. We focus on the domain of planar kinematic mechanisms, which can be viewed as a set of rigid bodies connected by joints, and use textbook graphics and images of hand-drawn sketches as input modalities. The goal of our algorithm is to automatically recognize the underlying mechanical structure shown in an input image by leveraging well-known computer vision methods for object recognition with the optimizing power of multiobjective evolutionary algorithms. Taking a raw image as input, we detect pin joints using local feature descriptors in a support vector machine framework. Improving upon previous work, detection confidence depends on multiple context-based classifiers of varying image patch size and greedy foreground extraction. The likelihood of rigid body connections is approximated using normalized geodesic time, and NSGA-II is used to evolve optimal mechanisms using this data. The present work is motivated by the observation that textbook diagrams and hand-drawn sketches of mechanisms exhibit similar object structure, yet have different visual characteristics. We apply our method using various combinations of images for training and testing, and the results demonstrate a trade-off between solvability and accuracy.**

Keywords-computer vision; evolutionary multiobjective optimization; kinematic simulation; object recognition

I. INTRODUCTION

The design of complex mechanical linkages is a challenging task involving the coordination of multiple rigid bodies to achieve a desired dynamic profile (see Fig. 1 for examples). The ability to visualize the kinematics of a mechanism is a valuable skill to improve mechanical intuition during design analysis and synthesis [1], yet current simulation tools may be insufficient for fast kinematic visualization. Currently, engineers will likely resort to one of three options. First, they may use mental simulations to infer mechanical behavior [2], but this is ineffective for people with low spatial ability [3] and is generally difficult for complex mechanisms [4]. Second, specialized software [5-6] may be used for simulations, but this task is often too time-consuming to be practical (e.g. students solving a dynamics homework problem, professional engineers brainstorming potential design concepts) and may require advanced programming skills, which hinders novice users. Third, engineers often use hand-drawn sketches

This work is supported by the National Science Foundation under Grant Nos. CMMI-084730, CMMI-1031703, and DUE-1043241.



(a) (b) (c)

Figure 1. Example mechanisms in (a) natural images of real-world objcets, (b) textbook graphics, and (c) hand-drawn sketches. Each mechanism contains a set of rigid bodies connected by kinematic pairs (e.g. revolute or prismatic joints) that constrain their motion. The present work focuses on automatically recognizing the number, location, and connectivity of joints in textbook graphics and hand-drawn sketches; this information is all that is required to fully specify the allowable motion of each rigid body.

to convey design ideas and visualize dynamic properties, perhaps abstracting the mechanism to a simpler form or using key annotations and arrows to demonstrate motion.

In a previous work [7], we developed an algorithm to bridge the gap between ineffective mental simulations and impractical computer simulations by automatically recognizing the underlying mechanical structure in a single image. At the heart of our approach was a novel combination of vision-based object recognition with multiobjective evolutionary optimization. The fundamental principle of the method was to consider mechanisms as a collection of connected joints, where each pairwise joint connection indicated that two joints were fixed to the same rigid body. We limited our study to planar mechanisms, in which the motion of every rigid body is constrained to the plane perpendicular to the viewer, and only considered examples made up entirely of revolute joints. With this representation, the task involved locating probable joints in an image using a sliding window object detector, assessing the likelihood of all pairwise joint connections using normalized geodesic time and maximizing image consistency and mechanical feasibility using the NSGA-II algorithm. The algorithm enabled the evolution of a small set of feasible mechanical structures based on local features in a single image, and only required a set of training images for joint detection.

We initially implemented the approach on textbook graphics due to their relative simplicity and wide availability.

In the present work, outlined in Fig. 2, we shift our focus to include sketches as valid input data to our algorithm. This is motivated by the idea that sketches are more directly related to design synthesis than textbook graphics. Someone creating a new mechanism may not be able to find a clean image depicting their design concept; indeed, they may not even know what they are looking for yet. With our technology, we hope to enable users to rapidly explore the design space using pencil and paper without being encumbered by existing designs.

We represent sketch data as an image, so that no modifications to the original algorithm are explicitly required to accommodate the new input modality. Regardless, we propose a couple key enhancements to the joint detection scheme in order to boost performance; details are provided later in this paper. Despite being of the same "form" as the textbook images used previously, we still consider sketches to be from a different *modality* because they were created in a different manner than textbook graphics. The evidence in support of this proposition is clear from the examples pictured in Fig. 1. Textbook graphics use consistent shapes, colors, and textures, while sketches are typically messier, have curvier lines, and include artifacts such as overtracing, tonal variation in stroke intensities, and cross hatching, among others [8]. Furthermore, depictions of mechanisms in textbooks may be surrounded by irrelevant text, annotations, highlighting, or other mechanisms that clutter the image; sketches, on the other hand, can be created without such distracting visual elements.

Even though they may be strikingly different in certain visual characteristics, textbook graphics and sketches of mechanisms adhere to the same structural principles. This poses an interesting problem: can we successfully use one input modality for training and the other for testing? More specifically, are we required to have a set of training sketches in order to correctly recognize test sketches of mechanisms? The answer may have important implications for future tools involving the recognition of visual objects with different input modalities.

The remainder of the paper is structured as follows: section II highlights related work in sketch recognition, computer vision, and evolutionary algorithms. Improvements made to our original algorithm are provided in section III. Experimental methods, including results and discussions, are given in section IV, followed by concluding remarks in section V.

II. Related Work

A. Object Detection

Object detection is a mature field of research in computer vision, spanning countless real-world applications. A typical object detector extracts salient features from sample images, learns a discriminative model from those features, and then scans test images using the model to locate instances of the object. Arguably the most critical step in developing a detection algorithm is feature selection. There are many well-known feature descriptors with reported success [9-11]; in the present work, locally normalized histograms of oriented gradients (HOG) over a grid of regions in the image are used. We follow the method outlined in the original work [9], which includes training a soft linear support vector machine (SVM) and mining hard negatives from sample images for subsequent re-training. We selected the HOG descriptor because it is a popular, dense, local feature set that has been successful for detecting various objects. However, our algorithm is not dependent on this choice; any feature descriptor and classifier can be incorporated into the overall recognition pipeline.

To our knowledge, kinematic mechanisms are a novel domain for object recognition. However, there is a breadth of ongoing research in recognizing similar objects comprising structured parts. Practical applications include face recognition [12], pose estimation [13], and 3D surface estimation [14]. The key difference, though, between previous work in this area and our present domain is that mechanisms do not have well-defined structural or spatial dependencies. For example, in face recognition, it is straightforward to learn that the forehead is not located below the mouth or that a nose should exist between the eyes; with kinematic mechanisms, it is less clear if a specific joint should be connected to another. Little knowledge is gained about the likelihood of other objects in the image just from knowing one object's location.

Object recognition across multiple modalities is a less well understood problem in computer vision. The most relevant works relate to face photo-sketch recognition [15-17], which attempts to match hand-drawn sketches of faces with samples in an image database. Various methods are used to find discriminating features between the two modalities; some even transform one modality to another (e.g. convert all photos to pencil sketches) in an effort to reduce the variance among the dataset. Our present work, by contrast, must not only recognize an object (mechanism) across modalities, but also should detect parts (joints) that make up the object across modalities.

B. Sketch Recognition

There are two important aspects of sketch recognition that relate to the present work: representation and complexity. With regard to representation, two classes of techniques have emerged in the literature. *Stroke-based* methods treat each sketch as a sequence of time-stamped strokes, each containing a series of sample points in space. While some works share similarities to our domain [18-23], stroke-based methods are ill-suited for our recognition framework, which was designed to work on images. Still, there are interesting parallels; for instance, [22] uses a graph representation to combine "low-level primitives into high-level shapes using geometrical rules". We also implement graphs in our recognition pipeline, but instead connect low-level joints to form high-level mechanisms based (partially) on mechanical feasibility rules. The other class of sketch recognition techniques is *image-based* approaches, including the present work, which neglect temporal information and only consider the spatial layout of pixels. This poses the additional challenge of grouping relevant pixels, depending on the object being recognized. With regard to sketch complexity, it is important to distinguish between isolated symbol recognizers and detecting objects in freehand sketches, which is a more challenging problem. The task of

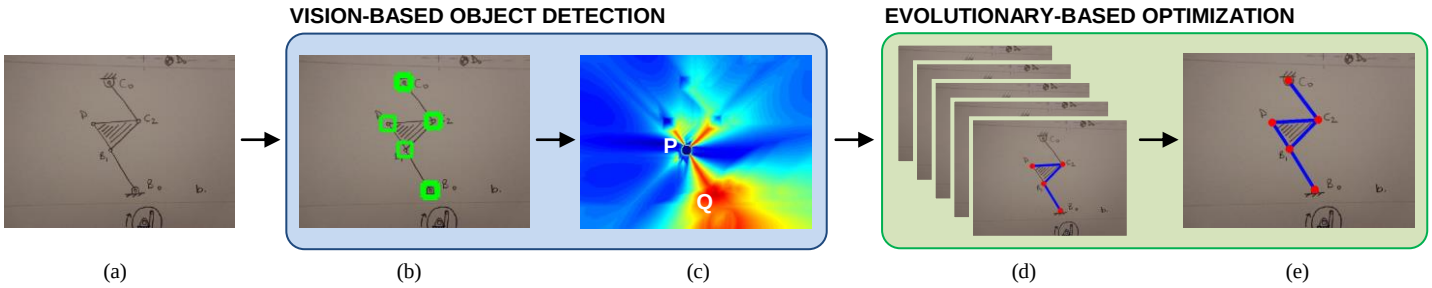


Figure 2. Overview of recognition pipeline. (a) The system takes as input a raw image. (b) To recognize the underlying mechanical structure, the first step is to detect all pin joints in the image; here, line thickness correlates positively with detection confidence. (c) Next, we compute the likelihood that pairwise detected joints are on the same rigid body using normalized geodesic time. In effect, this metric looks for short paths through dark regions in the image. Since there is a dark line connecting point P to Q in the original image, the normalized geodesic time from P to Q is high (indicated by the red color around Q); refer to [7] for more details. (d) Finally, the resulting data is optimized in an evolutionary framework, evolving a set of solutions (mechanisms) using conventional genetic operators and hopefully finding (e) the true solution over time.

symbol recognition can be treated as a template matching problem; some examples of successful approaches in this area include [23-29]. In some sense, the joint recognition algorithm used here is similar to a sliding window symbol recognizer. However, we do not use part templates and instead learn a discriminative model based on local image features.

It is widely agreed that robust sketch recognition algorithms require a large corpus of training images. Yet, acquiring such a large number of sample sketches can be a tedious task. One recent work [30] demonstrated the ability to automatically generate synthetic images from a small set of labeled examples. Another [31] investigates the effect of using isolated symbols for training a recognizer designed for freehand sketches. Our work contributes to this area by hypothesizing that sketch images may not be necessary for training if examples from another modality are more readily-available.

C. *Evolutionary Multiobjective Optimization*

Multiobjective evolutionary algorithms (MOEAs) are widely used in real-world applications that require optimization of several, often conflicting, objectives (fitness criteria). MOEAs operate by stochastically sampling the search space of candidate solutions and iteratively applying genetic operators such as crossover and mutation to evolve optimal solutions. To handle conflicting objectives, for which there is no single optimal solution, many MOEAs use the idea of Pareto dominance to rank solutions [32-34]. An individual solution is said to dominate another solution if it is at least as good for all objectives and better (more fit) for at least one objective. The Pareto front is defined as the set of all nondominated solutions. The algorithm selected for our approach, called the nondominated sorting genetic algorithm, was first introduced two decades ago (NSGA [32]) and improved several years later (NSGA-II [33]). We use the latter version, which is characterized by fast computation of nondominated sorting and inclusion of crowding distance to preserve diversity and showed promising results in prior work [7].

For the present domain, the feasibility of a predicted mechanism is governed by mechanical principles. These principles can be formulated as a series of constraints; in this way, large regions of the search space may become infeasible because one or more of the constraints fail. A critical step in MOEA design is determining how to handle such constraints. Constraint handling methods can be broadly categorized into two groups: (i) those that always prefer feasible solutions (hard constraints) [33,35] and (ii) those that treat constraints as objectives (soft constraints) [36]. We employ the latter method in order to allow infeasible, yet strong, solutions to persist because they may be near the constraint boundaries.

III. Technical Approach

The proposed framework for mechanism identification in images from various modalities largely relies on work previously developed in [7]. In this section, we provide a brief overview of the algorithm pipeline, followed by detailed descriptions of key modifications made to the original work. Unless stated otherwise, we use the same methods and parameters as [7].

A. *Overview*

The recognition framework (Fig. 2) has two primary stages: (i) vision-based detection of mechanical components, and (ii) evolutionary-based optimization of the mechanism structure. The algorithm was developed to be general in nature; any image type is a valid input to the system, any feature descriptor and classification method can be used to detect joints, any metric can be used to compute pairwise joint connection likelihood, and any genotype representation and genetic operators can be tested in the evolutionary algorithm. A basic outline of the algorithm is listed in Fig. 3; recent improvements are highlighted and will be discussed in the following sections.

B. *Using Multiple Context-Based Classifiers*

Previously, a fixed-window SVM classifier was used to detect likely pin joints in an image. The recall was generally high (i.e. very few false negatives), but the precision was sometimes low (i.e. too many false positives). Furthermore, the evolutionary algorithm does not optimize joints based on a simple binary decision; instead, it relies on the strength of classification, which we define as the distance to the SVM decision boundary. With this in mind, it should be clear to see that even a high-precision, high-recall classifier can be problematic for the optimization routine if even one false positive in an image has strong confidence. Also, previous

Algorithm 1 – Main

Pre-training:

1. Acquire sample images containing planar kinematic mechanisms.
2. Manually label all pin joints and pairwise joint connections in each image.
3. Separate data into training and testing sets.

Training:

1. Extract positive examples of pin joints in training images.
2. Augment positive examples by reflecting image patches about vertical/horizontal axes and rotating by {90,180,270} degrees.
3. Extract random negative examples from training images. Use tolerance of 32 pixels to ensure negative patches do not contain pin joints.
4. Compute HOG features for positive and negative image patches.
5. Train a soft (C=0.01) linear SVM.
6. Randomly extract 1000 additional patches per training image, classify using initial SVM, and add hard negatives to dataset.
7. Re-train the SVM.
8. Repeat training process using larger window sizes.

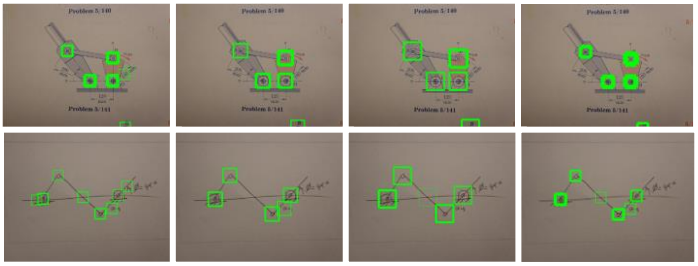
Testing:

1. Apply SVM to test image with sliding window of fixed size.
2. Suppress non-local maxima using mean shift algorithm [37].
3. Apply multiple classifers to detected joints to compute weighted confidence. Discard detections with confidence less than zero.
4. Apply foreground extraction to image. Discard background detections.
5. Store detected pin joint locations and confidence values.
6. For all pairs of detected pin joints, compute normalized geodesic time, which indicates the likelihood that those two joints are located on the same rigid body.
7. Store connection likelihood matrix.
8. Run NSGA-II using pin joint locations, associated confidence levels, and connection likelihood as input. Fitness evaluation includes image consistency measures and binary constraints for mechanical feasibility. The output is a set of Pareto-optimal solutions.
9. Discard solutions that are infeasible.
10. Remove duplicate solutions.
11. Prioritize the remaining solutions (currently using strength of joint connections).
12. Locate the ground truth (if applicable).

Figure 3. Algorithm details for the main recognition pipeline.

experiments revealed that execution time strongly depends on chromosome length, which is a function of joint detections. Therefore, it is highly desirable to decrease the number of false positive detections and increase the confidence of true positives relative to false positives.

To address this challenge, we implemented two significant modifications to the detection scheme. First, we incorporated multiple classifiers with varying window size with the idea that larger window sizes would pick up more global context cues regarding true joints. This design decision was primarily motivated by the observation that many false positives demonstrated strong local correlation to pins (e.g. text containing the letter 'o'), but lacked similarity in a global context (e.g. a pin usually has two rigid bodies emanating from its center, while the letter 'o' does not). For the current implementation, we used a root detection window size of 48 pixels, and two additional context classifiers with window sizes of 64 and 80 pixels, respectively. We increase the appropriate HOG descriptor parameters such that all image patches have the same number of features (in this case, 1764). In this manner, the larger classifiers have the same dimensionality, but coarser spatial binning due to increased window size. Contrary to some other approaches involving multi-scale classification,



(a) (b) (c) (d)

Figure 4. Using multiple context-based classifiers on sample textbook images and sketches. (a-c) Individual classifiers with increasing window size. (d) Final joint detection confidence. Line width positively correlates with confidence.

we do not run all classifiers on the full image. Instead, the root classifier is first used to find local maxima, after which patches centered at those locations are fed into the context classifiers. At that point, every detected pin joint has three values of confidence. A naïve approach might be to define the total confidence as the summation of individual detection strengths. This may yield a poor estimate of true confidence because the SVM decision boundaries do not incorporate normalization. Instead, we propose a weighted sum of confidence, in which the classifier weights, w_i, are given by,

$$w_i = \left(\frac{\mathrm{Pr}}{\mu} \right)_i \tag{1}$$

where Pr and μ are the precision of the classifier on training data and the average distance of true positives to the SVM decision boundary, respectively. Sample results from this improved detection scheme are shown in Fig. 4.

C. *Greedy Foreground Extraction*

The cascade of classifiers described above mainly improves the relative confidence of true positives with respect to false positives. To achieve high precision, we implement a greedy unsupervised foreground extraction method and discard any background detections. The approach is outlined in Fig. 5. Despite being a greedy approach, it performs exceptionally well on the images used in our experiments. Accuracy for both textbook images and sketches was 99%. The effect of this algorithm enhancement is depicted in Fig. 6.

Algorithm 2 – Foreground Extraction

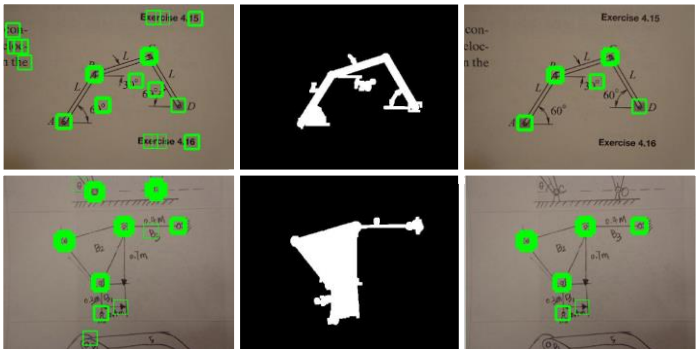
1. Run Sobel edge detector [38] over image.
2. Dilate edges by an 8-pixel radius.
3. Trace boundaries and extract connected regions.
4. Select the region with the maximum area.
5. Fill holes in the region.
6. Save region as foreground.

Figure 5. Algorithm details for foreground extraction.

IV. EXPERIMENTS

A. *Data*

Two image datasets were utilized in the experiments described in this paper (see Fig. 7 for examples). First, we use the MECH135 dataset [7], which includes 135 images of planar mechanisms from five different textbooks [39-43]. All mechan-



(a) (b) (c)

Figure 6. Greedy foreground extraction on sample textbook images and sketches. (a) Weighted sum of multiple classifiers. (b) Binary image showing foreground in white. (c) Joint detections after discarding background instances.

isms are closed kinematic chains and contain only revolute joints. In addition, we asked 25 engineering graduate students (21 male, 4 female; all 20-30 years old) to sketch ten randomly selected mechanisms from the MECH135 dataset on paper. In general, we did not restrict the sketching style nor the level of abstraction implemented by the students, as long as the true underlying mechanical structure was evident in the sketch. Pictures were taken of all hand-drawn sketches, yielding a second dataset of 250 images, approximately two samples per image in the MECH135 dataset. While it is assumed the full mechanism is shown in each image, no explicit restrictions were made regarding position, scale, or orientation of the object. Also, our approach does not require pre-processing of the images (e.g. cropping, filtering), so they may contain noise, illumination changes, and extraneous information such as text, annotations, pencil markings, or partial components from other mechanisms. Ground truth information, including joint location and pairwise connections, was manually provided for all images in both datasets.

B. Methods

The goal of the experimental studies was to assess the efficacy of our approach on various combinations of training and test images. To that end, six experiments were conducted using all permutations of textbook, sketch, or combined images for training and textbook or sketch images for testing. Each experiment comprised ten separate trials. For each trial, 100 training images and 20 test images were randomly selected, without overlap, from the appropriate datasets. The training images were used to generate a joint detector as described previously and subsequently applied to each test image to locate probable pin joints. Using this information, along with normalized geodesic time for pairwise joint connections, ten independent runs of the evolutionary algorithm were executed per test image, using the general settings listed in Table I. This amounts to 2,000 distinct instances for each experimental condition. Performance metrics of interest include accuracy and speed. Chi-square and Kruskal-Wallis tests were used to determine statistical significance. The implementation was developed in MATLAB [6], and all experiments were performed on an Intel(R) quad-core 3.40 GHz CPU with 8GB RAM.

TABLE I. General NSGA-II Parameters

Parameter	Symbol	Value
population size	μ	$200N^a$
number of offspring	λ	μ
maximum number of generations	n	20
crossover method	–	uniform
crossover probability	p_c	0.9
mutation method	–	uniform
mutation probability	p_m	0.1
tournament size	k	0.02μ

[a] N refers to number of detected joints

C. Results and Discussions

Example results on textbook graphics and sketches are shown in Fig. 7. The overall algorithm performance for each experimental condition is summarized in Table II, with the best values highlighted for each statistic. With regard to accuracy, several relevant metrics of success are presented that each contribute to one of two primary objectives: (i) was the true solution found by the evolutionary algorithm and (ii) if so, where in the prioritized Pareto-optimal set of solutions was it located?

An image is deemed *solvable* if the true solution is able to be found based on the data input to NSGA-II. Consider the test cases shown in Fig. 8. These instances are unsolvable; the underlying mechanical structure will *never* be correctly identified by our approach because the joint detector failed to locate one or more true joints. In this way, the percentage of solvable test images for an experiment reflects the quality of the joint detector on that dataset. For textbook images, the joint detector always performed relatively well (min. solvable = 88%), and it produced the least number of images with false negative detections when combining textbook and sketches for training (solvable = 94%). This is an interesting result that suggests sketching sample mechanisms may improve detector performance on textbook images. However, this does not guarantee improved performance of the entire algorithm, as evidenced by comparison of the remaining accuracy measures in rows A and E. For sketches, the joint detector did not perform as well, particularly when trained on textbook images (solvable = 47%). This is understandable given the high degree of variance in sketched pins compared to textbook graphics. Still, it is encouraging to note that the number of unsolvable sketch images decreases by more than 50% if sketches are added to the training set and is minimized when only sketches are used for training. A chi-square test revealed that the experimental conditions had a statistically significant effect on joint detector quality and hence the proportion of solvable images in a given test sample, $\chi^2(5, N = 1{,}200) = 180.75$, $p < .001$.

We classify an image as *solved* if our algorithm was able to correctly identify the underlying mechanical structure in at least one independent run. For both image datasets, a larger fraction of test images was solved when the training images were drawn from the same dataset. Using combined datasets for training does not appear to improve this performance measure. Once again, it should be noted that training on textbooks and testing on sketches resulted in a low number of solved images. The relative difference of solvable and solved

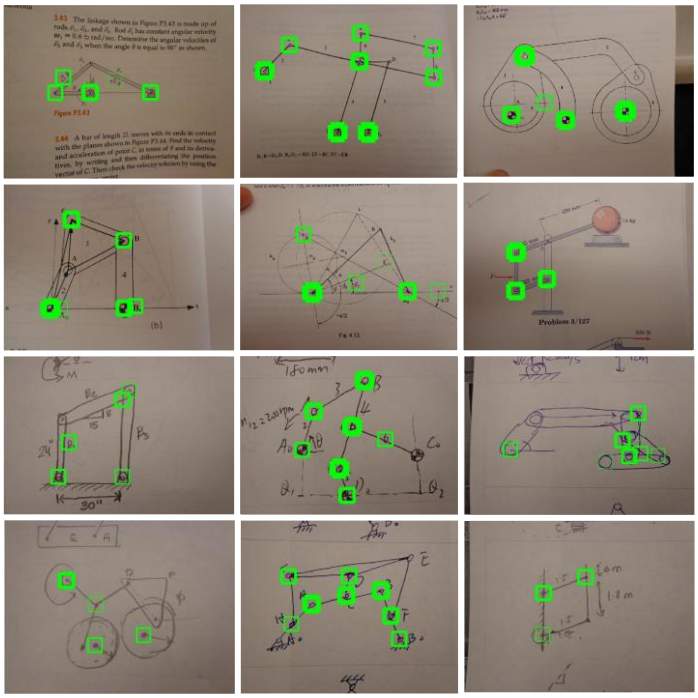
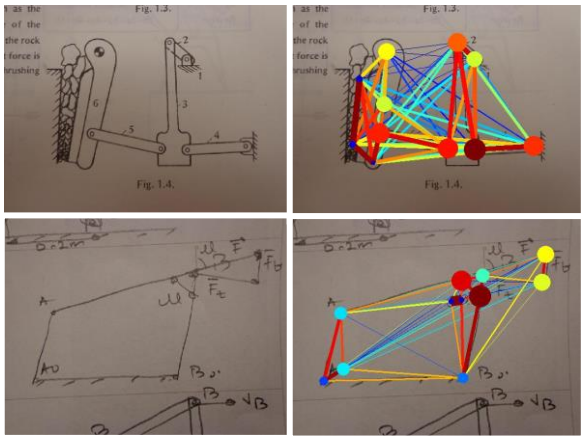


Figure 8. Unsolvable images, due to the presence of false negatives. Bounding boxes indicate pin joint detections, and line width positively correlates with confidence.

images yields the fraction of images that failed due to something other than false negative detections. The mean percentage of solvable, yet unsolved, images across all experiments is 32%. Experiment B had the best performance in this regard, with only one-fifth of solvable images failing to ever produce the correct mechanism, while experiment D was the worst-performing case, with over one-half of solvable images remaining unsolved. The observed differences in the percentage of solved images were found to be statistically significant, $\chi^2(5, N = 1{,}200) = 85.70$, $p < .001$. Some example failure cases and possible explanations are provided in Fig. 9.

The *overall success rate* (OSR), or the average number of runs in which the true solution was found, exhibits a similar trend to the *solved* metric. With the exception of experiment B, the true mechanism is identified in at least half of the runs. Also, recognition of textbook images is higher than sketch images in general. While overall success rate is a reasonable estimate of algorithm effectiveness for a given set of training and testing images, we suggest it is not the only meaningful metric because it is negatively skewed by unsolved images. With this in mind, we also compute the solved success rate (SSR), which characterizes the reliability of our approach for images that were correctly recognized at least once. The results are somewhat surprising; the most reliable experimental condition is the textbook/sketch case (93% SSR). In other words, if an image of a hand-drawn sketch is solvable, the algorithm presented in this paper is highly likely to correctly identify the pictured mechanism. One plausible explanation is that many sketches are less cluttered than their textbook counterparts; there is limited extraneous information that could be falsely identified by the algorithm and rigid bodies are typically drawn as simple dark lines, which is favored by our



(a) (b)

Figure 9. Example failure cases unrelated to false negative detections. (a) Raw images. (b) Visualization of data sent as input to NSGA-II. Higher confidence is indicated by darker (redder) color and thicker shapes. The first instance likely fails because the strongest joint connections are between false positives in the rocks. The primary failure in the second instance is that the strongest joint detections are false positives.

method for predicting pairwise joint connections. On a different note, experiments A, C, and E have nearly identical SSR; perhaps the modality of the training set is less critical when testing textbook images. Lastly, all experimental conditions produced a higher SSR than the previous work in [7], indicating that our modified pin detection method with foreground extraction appears to improve performance. Chi-square tests were performed and indicate there is a relationship between training and testing modalities and algorithm success rate; $\chi^2(5, N = 12{,}000) = 492.26$, $p < .001$ and $\chi^2(5, N = 7{,}210) = 39.29$, $p < .001$ for OSR and SSR, respectively.

While the ability of the evolutionary algorithm to find the true underlying mechanical structure is valuable, perhaps a more important performance measure is *where* the solution was found; that is, can we rank the Pareto-optimal set of solutions in such a way that the true solution has highest priority? The top-*N* accuracy refers to the percentage of successful runs in which the true solution was at least in the top *N* solutions. The obvious desired result is for top-1 accuracy to be high; however, this may not be very realistic. Even one false positive joint detection with confidence higher than any of the true joints will likely allow one or more incorrect solutions to be nondominated by the correct solution. Therefore, we think it is reasonable if the true mechanism is at least in the top 5 solutions generated by NSGA-II. At that point, the best solution could be extracted from this small set either by interactive user selection or feedback from full kinematic simulations. Looking at Table II, there are a few notable highlights to mention. First, experiments with sketches as the test set generally have higher top-*N* accuracy than similar experiments using textbook images (compare A↔D, B↔C, E↔F). Second, top-*N* accuracy on test images is lowest when training/testing modality differs and highest when modalities are the same. Finally, the best-performing case from this viewpoint is training on textbooks and testing on sketches, with over half of successful runs yielding the optimal scenario and top-5 accuracy of 84%. All observed differences with regard to

TABLE II. ALGORITHM PERFORMANCE

Experimental Setup			Accuracy (%)							Speed[a] (sec)	
ID	*training*	*testing*	*solvable*	*solved*	*success rate, overall*	*success rate, solved images*	*top-1*	*top-3*	*top-5*	*overall*	*per gene*
A	textbook	textbook	89.5	74.5	62.8	84.4	33.3	51.1	61.7	3.34±2.83	0.222±0.066
B	textbook	sketch	46.5	35.0	32.4	92.6	52.2	78.1	84.4	1.64±1.42	0.210±0.022
C	sketch	textbook	88.0	58.0	48.8	84.1	22.7	40.8	50.5	5.44±4.24	0.201±0.006
D	sketch	sketch	81.5	61.5	53.8	87.5	41.8	67.3	74.1	2.97±2.60	0.203±0.008
E	combined	textbook	94.0	73.5	61.9	84.1	29.3	44.9	53.6	4.25±4.61	0.201±0.011
F	combined	sketch	74.0	58.0	50.4	86.9	50.1	69.8	79.9	2.98±2.74	0.206±0.009

[a.] for running NSGA-II on one test image

top-*N* accuracy were found to be statistically significant; $\chi^2(5, N = 6{,}201) = 273.02$, $p < .001$ for top-1 accuracy, $\chi^2(5, N = 6{,}201) = 427.08$, $p < .001$ for top-3 accuracy, and $\chi^2(5, N = 6{,}201) = 414.30$, $p < .001$ for top-5 accuracy.

An accurate, reliable recognition algorithm is less useful if computationally expensive, so speed is another important performance characteristic to consider. Table II lists the average time it takes to complete a single run of NSGA-II, both overall and per gene; the fastest and slowest experiments are B and C, respectively. A Kruskal-Wallis test showed that the effect of experimental setup on overall runtime was statistically significant, $\chi^2(5) = 2221.6$, $p < .001$. Post-hoc analysis using Tukey's honestly significant difference (HSD) test demonstrates all pairwise experimental conditions have significantly different runtimes, with the exception of D and F. As expected, the time per gene is relatively constant (~200ms), so overall runtime becomes a function of chromosome length. Recall that the number of genes is directly related to the number of detected pins in the image. Thus, comparison of overall times reflects differences in joint detector performance. For example, the optimization is slower when testing textbook images, implying those images have more falsely-detected joints on average than sketches, which is understandable given the extra information (e.g. dimensions, annotations) usually present in textbooks. The computational cost of training the joint detector is overhead and therefore neglected from this analysis. Also, scanning a test image for likely joints and computing pairwise joint connections are independent of experimental conditions, so those metrics are not listed either; total testing time remains on the order of seconds.

V. CONCLUSIONS

In this paper, we explored the ability of an object detector trained on textbook graphics to positively contribute to the automatic recognition of kinematic mechanisms in images of hand-drawn sketches and vice versa. We improved our previous algorithm by incorporating weighted context cues from multiple classifiers and a greedy foreground extraction technique in the joint detection pipeline. Current experimental studies indicate a trade-off between solvability (whether an image can ever be solved by the evolutionary algorithm) and top-*N* accuracy (where a solution is found on the Pareto front). Sketches appear more likely to miss true joints, but less likely to be misled by extraneous information and false positives, resulting in high top-*N* accuracy. All test images benefited from the inclusion of textbook graphics during training. We think this a powerful idea that could be leveraged to create an intelligent sketch recognition tool to generate kinematic simulation models in a matter of seconds without ever needing a sample sketch for learning.

REFERENCES

[1] J. G. De Jalon and E. Bayo. *Kinematic and dynamic simulation of multibody systems*. Springer-Verlag New York, USA, 1994.

[2] M. Hegarty. Mechanical reasoning by mental simulation. *Trends in cognitive sciences*, 8(6):280-285, 2004.

[3] M. Hegarty. Mental animation: inferring motion from static displays of mechanical systems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(5):1084, 1992.

[4] M. Hegarty and V. K. Sims. Individual differences in mental animation during mechanical reasoning. *Memory & Cognition*, 22(4):411-430, 1994.

[5] Adams. *version 2013.2*. MSC Software Corporation, Newport Beach, California, 2010.

[6] MATLAB, *version 8.1.0.604 (R2013a)*. The MathWorks, Inc., Natick, Massachusetts, 2013.

[7] M. Eicholtz, L. B. Kara, and J. Lohn. Recognizing planar kinematic mechanisms from a single image using evolutionary computation, *GECCO '14,* July 12-16, 2014, Vancouver, BC, Canada. in press.

[8] K. Eissen and R. Steur, *Sketching: drawing techniques for product designers*. Bis, 2007.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 1, pp. 886-893, 2005.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision-ECCV 2010*, pp. 778-792, Springer, 2010.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis*, 60(2):91-110, 2004.

[12] A. L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1);59-70, 1991.

[13] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1385-1392, 2011.

[14] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2737-2744, 2011.

[15] X. Wang and X. Tang. Face photo-sketch syntehsis and recognition. *IEEE Trans Pattern Anal Mach Intelli*, 31(11):1955-1967, 2009.

[16] B. F. Klare, Z. Li, and A. K. Jain. Matching forensic sketches to mug shot photos. *IEEE Trans Pattern Anal Mach Intelli*, 33(3):639-646, 2011.

[17] D. Lin and X. Tang. Inter-modality face recognition. In *Computer Vision–ECCV 2006*, pp. 13-26. Springer Berlin Heidelberg, 2006.

[18] R. Davis. Magic paper: sketch-understanding research. *Computer*, 40(9):34-41, 2007.

[19] E. J. Peterson, T. F. Stahovich, E. Doi, and C. Alvarado. Grouping strokes into shapes in hand-drawn diagrams. *Proc AAAI Conf on Artificial Intelligence*, 974-979, 2010.

[20] C. Lee, J. Jordan, T. F. Stahovich, and J. Herold. Newtons Pen II: An intelligent, sketch-based tutoring system and its sketch processing techniques. *Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pp. 9-17, 2012.

[21] T. Hammond and R. Davis. Tahuti: a geometrical sketch recognition system for UML class diagrams. *Proceedings of the AAAI Spring Symposium*, pp. 59-66, 2002.

[22] T. Hammond and B. Paulson. Recognizing sketched multistroke primitives. *ACM Trans Interact Intell Syst*, 1(1):1-34, 2011.

[23] L. B. Kara, L. Gennari, and T. F. Stahovich. A sketch-based tool for analyzing vibratory mechanical systems. *Journal of Mechanical Design*. 130(10), 2008.

[24] G. Costagliola, M. De Rosa, and V. Fuccella. Recognition and autocompletion of partially drawn symbols by using polar histograms as spatial relation descriptors. *Computers & Graphics*, 39:101-116, 2014.

[25] L. B. Kara and T. F. Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics*, 29(4):501-517, 2005.

[26] W. Lee, L. B. Kara, and T. F. Stahovich. An efficient graph-based recognizer for hand-drawn symbols. *Computers & Graphics*, 31:554-567, 2007.

[27] L. Fu and L. B. Kara. Recognizing network-like hand-drawn sketches: a convolutional neural-network approach. *ASME International Design Engineering Technical Conference*, 2009.

[28] L. Fu and L. B. Kara. From engineering diagrams to engineering models: visual recognition and applications. *Computer-Aided Design.* 43(3):278-292, 2011.

[29] T. Y. Ouyang and R. Davis. A visual approach to sketched symbol recognition. *IJCAI*, 9:1463-1468, 2009.

[30] L. Fu and L. B. Kara. (2011). Neural network-based symbol recognition using a few labeled samples. *Computers & Graphics*, Elsevier, Volume 35(5):955-966, 2011.

[31] M. Field, S. Gordon, E. Peterson, R. Robinson, T. Stahovich, and C. Alvarado. The effect of task on classification accuracy: using gesture recognition techniques in free-sketch recognition. *Computers & Graphics*, 34:499-512, 2010.

[32] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221-248, 1994.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*, 6(2):182-197, 2002.

[34] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput*, 3(4): 257-271, 199.

[35] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191(11):1245-1287, 2002.

[36] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema. Constraint handling in multiobjective evolutionary optimization. *IEEE Trans Evol Comput*, 13(3):514-525, 2009.

[37] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intelli*, 24(5):603-619, 2002.

[38] P. E. Duda and R. O. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., New York, 1973.

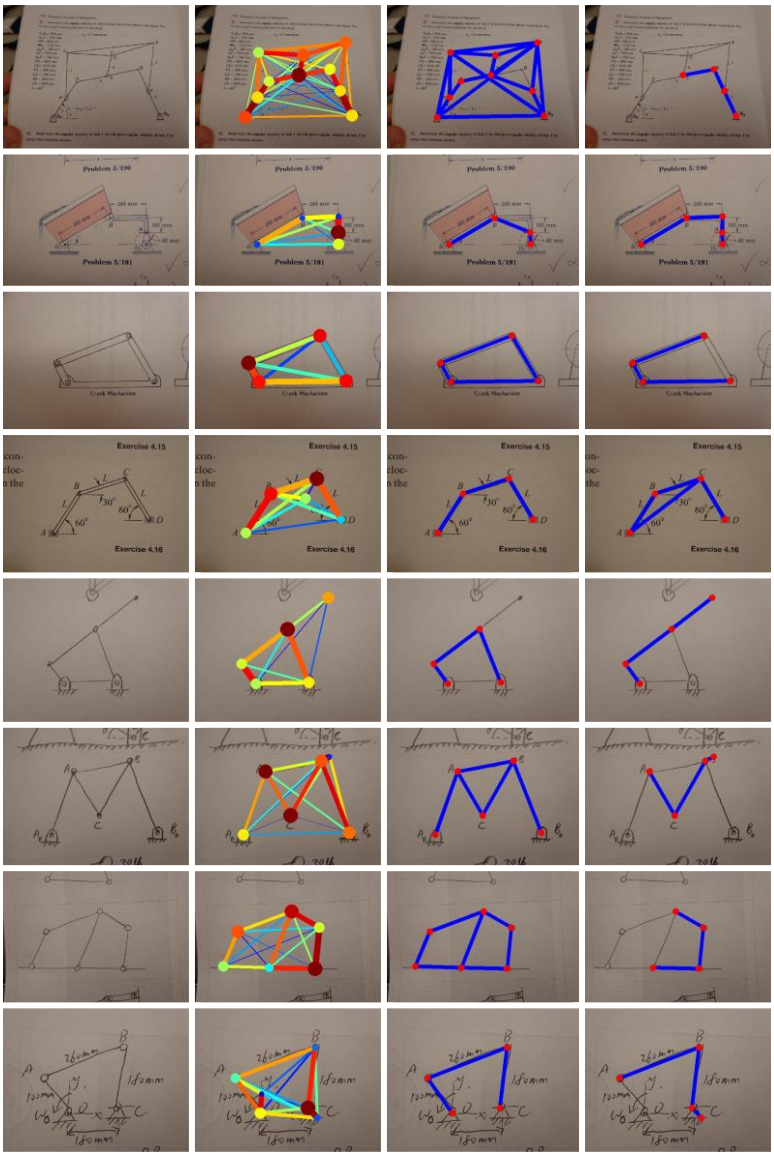
[39] J. Ginsberg. *Engineering Dynamics*. Cambridge University Press, New York, New York, 2008.

[40] D. J. McGill and W. K. King. *Engineering Mechanics: An Introduction to Dynamics*. Tichenor Publishing, Bloomington, Indiana, 2003.

[41] J. L. Meriam and L. G. Kraige. *Engineering Mechanics, Volume 2*. John Wiley and Sons, Inc., Singapore, 1993.

[42] J. L. Meriam and L. G. Kraige. *Engineering Mechanics: Dynamics*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2007.

[43] E. Soylemez. *Mechanisms*. Middle East Technical University, Ankara, Turkey, 1993.



(a) (b) (c) (d)

Figure 7. Example results on textbook and sketch images. (a) Raw images. (b) Strength of joint detections and pairwise connections; higher values indicated by darker (more red) color, thicker lines, and larger markers. (c) Correct solution found by the algorithm. (d) An incorrect solution on the Pareto front. All images depicted here were correctly solved, with the exception of the top row.