2D Matching Using Repetitive and Salient Features in Architectural Images

Brittany Morago, Giang Bui, and Ye Duan

***Abstract*— Matching and aligning architectural imagery is an important step for many applications but can be a difficult task due to repetitive elements often present in buildings. Many keypoint descriptor and matching methods will fail to produce distinctive descriptors for each region of man-made structures, which causes ambiguity when attempting to match areas between images. In this paper, we outline a technique for reducing the search space for matching by taking a two-step approach, aligning pairs one dimension at a time and by abstracting images that originally contain many repetitive elements into a set of distinct, representative patches. We also present a simple, but very effective method for computing the intra-image saliency for a single image that allows us to directly identify unique areas in an image without machine learning. We use this information to find distinctive keypoint matches across image pairs. We show that our pipeline is able to overcome many of the pitfalls encountered when using traditional keypoint and regional matching techniques on commonly encountered images of urban scenes.**

***Index Terms*— Keypoint matching, 2D registration, repetitive patterns, salient features, architectural imagery, dimension reduction.**

I. Introduction

REGISTERING architectural imagery poses several interesting challenges. Many architectural designs include repetitive elements such as rows of windows, doors, or balconies that all have similar appearances. If either local or regional descriptors are extracted for different elements of a repeating patten, they may take on near-equivalent values. This creates ambiguity when the descriptors from a series of images are compared. A matching algorithm being used to find the correspondence for a window in one image may not be able to distinguish between the descriptors of several repetitive windows in a second image. Figure 1 shows an example of how a common local keypoint descriptor and matching technique, SIFT [1], cannot distinguish between different areas of a building that look nearly identical.

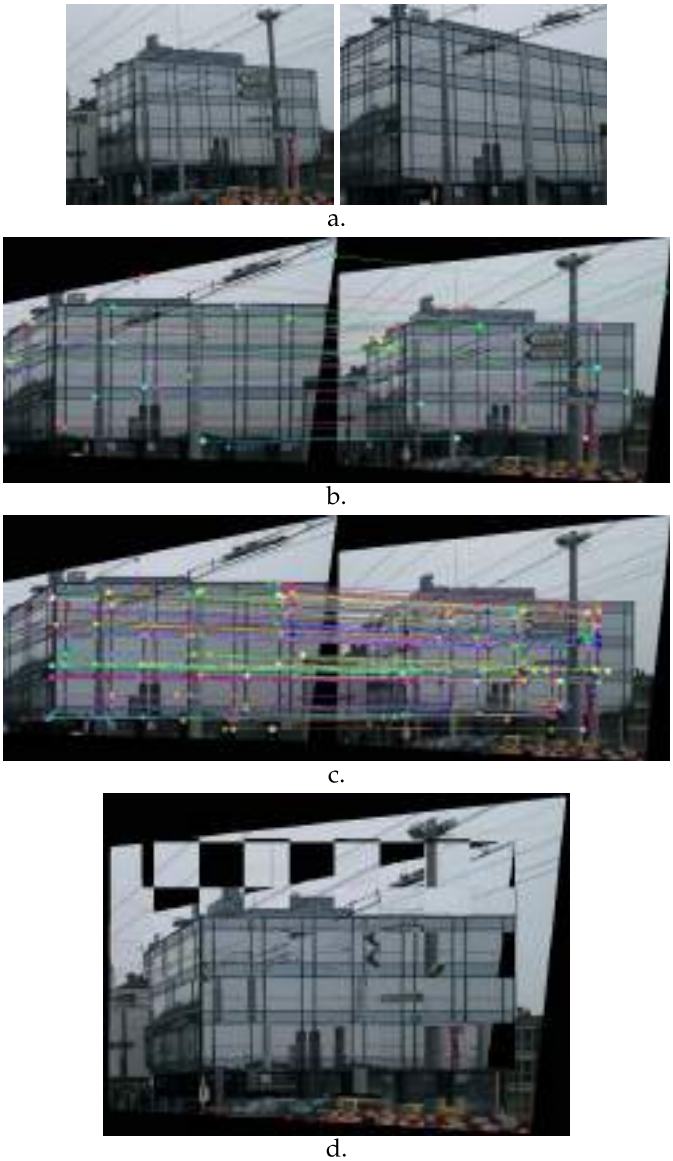

Fig. 1. Challenge of matching images containing repetitive features. (a.) Original image pair. (b.) SIFT keypoint matches on rectified images. Close inspection shows that the majority of the matches are incorrect and keypoints in the left image are matched to similar looking repetitive areas in the right image. We do not have enough correct SIFT matches for RANSAC to find a correct transformation. (c.) Keypoint matches after applying our proposed pipeline for handling repetition. (d.) Image pair aligned and overlaid using our pipeline.

We hypothesize that by using specific urban imagery-based criteria to reduce the search space during matching, we can bypass the difficulties encountered when matching

Manuscript received December 22, 2015; revised May 25, 2016 and July 19, 2016; accepted July 26, 2016. Date of publication August 8, 2016; date of current version August 29, 2016. This work was supported in part by NSF CC-NIE under Award 1245795, in part by NSF CMMI under Award 1039433, in part by the NSF Graduate Research Fellowship under Award 0943941, and in part by the U.S. Department of Education under Award P200A100053. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang-Su Kim. *(Brittany Morago and Giang Bui equally contributed to this work and are co-first authors.) (Corresponding author: Ye Duan.)*

B. Morago is with the University of North Carolina at Wilmington, Wilmington, NC 28403 USA (e-mail: moragob@uncw.edu).

G. Bui and Y. Duan are with the University of Missouri, Columbia, MO 65203 USA (e-mail: gdb338@mail.missouri.edu; duanye@missouri.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes additional results. The total size of the videos is 9.33 MB. Contact duanye@missouri.edu for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2598612

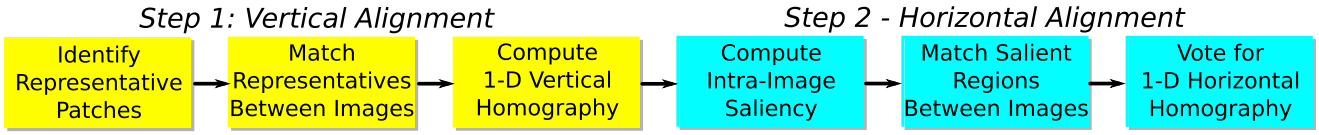

Fig. 2. Pipeline of our proposed registration method. Our method has two main stages: vertical alignment and horizontal alignment. For vertical alignment, we perform dimension reduction, identify representative patches for each image, and use the matching representatives to compute a 1D transformation that transforms the images to the same scale and aligns the pair along the vertical axis. For horizontal alignment, we compute the intra-image saliency and search for strong, salient matches along corresponding rows between images to vote for the correct displacement between the pair.

ambiguous data. We have observed that regardless of the height of a building, architectural repetition often occurs at least in the horizontal direction, if not also in the vertical direction [2]. Since we are focused on images of man-made structures that frequently have easy to identify vanishing lines, we rectify all images to be front facing [3], [4] so that horizontal repetitive elements have the same y-coordinate. Once images are in this front-facing format, we propose aligning images one dimension at a time following the idea that considering subsets of larger dimensional data independently can make an ambiguous global problem simpler to solve [5]. We first collapse each image along the vertical axis and use the y-coordinates of matching features to vertically align the image pairs. Once we know the relative scale and row alignment across images, we expand the data back out along the x-axis and compute the horizontal alignment.

To perform dimension reduction to compute a vertical alignment, we create an abstraction of each image by identifying the repetitive elements along the horizontal axis and choosing just one element to represent each group of repeated features and combine these with any found unique features. By choosing one representative patch for each row of repeated horizontal elements as well as identifying salient patches that do not repeat across the image, we can abstract an image into a vertical column of distinct regions and remove a substantial amount of potential repetition in the feature space. We match these representative regions across images to compute a pair's row-to-row correspondence.

This simplifies the second alignment stage since our only remaining unknown is the displacement along the horizontal axis. At this stage, we have a reduced search space to find distinctive matches that can be used to align repetitive facades. For a match to be distinctive, the matching regions must be unique within their own images. Using our assumption that the data can be rectified to be front-facing and vertically aligned, we propose a simple, but very-effective technique for computing the intra-image saliency[1] along the rows of urban images. We use this information to gauge the quality of keypoint matches. We are able perform this computation directly on the images being matched without relying on the existence of a large database of similar images and the use of learning techniques to determine the uniqueness of keypoints being matched. Our pipeline is outlined in Figure 2.

II. Literature Review

Several groups have developed techniques to use different types of structural and high-level information to guide image matching. The goals of these methods are generally to overcome pitfalls of locally or even regionally defined descriptors [1], [6]–[10] that are ambiguous in the presence of repetitive features. While locally defined descriptors can provide very accurate information that is invariant to scale changes and scene clutter, in certain cases they do not take into account large enough image regions to be truly distinct [11].

To make matching images with repetitive architectural patterns a more tangible task, researchers have explored using larger-scale properties such as feature symmetry [12], [13] and repetition to identify corresponding regions [14]. Hauagge and Snavely [15] search for horizontal, vertical, and rotational symmetries about various axes and scales across images. Self-similarities can be identified in patterns of colors, edges, and repeated visual elements by measuring how similar a small region is to parts of the larger surrounding region [16]. Wu *et al.* [2] match SIFT features within an image to identify repeating and symmetrical elements that occur at regular intervals and the boundaries between repetitions. Kushnir and Shimshoni [17] also match SIFT features across images to find repeating elements and use agglomerative clustering to group similar regions. This group presents two versions of their matching algorithm. Similar to [2], one version focuses on images where grids of periodic elements can be identified. The other handles cases where repetition exists in a less predictable fashion. Matches between non-repetitive features are identified and used to solve a transformation between an image pair. Chung *et al.* [18] use graph matching to globally align images of buildings that capture the same portion of a structure's facades. This work identifies repetitive MSER patches and constructs a semantic sketch of a building to match between highly varying viewpoints. Bansal *et al.* [19] also work to align images of buildings taken from highly different viewpoints such as aerial to ground level. Aligning images of high-rise buildings that have many repetitive elements and appearance differences due to view changes poses difficulties. They construct an embedding of a facade that captures how similar or dissimilar image regions are to their neighbors. This group uses the periodicity of a repeating element as an estimate of the scale of interest. This group's contribution of a "scale-selective self-similarity" can be extracted from any pixel in the image bypassing the need for a repeatable detector. PatchMatch [20], [21] provides a framework for matching patches across images by randomly assigning and searching for correspondences. This method uses a combination of searching through different scales, random sampling, and distance propagation to guide the matching process using a cooperative hill climbing strategy

[1]We use the term saliency to refer to image regions that do not look like any other regions within the image.

to find a patch's set of nearest neighbors. Ceylan *et al.* [14] construct grids of repetitive elements within images and limit an aligning transformation to be one of a discrete set of possible solutions that overlay different images' grids.

A different perspective for matching data that has repetitive elements is to find the salient regions in images and use these areas to guide registration. Machine learning techniques have been used to identify these types of unique regions in photographs, paintings, and 3D models [22]–[24]. These mid-level discriminative features ideally occur often enough across a dataset to be learned, but are different enough from the rest of the imagery to be clearly located in photographs and models with visual dissimilarities. Training a system to learn these features tends to be time consuming and computationally intensive. In this paper, we share a similar spirit of finding salient regions. However, unlike existing work that requires supervised training, our proposed salient region detection method is applied directly to and only on the images being matched and is unsupervised.

III. Contributions

Our main contributions to the area of architectural imagery registration include methods for:

1) aligning rectified images one dimension at a time, reducing the search space and increasing the image matching robustness,
2) computing intra-image saliency maps in an unsupervised manner with no training set, and
3) directly accounting for image region saliency when quantifying keypoint match quality.

Our pipeline is designed to overcome several of the limitations and assumptions made in the work described in Section II. We use a technique for identifying repetitive patterns in images regardless of the spatial distribution of similar elements and we are able to match images of repetitive facades that only partially overlap. We also remove the ambiguity in the feature space that is common in architectural imagery by collapsing repetitive elements into single representative patches. Finally, instead of only using either repetitive or salient features for matching, we take advantage of all the information available in images, and incorporate both types of features in our registration pipeline.

IV. Two-Step Approach

The main steps of our registration methodology are identifying both repetitive and salient elements within images, using this information to first align image pairs vertically, and finally computing a horizontal alignment using a dramatically reduced and unambiguous search space. The first step of this process is solving for a 1D affine transformation that encodes the relative scale change and vertical shift between images and matches rows to rows as shown in Figure 12c. The second step is determining a second 1D transformation that represents the horizontal translation overlaying the image pair. Equation 1 shows the forms of these vertical and horizontal transformations (T_v and T_h) where s is the scale change, d_y is the displacement along the y-axis, and d_x is the displacement along the x-axis.

$$T_v = \begin{bmatrix} s & d_y \\ 0 & 1 \end{bmatrix} \quad T_h = \begin{bmatrix} s & d_x \\ 0 & 1 \end{bmatrix}. \tag{1}$$

A. Vertical Alignment

We begin by performing dimension reduction and selecting one representative patch of each type of region in an image so that all the chosen representative patches are distinctive. This entails identifying portions of an image that are duplicates of each other and represent a repeating pattern as well as image portions that are truly unique. We use a combination of local and regional descriptors to accomplish this. Doing so helps overcome some of the inherit ambiguity of very-locally defined descriptors [10], [11] and increases our confidence as we categorize regions. Incorporating regional descriptors at varying scales also allows us to explore how different image regions may switch between repetitive and salient as the considered area changes.

To initially identify repetitive elements, we extract Psuedo Corners [10] which are sparse, repeatable local keypoints that are found at the intersection of line segments. Our design choice of using Pseudo Corners as anchor points for finding repetition and matching images is discussed further in Section V-B3. We extract multi-scale patches centered at each anchor keypoint and use local SIFT and regional HOG descriptors [10] to group repetitive elements that are centered on the same image row. For each group, we use the element that is most similar to the group's mean patch as the representative patch. Figure 3 provides a visual example of how we identify repetitive elements in an image and choose a group's representative patch. We use the y-coordinate of these representative patches to solve for T_v.

When we match just the representative patches of groups (that may have either one or multiple elements), we can focus on matching unique areas without worrying about horizontal ambiguity. To actually match representative patches, we again use the SIFT descriptor at the center point of each patch and the HOG descriptor of the entire representative region. For each representative patch in I_1, we compare its SIFT descriptor to all of the representative patches in I_2. If the patch in I_2 with the most similar SIFT descriptor to the patch in I_1 also has Euclidean distances between the center points' SIFT descriptors and the patches' HOG descriptors below a threshold (we use 0.5), we label this as a group match. We consider all of the representative patches in I_1 and I_2 that were constructed at different scales in one round of matching. Figures 3 and 12b show examples of matching groups between an image pair.

Using RANSAC [25], we randomly choose sets of two matching representative patches between the image pair to compute a series of potential T_v's. For each T_v, we check how many other representative patch matches have a symmetric transfer error [3] lower than $0.5 * winSize_1$ and choose the T_v with the largest support. We then re-estimate T_v using this set of inliers.

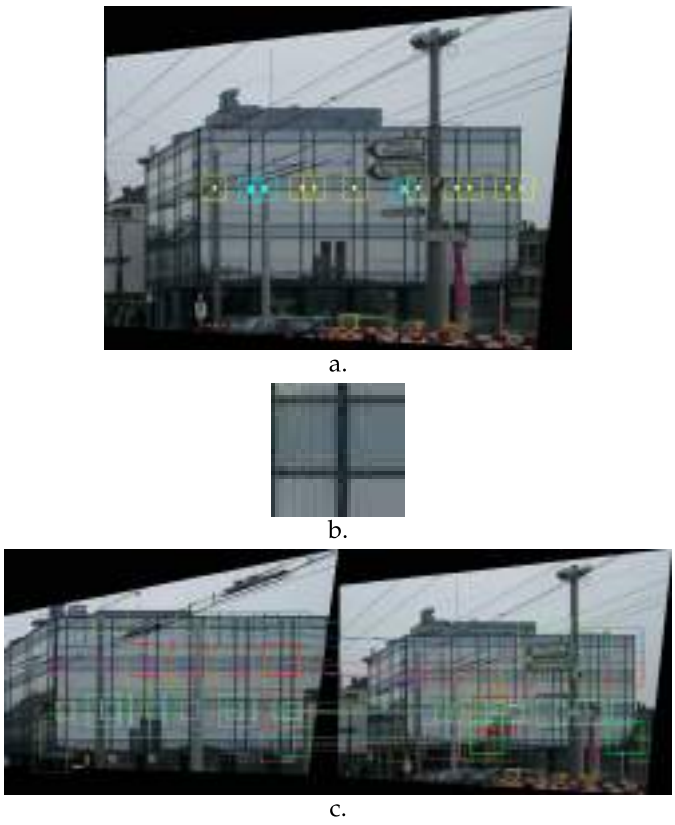

Fig. 3. Identifying groups of repetitive elements and matching groups. (a.) Similar patches are identified along a row of an image. Teal boxes surrounding a larger dot show the matching anchor keypoints. Yellow boxes surrounding a smaller dot show the dense features that were matched to the anchor keypoint for the row. For this example, we use Pseudo Corners as anchor keypoints as is discussed in Section V-B3. (b.) The chosen representative patch of the group. This patch is most similar to the average of all the elements contained in the group. Note that sometimes in architectural designs, the repetitive elements may not occur at regular intervals as shown in this example. Our method is designed to search for repetition regardless of the regularity or irregularity of the interval. (c.) Subset of the corresponding groups after computing a 1D transformation are highlighted in matching colors. Every element of the groups are displayed.

B. Horizontal Alignment

Since we are working with data containing repetition, especially images with patterns along the horizontal direction, we need to identify matching areas where we can be sure the correspondences are non-ambiguous to compute d_x. This is especially important because we do not require that our image pairs contain completely overlapping facades, so we can not rely on the relative arrangement of repetitive elements to determine the horizontal alignment. Figure 4 shows an example of the problem of relying on the relative arrangements of repetitive elements to align images. Our goal is to find matching pixels (p_1 in I_1 and p_2 in I_2) that satisfy the following criteria:

1) p_1 and p_2 have very similar HOG descriptors
2) p_1's closest match within I_1 (q_1) is very dissimilar to p_1
3) p_2's closest match within I_2 (q_2) is very dissimilar to p_2

These conditions help us ensure that we find matching pairs of pixels that are both very similar to each other, but are distinct within their respective images, making the match distinct. We have two steps for identifying salient matches.

Fig. 4. Motivation for searching for salient matches for horizontally aligning images. The highlighted image patches in the above image pair shows a set of matching repetitive elements. If we choose the horizontal alignment that maximizes the number of overlapping elements, we will have the wrong image pair transformation in situations like this where the images do not fully overlap. The highlighted patches are colored according to which patches will match if we try to maximize the number of overlapping patches. This is why we use our reduced search space (after vertical alignment) to identify distinct areas and salient matches to choose potential horizontal alignments.

The first is to compute an intra-image saliency map to identify distinct areas in a single image. The second step uses this information to compute the pairwise saliency of each match.

1) Intra-Image Saliency Maps: The goal of our intra-image saliency maps is to display how salient each pixel and its surrounding region in an image are compared to other regions in the image. Since at this point in our pipeline, for a given pixel in I_1, we are only looking for its match on a single row in I_2, we are only concerned with how similar each region is to all other regions centered on the same row. This reduces our intra-saliency computation by requiring that we only compare each region only to the other non-adjacent regions on the same image row. We consider regions whose HOG windows do not overlap to be non-adjacent. Equation 2 shows how we compute each pixel's saliency. We use s_{p_i} to denote the saliency of a pixel, p_i, and its surrounding patch. q_i is the pixel on the same row as p_i whose surrounding HOG patch has the lowest L^2 distance to the HOG patch surrounding p_i and is not adjacent to p_i. i denotes the image index.

$$s_{p_i} = \arg\min_{q_i} \|HOG(p_i) - HOG(q_i)\| \tag{2}$$

Using this equation, we gauge the saliency of each pixel and its surrounding region by the distance between its HOG descriptor and its closest match on the same row. Figures 5b and 12d show visualizations of the range of intra-image saliency values found in each image. We compute the saliency at multiple scales or HOG window sizes as shown in Figure 12d to consider the saliency of different image region sizes. The ranges shown in these figures are colored with a standard jet colormap where the minimum values in the image are assigned a blue color, the maximum values are assigned a deep red, and the values in between interpolate through blue, green, yellow, and red. The specific colors chosen for visualizing the range of values are arbitrary. We use them as a tool to easily study the numeric values of how salient each region in the image is. We can see in these examples how the more distinctive features and objects that are only seen once in an image stand out from the repetitive elements.

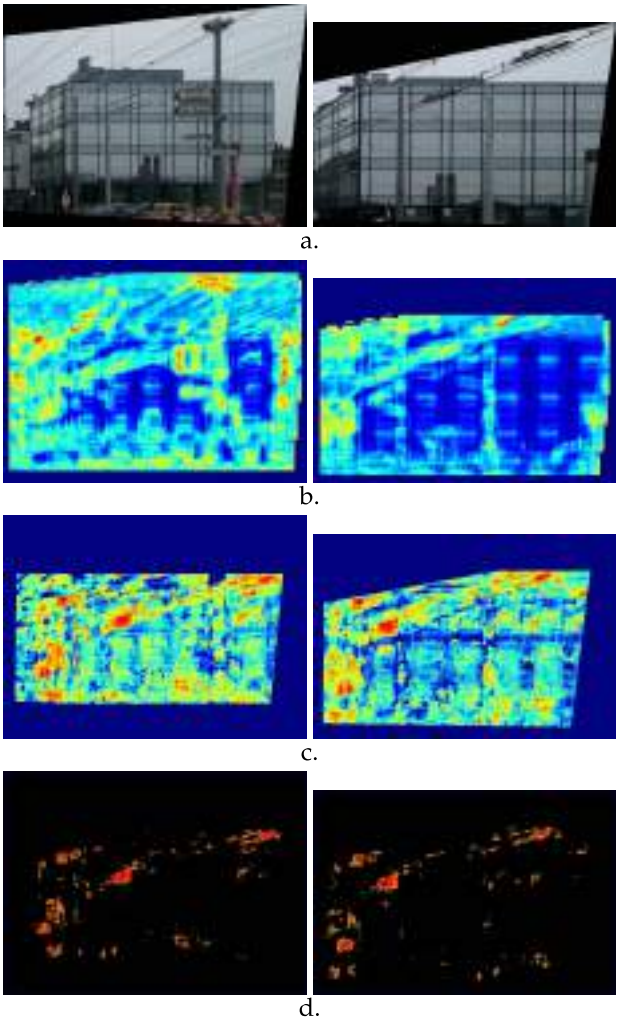

Fig. 5. Visualization of match saliency equation. (a.) Original images being matched. (b.) Intra-image saliency map of each image. More salient areas appear in red and more repetitive areas appear in blue. (c.) Color fields of match saliency scores shown from the perspective of both images. (d.) Result after applying Otsu thresholding. All keypoint matches that are contained within the remaining colored regions are used to vote for a hortizontal alignment. The final aligned result using these matches is shown in Figure 1.

2) Pairwise Match Saliency: Once we know how salient each pixel is within its own image, we can use this information to gauge how confident we are in the saliency of keypoint matches found between the image pair. We define this value, m_s, in Equation 3. $HOG(p_1)$ is the HOG descriptor of a pixel, p_1, in I_1. We compare p_1 to the HOG descriptors of all pixels lying on its corresponding horizontal line in I_2. $HOG(p_2)$ is the HOG descriptor surrounding a pixel on this line that is most similar to $HOG(p_1)$. We enforce a bi-directional constraint on the matches here, meaning that p_2 is the best match for p_1 in I_2 and p_1 is the best match for p_2 in I_1. ϵ is a very small value to ensure both that m_s does not go to infinity and that a perfect match that is also repetitive does not have a high saliency score.

$$m_s = 1 - \frac{\|HOG(p_1) - HOG(p_2)\| + \epsilon}{\|s_{p_1} + s_{p_2}\| + \epsilon} \tag{3}$$

If either p_1 or p_2 are elements of a repetitive pattern within I_1 or I_2, we cannot be confident that p_1 and p_2 are a unique and correct match. For this reason, we use the intra-image saliency scores s_{p_1} and s_{p_2} of both p_1 and p_2 respectively. We are confident in keypoint matches with high m_s values since this indicates that the matching pixels have similar descriptors and that they are both distinct within their respective images, leaving little room for ambiguity in the pair match.

To distinguish the pixel matches that we are confident in from the ones that are either poor matches or ambiguous, we create an image visualizing the range of m_s values in the same manner we used to visualize the intra-image saliency values. The color map of m_s values for different image pairs are shown in as shown in Figures 5c and 12e. We apply Otsu thresholding [26] to identify the "foreground" pixels in this image, or those whose saliency values stand out from the rest of the image. Figures 5d and 12f show the saliency match fields we are left with after applying this thresholding. We choose to use Otsu thresholding since it is an automatic method that is adaptable to different types of images and does not require that we set one threshold value for all datasets.

We use the pixel matches remaining in the pairwise saliency maps after thresholding to vote for T_h, with each match voting for a potential d_x value. Each match's vote is weighted by it's m_s score so that the most salient matches have a stronger voice in choosing the horizontal alignment.

We can encounter a potential obstacle during the horizontal shift voting stage if the scene in the image actually contains multiple planes that cannot be represented by a single transformation. For instance, as shown in Figure 12h., if an object, such as a tree, is in front of the main building being photographed, one transformation might align the two views of the tree captured between images, while a second transformation will align the building of interest. Since, in general, foreground items like this also tend to be salient, we may get a large number of pixel matches voting for the shift to align the salient foreground as well as a strong support for aligning the salient features on the more repetitive building facades.

To handle this scenario, we consider several different potential d_x values. We have observed that the horizontal shift that aligns the salient foreground object does not also align the repetitive elements on the facade of interest, except by coincidence. Using this observation, if multiple d_x values have strong support, we choose the one that best aligns the repetitive facades in the images. We have already identified the image regions showing building facades when we extracted and matched groups of repetitive elements in Section IV-A.

To decide how many d_x values have strong support, we consider all shifts that have at least 50% of the number of votes as the shift with the most votes. We use each of these shifts to project all the members from the repetitive groups extracted at the beginning of our pipeline in I_1 to I_2 and vice versa. For all of the members that are projected within the matching image's plane, we maintain a counter. The counter is incremented every time a repetitive element is projected onto the location of a member of its matching group. If there is no matching group member at an element's projected location, the counter is decremented. After projecting all the group members between images, we divide the counter value by the

number of group members that were projected within their matching image's boundaries. We choose the horizontal shift which has the largest counter score. An example of using this criteria to choose the horizontal shift that best aligns a building facade with salient objects in front of it is shown in Figure 12h.

3) Full 8 Degree of Freedom Transformation and Refinement: At this point we have two 1D transformations that can be merged into one 3x3 homography that encodes the scale change between the image pair and the translations along the x and y axes. If both images have been perfectly rectified, this is all the information we need to align them. However, since we are working with real-world data and applying an automatic rectification algorithm to large sets of images, there is naturally still some warping and distortion in the rectified results. To handle these effects, we can switch to a traditional 8 degree of freedom (DOF) homography at the end of our pipeline, using the original two 1D transformations as an initial alignment estimate. We refine our transformation [10] using the iterative closest point algorithm [27], switching to a full homography in the process.

V. Evaluation and Discussion

A. Experimental Setup

We have evaluated our pipeline on the ZuBuD building dataset [28] and two datasets containing symmetrical elements in urban scenes [14], [15] which we will refer to as the SymFeat dataset and the SymUrban dataset respectively. All of these datasets consist of a number of photographs of buildings under changing photographic conditions. In the ZuBuD dataset, there are five images of each building with each image varying in some aspect such as perspective, scale, or rotation. We used the software provided by [4] to rectify the images in the ZuBuD and SymUrban datasets. The majority of the images in the SymFeat dataset are already rectified and front facing. For the ZuBuD dataset, using the images that were automatically rectified correctly, we have 716 pairs of images for testing. The SymUrban dataset consists of 9 image sets, each focused on a single building. There are around 30 photos of each view. From the subset of images that were correctly rectified, we chose about 50 image pairs that represented all of the view changes presented by the dataset. We use selected images with the image pairs in the SymFeat dataset that had reliable ground truth information to test our method to see if we can overcome some of the common challenges of aligning urban images. We ran SIFT, the contextual regional matching method described in [10] on all these datasets. We chose these methods for comparison because both have been shown to be robust under changes in photographic conditions, but due to their relatively local nature (compared to using some sort of grouping and image abstraction), can be confused by repetition ambiguity in images. We also compare our method to the symmetrical feature-based methods described in [15] to look at other approaches specifically designed for urban imagery. A list of the names and thumbnails of the SymFeat and SymUrban images we used as well as the especially challenging images from the ZuBuD dataset are provided in the Supplementary Material. To obtain ground truth for the ZuBuD and SymUrban images, we manually selected a set of four matching points between image pairs to compute an aligning homography. We use these ground truth homographies to quantitatively evaluate our results in Figures 6-10 and Table I.

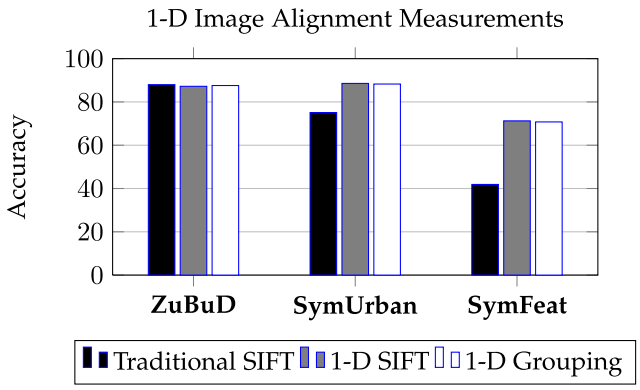

Fig. 6. Accuracy measures for aligning images vertically, which is the first stage of our two-step pipeline. We show the results on our three datasets of using the traditional SIFT method, only the y-coordinates of the SIFT matches (1D SIFT), and using our proposed method for vertical alignment (1D Grouping).

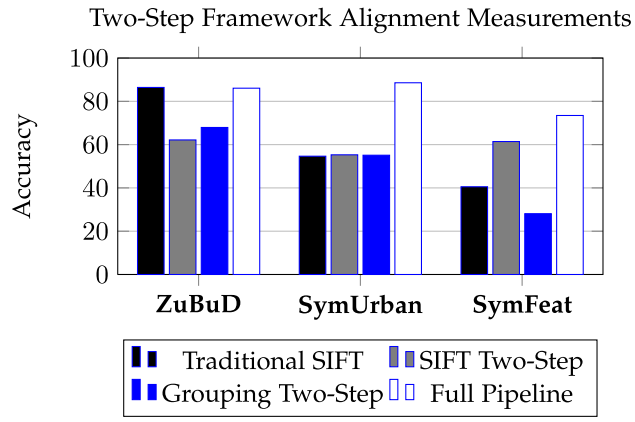

Fig. 7. Accuracy measures for aligning images both vertically and horizontally. We show the results on our three datasets of using the traditional SIFT method, using SIFT matches to align the images first vertically, then horizontally (SIFT Two-Step), our proposed method for vertical then horizontal alignment (Grouping Two-Step), and our full pipeline.

B. Results

We ran a variety of tests to evaluate both different design choices in our pipeline and how they build on each other as well as the overall effectiveness of our proposed methodology. Our first set of tests looks at the contribution of breaking down the transformation estimation into two steps, focusing on aligning one dimension at a time. We next look at how matching accuracy increases as we progress through the steps of our pipeline, adding in the intra-image saliency and pairwise match saliency computations to the two-step framework. Included in our results is also an investigation of how the type of anchor point used for vertical matching impacts our results.

For all of our alignment accuracy tests, we compute a transformation matrix for each tested method. We translate

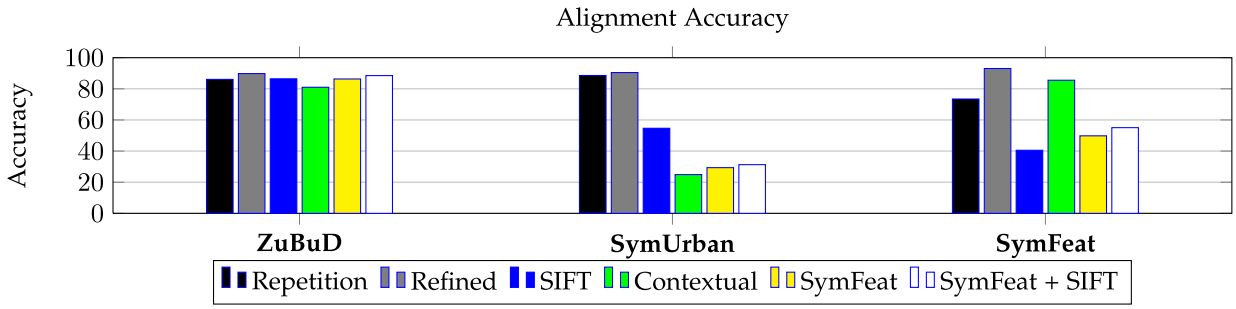

Fig. 8. Accuracy measures for fully aligning images using our complete pipeline compared to state-of-the-art methods. We show the results on our three datasets of using our proposed method (Repetition), our proposed method with refinement (Refined), the traditional SIFT method, the contextual method described in [10], the two vartions of the symmetry-based features method described in [15].

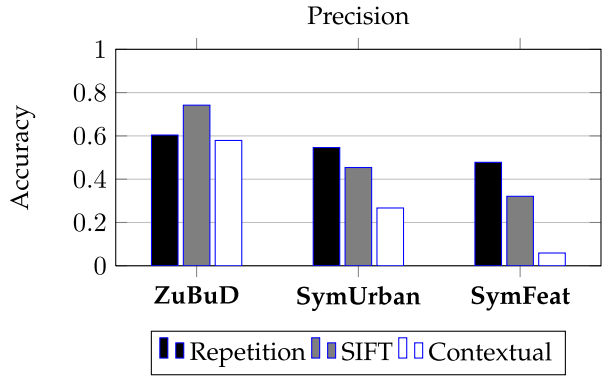

Fig. 9. Precision measurements of matches found in the keypoint matching methods tested and shown in Figure 9. We show the results on our three datasets of using our proposed method (Repetition), the traditional SIFT method, and the contextual method described in [10].

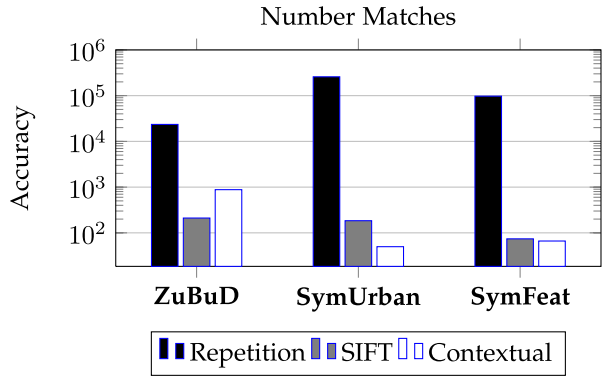

Fig. 10. The number of matches found with the keypoint matching methods tested and shown in Figures 9 and 8.

every pixel in I_1 to I_2 using both the ground truth homography and the test transformation. We compute the percentage of pixels that are transformed to the same coordinate. To allow for rounding errors and small inaccuracies in the ground truth data, we set a threshold on what the 2D distance between projected pixels can be to label a pixel as having been transformed correctly. This threshold is 0.6% of $max(image_{width}, image_{height})$ which is the same threshold used on 2D symmetrical transfer errors for identifying inliers in [29].

TABLE I

REPEATABILITY COMPARISON ON ZUBUD DATASET

Pseudo Corners	SIFT	Harris-Laplacian
28.645	13.803	10.885

After running these tests on the full datasets, we take the average values and report them in Figures 6-10 to provide an overview of the trends in the results. The Supplementary Material contains more detailed results and shows the evaluation for individual image pairs. In the Supplementary Material, since it would be unintuitive to show quantitative results for the over 700 image pairs from the ZuBuD dataset, we focus on about 50 image pairs that were identified as especially challenging because of the amount of architectural repetition present in these specific scenes. These images pose a number of problems for state-of-the-art techniques and we show how our pipeline can overcome these challenges. In the following tables, however, since we are showing average scores for each method on each dataset, we have included the results for the full ZuBuD dataset. The additional 650+ images included from the ZuBuD results in the main paper do not necessarily contain repetition or any specific challenges for the state-of-the-art methods. Many of the buildings in these additional images have different types of features such as curved walls and domes that prevent the full building facades from being correctly rectified. Many of these images also do not require or fit the repetition identification and two-step alignment methods described in this and other repetition identification papers. Therefore, we encourage the reader to also study the results in the Supplementary Material that focus on the types of images this work is intended to benefit.

1) Two-Step and Saliency Computation Evaluation: We begin by performing tests to study the usefulness of our proposed two-step method for image alignment. We show that by aligning a single dimension first, instead of registering the full coordinate systems of an image pair, we can achieve a higher accuracy along that dimension. First, we aligned the test data using traditional SIFT matches with the 8-point algorithm [3] and RANSAC [25] to compute a full 8 degree of freedom homography. We then check the 1D error of the matches along the y-axis. Then we use the y-coordinates of the SIFT matches to compute a 1D alignment (as shown in Equation 1) and again check the error along the y-axis. Our third test is to check the accuracy along the y-axis of

the 1D transformation computed using the grouping method described in this paper. Fig. 6 shows the results of these tests on our three datasets. Both of the 1D alignments generally yielded better results than the traditional 8 DOF computation across the test data. The benefit of our grouping method over both of the SIFT-based tests is seen especially on image pairs that have large scale, rendering, or lighting changes such as those encountered in the ZuBuD and SymFeat datasets.

Next, we look at what happens when we get to the second stage of the two-step approach. Fig. 7 shows results after using the SIFT-based and grouping-based techniques described above in combination with different methods for determining the horizontal alignment. We first align the images vertically (using SIFT features or grouping features) and then match SIFT features only along corresponding rows. In both tests, distinctive SIFT matches are used to vote for a horizontal alignment. We can see that this minimal version of the two-step approach tends to outperform the traditional technique of solving a full 8 DOF matrix on challenging, repetitive images. We also show the results of using our full pipeline, consisting of the two-step approach in combination with the intra-image saliency and pairwise match saliency computations. The general trend is that the basic two-step approach outperforms traditional SIFT matching and our full pipeline outperforms the basic two-step approach, showing the strength of our full pipeline. These results are shown in Fig. 7.

2) Overall Pipeline Evaluation: To further test the overall effectiveness of our full pipeline, we compute transformations for aligning the test image pairs using our pipeline, SIFT keypoint matches, contextual keypoint matches using the method discussed in [10], and symmetry-based features as described in [15]. Following the symmetry-based features paper [15], we tried matching the images using only symmetry descriptors and symmetry and SIFT descriptors concatenated. The alignment accuracies for these methods are shown in Figure 8 Top. We also look at the accuracy of the keypoint matches obtained with each of the first three test methods, the number of matches obtained with each method, and the percentage of these keypoint matches that are correct. Figure 10 shows the number of matches we obtain for each method and Figure 9 shows the precision of these matches. We can see from all of these results that our proposed pipeline tends to outperform the comparison methods. We do have several cases where the refined result has a significantly higher accuracy than the pre-refined result. Often times in these cases, the rectified images still have a small amount of warping which is not captured in our 1D homographies. It takes the full 8 DOF transformation to model these distortions. Given the fact that our horizontal alignment stage takes into account dense matches in salient regions of the image pairs, as we saw in Figures 5 and 12, our pipeline also tends to have a dramatically higher number of keypoint matches in the end than our comparison methods. Despite the high number of matches we consider, they are comparably more precise to matches from other methods.

We show several visual examples of the result of our pipeline for qualitative evaluation in Figure 11. The images shown here are a sampling of the different types of data we handle. These image pairs have varying ratios of salient and repetitive regions. Several of the pairs also have changes in rendering style, time captured, scale, and lighting, all image characteristics that are known to cause challenges for many traditional keypoint matching methods [7]. Our method is able to handle many of these problems on top of the issues presented by repetitive data.

TABLE II

Average Runtimes in Seconds AND *Number of Features* FOR STAGES OF OUR PIPELINE ON ZUBUD DATASET

Pipeline Steps	-
Average Image Dimensions	*628 x 457*
Line Segment Extraction (Section 5.2.3)	***0.5115***
# Line Segments	*1313*
Pseudo Corner Extraction (Section 5.2.3)	***1.2838***
# Pseudo Corners	*566*
SIFT Descriptor Extraction (Section 4.1)	***8.2625***
HOG Extraction (Section 4.1)	***3.9168***
# HOG Patches	*141493*
Repetition Identification (Section 4.1)	***11.9155***
# of Groups	*259*
Group Matching (Section 4.1)	***8.1449***
Vertical Alignment (Section 4.1)	***0.5165***
Intra-Image Saliency Computation (Section 4.2)	***17.5684***
Saliency Match Computation (Section 4.2)	***11.9831***
Horizontal Alignment/Voting (Section 4.2)	***0.0165***
Refinement (Section 4.2.3)	***5.9628***

3) Anchor Keypoint Evaluation: To identify repetitive and salient regions for initial matching, as was discussed in Section IV-A, we need anchor keypoints to use for searching for and comparing regions throughout an image. There are a number of feature extraction methods that can be used for this purpose such as SIFT or Harris-Laplacian corners. Since most architectural images have abundant linear features that capture a great deal of a building's structure, we take advantage of this information to find anchor keypoints. We intersect line segments extracted from these images to estimate corner locations, which we refer to as Pseudo Corner keypoints [10].

To test the usefulness of Pseudo Corners in our pipeline, we computed its repeatability scores on the images we used for evaluation from the ZuBuD dataset. We compared the repeatability of our Pseudo Corners to that of SIFT and Harris-Laplacian. These values are provided in Table I which shows that our Pseudo Corner detector tends to out perform SIFT and Harris-Laplacian on our test images. Given that Pseudo Corners are based on line segments, it is very well-suited for architectural imagery. Since it tends to be more repeatable on this type of data, we chose to incorporate it in our work to help increase the chances of extracting groups for matching that will exist in both images. However, depending on the type of data being used and the designers' preferences, other feature detector may be used in its place with the rest of our pipeline being run exactly the same. To demonstrate this, we also tested out our pipeline on the ZuBuD dataset using SIFT features as the anchor keypoints for grouping. The result of doing so is shown in Figure 7 with the labels "Repetition SIFT" and "Refined SIFT" in the Supplementary Material.

4) Runtime: Table II shows the average runtime of our pipeline on the tested ZuBuD images. These images were matched on an Acer Laptop with 6 GB of memory and a 2.2 GHz Intel Core i7 Processor running Ubuntu 12.04.

5) Limitations and Future Work: One of the main bottle necks of our pipeline is the requirement that the image

Fig. 11. Visual alignment results. The original image pair is shown above each of the checkerboard views of the aligned result. These image pairs poses a variety of challenges to matching in addition to being repetitive such as different rendering styles, time changes, partial facade overlap, scale changes, and lighting changes. Image pairs shown are from datasets provided by [14], [15], and [28].

be rectified. While this is an acceptable assumption for urban images, we cannot guarantee that the method we use will also be able to perfectly rectify each image. Other repetition matching pipelines use an initial estimation of the repetition to refine rectification [2], [17]. We can apply a similar approach to our work. We also assume that the images do not have severe radial distortion. Images that do have this type of distortion may not be rectified correctly. A radial undistortion

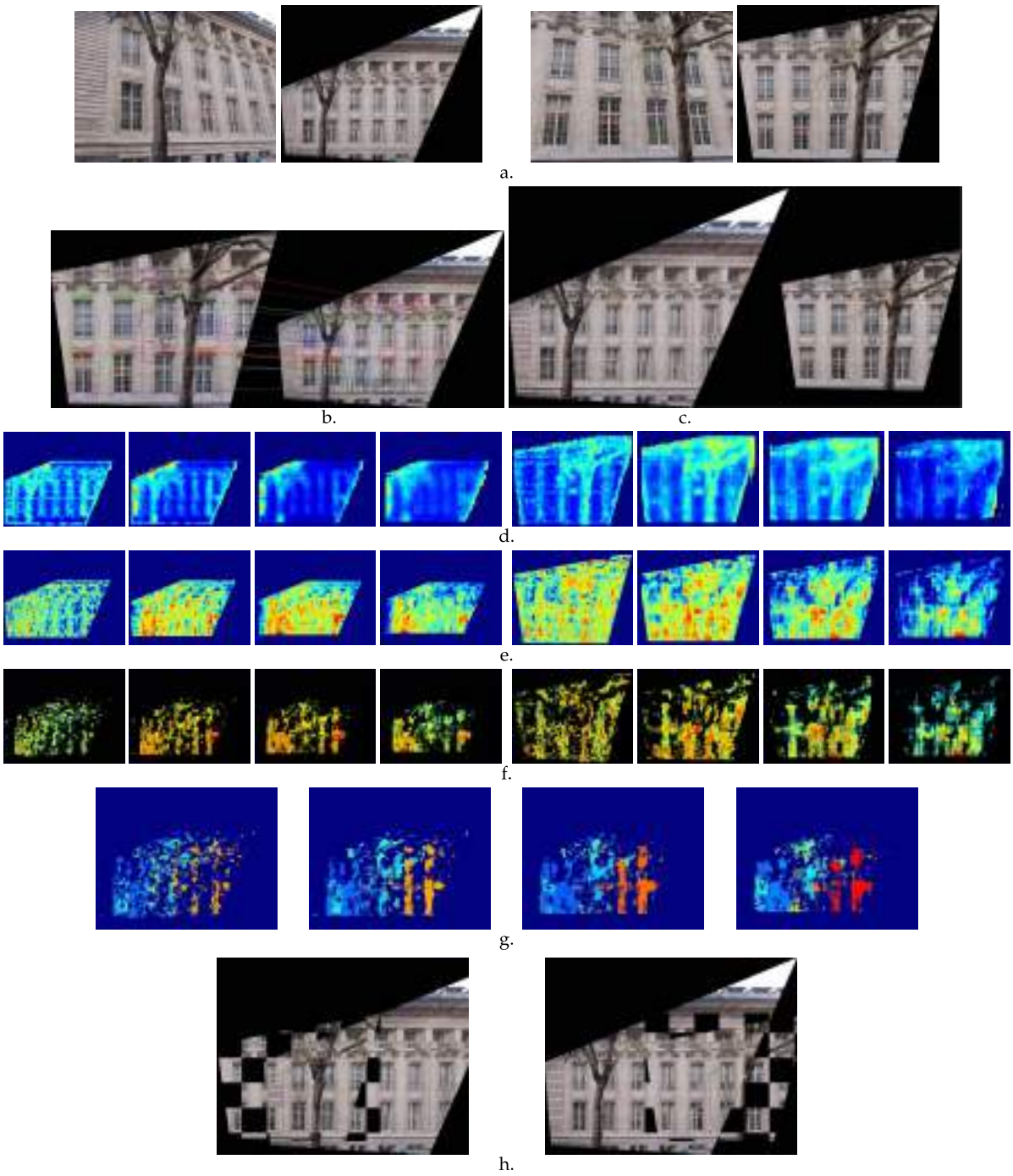

Fig. 12. Alignment of images containing multiple planes. (a.) Original images, before and after rectification, that are being matched. (b.) Subset of the corresponding repetitive groups after computing a 1D transformation (Section IV-A) are highlighted in matching colors. Every element of the groups is displayed. (c.) Images shown after they have been vertically aligned. Note that they now have the same scale and matching rows are aligned with each other. (d.) Intra-image saliency map of each image. The four color fields for each image show the inner-image saliency computed at different scales. The scale increases from left to right for each image. More salient areas appear in red and more repetitive areas appear in blue. (e.) Color fields of match saliency scores shown from the perspective of both images. Again, the color fields are shown at four different scales for each image, increasing from left to right. (f.) Result after applying Otsu thresholding to match saliency score images. All keypoint matches that are contained within the remaining colored regions are used to vote for a hortizontal alignment. (g.) Displacement maps. Each match visualized in (f.) votes for an alignment. The range of these shifts is scaled from [0-1] and displayed as a color map. By doing this, we can see that at each of the four levels, two different displacements have strong support. (h.) Result of aligning the image pair using the two different horizontal displacements chosen in (e.) The left result aligns the salient tree that is in front of (and on a different plane than) the building of interest. The right result uses the salient matches on the building facade to correctly align the region of interest. Our selection criteria chooses the alignment on the right side as the one that best aligns the repetitive elements, and the thus, the building facade.

step could be applied in this case [3]. The effect that radial distortion has on our evaluation is discussed in greater detail in the Supplementary Material.

This pipeline can easily be expanded to handle images in which multiple buildings of a facade are visible. Multiple rectified versions of a photograph that are each arranged to

have a different building plane be front facing can be matched to a query image. We can also alternate between collapsing an image along the horizontal direction to collapsing it along the vertical direction and computing a 1D horizontal transformation first. This may be a good option for data of high-rise buildings if the majority of repetitive features lay along columns instead of rows.

VI. Conclusion

We have presented a method for registering images of architecture that contain repetitive features that have a tendency to create ambiguity for general keypoint matching methodologies. In contrast to several state-of-the-art approaches to registering urban data, our method does not try to fit an evenly-spaced grid to repetitive elements [2], considers both salient and repetitive regions simultaneously [17], and does not require that the same portion of a building's facade be captured in different images [18]. Our method takes a two-step approach to align images one dimension at a time to reduce the search space during each matching stage. We have discussed how we perform a dimension reduction amongst unorganized repetitive features to further remove ambiguity from the search space when matching regions across images. We have also proposed equations for determining the saliency of matching regions between an image pair without requiring machine learning techniques to learn what elements are salient on a large dataset [24]. We believe that by developing a pipeline without these constraints, our method is more robust than many other proposed registration approaches, allowing us to align challenging, ambiguous architectural photographs in a more general manner.

Acknowledgment

The authors would like to thank the reviewers of this paper for all of their helpful comments and suggestions. They are grateful to the researchers who provided the public datasets in [14], [15], and [29] and code [15] that we used for testing and evaluating the work in this paper as well as the engineers and researchers who designed and implemented the OpenCV [30] and VLFeat [31] libraries and image rectification software [4] which were incorporated into this work.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] C. Wu, J.-M. Frahm, and M. Pollefeys, "Detecting large repetitive structures with salient boundaries," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 142–155.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[4] J. Lezama, R. G. von Gioi, G. Randall, and J.-M. Morel, "Finding vanishing points via point alignments in image primal and dual domains," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 509–515.

[5] K. Wilson and N. Snavely, "Robust global translations with 1DSfM," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 61–75.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Distinguished regions for wide-baseline stereo," Center for Machine Perception, K333 FEE Czech Tech. Univ., Prague, Tech. Rep. CTU-CMP-2001-33, Nov. 2001, [Online]. Available: ftp://cmp.felk.cvut.cz/pub/cmp/articles/matas/matas-tr-2001-33.ps.gz

[7] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[8] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1, pp. 43–72, 2005.

[9] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.

[10] B. Morago, G. Bui, and Y. Duan, "An ensemble approach to image matching using contextual features," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4474–4487, Nov. 2015.

[11] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. Trends Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, Jan. 2008.

[12] M. Park, S. Lee, P.-C. Chen, S. Kashyap, A. A. Butt, and Y. Liu, "Performance evaluation of state-of-the-art discrete symmetry detection algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert, "Detecting and matching repeated patterns for automatic geo-tagging in urban environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[14] D. Ceylan, N. J. Mitra, Y. Zheng, and M. Pauly, "Coupled structure-from-motion and 3D symmetry detection for urban facades," *ACM Trans. Graph.*, vol. 33, no. 1, 2014, Art. no. 2.

[15] D. C. Hauagge and N. Snavely, "Image matching using local symmetry features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 206–213.

[16] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[17] M. Kushnir and I. Shimshoni, "Epipolar geometry estimation for urban scenes with repetitive structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2381–2395, Dec. 2014.

[18] Y.-C. Chung, T. X. Han, and Z. He, "Building recognition using sketch-based representations and spectral graph matching," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2014–2020.

[19] M. Bansal, K. Daniilidis, and H. Sawhney, "Ultra-wide baseline facade matching for geo-localization," in *Proc. Eur. Conf. Comput. Vis. Workshops Demonstrations*, 2012, pp. 175–186.

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[21] C. Barnes, E. Shechtman, and D. Goldman, "The generalized patchmatch correspondence algorithm," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 29–43.

[22] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, "Data-driven visual similarity for cross-domain image matching," *ACM Trans. Graph.*, vol. 30, no. 6, p. 154, Dec. 2011.

[23] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 73–86.

[24] M. Aubry, B. C. Russell, and J. Sivic, "Painting-to-3D model alignment via discriminative visual elements," *ACM Trans. Graph.*, vol. 33, no. 2, p. 14, Mar. 2014.

[25] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, nos. 285–296, pp. 23–27, 1975.

[27] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.

[28] H. Shao, T. Svoboda, and L. V. Gool, "Zubud-zurich buildings database for image based recognition," Comp. Vis. Lab, Swiss Fed. Inst. Technol., Switzerland, Tech. Rep. vol. 260, 2003.

[29] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from Internet photo collections," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 189–210, 2007.

[30] G. Bradski, "Opencv library," Dr. Dobb's J. Software Tools, 2000, Art. no. 2236121.

[31] A. Vedaldi and B. Fulkerson. (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. [Online]. Available: http://www.vlfeat.org/

Brittany Morago received the B.S. degree in digital arts and sciences from the University of Florida in 2010, and the Ph.D. degree in computer science from the University of Missouri, Columbia, in 2016. She is currently an Assistant Professor with the Department of Computer Science, University of North Carolina at Wilmington. Her research interests include computer vision and graphics. She was a recipient of the NSFGRF and GAANN fellowships.

Giang Bui received the B.S. and M.S. degrees from the Vietnam National University of Hanoi in 2004 and 2007, respectively. He is currently pursuing the degree with the University of Missouri, Columbia. He was a Research Assistant with the Computer Graphics and Image Understanding Laboratory under the supervision of Dr. Y. Duan. His research interests include image and video processing, 3-D computer vision, and machine learning.

Ye Duan received the B.A. degree in mathematics from Peking University in 1991, and the M.S. degree in mathematics from Utah State University in 1996, and the M.S. and Ph.D. degrees in computer science from the State University of New York, Stony Brook, in 1998 and 2003, respectively. From 2003 to 2009, he was an Assistant Professor of Computer Science with the University of Missouri, Columbia. He is currently an Associate Professor of Computer Science with the University of Missouri, Columbia. His research interests include computer graphics and visualization, biomedical imaging, and computer vision.