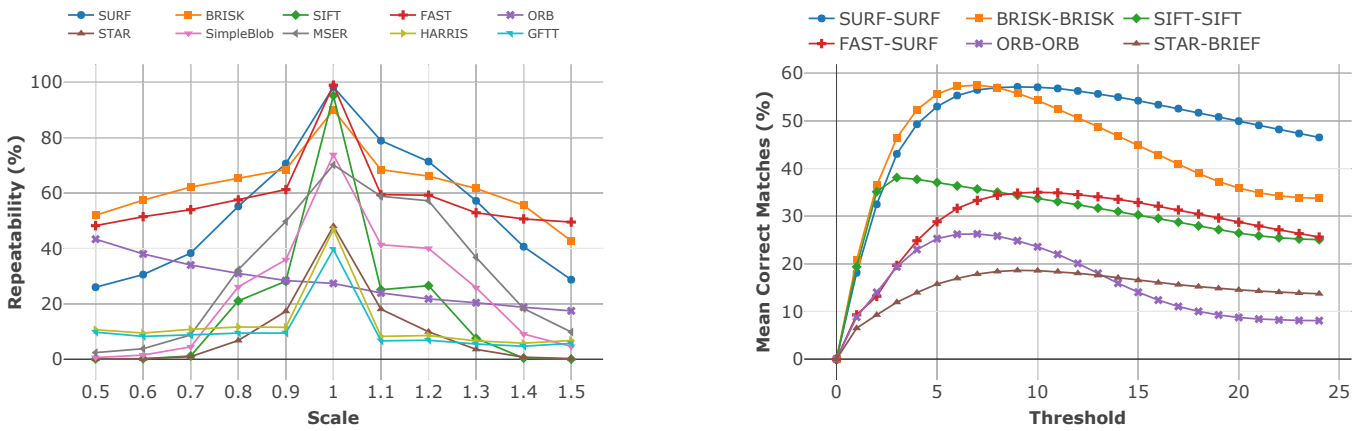

Figure 2. ***Repeatability*** **vs scales, and** ***Mean Correct Matches*** **vs thresholds of our dataset.**

static documents. In this section, we compare the performance of various feature matching algorithms against these needs.

Feature matching between two images A and B can be described as a 5-step process:

1. For each image, a first algorithm called a *Detector* is applied to extract a certain number of key regions (described by an x;y coordinate and a diameter).
2. Then, a second algorithm called a *Descriptor* takes the regions identified by a given detector and returns a *descriptive vector* of numbers describing each of these regions.
3. Next, either via brute force or a dedicated algorithm, the system measures a *distance* between the *descriptive vector* of each key region of image A and B (using Hamming or Euclidean distances), and returns a *list of matches* between both images (tuple, one for each image).
4. After that, only the *matches* with a *distance* below a specific *threshold* are kept in order to eliminate most incorrect matches. This results in a *list of associated regions*.
5. Finally, the system takes the *list of associated regions*, as well as the x;y coordinates of these key regions to identify an appropriate homographic transformation using the RANSAC algorithm [18].

Comparing feature matching algorithms

Feature matching algorithms are usually compared with a set of natural images using the following metrics:

Repeatability: Evaluates the performance of *detectors* (step 1 above) by measuring the proportion of similar key regions extracted from two images containing the same object [39].

Precision and recall: Evaluates the performance of a *detector-descriptor* pair (step 2 and 3 above) by measuring the number of key regions from a first image that are correctly associated to key regions from a second image [38].

However, already published comparisons of feature matching algorithms suffer from several limitations making them unhelpful to find the most adapted algorithm for Chameleon's needs. First, algorithms are tested on natural images and photos, while documents contain a wider variety of figures (e.g. state diagrams, photos, bar charts, and other data visualizations). Second, robustness to scale is seldom measured during evaluations of algorithms, whereas it is the main deformation that Chameleon faces since documents are displayed at different scales. Third, precision and recall are less adapted to compare detector-descriptor pairs when different detectors are tested: some detectors yield no key regions for harder-to-detect objects. Thus, descriptors associated with those detectors would be given an unfair advantage because they would not be tested on figures with harder-to-detect features. To address this issue, we use (instead of the precision and recall) a *Mean Correct Matches* metric that computes the mean percentage of correctly associated regions per image. This metric takes into account detectors yielding no key regions and decreases the overall percentage of correct matches accordingly.

Dataset

We evaluate the different feature matching algorithms on two datasets. The first is a *scientific papers* dataset introduced by Clark et al. [10], composed of 150 research articles from 3 different conferences (NIPS, ICML and AAAI) from 2008 to 2014. All figures in this document set were annotated manually. In order to add diversity to the figures, we gathered a second *presentation* dataset. 100 presentations were randomly selected in 10 different categories from the SpeakerDeck website [17]. We extracted from the first 20 pages of each presentation all images whose height and width were larger than 50 pixels. In total, the dataset comprises 1660 figures from 2741 pages. Each of these figures was matched against the PDF page containing the figure, rasterized at 72 DPI. To evaluate the influence of scaling on the results of feature matching algorithms, we applied a scale transformation to the rasterized PDF pages. Tested scales were comprised of every 0.1 scaling step in [0.5, 1.5], and the resizing was based on a bilinear interpolation, the technique observed in Adobe Acrobat Reader DC version 2018 on macOS Sierra running OS X version 10.12.6. Knowing the position and the size of the figure in the scene, we compute as ground truth the homography relating a figure to its PDF page.