Bridge Inspection Component Registration for Damage Evolution

Transportation Research Record
2021, Vol. XX(X) 1–13

DOI: 10.1177/ToBeAssigned
journals.sagepub.com/home/trr
SAGE

Eric Bianchi[1], Nazmus Sakib[2], Craig A. Woolsey[2], Matthew Hebdon[3]

Abstract

There have been great advances in bridge inspection damage detection involving the use of deep learning models. However, automated detection models currently fall short of giving an inspector an understanding of how the damage has progressed from one inspection to the next. The rate-of-change of the damage is a critical piece of information used by engineers to determine appropriate maintenance and rehabilitation actions to prevent structural failures. We propose a simple methodology for registering two bridge inspection videos, collected at different stages of deterioration, so that trained model predictions may be directly compared for measuring the changes of any new damage and damage progression. The changes may be documented and presented to the inspector so that they may quickly evaluate key segments in the inspection video. Three approaches comprised of rigid, deformable, and hybrid image registration methods were experimentally tested and evaluated based on their ability to preserve the geometric characteristics of the referenced image. It was found in all experiments that the rigid, homography-based transformations performed the best for this application over a state of the art deformable registration method, RANSAC-Flow.

INTRODUCTION

Routine, comprehensive bridge inspections are a necessary component of the life-cycle maintenance and evaluation to preserve the safety of our highway infrastructure. One of a bridge inspector's jobs is to identify defects (both large and small) and potential problem areas before they become critical to the structural or functional integrity. If not monitored and appropriately corrected, deterioration can grow to a point where remediation is extremely costly, time-consuming, and disruptive. In the most severe case, a catastrophic failure or partial collapse may result, possibly causing deaths or serious injuries. The deadly collapse of the 51-year-old Morandi bridge in Genoa, Italy (2018) is a reminder of the damage to properties and human lives that failures caused by excessive deterioration. Maurizio et al. (*1*) investigated the collapsed bridge and found that their fatigue models predicted the collapse of the bridge 2-4 years earlier than the actual collapse in 2018. Poor maintenance of the structure led to undetected cable corrosion that tragically took 43 lives. On May 11, 2021 in Memphis, Tennessee inspectors found "a significant fracture in one of two 900-foot horizontal steel beams on interstate I-40 that were crucial for the bridge's integrity" (*2*). Investigations showed that the inspectors in previous bridge inspections had missed the crack development completely; a catastrophic event could have occurred had authorities not closed the bridge to traffic following the later investigation.

Unmanned aerial vehicles (UAVs) have already proven to be an effective tool in many civil engineering applications like: construction safety and progress monitoring, geotechnical engineering, and post-disaster reconnaissance (*3*). Chen et al. (*4*) showed that non-contact inspection using aerial video imagery can identify large cracks and defects in roads. Another study conducted by Reagan (*5*) concluded that a fully developed aerial surveillance system could increase the inspection efficiency and the inspection frequency of the inspectors allowing for faster maintenance of defective bridges. Typical bridge inspections rely on collecting and quantifying data to describe the quantity and severity of deterioration on individual bridge elements such as damage condition states, or estimated areas of damage (concrete spalling, crack length, deflection, etc.). This data provides a historical snapshot which describes both the localized regions on a bridge as well as the global bridge health as a whole. A series of inspection reports provide a time-history

[1]Department of Civil and Environmental Engineering, Virginia Tech, Blacksburg, Virginia 24061
[2]Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia, 24061
[3]Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, Austin, Texas, 78712

Corresponding author:
Eric Bianchi, beric7@vt.edu

of the health of the structure and rate of degradation. Most inspection work is done by observing the bridge visually. The work described in this paper aims to assist bridge inspectors with their visual inspection tasks so that the inspection can be done more accurately and efficiently. Specifically, this technology will assist an inspector with the on-going monitoring of the condition and deterioration progression over time. With that goal in mind, this paper proposes an algorithm which combines three different computer vision tasks – image matching, image alignment and image comparison – that can greatly improve the inspection process. The initial assumption being that the algorithm finds the closest image-match of any bridge section, bridge element, or structural bridge detail identified from previous inspections to those images from the current inspection. The matched images are then perspectively warped to align with the images from the previous inspections. Then the aligned images are compared and the damage progression and severity is quantified. This quantitative value indicates how much the condition of the problematic region has deteriorated compared to the previous inspections thus reducing some of the workloads from the inspectors.

The primary contributions of this paper consist of:

1. A methodology for aligning inspection video content for monitoring the progression of existing damage, and detecting new instances of damage using state of the art optical flow and keypoint detection algorithms.

2. Experiments on the effect of camera attitude on error, and the important conclusion that a rigid image registration method, homography transform, grants the most success for this bridge inspection application.

The first stage of the proposed process is to identify key feature points between target and source images - the *source image* is the image being aligned to the *target image* - for estimating 3D structure and camera poses. Large shifts in perspective, illumination, occlusion, blur, and a lack of texture are all characteristics that make 2D-to-2D data association difficult in a moving camera frame particularly in the task of a bridge inspection. A recently developed Graph Neural Network (GNN) called "SuperGlue," which has been proved to outperform other feature matching techniques (*6*), is used to match feature points in this paper. After obtaining the feature points, the next step is to geometrically align the two images. A dense image alignment method (also known as image registration) called "RANSAC-Flow" (*7*) is used in this step to geometrically align the newly obtained image with the previous inspection image. Finally, the aligned source image is compared to the target image by using a semantic damage detection model. The change in damage is quantified to determine the progression of deterioration for that target location. The model, DeeplabV3+ (*8*), was trained on a novel corrosion condition state dataset - the trained weights for the model may be accessed at https://github.com/beric7/structural-main.

Results are presented here using a contrived scene, for evaluation and sensitivity analysis, and for an actual bridge element – a beam – as a more realistic demonstration. The steps above were performed on pre-recorded videos on the ground by the research team, and are expected to be scalable to inspection videos recorded by unmanned aerial systems. As the hardware available to the investigators is not sufficient to perform these tasks during a live inspection, thus, running the algorithm described above on a pre-recorded video is synonymous to running it on a sequence of images. As such, most of the experiments to identify the strengths and weaknesses of the algorithm are done on sequences of image and then its performance is evaluated using a pre-recorded video stream of an actual bridge element, i.e, a beam. The algorithm along with all related code and validation data can be found at: https://github.com/beric7/Inspection-Image-Registration.

Related Work

Building upon the early contributions of Sand and Teller (*9*), the scale-invariant feature transform (SIFT) algorithm (*10*) was developed to find keypoints in an image that can be used for algorithms such as object tracking, image stitching, or simultaneous localization and mapping (SLAM). Focusing on image stitching, Cui *et al* (*11*) took slices of images at different points in time, aligned them, and stitched the resulting sub-images together. The goal was to add or remove objects within the image to address the problem of occlusions.

An algorithm titled learning to align and match videos (LAMV) temporally aligned a specified instance in a video, or a specific video frame, from several different cameras and camera poses using kernelized temporal layers (*12*). This approach looked for key characteristics of each of the frames and determined if they matched. Thus, the algorithm did not align the frames themselves, but it aligned the same temporal point in time at which the frames were taken. Building on the LAMV algorithm, the bi-directional attention flow (BDAF) algorithm very accurately aligned video frames (*13*). This BDAF algorithm demonstrated a 10% and 14% increase in accuracy on the Climbing (*14*) and Madonna (*14*) datasets, respectively.

There are two general approaches to image registration: rigid image registration (RIR) and deformable image registration (DIR). RIR uses only rigid transformations while DIR allows regions within images to deform and warp in ways that cannot be described using only translations and rotations (*15*). Both RIR and DIR techniques have their advantages, but DIR techniques have been growing in popularity, especially in the medical field (*16, 17*). Medical imagery focuses more on the DIR techniques because the 3D scene that is being imaged (e.g., the tissue within a

human body) may deform between consecutive images (*15*). One example of image registration in the medical field is the fusion of images obtained using different modalities, such as positron emission tomography (PET) and computed tomography (CT) scans, Figure 1. However, because bridge structures are much less likely to deform to the extent that the human body does the research team hypothesized that a RIR approach would be more appropriate for this application.

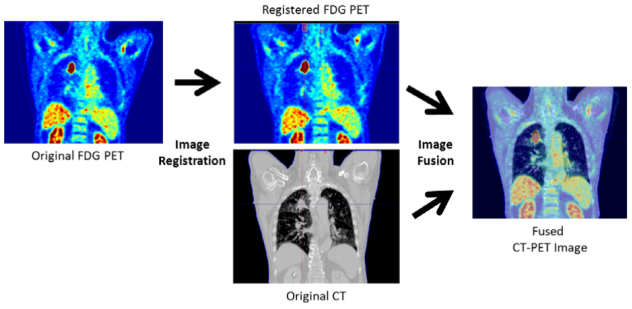


Figure 1. Fusion image registration of multiple image modalities (*17*).

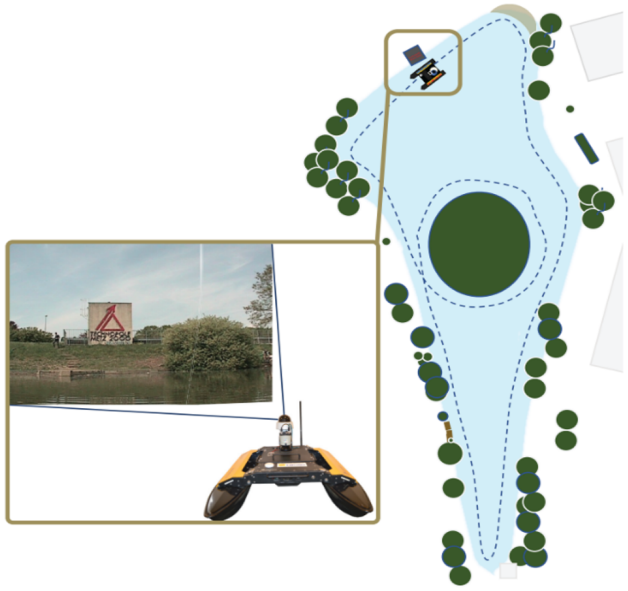


Figure 2. This is the GeorgiaTech-Loraine (GTL) Clearpath Kingfisher which collected surveys on the Symphony Lake Dataset. It navigated the path shown as the dotted line with its camera facing towards the shoreline (*18*).

To the authors' knowledge, the research that is most closely related to the work described here involved aligning images taken from a robot on a lake for a land survey repeated over many seasons (*18*); see Figure 2. This dataset of land surveys is called the Symphony Lake Dataset (*19*). The researchers used camera position information, SIFT keypoints, and an optimization process to warp images to fit to the target image. The result was a time-lapse of specific locations along the route for each target location. Griffith (*18*) recommended to include a more robust and modern method for collecting keypoints in the future. The research presented in the present paper uses a more modern keypoint finder, SuperGlue (*6*), with almost the same goal, except that instead of creating time-lapse imagery, the aim here is to detect and document damage progression over time.

METHODOLOGY

The overarching methodology of this research was structured to evaluate deterioration observed in individual frames from two videos taken at different points in time. Let's call the most recent bridge inspection $\texttt{inspection}_{\{N\}}$ and let's call the next bridge inspection $\texttt{inspection}_{\{N+1\}}$. The video frames from $\texttt{inspection}_{\{N+1\}}$ are the source images, which are aligned with target images from $\texttt{inspection}_{\{N\}}$ to determine the change between the two sets of images. The detected damage trends could be used to find which points in the $\texttt{inspection}_{\{N+1\}}$ video that there was significant change from $\texttt{inspection}_{\{N\}}$. This type of analysis would be an effective way to automatically find crucial periods of interest, especially when the inspection video is extensive. To explain how our algorithm accomplishes this feat we break it down into three main parts: **Finding candidate images, aligning candidate images, and detecting changes in damage progression**.

Throughout the methodology we reference two reasonable assumptions described below.

I. The traversed inspection path is approximately the same between each inspection.

II. The elapsed recorded time is approximately the same between inspections.

For assumption (I), while there may be slight deviations in the camera offset from structural elements, the attitude (yaw, pitch, and roll) of the captured frames, the relative path stays the same. In regarding a video feed, a similar relative path would mean that the trajectory of the source and target inspection videos are similar, with an offset from the bridge elements approximately the same, and that they start and end in the same location. This is a crucial detail since the efficiency of the image-matching algorithm relies on the approximate sequential ordering of the two inspections. This assumption becomes especially important given the repetitive patterns of bridge elements, which sometimes only leaves slight visual nuances between two different locations on the bridge. For assumption (II), having the elapsed time be approximately the same between the inspections is another useful assumption to make for the efficiency of the image matching search algorithm, and will be further explained in the methodology.

Finding Candidate Images (SuperGlue)

All feature matching techniques include the following steps: (i) obtaining interest points (well defined pixel locations in a 2D image which can be robustly detected); (ii) computing descriptors (descriptors are descriptions of the visual features at or in the neighborhood of those interest points containing elementary characteristics such as the shape, the color, the texture, the motion etc.); (iii) for each of these points, matching the descriptors with nearest neighbor (NN) searches; (iv) filtering out the incompatible searches (for example, some points in one image might not appear in another image due to occlusion); and (v) estimating a valid transformation between the two images. Classical feature matching techniques often involve finding feature points using SIFT (*10*), filtering out unwanted points using techniques like Lowe's ratio test and neighborhood consensus (*10, 20–22*), and finally finding the necessary affine or perspective transformations using robust outlier rejection algorithms like RANSAC (*23*). With the advent of deep learning methods, the focus shifted to learning better sparse detectors and local descriptors (*24–27*) using convolutional neural networks (CNNs). SuperGlue (*6*) is such a CNN which wraps the "attentional graph neural network" heuristic with an "optimal matching layer". In the attentional graph neural network the keypoint positions and their corresponding descriptors are passed through repeating self- and cross-attention neural network layers to obtain the common descriptors between two input images. The optimal matching layer then creates a score matrix using the Sinkhorn algorithm (*28*) between the common descriptors to match them appropriately.

SuperGlue Architecture Computer vision theories assume that images of the physical world satisfy certain conditions. For example, each 2D image point is the planar projection of some 3D world point and each point in 3D space is represented by at most one corresponding point in the 2D image. Because of occlusions, one 2D image may exhibit the projection of a particular 3D world point that is not represented in another 2D image of the same scene. Let I_s and I_t indicate source and target images respectively with M and N local features. The ith feature point in an image can be represented as a vector $\boldsymbol{p}_i := (u, v, c)_i$, comprising its scalar pixel coordinates u and v and the scalar detection confidence c. The associated descriptors $\boldsymbol{d}_i \in \mathbb{R}^D$ can be extracted by using any common descriptor matching algorithm like SIFT or CNNs like SuperPoint (*29*). The length D of each of the descriptor vectors depends on the type of descriptor used. For example, $D = 128$ for a SIFT descriptor. According to (*29*), the SuperPoint detector and descriptor obtains more feature points than traditional detectors and descriptors like LIFT (*30*), SIFT (*10*) and ORB (*31*). SuperPoint is used in this paper to extract keypoints and their corresponding descriptors; SuperGlue is then used to to find the matches. The function of SuperGlue is to predict a soft assignment matrix: $\boldsymbol{P} \in \mathbb{R}^{M \times N}$ whose ij^{th} element $P_{ij} \in [0, 1]$, representing the confidence value of each possible correspondence, that satisfies:

$$\boldsymbol{P}\mathbf{1}_N \leq \mathbf{1}_M \quad \text{and} \quad \boldsymbol{P}^{\mathrm{T}}\mathbf{1}_M \leq \mathbf{1}_N$$

Here, $\mathbf{1}_M$ and $\mathbf{1}_N$ represent column vectors of ones of dimensions M and N respectively.

Humans match features between two given images by examining the two images and looking for contextual cues. Mathematically, this is similar to running an iterative process that focuses attention on important regions. SuperGlue does the exact same thing using an attentional graph neural network which reasons about both the appearance of an image and the pixel positions and assigns "attentional weights" for each of the matched features. The feature descriptor vector is then constructed based on these weights and is then optimized using the Sinkhorn algorithm to produce the soft assignment matrix $\boldsymbol{P}$ under some constraints.

Selection by Keypoints The image sequence for $\texttt{inspection}_{\{N\}}$ consists of some number of target images which are compared to the source images for $\texttt{inspection}_{\{N+1\}}$. The comparisons used the number of keypoints detected by the SuperGlue algorithm to determine the strongest match. If the number of keypoint matches were less than 100, then that image was not considered. To determine the best starting position, every image in the source dataset ($\texttt{inspection}_{\{N+1\}}$) was compared to the first target image ($\texttt{target}_{\{0\}}$) of $\texttt{inspection}_{\{N\}}$. Once the starting image ($\texttt{source}_{\{0\}}$) from the source dataset ($\texttt{inspection}_{\{N+1\}}$) was chosen, the next source image comparison began 10 frames before ($\texttt{source}_{\{0\}}$) to provide some overlap. The next source image, $\texttt{source}_{\{1\}}$, and each subsequent source images, were chosen after the largest peak of detected keypoints was found. For each target image the top K strongest matched source images from the keypoint peak were chosen, in our case $K = 3$. These top-3 best matched group of source images are referred to as the *source image ensemble* throughout the rest of the paper. A diagram of this process is shown in Figure 3.

Aligning the Candidate Images (RANSAC-Flow)

Each candidate source image in the ensemble must be aligned to the target image before the damage detection algorithm can compare the two images to detect changes. As discussed earlier, there are two approaches to image alignment: rigid image registration (RIR) and the more recent deformable image registration (DIR) method. In the RIR method, it is assumed that the image being compared is related to the target image by a global affine or homographic transformation. If the actual transformation can indeed be described as a rigid transformation (a translation and a rotation), then this transformation can be obtained fairly accurately from

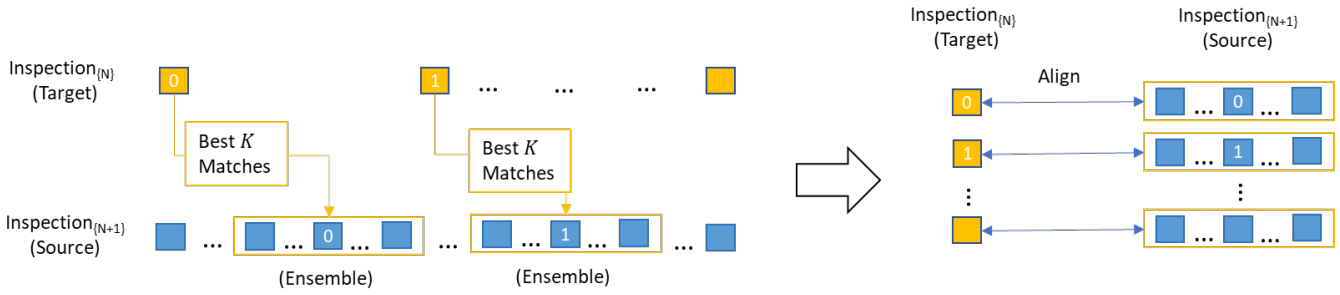


Figure 3. Selecting Best Matches Between Inspections

the matched feature points by rejecting outliers using robust methods like RANSAC (*23*). Modern methods try to optimize some pixel alignment metrics directly using CNNs without making any assumption about the nature of the transformation. Either approach has limitations: RIR methods fail to completely resolve occlusions whereas DIR methods do not work properly if the overlapping region between the two compared images is small. RANSAC-Flow (*7*) is a hybrid strategy that combines the advantages of using both the RIR and the DIR methods together.

RANSAC-Flow Architecture RANSAC-Flow first aligns the source image with the target image coarsely, then it performs fine-tuning on the coarsely aligned image to produce the final aligned image. During the coarse alignment the algorithm uses RANSAC, as the name suggests, to fit a homographic transformation on a set of probable matches between the source and the target image. The matched features are extracted from the two images at some number of different scales. The authors in (*7*) used seven different scales: 0.5, 0.6, 0.88, 1, 1.33, 1.66, and 2. Matches are discarded if they are not consistent across all scales. Using the obtained homography, the source image and the target image are warped and fed into the fine tuning algorithm. The objective of the fine alignment is to obtain a mapping function $\boldsymbol{F}_{s\to t}$ that warps the coarsely aligned source image into an image similar to the target image. For this to happen, an objective function is formulated that includes: reconstruction loss ($\mathcal{L}_{\mathrm{r}}$), matchability loss ($\mathcal{L}_{\mathrm{m}}$), and cycle-consistency loss ($\mathcal{L}_{\mathrm{c}}$). The total loss for a pair of source and target images (I_s, I_t) then becomes:

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{\mathrm{r}}(I_s, I_t) + \lambda\mathcal{L}_{\mathrm{m}}(I_s, I_t) + \mu\mathcal{L}_{\mathrm{c}}(I_s, I_t)$$

where $\lambda = .01$ and $\mu = 1$ are the weights of the matchability and cycle losses respectively used in (*7*). The values were obtained empirically as they provided the best performance for the algorithm while aligning two images. Since image alignment is an objective of this paper, these values are kept unchanged.

The total loss function is then optimized using a self-supervised neural network. The process described thus far works well on images that can be related almost perfectly using a single homography matrix. This is the case when the relative camera position changes by a small amount and there exists minimal occlusion. However, for image pairs with large displacements, strong 3D effects, and high occlusions between the two, a single homography transformation is not sufficient to align the two images properly. For such cases, the algorithm iterates over the pair of images to obtain multiple homographic transformations corresponding to some regions. At each iteration, the matched feature points, along with their associated regions, used to compute the previous homographic transformation are removed. Thus every iteration produces a new mapping function, $\boldsymbol{F}_{s\to t}$, with some associated regions. The overall transformation is then simply the combination of all the mapping functions, $\boldsymbol{F}_{s\to t}$, and their respective regions.

The keypoints detected from the SuperGlue algorithm were used to generate a homography matrix. The homography matrix was applied to the source image to warp and align it with the target image. Additionally, RANSAC-Flow's coarse alignment was applied to the image for an iterative alignment beyond what the homography matrix could provide. While RANSAC-Flow does offer a further processing step of fine alignment, we found that fine alignment was not a good choice for this application. The reasons are discussed in Section .

Damage Detection

Damage detection was the final step in the process. One of the typical tasks for the inspection of steel bridges is to locate and monitor corrosion damage. The damage must be quantified and the severity of its condition state must be rated on a scale of 1 to 4 (good, fair, poor, severe), based on prescriptive guidelines. In order to automate this part of the inspection, a trained deep neural network damage detection model for corrosion and cracks was used to semantically segment the damage in each target and aligned image. The chosen model architecture was the DeeplabV3+ (*8*), which was trained on a novel corrosion condition state dataset - the trained weights for the model may be accessed at https://github.com/beric7/structural-main (CITE). The trained model semantically predicted the

corrosion condition states for each pixel in the image. Thus, the extent of damage and its severity was quantified in the most granular scale possible. Because both the damage from past and current inspection images can be quantified, the relative change or progression of deterioration between the two inspections can be approximated. Figure 17 shows an example of the damage detection algorithm.

EXPERIMENTS

There were three experiments conducted, a preliminary experiment, a sensitivity study, and a field test. The preliminary experiment was completed in a classroom at the Thomas Murray Structures lab at Virginia Tech. The preliminary experiment evaluated the effectiveness of RANSAC-Flow. The sensitivity study was performed in the Virginia Tech's Kentland Experimental Aerial Systems (KEAS) lab. The objective of the sensitivity study was to evaluate the effects of changing the offset distance and camera orientation on the ability to geometrically align two images. Finally, field tests were conducted outside of the Thomas Murray Structures lab in the downed steel and concrete beam yard. The field test was performed to measure how the performance of the three methods (coarse, homography, and hybrid) in a more realistic inspection setting.

Preliminary Observations

In our preliminary experiments, when evaluating the effectiveness of using RANSAC-Flow to geometrically align images, we noticed two main issues. The first issue was with the algorithm's ability to geometrically align images when the distance from the camera to the scene varied between the two images. For example, when the target image was obtained from nearby (2 ft) and the source image was obtained from far away (12 ft) (Figure 4), the source image did not geometrically align adequately (Figure 5). Therefore, we applied a generic homography transform before applying RANSAC-Flow, which improved the performance of the latter algorithm. The second issue with RANSAC-Flow was with the fine-tuning alignment step applied after the coarse-alignment. It was observed that that the fine-alignment warped the source image to deform back to the target image. Figure 7 depicts this phenomenon. While this result may be desirable in some situations, it is undesirable for inspection applications where the objective is to detect changes in the scene. This type of deforming alignment would effectively hide damage as it evolves. Therefore, only RANSAC-Flow's coarse alignment was used since it does not reverse minor changes (Figure 6) the way that fine-tuning alignment does (Figure 7). Another example of these warping and deforming changes is shown in Figure 13 in the Appendix.

Figure 4. Target image *near* - 2 ft (left), source image *far* - 12 ft (right)

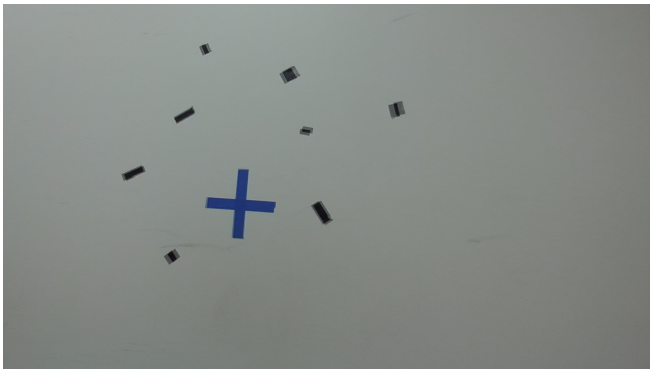


Figure 5. Coarse alignment overlay image on target image. Miss-alignment perceived as *blurring*, a perfectly aligned image would show no *blur*.

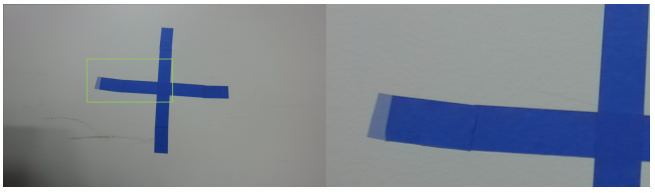


Figure 6. Small growth change. Coarse alignment global image (left), and coarse alignment zoomed-in portion (right).

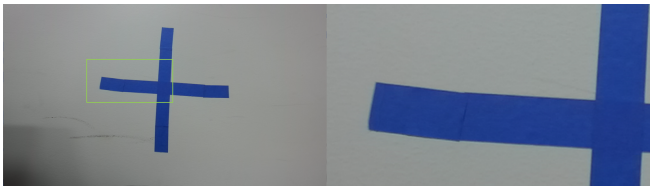


Figure 7. Small growth change. Fine-tune alignment global image (left), and fine-tune alignment zoomed-in portion (right). The growth has been snapped back to the original size.

Sensitivity Study

Given the preliminary results, we wanted to understand the effects of changing the offset distance and camera orientation on the ability to geometrically align a region in a source image to that of a target image because if the alignment step is not good then the damage prediction algorithm will be incorrect. During an inspection it is expected that camera pose and offset may vary from the target image - especially if a drone is being used to capture the data. Therefore, we

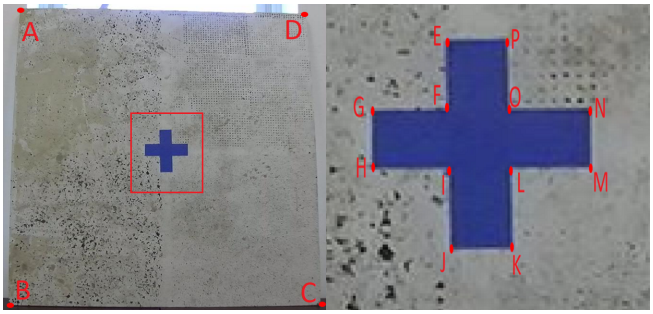


Figure 8. Example image used for the sensitivity study. Points A-P indicate the location of the pixel coordinates used for analysis.

examined the sensitivity of aligning source images to a target image by varying the camera offset distance and attitude (roll, pitch, and yaw angle). Additionally, three image alignment methods were tried: (1) homography, (2) coarse-alignment, and (3) a hybrid method which first applied homography and then coarse-alignment. The goal of this sensitivity study was to discover which method performed the best and to determine if there were particular types of camera attitude that introduced more alignment error than others. For this investigation, a "plus" symbol was created on a white board using blue masking tape, as shown in Fig. 8. The methods were evaluated using root mean squared error (RMSE) in the keypoints A through P (indicated in Fig. 8) and measuring the difference in the area of the plus sign between the source and the target images. Eight different camera poses were tested, relative to the given scene, at four different offsets of approximately $4\,\text{ft}$, $6\,\text{ft}$, $8\,\text{ft}$, and $10\,\text{ft}$. Of the eight different poses one had minimal roll, pitch or yaw orientation and was termed "normal". The "pitch up" and "pitch down" orientations were about 20 and $-25\,\text{degrees}$ respectively whereas the "roll left" and the "roll right" orientations were about $16\,\text{degrees}$ from the normal camera orientation. The other three orientations were due to different yaw angles and were named "yaw 7.5", "yaw 15", and "yaw 30" based on the approximate measurement of yaw angles. All of the eight different poses were repeated at each of the four offset distances.

To test the accuracy of the alignment, 16 fixed points A through P were selected on every aligned source image (marked in Fig. 8 for example) manually for comparison. The total number of alignment cases for comparing against one target image thus became: 4 from four different offset levels $\times$ 3 from three different alignment methods $\times$ 8 from eight different attitudes = 96. Three different target images were used: normal poses at $4\,\text{ft}$, $6\,\text{ft}$, and $8\,\text{ft}$ respectively against which the above mentioned 96 cases were compared. Thus the total number of images analyzed were $3 \times 96 = 288$ for all the three target images. To select the true pixel values of the 16 points: A-P, as shown in Fig 8, each of the target images was looped 20 times and the corresponding pixel values of the 16 points were recorded. Then the pixel values were averaged to get the true pixel coordinate for each of the 16 points. Afterwards, the pixel coordinates of the same 16 points in each of the warped source images were selected. The RMSE between the selected pixel points and the true pixel points was then obtained. Additionally, using the points E through P, the area of the plus sign was obtained. Similar to the RMSE, the areas of the source images were compared to the true area of the target images and the error in area was obtained.

Field Test

We ran field tests on two grounded steel beams (beam 'A' and beam 'B') in a steel yard. The steel beams A and B were passed four and two times respectively, with a ZED2 camera starting and stopping at approximately the same place. The ZED2 is a stereo camera, however, only the "left" image was used. The resolution for the left image was 1920x1060. Offsets from the face of the beam were kept at roughly the same distance. The first goal was to verify results found in our sensitivity study. The second goal was to demonstrate how this application fit into the inspection process using a real structural element. The three methods from the sensitivity study: coarse, homography, and hybrid, were evaluated using RMSE between matched keypoints and segmentation (F1 scores, intersection over union - IoU) using the trained corrosion model's predictions between the target and source images.

The keypoint pairs were automatically detected and matched using SuperGlue. Thus, hundreds of keypoint pairs were found automatically instead of the manual keypoint selection process used in the sensitivity study. The error measured was the RMS pixel error from the expected keypoint location in the target image to the detected keypoint location in the source image. This error was averaged across the ensemble of source images (top-3 best matched). Similarly, the average prediction on the ensemble of source images was used when calculating the F1 and IoU score. The F1 and IoU scores are typical metrics for quantifying the success of a semantic segmentation model prediction. The F1 score is a weighted class accuracy and precision score which also accounts for imbalances in class categories. The IoU score observes the intersection over the union of class predictions to the ground truth labels. F1 and IoU scores are positively correlated, but the IoU score tends to amplify discrepancies, which is useful to highlight a particular method's superiority.

RESULTS and Discussion

Sensitivity Study

The manipulation techniques (distance offset, attitude) and alignment methods (RANSAC-Flow's coarse, classical homography, and hybrid) were compared using a composite

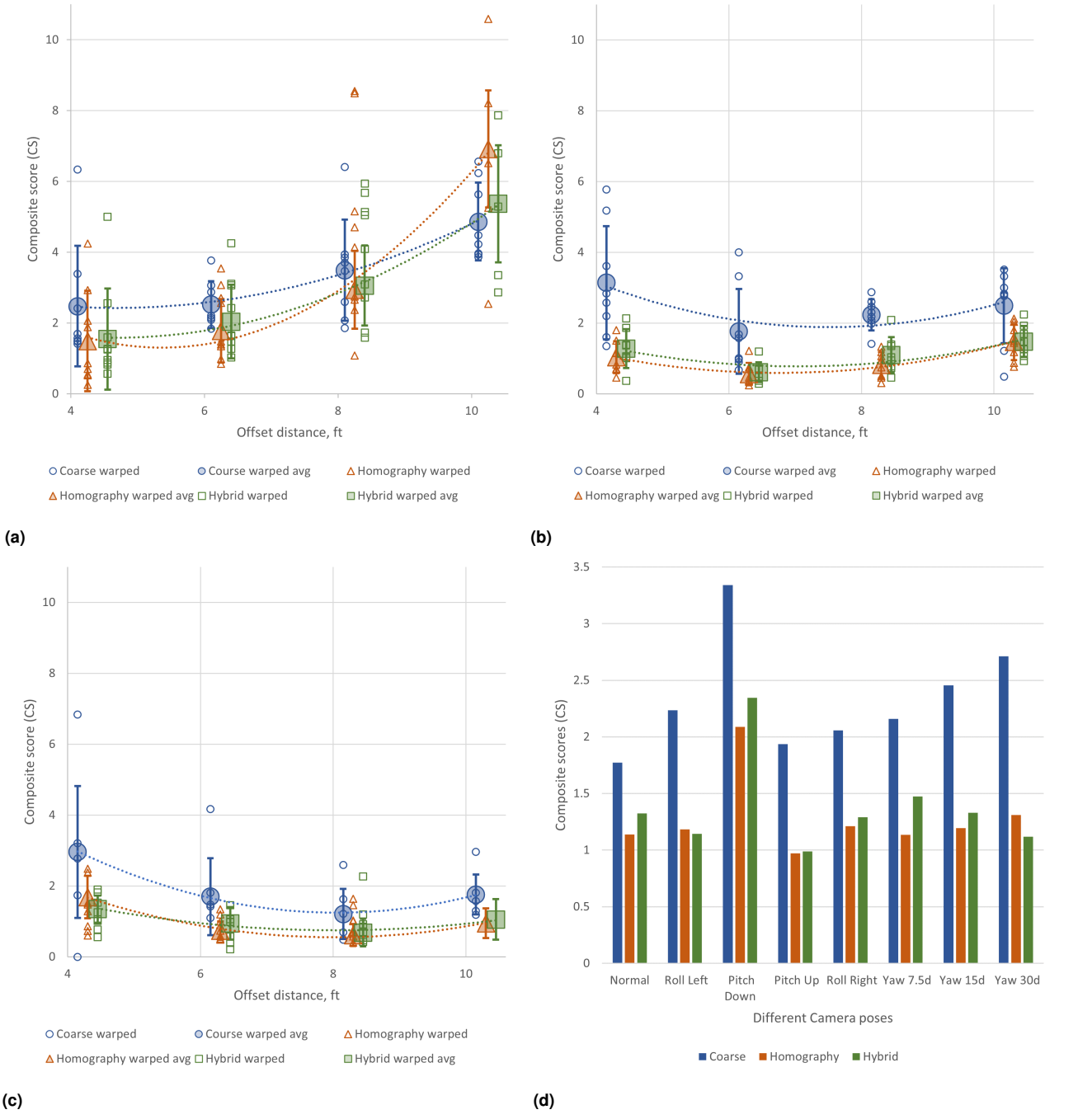


Figure 9. Composite scores (CS) for (a) target image at 4 ft, (b) target image at 6 ft, and (c) target image at 8 ft with the source images at different offsets (the large markers indicate the average of the scores at each offset location). Note: The homography and the hybrid data are shifted to the right along the x-axis for better clarity; (d) average sensitivity of all the images categorized based on the respective camera poses. The lower the score, the better the performance.

score containing both the keypoint average pixel error and the average area error of the tape. The composite score (CS) was calculated by normalizing each of the RMS pixel errors with the average RMS pixel error of the entire set of 288 images and adding them to the corresponding normalized area errors, mathematically:

$$\mathrm{CS} = \frac{\text{RMS error}}{\text{Total average RMS error}} + \frac{\text{Area error}}{\text{Total average area error}}$$

Figures 9a, 9b, and 9c show the composite score for each set of 96 source images against target images at 4 ft, 6 ft, and

8 ft respectively. The large markers indicate the average of all the CS (that is, average of eight different poses) at each of the offset distances. In most of the attitude manipulations, homography (orange) had the lowest pixel error, followed by hybrid (green), and coarse alignment (blue). It is also apparent from 9a - 9c that even though there are lots of deviations in the CS, taking the average is a good way to reduce the effect of potential outliers.

In a practical scenario, a drone would be used to take videos of defects near a bridge flying in presence of wind gusts or disturbances. This would mean that the images obtained from the camera would include some amounts of offset and attitude from an ideal image. So, to quantify defects, it is recommended that multiple source images should be warped and aligned with respect to each target image and then the model predictions of the defect locations should be averaged across all the warped source images to obtain an ensemble of predictions for a better representation of the damage with greater certainty. We recommend that the size of the source image ensemble be determined by having at least 95% of the quantity of matched keypoints in the best-matched image. It is expected that the change in attitude or offset due to wind turbulence, obstacle, and flight path from one image frame to the next one, would not be as extreme as the cases tested in this sensitivity study, so the performance should actually improve with lower standard deviation when compared to the cases presented in the sensitivity study. The performance can be improved further by implementing outlier rejection techniques like RANSAC or three sigma bounds and rejecting some of the image frames which generate pixel points that are outliers while using the average measurements from the rest.

Figure 9d shows the effectiveness of three different methodologies (coarse, homography and hybrid) against different camera poses. Here the CS for a particular pose at different offset distances are averaged. The data corresponding to source images at 10 ft and target image at 4 ft are ignored in this calculation as the relative offset between the two images are higher for this case than for any other tested cases and the comparisons would not be accurate.. With the exception of the "Pitch down" case, all the other orientations are seen to be fairly close with minimum deviation for both the homography and the hybrid methods. These two methods are calculated to be performing on an average 1.98 and 1.85 times better than the coarse method respectively. The standard deviation of the CS displayed in Fig. 9d for the three methods coarse, homography, and hybrid are .32, .10, and .16 respectively indicating that homography and hybrid methods are much more robust in response to the change in camera pose.

Homography handled the distance manipulation changes the best until the target and source images had relative offset of more than 6 ft (Fig. 9a). At a higher offset distance, the coarse alignment performs well because in coarse alignment multiple candidate homographies are computed between image pairs at different scales and the outlier homographies are rejected. This generates a more accurate homography matrix than the classical case where only one single pair of image is used to obtain homographies. An expected observation from Figs. 9a, 9b, and 9c is that the best performance is obtained for those cases where the source images have almost the same offset distance as the target image. By analyzing The CS of 288 images it is apparent that approximately $5-7$ ft distance is the ideal offset for both the target images and the source images that produces the best result. From a practical point of view, there is a possibility of crashing into the structure if the offset distance is less than 4 ft and the smaller defects might get undetected if the offset distance is greater than 8 ft.

Field test

For our field test we used two grounded steel beams, (Beam A and Beam B - Figure 14), from the steel yard by the Virginia Tech's structures laboratory. For each beam, multiple videos were recorded to simulate multiple inspections taken at different inspection cycles. The goal was to observe the effectiveness of the three methods (RANSAC-Flow's coarse alignment, homography, and a hybrid of the two) on aligning image frames from the video captures. In this case, since the inspection videos were taken immediately after one another, a perfect method would produce 0% RMS pixel keypoint match error and the best-matched corrosion condition state prediction. This is because the image which is best-aligned would present the damage detection model with an image whose defects were closest in size and position as the target image. Through our experiment, we found that the homography alignment performed the best in every evaluation method in the field test. It had the lowest RMS pixel error for keypoint matches between the target and source image and the highest F1 and IoU score for the detectable corrosion damage condition states, Table 1.

		F1 ↑	IoU ↑	RMSE ↓
Beam A	Coarse	0.987	0.977	2.290
	Homography	**0.989**	**0.980**	**1.228**
	Hyrbid	0.988	0.978	2.304
Beam B	Coarse	0.958	0.924	2.804
	Homography	**0.974**	**0.952**	**1.899**
	Hybrid	0.957	0.921	2.899

Table 1. Summary of F1 scores, intersection over the union (IoU), and root mean squared error (RMSE) for Beam A and Beam B. The RMSE shown in the table is average RMSE across all frames sampled.

The F1 score and IoU scores utilized predictions from a trained corrosion damage condition state model. Figure 17 of the Appendix is an example of predictions from the corrosion model on Beam B. The diagram shown in Figure 17 of the

Appendix describes the prediction averaging process from the K number of 'best-matched' source images compared to the target image. The predictions on the target images were used as the ground truth predictions and they were compared against the average prediction for the three best-matched images after they were aligned to the target image. A heatmap of the average source ensemble outputs demonstrate how false-positive noise is reduced to produce a more consistent prediction.

The majority of the pixels in the video frames taken from both Beam A and Beam B did not contain damage, 98.5% and 87% respectively. This meant that the image sequences contained mostly background, thus, the F1 and IoU scores were all very high (close to 1.0) because they were buoyed by the large amount of background pixels. Therefore, the small differences between the F1 and IoU scores for each of the methods (coarse, homography, and hybrid) were actually were significant in terms of comparing performance. The significance of the small differences in the F1 and IoU scores are highlighted in Table 2 by breaking down the error between the number of detected corrosion pixels in the target and source images. There was a massive difference in F1 and IoU error found between Beam A and Beam B (Table 2). This was because there was a smaller sample of detected corrosion in Beam A (1.5%) compared to Beam B (13%). Therefore, the corrosion error tended to inflate drastically when there were any false-positive corrosion damage predictions. Still, Table 2 highlights that homograhpy was indeed the best method in terms of returning consistent model predictions.

	Coarse		Homography		Hybrid	
	Fair	Poor	Fair	Poor	Fair	Poor
Beam A	26.422	---	**20.447**	---	25.195	---
Beam B	0.573	0.080	**0.138**	**0.068**	0.568	0.188

Table 2. The average pixel quantity error across all frames for model detected damage. *Fair* and *Poor* corrosion condition states, which define the severity of damage on a scale of *Good* (background), *Fair*, *Poor*, *Severe*. Because the model did not find corrosion with a condition state of 'poor' on Beam A, it was excluded as indicated by '—'.

Proposing Usable Alignment Regions

Even though a RIR method was used to register the images, there will still always be image distortions present in the image. For bridge inspectors, knowing the overall extent of distortions in the aligned image would be important. Possibly even more useful would be knowing which specific regions within an aligned image were most reliable when analyzing the change between the inspections. Knowing this distortion information would build more trust between the inspector and algorithm.

It was observed in both the sensitivity study and the field test that certain areas or regions within the images did not align well, while others did very well. A reliable region has been defined by the authors in this paper as $0-5\,\text{pixel}$ error between detected keypoints, but could be at the discretion of the engineer. Some of these less-reliable regions also contained many inconsistent corrosion condition state model predictions. These observations motivated the us to investigate the regions within the target image and source images which poorly aligned. The goal was to take an aligned source image, and determine which regions ~~or sub-images,~~ were most reliable based on the keypoint match error from the target image.

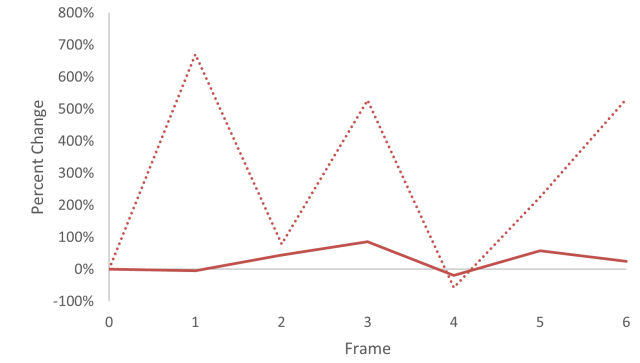
To achieve this, the ensemble of source images were divided into a grid of sub-images. Each of the sub-images from the target image were compared with the sub-images from the new aligned image to determine how well each of them matched. In each of the sub-images, the mean pixel error difference from the matched keypoint pairs was calculated (keypoint pairs were found using the SuperGlue algorithm). If a sub-image did not contain any keypoints, then that sub-image was deemed to be unreliable.

The criteria for determining a reliable sub-image would be at the discretion of the inspector, where they would have the ability to choose the acceptable pixel error threshold. Figure 16 of the Appendix depicts how the ensemble of source images were divided and whittled down to show only the most reliable sub-images to an inspector. Then the average masked image from the ensemble was used for making final predictions. This technique was applied to a subset of images from two videos taken of Beam B (Figure 18 of the Appendix). A perfect result would be that there would be no change between the two videos since they were taken of the same beam. Using the reliable sub-image approach, figures 10a and 10b show that there was a much more stable damage prediction (closer to 0% change) for both fair and poor corrosion condition states.

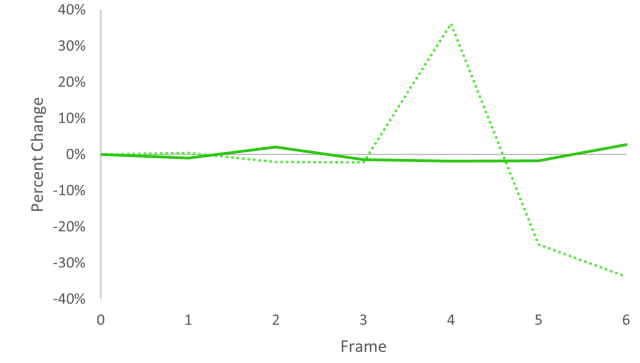
Simulated Demonstration

The authors wanted to demonstrate how the method would respond in practice when a new defect was found, or if there was progression of existing damage. To do this we would have had to either record and return to an existing bridge element after deterioration was noticeable, or create an environment to physically corrode the test-beam ourselves - both would have been costly. Instead, the authors chose to simulate the change by altering a single video frame containing an existing damaged location on beam B (Figure 14 of the Appendix). Figure 11 describes editing process to create the edited data pair.

The simulation was performed by altering a known corrosion damage location in one of the inspection video frames, frame 3 of Figure 18, and roughly restoring it as shown in Figure 11. The single frame was synthetically edited, where the damage was effectively removed using a state of the art in-painting method, the recurrent feature



(a)



(b)

Figure 10. Corrosion condition state *fair* (a) and *poor* (b) model prediction error between target and source images. The dotted line refers to the predictions on the entire image, solid line refers to the predictions on only regions which had an average keypoint error $\leq 5\,pixels$. Frames were used from the Beam B data subset (Figure 18).

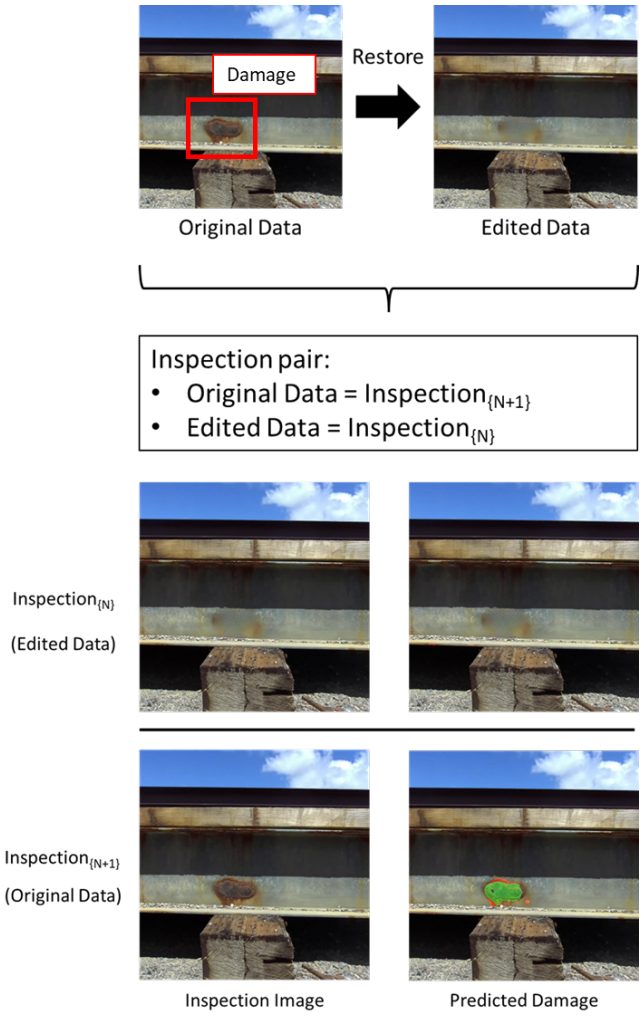


Figure 11. Inspection pair of $\texttt{inspection}_{\{N\}}$, and $\texttt{inspection}_{\{N+1\}}$ (Frame 3 in Figure 18 of the Appendix).

reasoning (RFR) in-painting (*32*), trained by the authors on structural bridge data. The top row in Figure 18 of the Appendix serves as the $\texttt{inspection}_{\{N\}}$ and the bottom row image from Figure 18 serves as $\texttt{inspection}_{\{N+1\}}$. Frame 3 from Figure 18 is highlighted in Figure 11, contrasting the edited data with the original data.

Clearly, the damage from $\texttt{inspection}_{\{N+1\}}$, as shown in Figure 11, was detectable and was significantly more than $\texttt{inspection}_{\{N\}}$ by a qualitative observation. This qualitative observation is quantified in the percent change bar graph shown in Figure 12. This change was especially exaggerated (38,000%) since the algorithm detected a condition state ("poor") in $\texttt{inspection}_{\{N+1\}}$ that did not exist in the prior $\texttt{inspection}_{\{N\}}$. In reality the spike would not be localized to a single frame, as indicated by our example in Figure 12, but it would be more like a parabola since the first detection of the new or progressed damage would only be on the edge of an extracted frame.

The spikes in change would indicate that there was a series of noteworthy frames, which correspond to a time range in the video. Each of these flagged time ranges could be quickly sorted through and checked by the inspector instead of watching the entire lengthy inspection recording. These shortened video segments would also help the inspector maintain a higher level of vigilance as opposed to reviewing the entire recorded video. Again, this simple example in Figure 12, when extrapolated on a large-scale, could prove to be quite important when inspection videos are several hours long, where fatigue may otherwise cause an inspector to miss new or significant changes from past inspections.

CONCLUSIONS

In this paper we have proposed a method to align two bridge inspection videos for damage progression and change detection. We developed a method for aligning target images

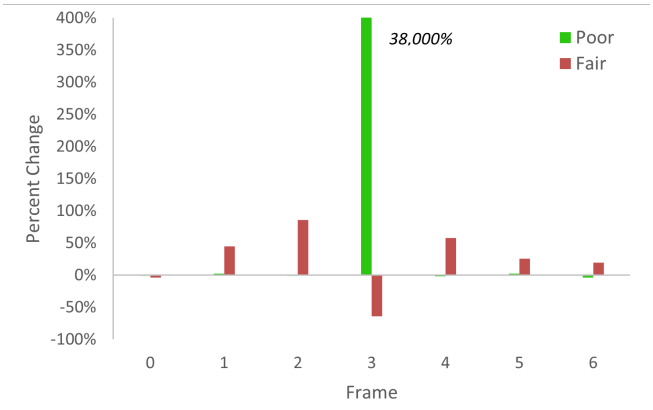


Figure 12. Percent change of corrosion condition states between two inspections.

from a prior inspection video to a current inspection video using a keypoint matching algorithm, SuperGlue. We compared a state of the art optical flow method, Ransac-Flow, against a traditional method, homography, when applied to a bridge inspection image registration problem. Through our experiments we found that homography produces the least amount of pixel-level errors and gives the best fidelity for our damage condition state segmentation model. Additionally, homography transformations did not require the large GPU requirements, which limited the image size, needed for the RANSFAC-Flow algorithm. We have also observed that having images taken from offsets of $5 - 7$ ft tends to produce the lowest errors. We expect that by implementing this algorithm we can increase the efficiency and the accuracy of infrastructure inspection tasks and save these structures from critical or fatal failures.

Future Work

An obvious next step in this research would be to pursue a real-time image-alignment feedback control-loop during the video capturing process. Effectively, if the match strength was controlled at the collection source, the matches would be much stronger in the post-process image warping. This could be achieved by using inputs from the prior inspections like depth information and camera pose, coupled with control loop feedback, like the number of SuperGlue keypoints, detailing the strength of the camera position. Maintaining a consistent positioning of the camera attitude and offset from target surface between inspections is especially crucial since the necessary method is a form of rigid image registration.

References

1. Morgese, M., F. Ansari, M. Domaneschi, and G. Cimellaro. Post-collapse analysis of Morandi's Polcevera viaduct in Genoa Italy. *Journal of Civil Structural Health Monitoring*, Vol. 10. doi:10.1007/s13349-019-00370-7.
2. Brantley, M. I-40 bridge fix may take 'considerable amount of time,' senator says. Crack 'significant,' engineers say. *Arkansas Times*. URL https://arktimes.com/arkansas-blog/2021/05/12/i-40-bridge-fix-may-take-considerable-amount-of-time
3. Greenwood, W. W., J. P. Lynch, and D. Zekkos. Applications of UAVs in Civil Infrastructure. *Journal of Infrastructure Systems*, Vol. 25, No. 2, 2019, p. 04019002. doi:10.1061/(ASCE)IS.1943-555X.0000464.
4. Chen, S.-E., C. Rice, C. Boyle, and E. Hauser. Small-Format Aerial Photography for Highway-Bridge Monitoring. *Journal of Performance of Constructed Facilities*, Vol. 25, No. 2, 2011, pp. 105–112. doi:10.1061/(ASCE)CF.1943-5509.0000145. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CF.1943-5509.0000145. https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CF.1943-5509.0000145.
5. Reagan, D., A. Sabato, and C. Niezrecki. Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges. *Structural Health Monitoring*, Vol. 17, No. 5, 2018, pp. 1056–1072. doi:10.1177/1475921717735326. URL https://doi.org/10.1177/1475921717735326. https://doi.org/10.1177/1475921717735326.
6. Sarlin, P.-E., D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4937–4946. doi:10.1109/CVPR42600.2020.00499.
7. Shen, X., F. Darmon, A. A. Efros, and M. Aubry. RANSAC-Flow: generic two-stage image alignment, 2020. 2004.01526.
8. Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*. 2018.
9. Sand, P. and S. Teller. Video Matching. *ACM Trans. Graph.*, Vol. 23, No. 3, 2004, p. 592–599. doi:10.1145/1015706.1015765. URL https://doi.org/10.1145/1015706.1015765.
10. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, Vol. 60, No. 2, 2004, p. 91–110. doi:10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.
11. Cui, Z., O. Wang, P. Tan, and J. Wang. Time Slice Video Synthesis by Robust Video Alignment. *ACM Trans. Graph.*, Vol. 36, No. 4. doi:10.1145/3072959.3073612. URL https://doi.org/10.1145/3072959.3073612.

12. Baraldi, L., M. Douze, R. Cucchiara, and H. Jegou. LAMV: Learning to Align and Match Videos with Kernelized Temporal Layers. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7804–7813. doi:10.1109/CVPR.2018.00814.
13. Abobeah, R., A. Shoukry, and J. Katto. Video Alignment Using Bi-Directional Attention Flow in a Multi-Stage Learning Model. *IEEE Access*, Vol. 8, 2020, pp. 18097–18109. doi:10.1109/ACCESS.2020.2967750.
14. Revaud, J., M. Douze, C. Schmid, and H. Jegou. Event Retrieval in Large Video Collections with Circulant Temporal Encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.
15. Crum, W., T. Hartkens, and D. Hill. Non-rigid image registration: Theory and practice. *The British journal of radiology*, Vol. 77 Spec No 2, 2004, pp. S140–53. doi:10.1259/bjr/25329214.
16. Yuen, J., J. Barber, A. Ralston, A. Gray, A. Walker, N. Hardcastle, L. Schmidt, K. Harrison, J. Poder, J. R. Sykes, and M. G. Jameson. An international survey on the clinical use of rigid and deformable image registration in radiotherapy. *Journal of Applied Clinical Medical Physics*, Vol. 21, No. 10, 2020, pp. 10–24. doi:https://doi.org/10.1002/acm2.12957. URL https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/acm2.12957. https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/acm2.12957.
17. Brock, K. K., S. Mutic, T. R. McNutt, H. Li, and M. L. Kessler. Use of image registration and fusion algorithms and techniques in radiotherapy: Report of the AAPM Radiation Therapy Committee Task Group No. 132. *Medical Physics*, Vol. 44, No. 7, 2017, pp. e43–e76. doi:https://doi.org/10.1002/mp.12256. URL https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12256. https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/mp.12256.
18. Griffith, S., F. Dellaert, and C. Pradalier. Transforming multiple visual surveys of a natural environment into time-lapses. *The International Journal of Robotics Research*, Vol. 39, No. 1, 2020, pp. 100–126. doi:10.1177/0278364919881205. URL https://doi.org/10.1177/0278364919881205. https://doi.org/10.1177/0278364919881205.
19. Griffith, S., G. Chahine, and C. Pradalier. Symphony Lake Dataset. *The International Journal of Robotics Research*, Vol. 36, No. 11, 2017, pp. 1151–1158. doi:10.1177/0278364917730606. URL https://doi.org/10.1177/0278364917730606. https://doi.org/10.1177/0278364917730606.
20. Tuytelaars, T. and L. Van Gool. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. doi:10.5244/C.14.38.
21. Cech, J., J. Matas, and M. Perdoch. Efficient Sequential Correspondence Selection by Cosegmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 32, No. 9, 2010, p. 1568–1581. doi:10.1109/TPAMI.2009.176. URL https://doi.org/10.1109/TPAMI.2009.176.
22. Bian, J., W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng. GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2828–2837. doi:10.1109/CVPR.2017.302.
23. Fischler, M. A. and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, Vol. 24, No. 6, 1981, p. 381–395. doi:10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692.
24. Yi, K. M., E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.). Springer International Publishing, Cham, 2016, pp. 467–483.
25. DeTone, D., T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 337–33712. doi:10.1109/CVPRW.2018.00060.
26. Dusmanu, M., I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8084–8093. doi:10.1109/CVPR.2019.00828.
27. Ono, Y., E. Trulls, P. Fua, and K. M. Yi. LF-Net: Learning Local Features from Images. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, p. 6237–6247.
28. Sinkhorn, R. and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, Vol. 21, 1967, pp. 343–348.
29. DeTone, D., T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–33712.
30. Yi, K., E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. 2016, pp. 467–483. doi:10.1007/978-3-319-46466-4_28.
31. Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544.
32. Li, J., N. Wang, L. Zhang, B. Du, and D. Tao. Recurrent Feature Reasoning for Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

APPENDIX

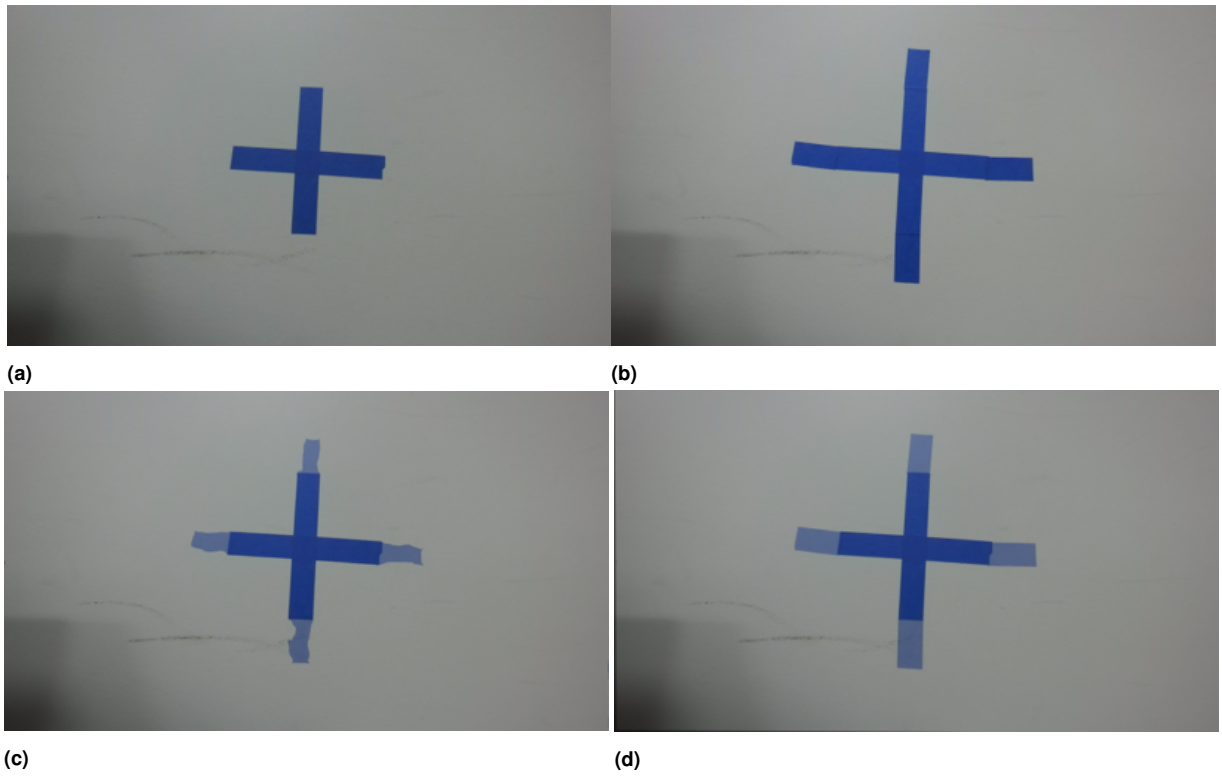


Figure 13. Target image *default* (a), source image *growth* (b). Overlay of warped image and target image using (a). RANSAC-Flow fine-tune warped image (c) and RANSAC-Flow coarse warped image (d) - a perfect alignment would be the overlay of (a) and (b) image.

Figure 14. Target images: beam A (top) and beam B (bottom).

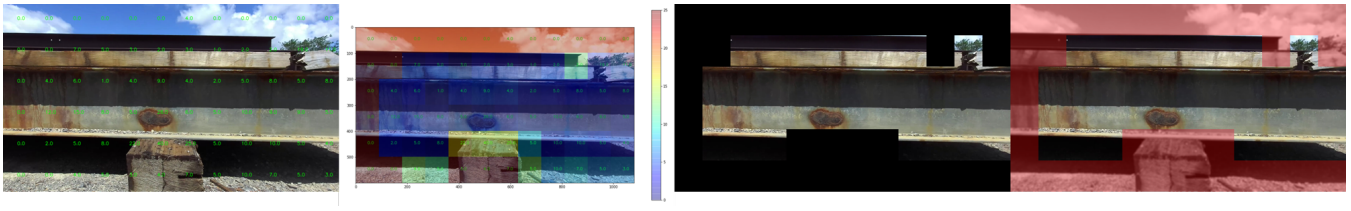


Figure 15. Usable regions on Beam B, $\texttt{target}_{\{0\}}$. The left image shows the number of keypoints in each sub-image grid. The next image to the right shows an error heatmap between the target and source image. The following image masks the unreliable regions which have an average of a $5\,\mathrm{pixel}$ error or greater. The final image includes the background for context, but blurs and tints the unreliable

regions red.

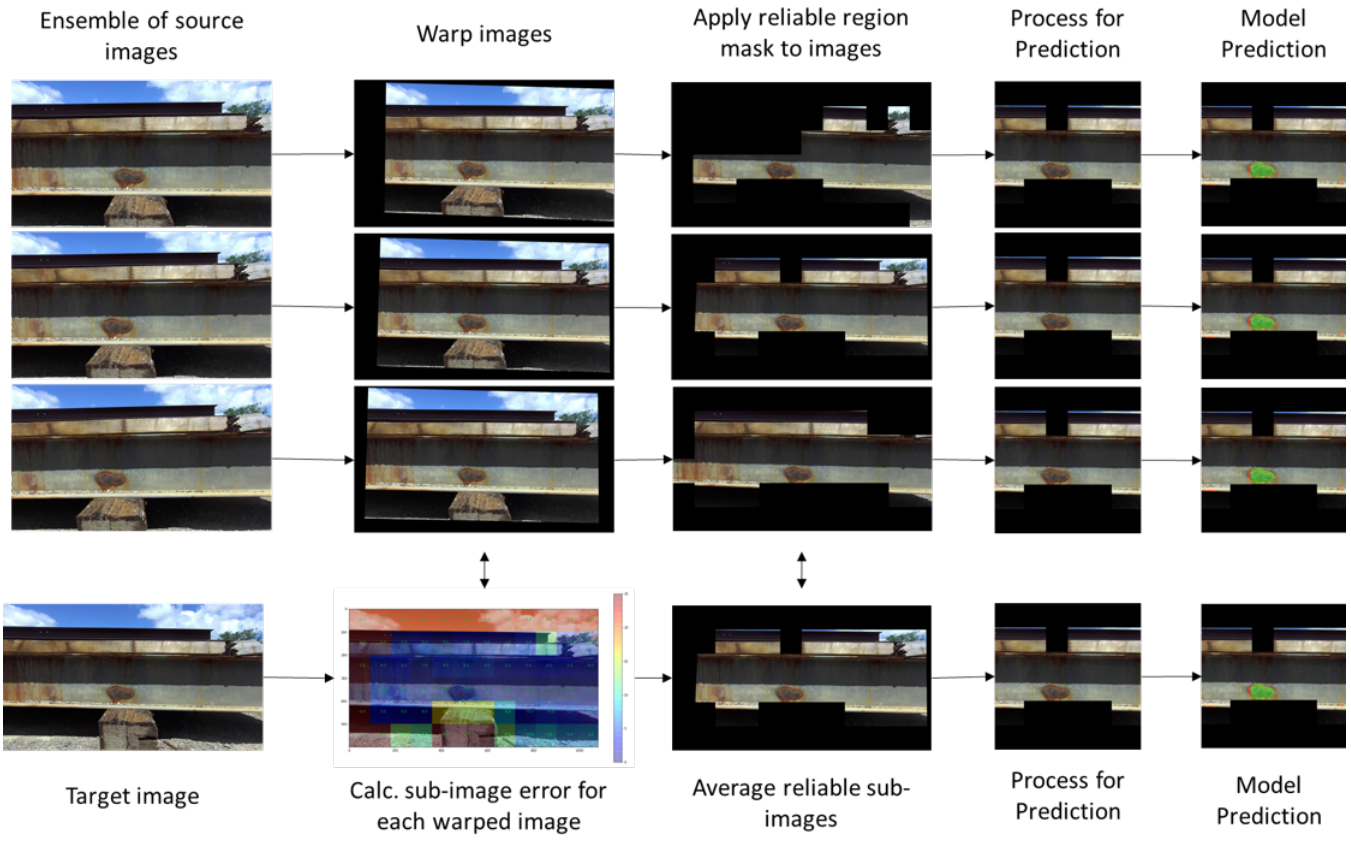


Figure 16. Flow diagram of the image processing once candidate images have been identified. Three candidate best matched images are selected. The source images are warped to the target image. The average keypoint error for each sub-image is determined using SuperGlue. A mask is generated for all sub-image errors greater than the set threshold. The average mask is used. The source and target images are processed and cropped for preparation of the damage detection model. The damage detection model is run on the masked source and target images, and the results from the source images are averaged and compared for changes in the progression of condition states and damage area.

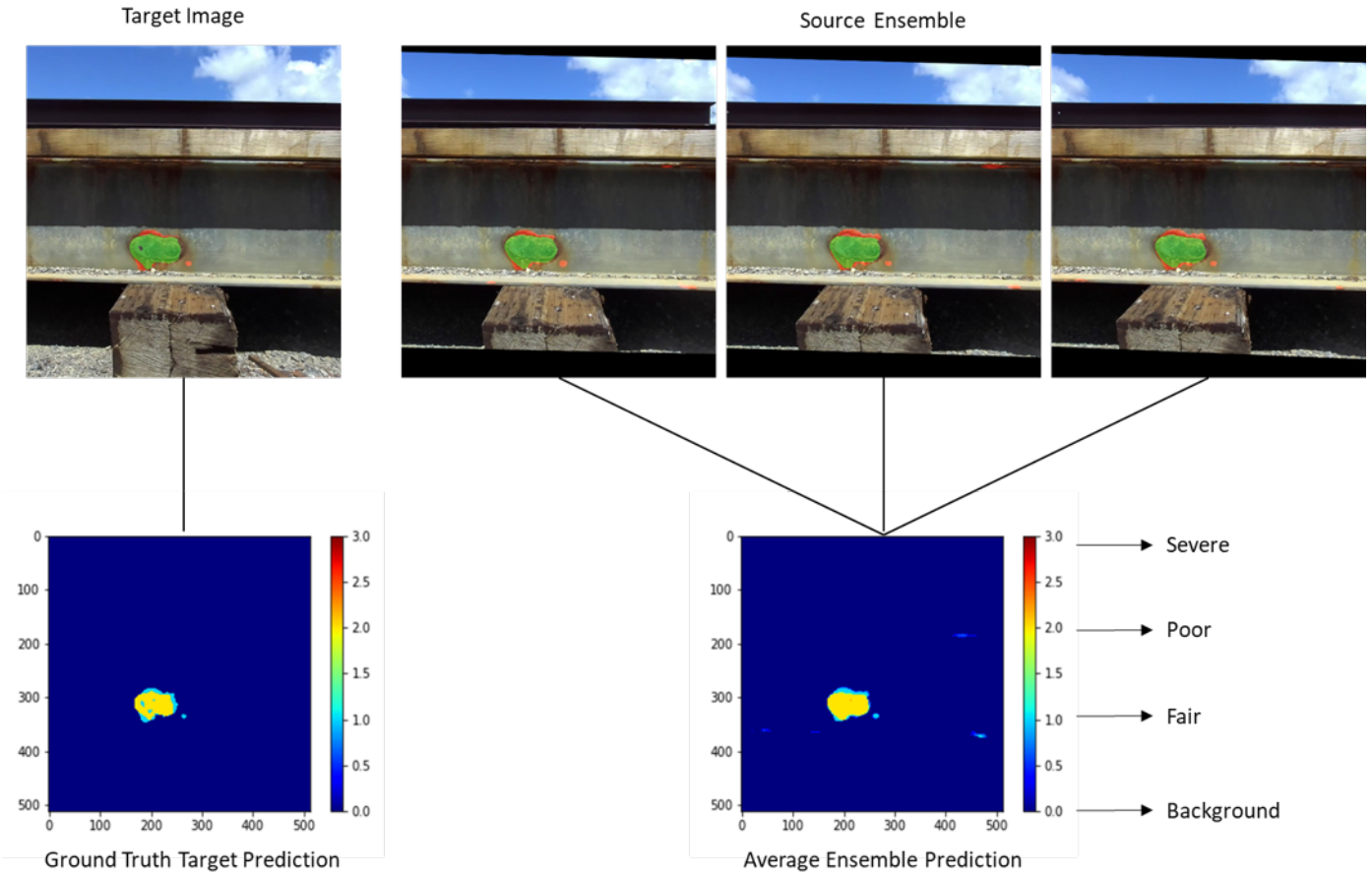


Figure 17. Comparing an ensemble of predictions from $\texttt{inspection}_{\{N+1\}}$ to $\texttt{inspection}_{\{N\}}$. Ground truth target image (top-left), and top three best-matched source images with their respective homography transformations and predictions (top-right).

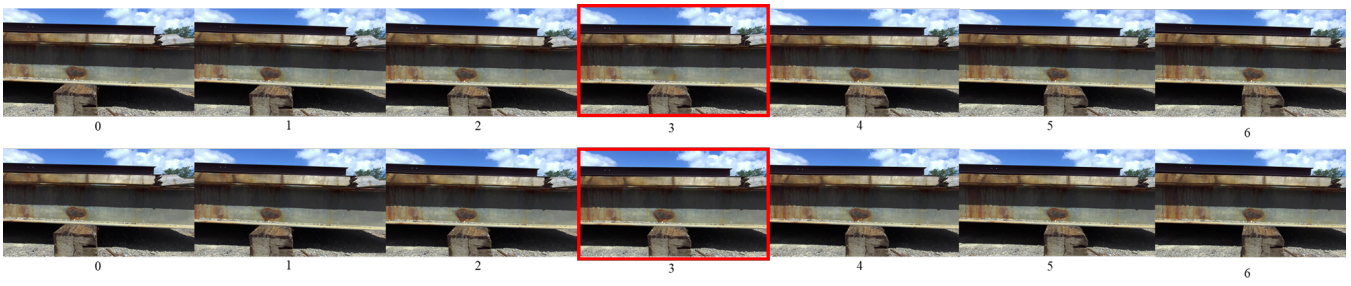


Figure 18. Top row is the $\texttt{inspection}_{\{N\}}$ subset of target images, it contains an edited image by in-painting the damage using the RFR in-painting algorithm. The bottom row is the $\texttt{inspection}_{\{N+1\}}$ subset of source images. The values below the images indicate the frame number. Frame 3 is outlined in red because it is the frame which is altered.