Improving Robustness of Homography Estimation for Ice Rink Registration

by

Jia Cheng Shang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 3 contains content from the previously published paper [34], of which I was the main author.

Chapters 2 and 4 contains content from a paper currently under review at ACM MMSports '23 named "Rink-Agnostic Hockey Rink Registration," of which I was the main author.

Abstract

Hockey analytics involves obtaining information from games so that coaches, managers, and teams can make better decisions in training, playing, and hiring. As there is a large amount of information available in each game, manual analysis is difficult and tedious, so automated computer vision techniques have been developed to acquire and process data more efficiently.

One key component to such analysis is the location information of players and events. This information can be obtained using a technique called rink registration, which involves estimating the homography matrix needed to warp an overhead template of the rink onto video frames, or vice versa. By doing this, we can obtain the location of objects in video with respect to the fixed reference frame of the overhead template. Current methods focus on NHL rinks, which have a standardized size and have similar appearances. However, the quality of results drop when other types of rinks are used, because the existing methods are not trained to work on non-NHL rinks.

This work seeks to improve the rink registration process by making it more robust to differences in rinks, while maintaining good accuracy. It also tries to develop a generalized system that can work on a variety of rink types, such as NHL, Olympic, and European, without the need for additional rink-specific training or expensive annotations. By developing this rink-agnostic system, it can provide rink registration results regardless of rink, making analysis more equitable for smaller groups. It also reduces the cost needed as it only requires broadcast video and the overhead rink template, without the need for additional technology to be installed or annotations to be made. The results of this rink-agnostic system are competitive with the results of an NHL-only baseline on NHL rinks and are noticeably better than the baseline on non-NHL rinks. The rink-agnostic system achieves a 1.1% IOU_{part} improvement on the Olympic 2014 rink and a 8.8% IOU_{part} improvement on the Berlin Mercedes-Benz Arena rink.

Acknowledgements

I would like to thank my supervisors Prof. David Clausi and Prof. Javad Shafiee for their guidance and support. Thanks to Mehrnaz Fani for helping me get started in research for computer vision. Thanks to everybody in the Vision and Image Processing Lab for making it a great place to research for the past 2 years.

Thanks to my friends and family for being with me through thick and thin.

Dedication

This is dedicated to my friends and family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Premise

Sports analytics is a field that involves extracting information from sports games to analyze and improve player and team performance. It has grown and advanced as technology developed to cater to the large sports market. Different sports are popular around the world with millions cheering at events of various scales, from local teams playing all the way up to games played at the international stage. One such sport is hockey. It has a large following especially in North America, where the NHL's yearly revenue was $2.3 billion in the 2020/21 season [13]. Each hockey team strives to outperform each other, and sports analytics can help with this by providing statistics about individuals and teams.

One field that can be linked to hockey analytics is computer vision, which analyzes images and videos to obtain information. The vast quantity of data from games means these are difficult to process manually, so improvements in computer vision can help automate this process. Computer vision techniques can provide information that would otherwise be almost unobtainable due to the sheer scale of the data and the fast paced nature of the game. This extracted information is then used for further analysis.

1.2 Motivation

Hockey analytics can provide teams and managers with information about various players and their interactions. This can affect training, coaching, and hiring decisions, such as

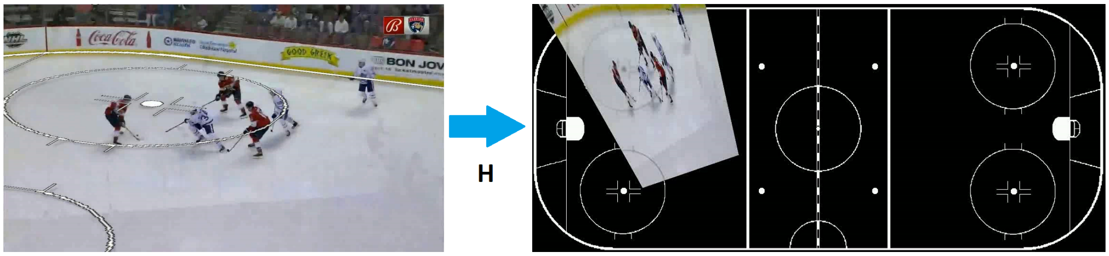


Figure 1.1: Warping frame to overhead. The inverse of the homography matrix can be used to warp overhead to frame.

determining areas of improvement for players, or developing new strategies. It can also affect minor leagues or junior leagues, as NHL teams recruit players from these leagues based on analysis of their performance. In such a competitive sport with high revenue and popularity, each team will definitely try to use whatever they can get to outperform each other, and data-driven solutions are one way to optimize player and team performance. For example, statistics involving shots, possession, and blocking are measured to determine player performance [28].

Providing automatic analysis from broadcast video allows for information to be obtained regardless of rink. This improves access to such information for smaller leagues or teams. They do not need to have specialized technology in their rinks and do not need to annotate data manually, which would have required large amounts of time, funding, and effort. This makes access to the information more equitable for smaller groups and minor leagues, while also saving time and cost for larger teams.

1.3 Scope

This thesis primarily focuses on the task of rink registration. This involves estimating the homography transformation required to warp an overhead template to the broadcast video feed, or vice-versa using the inverse transformation. Once the homography matrix is obtained, the true location of everything in the video frame can be identified with respect to a fixed reference frame. An example of rink registration can be seen in Figure 1.1.

This is an integral part of hockey analytics, as knowing the location of the players and events is one of the first steps for further analysis, such as determining missed opportunities. The only input required is the broadcast video feed and the overhead template of the rink.

This thesis first describes improvements that are made to an existing homography estimation work. It then describes a method that generalizes homography estimation to work on all rink types. This rink-agnostic system would work on multiple rink types without the need for additional rink-specific training or the need for expensive annotations of non-NHL data. The system is tested on non-NHL rinks such as Olympic or European rinks, without additional training or setup. Finally, this work will discuss a method to generate the overhead rink templates that are used in some homography estimation methods, for cases where the rink template may not be readily available. Overall, this thesis seeks to make advancements in the field of homography estimation for the purpose of rink registration, and generalizes it to be more robust to rink type and appearance.

Chapter 2

Background and Related Works

This chapter covers works relating to the field of homography estimation for the purposes of sports field registration.

2.1 Homography Estimation

Homography is an aspect of image processing where one plane is warped onto another plane. It can be used in a variety of different tasks, such as image stitching and structure from motion [38]. It can also be used to describe how a plane can be viewed from different perspectives, as seen in Figure 2.1.

The matrix itself is a 3 × 3 matrix, H, that has eight degrees of freedom (DoF), and relates the transformation between two planes up to a scale factor. Specifically, the matrix transforms points in one plane (x, y) to points on another plane (x', y'), with an unknown scaling s. To remove this scaling, homography matrices are often normalized so that the final value, h_{33}, is fixed to 1. When used on a set of points of one plane, it is able to calculate their locations in another plane, as seen in equation 2.1. In rink registration, it can be used to warp the broadcast frame points onto an overhead template, or vice-versa using the inverse matrix. In 2.1, $[x, y, 1]^T$ can represent the frame pixels, while $[x', y', 1]^T$ would represent coordinates in the overhead template space.

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

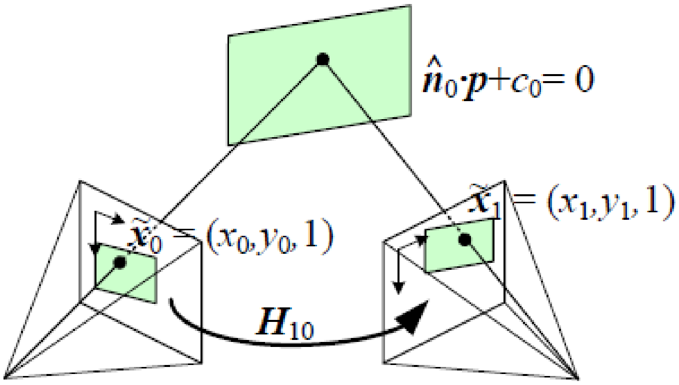

$\hat{n}_0 \cdot p + c_0 = 0$
$\tilde{x}_0 = (x_0, y_0, 1)$
$\tilde{x}_1 = (x_1, y_1, 1)$
H_{10}

Figure 2.1: How a plane can be viewed from two different perspectives, and the homography that can be used to warp between the perspectives. From Szelinski et al. [38].

Homography transformations are a superset of affine transformations such as rotation, translation, scaling, and shearing [14]. However, while affine transformations keep straight lines straight and parallel lines parallel after the transformation, homography transformations only need to keep straight lines straight.

Traditional homography estimation techniques often involve identifying feature pairs from image pairs using methods such as SIFT [27] and ORB [32]. Once we obtain at least four matches, direct linear transformation (DLT) can be used to calculate the terms of the homography matrix [14]. To handle noise and possible mismatches, the feature matching stage acquires many point matches and uses random sample consensus (RANSAC) to determine accurate sets of 4-point matches to feed into the DLT [9] .

Homography could also be estimated directly from two images if they were close enough in perspective and appearance. This can be done using pixel-based direct homography optimization, as described by Szeliski et al. [39]. One method involves using gradient descent to minimize the sum of squared differences (SSD) between pixels. To improve this process when there is a larger difference in position between images, a hierarchical image pyramid is set up. The optimization is first performed at a coarse level before being performed at finer levels. However, even with this, the difference between images must be small, to ensure there is a large enough overlap to perform direct homography optimization effectively. Furthermore, this can still take awhile to perform, which makes it difficult to use when speed is a concern.

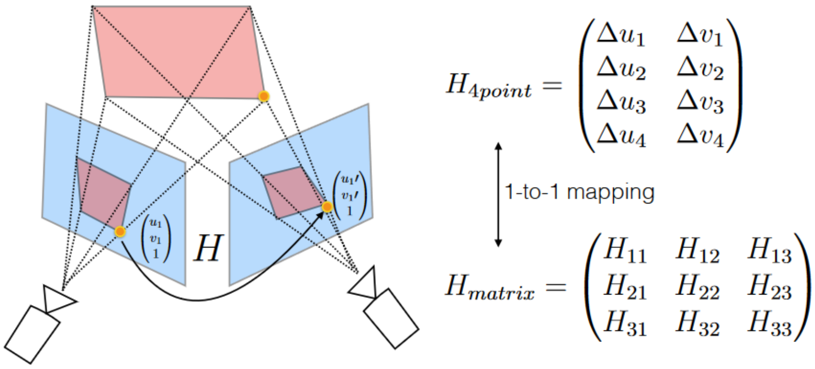

$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix}$
1-to-1 mapping
$H_{matrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$
$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$
$\begin{pmatrix} u_1' \\ v_1' \\ 1 \end{pmatrix}$
H

Figure 2.2: How the 4-point method from Detone et al. works [8]. Four points can be converted into a homography matrix using DLT. Δu and Δv represent the differences between the points in the image, such that $\Delta u = u' - u$, $\Delta v = v' - v$.

Aside from these traditional techniques, deep learning methods were also developed. DeTone et al. [8] were one of the first to estimate homography via deep learning. They used a Visual Geometry Group (VGG) based model to estimate the location of four corners of one image in the image space of the other image. These sets of point estimates can then be converted into a homography via DLT [14] [8], as matching four (x,y) pairs is enough to solve the eight unknowns in the matrix. An example of this approach can be seen in Figure 2.2.

Zhou and Li [48] describe how directly estimating homography parameters via deep learning was difficult due to the different scaling needed for each parameter. They described that as the reason for the 4-point approach used by DeTone, and proposed an alternate method where the normalized homography was directly estimated instead. By using normalization, the homography matrix parameters were altered to have relatively similar distributions, making them more suitable for the loss functions used by deep learning models [48].

2.2 Homography Estimation in Sports

In hockey rink registration and other sports registration tasks, homography estimation is difficult using traditional approaches [29]. Hockey rinks can be quite featureless, with markings being sparse or occluded by players. Thus traditional feature detection models

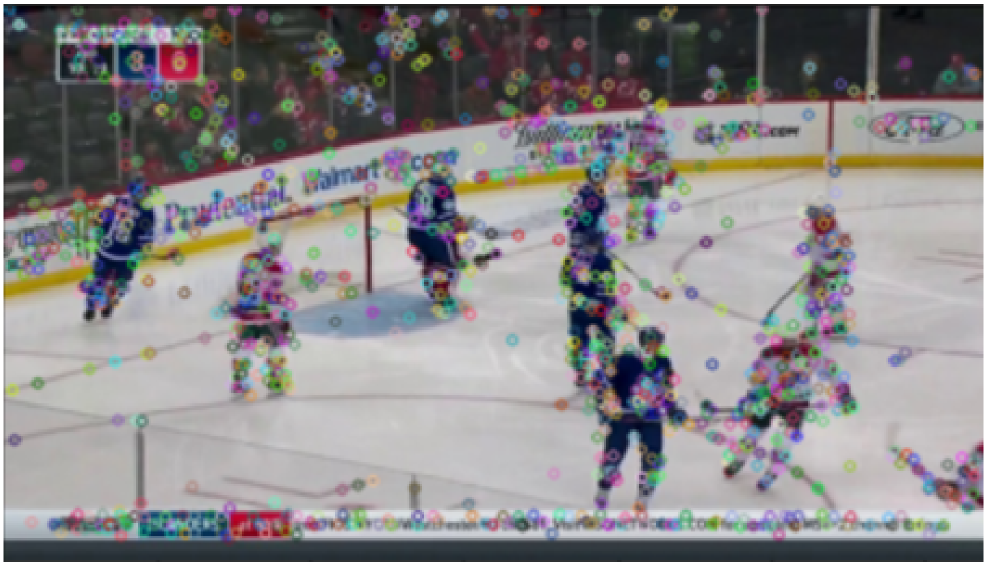


Figure 2.3: SIFT [27] features on a hockey broadcast frame. Unfortunately, the vast majority of detected features do not relate to rink features.

often choose points on players, audience, or shadows, which are not desirable because these are not part of the fixed rink template. This issue is shown in Figure 2.3.

Direct homography estimation using pixel-based alignment is also not viable due to its long calculation time and the fact that the inputs needs to be close together in perspective [39]. The movement of the players and audience, along with screen elements such as scores and advertising may also cause incorrect solutions to be generated.

With the advent of deep learning, new models have been developed for homography estimation. Various models and pipelines specialized for sports field registration were also developed to overcome the above issues. Homayounfar et al. [15] utilize semantic segmentation to isolate field marking data to use in a Markov random field. Chen and Little [3] set up a camera-pose database with predefined poses. Then they select the best pose by comparing it to features and edge images extracted from the input image, before refining the pose. This can be seen in Figure 2.4. Sha et al. [33] also use a dataset system, except they replace the edge images with semantic segmentation maps for use in the comparison.

Nie et al. [29] use a U-net model to estimate the location of a series of keypoints uniformly spread across the rink template, which can then be used to calculate the homog-

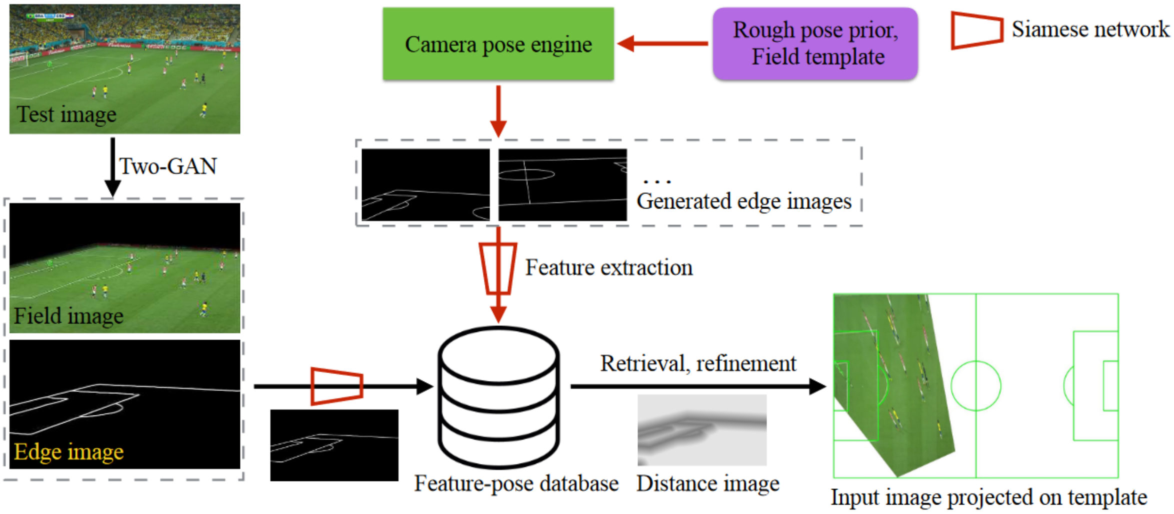


Figure 2.4: Pipeline for the dataset approach described by Chen and Little [3].

raphy. This keypoint-based estimation is then further refined based on feature heatmaps extracted from the input image, alongside the previous input frame's heatmaps [29]. This approach is visualized in Figure 2.5. The keypoint method was expanded upon by Chu et al. [7], who used better keypoint estimation based on dynamic filter learning, and removed the refining step as they found it no longer necessary.

Jiang et al. [21] use a deep neural network to estimate the locations of four points of the input frame on an overhead template, similar to Detone et al. [8]. This estimated homography is used to warp the template, resulting in a second input that can be used by a refinement network. The refinement model then calculates the relative homography between the original image and the initial estimate [21]. This approach was improved by Shi et al. [35], who utilize a self-supervised method of warping dataset images to provide a greater set of synthetic data for training.

2.3 Metrics To Assess Homography

Intersection over union (IOU) is a common metric in the field of computer vision when identification of areas is being performed, such as segmentation and homography. The area of the ground truth is compared with the area from the estimate, and the ratio of

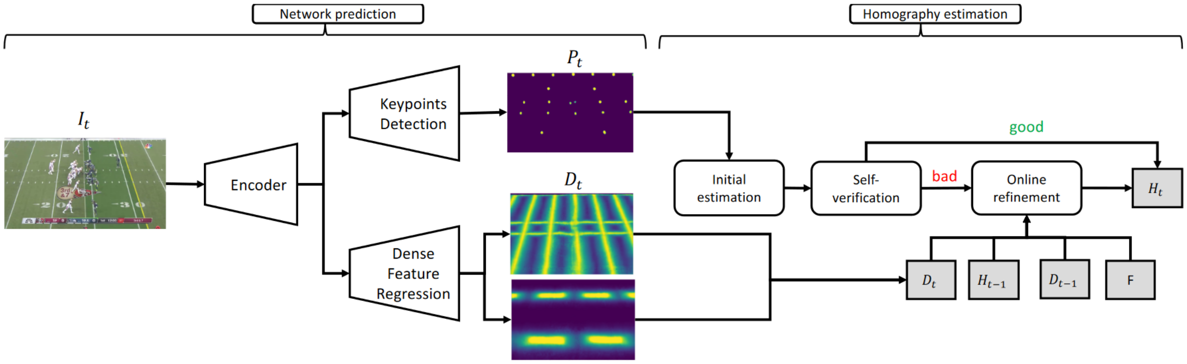


Figure 2.5: Pipeline for the keypoint approach described by Nie et al. [29].

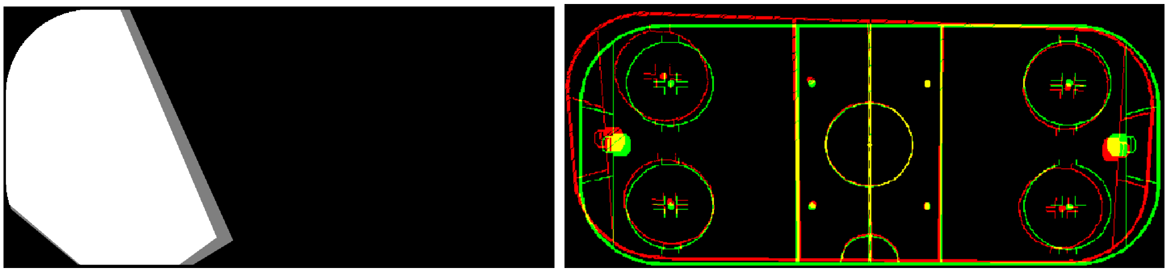
the intersection over the union of the areas is returned. For evaluating the quality of homography in sports registration, two variants are used: IOU_{part} and $\text{IOU}_{\text{whole}}$.

IOU_{part} involves warping the image frame onto the overhead template using the homography estimate and the homography ground truth. The areas are then compared via IOU. In this case, only the area of the rink visible in the frame is compared.

For $\text{IOU}_{\text{whole}}$, the ground truth homography is matrix multiplied with the inverse of the estimated homography matrix. The resulting matrix product can then be used to warp the overhead template to show the differences between the homographies even on portions out of view of the camera. An example showing both IOU_{part} and $\text{IOU}_{\text{whole}}$ can be seen in Figure 2.6. Note that both IOU types use the areas of the rink rather than the area covered by only the lines in the rink.

2.4 Conclusion

This chapter covers different works relating to the problem of homography estimation for the purpose of rink-registration. It also describes some of the background metrics involved, namely IoU_{part} and $\text{IoU}_{\text{whole}}$. Different techniques relating to homography estimation have achieved competitive results, and the lack of publicly available code and data makes it difficult to compare them fully. However, the fact that different techniques exist show that there is no consensus on the proper approach as of yet. Furthermore, there are other ways to contribute to the field apart from just improving accuracy, and we make such a contribution with a rink-agnostic homography estimation pipeline.



(a) $\mathrm{IOU}_{\mathrm{part}}$

(b) $\mathrm{IOU}_{\mathrm{whole}}$

Figure 2.6: $\mathrm{IOU}_{\mathrm{part}}$ vs $\mathrm{IOU}_{\mathrm{whole}}$ for hockey rink registration. The slight grey regions in image (a) around the white shape show the region that is not overlapped for $\mathrm{IOU}_{\mathrm{part}}$. For image (b) we see that the red region is not overlapping the green region at the edges, even though the center region is relatively aligned.

Chapter 3

Improving Homography Estimation

We first replicate a recent model from Shi et al. [35] to determine the efficacy of existing techniques on our data. We will then describe some contributions that improve the augmentations and pipeline to increase model accuracy.

3.1 Method

The method used in this section is based on Shi et al. [35]. However, no data or code is publicly available, so the reproduced method and model itself has been replicated based on available information from the paper but cannot be assured to be exactly the same.

A two part approach is used to estimate the homography of incoming broadcast feed frames needed to warp the frames onto the overhead template (and vice-versa). The first stage is an initial estimator module, which takes the original video frame, and identifies an estimate of the homography. This estimate is used to warp the overhead template and produces a warped template. This is fed alongside the original frame as input into the refinement module. The refinement module produces a homography that corrects the initial homography estimate, and can be combined with it to produce the final homography we want. The full pipeline is shown in Figure 3.1.

The initial estimator module is a Resnet18-based regressor that takes in the broadcast frame as input, and estimates the location of four frame points on the overhead template, similar to previous works such as Detone et al. [8], Jiang et al. [21], and Shi et al. [35]. The points used are the two bottom corners of the image, and two points 60% of the way up the image sides. The top portion was avoided to ensure the points were on the

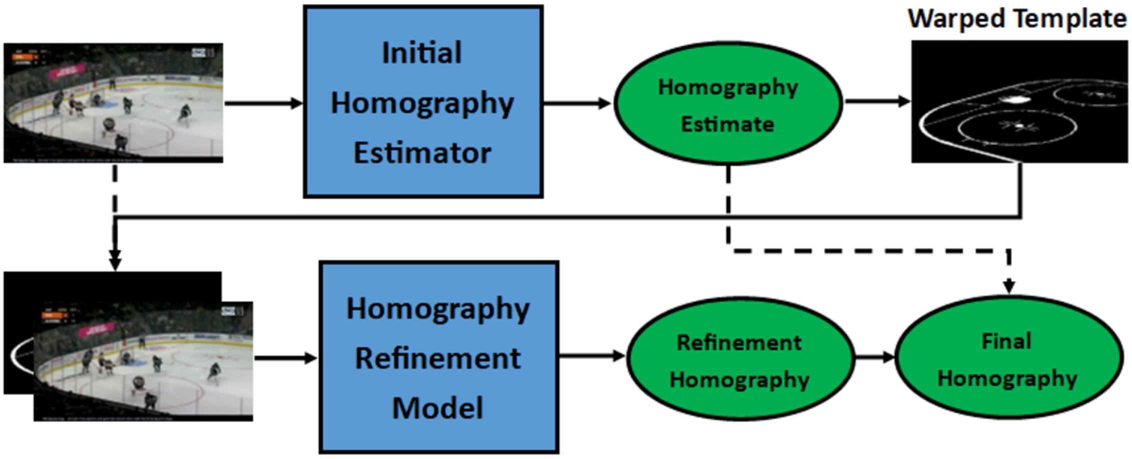


Figure 3.1: Full pipeline of the initial estimator + refinement model approach. It consists of two models: one that estimates an initial homography, and one that refines the estimate.

rink itself, which is usually closer to the bottom of the image for broadcast video, rather than on the audience. These four points and their estimated new locations can be used to calculate homography using the direct linear transformation (DLT) [14]. This generates the homography needed to warp from the frame to the overhead, and the inverse matrix is used to warp the overhead template to frame view to generate a warped template. This process can be seen in Figure 3.2.

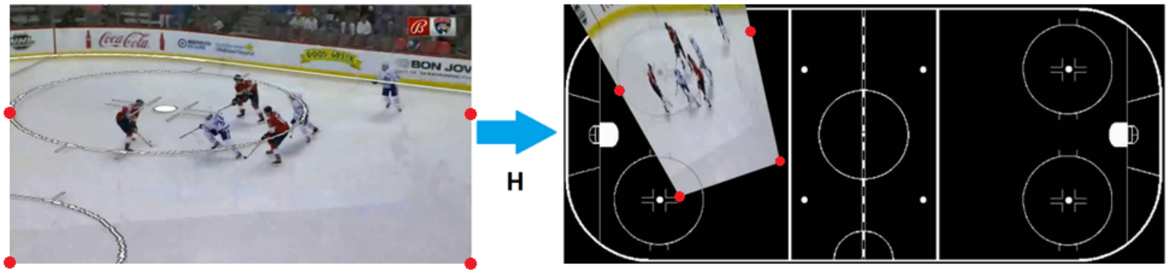


Figure 3.2: Estimating the location of four points from the initial image in the overhead view. Once the four points in the overhead frame are obtained, DLT can be used to obtain the homography needed to perform the warp.

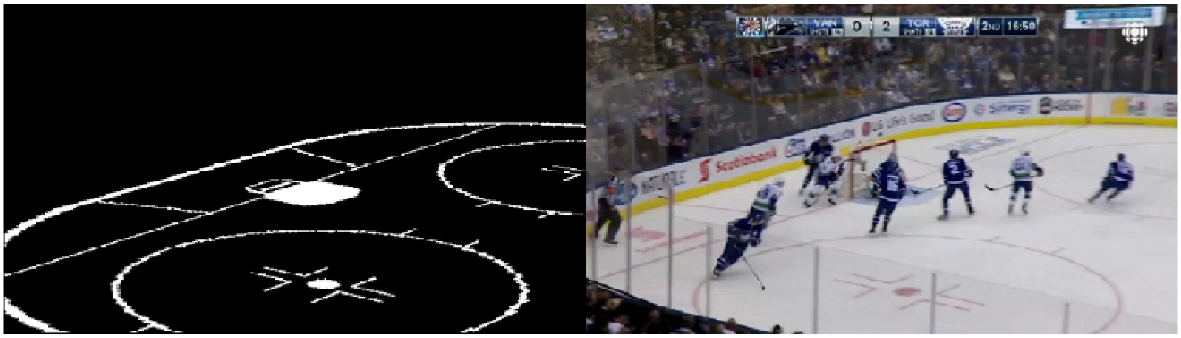


Figure 3.3: In this case, the overhead template is warped using the result of the initial estimator module, to produce the warped template seen on the left. The refinement module takes the warped template and the actual video frame, and tries to calculate the homography needed to warp between the two using the 4-point approach.

This warped template is concatenated with the video frame and fed into the refinement model as its input. The refinement model is also a Resnet18-based regressor that estimates the homography difference between the frame and the warped template. It uses the 4-point approach as well, although the points used are positioned 25% or 75% of the way across the image, such that each point is 25% of the way from the two closest edges. An example is seen in Figure 3.3.

The refinement model has two input branches, similar to the one from Shi et al. [35], except without the scoring portion. A diagram can be seen in Figure 3.4. The left branch processes the hockey frame while the right branch processes the warped template. Due to the lower amount of detail in the warped template, the usual Resnet blocks on the right branch are replaced by convolutions with fewer parameters. Non-local blocks were also added to improve the model's ability to capture long range dependencies [43]. These blocks are similar to attention as they compare features from all positions using weighted averaging [43].

The end result of the refinement portion is a refinement homography that corrects the difference between the initial estimate and the video frame. Thus, we can perform matrix multiplication using it and the initial estimate homography to obtain a more accurate final homography.

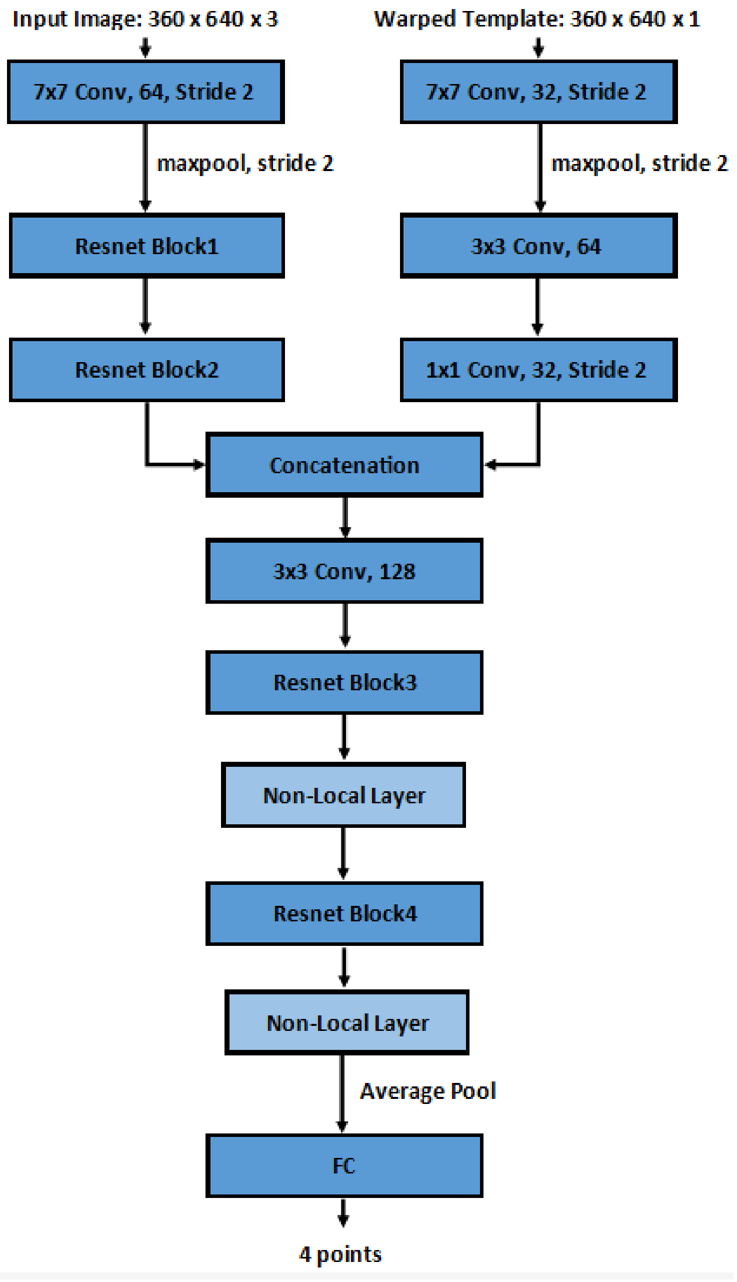


Figure 3.4: Diagram of refinement model. It is based on the one created by Shi et al. [35], with slight differences such as the addition of non-local layers.

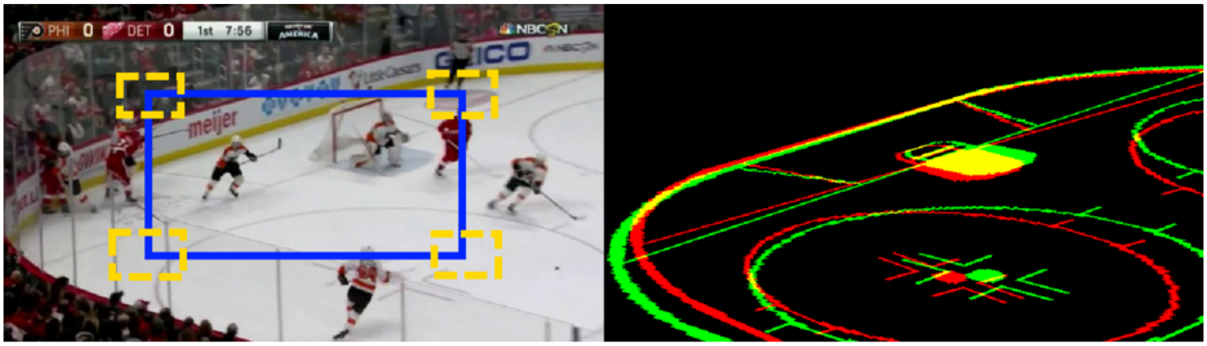


Figure 3.5: An example perturbation used to train the refinement module. The blue rectangle on the left image shows four original points, while the yellow show the range of possible perturbations. The right side shows how the perturbation affects the warped template, with the green representing the original location and the red representing the warped version after perturbation.

3.1.1 Training Process

For the initial estimator, the model was trained using roughly 4500 broadcast frames from NHL videos, along with their ground truth homographies. The model was trained for 400 epochs with smooth L1 loss, a batch size of 16, and an AdamW optimizer with an initial learning rate of 0.0001. A step scheduler was also included, that reduced the learning rate by 10× at 200 and 300 epochs.

For the refinement model, the training process is more complicated, as our goal was to teach it to calculate the small homography warps between the warped templates and the video frame. In order to do so, we use perturbations to generate synthetic warped templates.

To generate a perturbation homography, we select a rectangle in the image, and shift each of the four points by a small amount. This can be seen in Figure 3.5, where the blue rectangle represents the original four points, and the yellow rectangles represent the ranges to which each point can be perturbed. The original and perturbed sets of points can be used to generate a perturbation homography. When combined with the frame's original homography via matrix multiplication, this allows us to generate a synthetic warped template by warping the overhead template.

This process is seen in Figure 3.5 with the red line template representing the synthetically perturbed warped template, and the green lines representing the original frame's

Table 3.1: Number of frames and games used for the training, validation, and test data.

Split	Number of Frames	Number of Games
Training	4501	16
Validation	914	4
Test	756	4

homography. We feed the randomly generated perturbed warped templates (red template lines in the example) and the input frames during the training, and the model learns to identify the perturbation warp needed to go from the red line template to the green line template (which is not seen by the model). Note that in places where the templates overlap, the example image becomes yellow due to how it is visualized.

The refinement model was trained for 350 epochs with smooth L1 loss, a batch size of 16, and an AdamW optimizer with an initial learning rate of 0.0001. A step scheduler was also included that reduced the learning rate by x10 at 150 and 250 epochs. Training for this model and all subsequent models was done using one to two NVIDIA RTX 2080 Ti GPUs.

3.1.2 Data

The NHL data used here includes frames from 24 different games. The splits are listed in Figure 3.1.

3.2 Augmentations and Improvements Made

We further improve the model from Shi et al. [35] by making several adjustments during model training. Increased augmentation such as zoom augmentations helped cover cases that were rare in the dataset, such as closer camera zooms. This was done during training by scaling the images randomly by a factor of $1 - 2\times$, and calculating the corresponding homography needed to perform that scaling as well. This would allow us to augment the images while providing the necessary ground truth alteration to match the changes, as the scaling homography can be matrix multiplied with the ground truth homography.

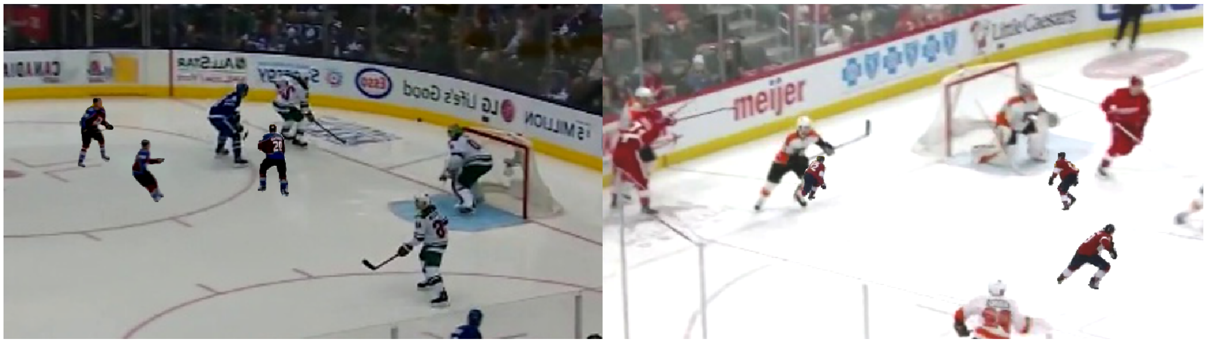


Figure 3.6: Examples of copy-paste augmentation. Here, players from other scenes are added to simulate how rink features could be occluded naturally during hockey videos. This is used to improve the model's robustness against occlusion.

Non-local blocks were added to the refinement model, to capture long-ranged dependency information, as described by Wang et al. [43]. These include context information relating different features across the image, such as relationships between the faceoff circles and different lines.

We also add copy-paste augmentation to provide a way to improve the model's robustness against occlusions. This augmentation is based on the work of Ghiasi et al. [12], which takes portions of other images to augment current images with new objects to segment. In our case, however, we use this to paste players from other images into current frames during training. These players provide a natural way to block parts of the rink, and help improve the model's robustness against occlusions of rink features by players that occur naturally throughout a game. Examples can be seen in Figure 3.6.

3.3 Temporal Approach to Pipeline

Another improvement that was investigated was the temporal approach to this pipeline. Although briefly mentioned in [35], we delve deeper to investigate the effects this method has on the resulting homography estimates.

The temporal approach of this pipeline involves using the refinement model to calculate the homography needed to warp successive frames in video. This allows the system to skip the homography initial estimator, resulting in faster inference. The downside is that this only works when the video has sufficient fps such that there is only a small amount of movement between successive frames. However, 30 fps has been seen to be sufficient during tests.

A diagram illustrating the overall pipeline can be seen in Figure 3.7.

This approach works because the refinement model is trained to identify small changes in homography, and successive frames in videos usually only have a small change in perspective. However, there needs to be a system to determine if any major break occurs in the video, such as if the broadcast video changes to a closeup. Thus, the video should be split beforehand into sections that focus on general gameplay, and avoid commercial breaks or replays from behind the net/close-ups. This can be done via shot transition detection.

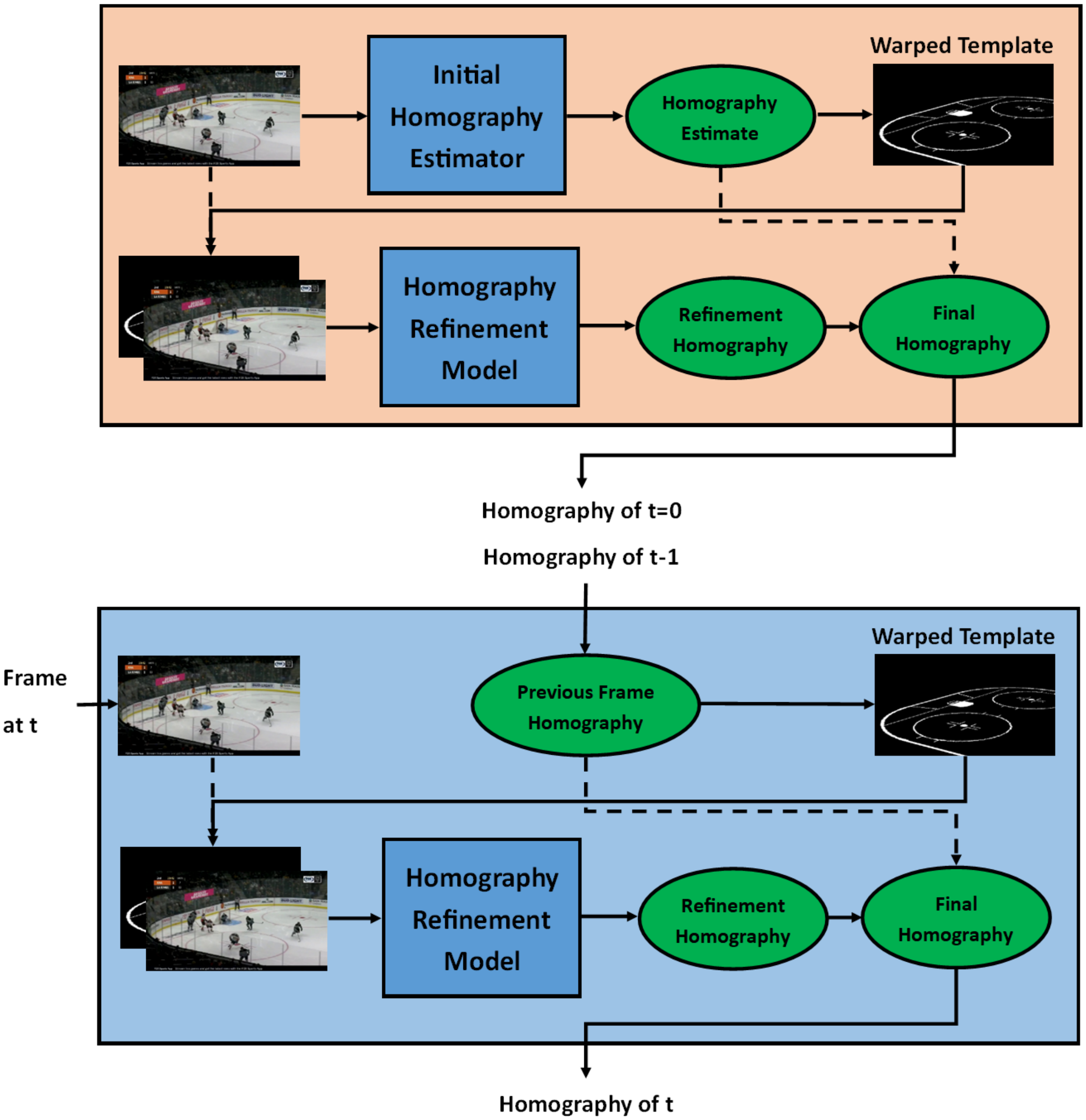


Figure 3.7: Pipeline demonstrating the temporal method of inference. In this system, the initial homography estimator is only used in the first frame of the clip, and only the refinement module is used for later frames. It calculates the warp needed to go between successive frames in the video.

3.4 Results

The results using the initial homography estimator and the refinement model can be seen in Tables 3.2, 3.3 and 3.4. In Table 3.2, we see how the homography and color based augmentations added improved test accuracy by 0.2-1.4%. These results were acquired from a single seed, so different seeds may vary the results by a small amount. However, the overall trends are still valid as runs with slight differences in setup still yielded similar results.

In Table 3.3, the addition of non-local blocks increased the accuracy due to improvements in capturing long range dependencies, albeit very slightly. The fact that the accuracies only increased slightly suggests that the long-range dependencies were either captured to a sufficient degree beforehand, or were not too relevant.

In Table 3.4 we see that the refinement model is needed to improve the initial estimate as the initial estimator results can often be off. We also see how copy-paste augmentations further improved the IoU_{part} by a small amount.

During some segmentation experiments, the copy-paste augmentation helped improve the model's ability to identify small features such as faceoff dots when they were occluded. An example can be seen in Figure 3.8. Thus, we included it in the final model even though the accuracy increase was relatively minor, since it improved the model's robustness against occlusion.

Examples of the homography estimation results can be seen in Figure 3.9. The middle row shows examples of cases where the initial estimate was off, but the refinement model was able to correct the mistake. The bottom row shows a couple of examples where even the refined model was unable to align some lines correctly.

The temporal approach resulted in roughly the same IoU as the initial estimator + refinement method. However, it was seen in some cases that it improved estimation results

Table 3.2: Accuracy increases due to added homography and color augmentation on refinement model.

Model	IoU (part)	IoU (whole)
Refined Model (base)	96.9%	86.4%
Refined Model (with augmentation)	**97.1%**	**87.8%**

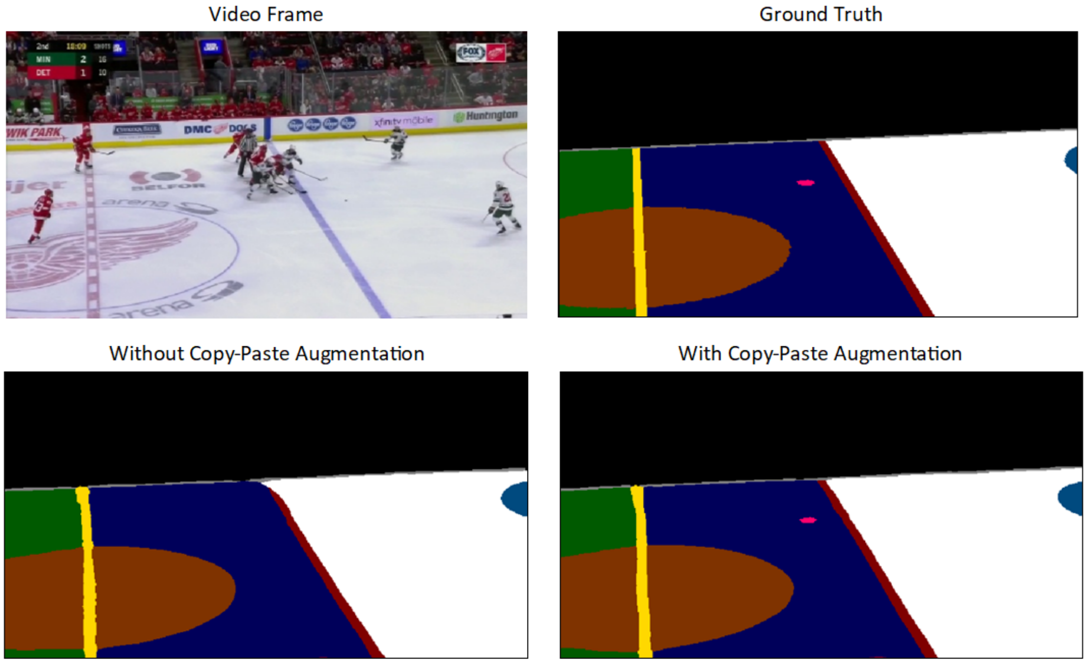


Figure 3.8: Segmentation model experiments involving copy-paste augmentation. The bottom row highlights the effect copy-paste augmentation has on the model's ability to deal with occlusion. The model trained with copy-paste augmentation was able to segment the faceoff spot (seen in pink), while the model without the augmentation was affected by the occlusion.

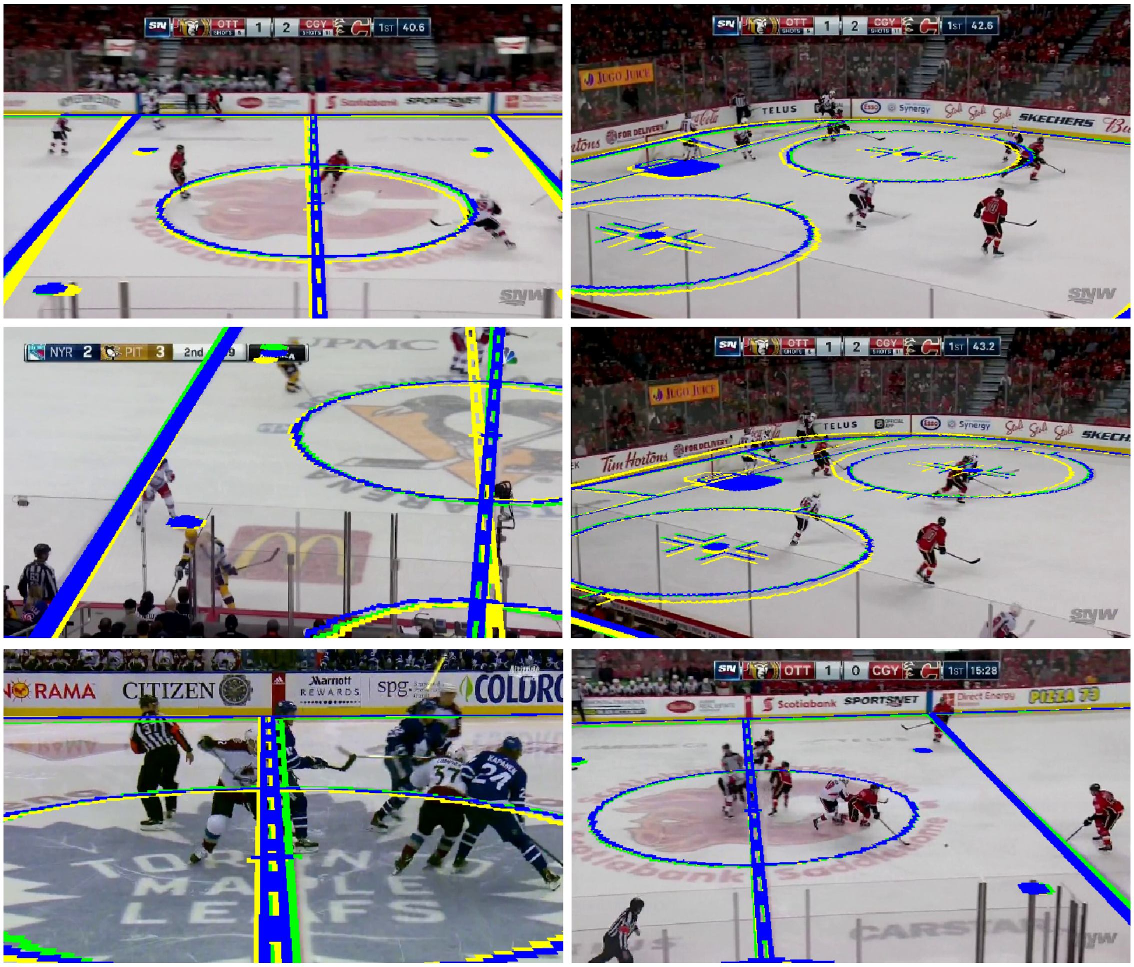


Figure 3.9: Examples of homography results. The green lines represent ground truth, the yellow lines represent the initial estimator result, and the blue lines represent the refined estimate result. Bottom row shows a couple of cases where lines are still not aligned even after refinement.

Table 3.3: Accuracy increases due to non-local blocks.

Model	IoU (part)	IoU (whole)
Refined Model (no non-local)	97.1%	87.8%
Refined Model (with non-local)	**97.2%**	**87.9%**

Table 3.4: Accuracy increases due to copy-paste augmentation on both initial estimator and refinement model.

Model	IoU (Part)	IoU (Whole)
Initial Estimator (no copy-paste)	96.0%	86.2%
Initial Estimator (with copy-paste)	**96.2%**	**86.3%**
Refined Model (no copy-paste)	97.2%	87.9%
Refined Model (with copy-paste)	**97.3%**	**88.1%**

when screen advertising and other screen elements occluded the rink. This is due to skipping the initial estimator, as that model has to estimate a large warp from the overhead view to the image view. This is thought to be a more difficult problem than what the refinement model solves, which is calculating a small warp between two similar perspective images. Thus the initial estimator may not have been robust enough to handle the unexpected screen occlusion, whereas the refinement model handled it just fine. An example of this case is seen in Figure 3.10.

3.5 Conclusion

This chapter shows the results of a model based on [35] trained on our data. It highlights some improvements that can be made during augmentation to improve robustness against rarer camera orientations and occlusions. It also demonstrates the benefits of the temporal method, which can improve speed and provide increased robustness against difficult cases that the initial estimator is unable to deal with.

Some future research into this field include architecture improvements to produce similar results in a smaller model, or with faster inference. As well, efficient smoothing mechanisms can also be applied to produce cleaner results during video inference.

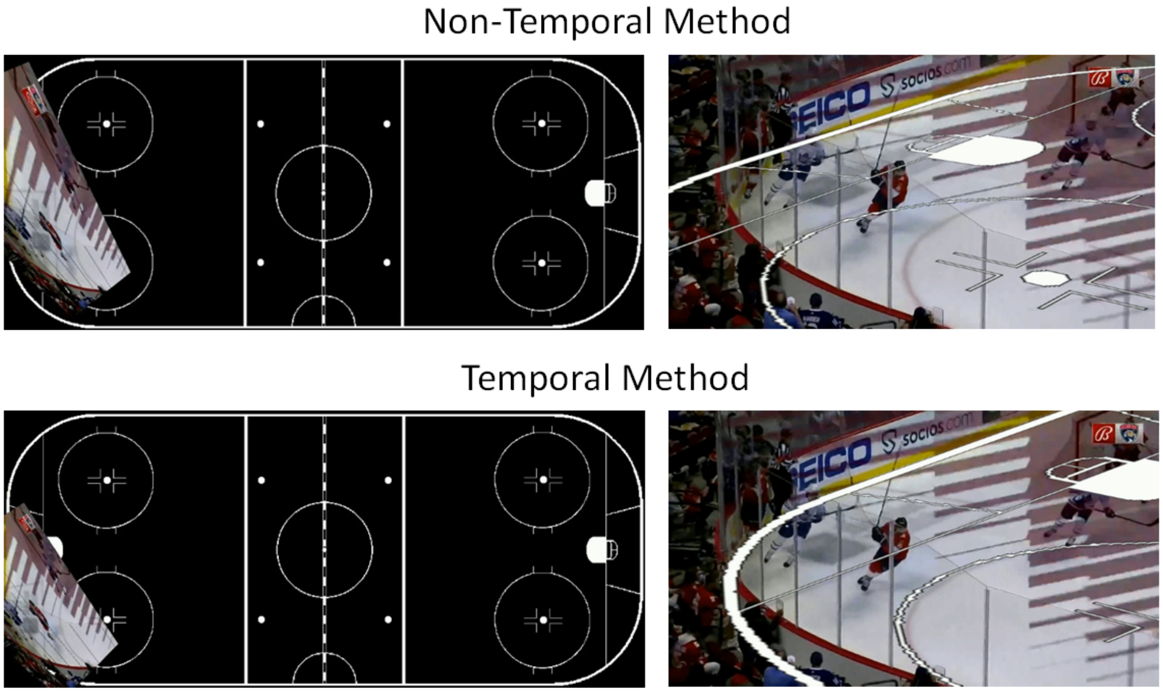


Figure 3.10: Example case where temporal approach improved results. In this case, the temporal approach managed to warp the scene correctly, even though it was partially occluded by visual elements (such as advertising or transitions between clips).

Chapter 4

Rink Agnostic Homography

This chapter describes a method for rink-registration that is agnostic to the type of rink used, and works in a wide variety of rinks without any additional training data.

Most existing rink registration systems focus on NHL rinks, which have a strict standardization system [44] [35] [21] [29]. This means that each rink is the same size, with the same positions for features such as faceoff circles, blue lines, and goal lines. However, non-NHL rinks also exist. For example, many European rinks follow the International Ice Hockey Federation (IIHF)/Olympic hockey rink format, which is wider than the NHL standard [1]. This standardization is not as strict, resulting in varying rink sizes and feature location changes in different rinks. For example, some arenas in Finland have sizes that fall between IIHF and NHL sizes [10]. Also, minor leagues and recreational rinks may not follow standards as strictly, resulting in more differences. Examples of different rinks can be seen in Figure 4.2.

Models trained on NHL rinks, due to geometric differences, sometimes will not work during inference on rinks that that have geometry or feature placements different than the NHL rinks. For example, running the model in Chapter 3 on Olympic data often results in errors in the warping process. This is due to the system not being designed to work with other rink types. Examples can be seen in Figure 4.1.

Training on different rink setups using the existing methods would require labelled ground truth for those rink types, which is time-consuming and expensive to produce. Furthermore, such data would produce a model that is designed to work on that rink type, which results in a different model and training data being needed for each rink variation.

Thus, the rink-agnostic pipeline aims to address these problems. It is designed to work on a wide variety of rinks without the need for additional training data beyond the NHL

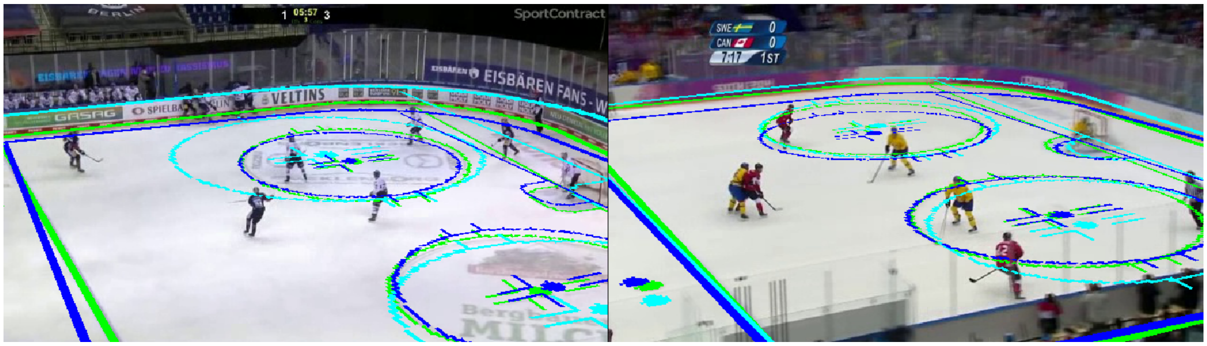


Figure 4.1: Examples of baseline results on non-NHL rinks. Green is ground truth, light blue is initial estimator, and dark blues is the final estimate. We see that the NHL-specific model sometimes does not work in these conditions, even after scaling results to match the template size differences.

data used to train the models in Chapter 3. This is done using domain adaptation and synthetic data methods to improve the model's generalization ability on different rinks. All that is needed during inference is the video frames and an overhead view of the rink template, which can be generated using various rink dimensions.

We propose a novel pipeline with three main modules (models) to resolve the aforementioned issues. The first model performs semantic segmentation on the input image to produce a segmentation map. The following two models estimate and refine a homography estimation based on the segmentation map and the corresponding rink template. To address the lack of data for non-NHL rinks, we implement domain adaptation techniques, use improved augmentations, and use synthetic data to simulate different possible rinks.

To the best of our knowledge, this pipeline is the first system designed for sports rink registration that is able to work on a variety of rink types, making it rink-agnostic. It is able to estimate homography for multiple rink types with competitive accuracy, despite only having labelled data for a single rink type.

4.1 Related Works

The rink-agnostic pipeline consists of modules that perform segmentation and are trained with domain adaptation. Thus this section will provide a brief overview on those fields.

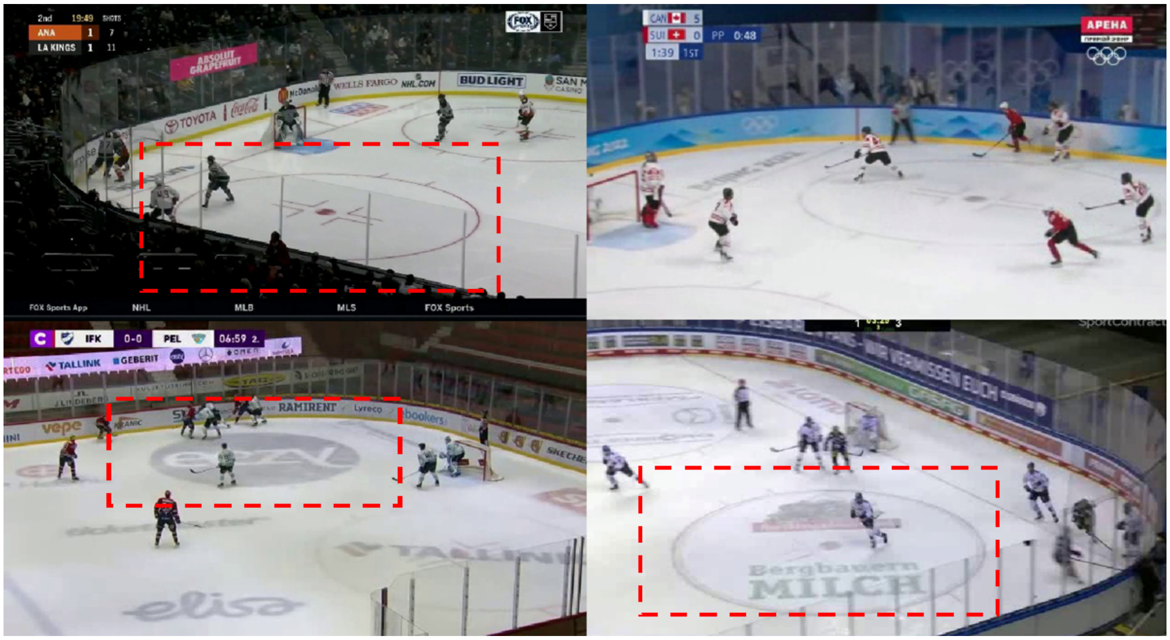


Figure 4.2: Examples of different rinks. On top of the differences between rink shape and feature positioning, there are also differences in color, advertising frequency, and how faceoff circles were filled. Faceoff circle differences are highlighted using dashed boxes.

4.1.1 Semantic Segmentation

Semantic segmentation involves classifying each pixel in an image into several provided categories. With the advent of deep learning, many models were developed to do this for fields such as autonomous driving and remote sensing.

Long et al. [25] popularized the use of fully convolutional networks (FCNs) for the purpose of semantic segmentation, which was a major milestone in the development of deep learning models for semantic segmentation [25]. To obtain the necessary information to accurately segment the image, Long et al. reinterpret the final fully connected layers of a CNN as convolutional layers with kernels covering their entire input region [25]. This results in the model outputting a classification map, which can be used for segmenting each pixel. Due to the downsampling however, this map is generally smaller than what is needed, so a form of upsampling is needed to recover the original size. Long et al. interpret upsampling with factor f as convolving with fractional stride $1/f$. This can be done via backward convolution (also known as deconvolution) with an output stride of f, which is simple and efficient to implement [25]. The filters used for this deconvolution can be learned to improve the upsampling results. Skip connections were also included to link semantic information from previous layers to the output.

Ronneberger et al. [31] designed the U-net model which builds upon the fully convolutional network by setting up a dedicated encoder and decoder structure. In U-net, the x-y sizes of the feature maps first decrease in the encoder, before expanding in the decoder, and connections are made between equivalent sized blocks in the encoder and decoder in order to provide more information during the expansion process [31].

The DeepLab series of models further build upon the U-net structure by adding various techniques such as Atrous Spatial Pyramid Pooling (ASPP) and image level pooling, to improve long range and global context information acquisition [4] [5] [6]. DeepLabV3+ in particular, combined benefits of various past models together [6]. The spatial pyramid pooling in the encoder allows for contextual information to be captured at multiple scales. The decoder portion helped create accurate segmentation boundaries during the upscaling process by using skip connections to obtain information from the encoding stages.

Various vision transformer based approaches were also used for semantic segmentation, taking advantage of the improvements transformers provided to the field of image processing. SETR was one the earliest, adapting the Vision Transformer (ViT) by using a classification ViT as the encoder and using convolutional layers in the decoder to produce an output map [47].

Further improvements to the ViT encoder were done by Wang et al. in their Pyramid

Vision Transformer, which naturally introduces a pyramid structure to ViTs for higher resolution predictions in tasks such as segmentation [42]. Swin Transformers took a different approach compared to traditional ViTs by using a hierarchical approach with shifted windows, to model at different scales and improve complexity when larger images are used [24].

Segformer also used a hierarchical approach in their encoder to provide multi-scale context information [45]. They also incorporate convolutions during their transformer blocks to provide implicit position encoding, removing the need for explicit position-encoding during the embedding stage. This allows for the resolution at test time to differ from the resolution in training without needing to interpolate position encodings [45].

4.1.2 Domain Adaptation for Semantic Segmentation

Unsupervised domain adaptation (UDA) involves trying to bridge the domain gap caused by differences between the labelled training data (source domain) and unseen test data (target domain). UDA tries to mitigate this issue by training on both labelled source data and some unlabelled target data and using techniques to improve the model's performance in the target domain. Techniques such as maximum mean distances [26], adversarial learning [11], and self-training [41] have been used for deep learning to improve the model's ability to bridge the domain gap.

Self-training methods in particular seem to perform well for UDA in the field of semantic segmentation, with several recent works using it [16] [17]. DAformer by Hoyer et al. use a teacher-student approach for self-training, where a teacher model is gradually updated using exponential mean average of the student weights, and is used to produce pseudo-labels of the target data for the student to train with [16]. Masked Image Consistency (MIC) uses a similar approach that can be added on top of existing domain adaptation methods [18]. It involves masking the target images fed into the student model to train it to learn contextual relations between different components in the target image. The loss is then computed between the predicted heatmap and a pseudo-label generated by the teacher model, which has access to the entire image.

The rink-agnostic pipeline takes techniques from this field into account to improve performance on the unseen target domain of Olympic and other non-NHL rinks, especially during the segmentation stage of the pipeline.

4.1.3 Semantic Segmentation and Homography

Some models have used homography to improve the results of semantic segmentation, especially in cases where the resulting segmentation is expected to follow a structure that is known beforehand. Examples can include organ semantic segmentation in biology, where the organ components have a roughly known structure, and this prior can be used to provide a better segmentation.

Lee et al. [22] develop an Image-and-Spatial Transformer Network (ISTN), which consists of two components: an image transformer network (ITN) that generates a representation of two input images, and a spatial transformer network (STN) that is trained to find the affine transform needed to align the resulting feature representations together[20]. Sinclair et al. build upon this work in their Atlas-ISTN by setting the ITN to be a semantic segmentation network, and using the result of that in an STN to warp an "atlas" template to a proper orientation [36]. The main goal of the Atlas-ISTN is to develop a semantic segmentation of the organ that is free of artifacts or noise thanks to the final result being a warp of the prior template, guided by the initial ITN segmentation [36].

Our pipeline uses a similar approach of performing segmentation before estimating a warp matrix. However, the main goal is estimating the matrix used to warp the template, rather than getting the warped segmentation itself. Furthermore, we require a homography matrix rather than an affine matrix in order to map one plane onto another. We only have ground truth training data for a single source domain, and use UDA techniques to improve results on other rink types.

Other sports registration systems have used segmentation to extract feature information before further analysis [15] [46]. However, none of them do so for the purpose of performing rink-agnostic homography.

4.2 Methodology

We propose an end-to-end system for rink-agnostic homography estimation. It takes in video frames and the overhead template of the rink as input, and outputs the homography needed to warp the template onto the frame. Our pipeline consists of three components:

1. A semantic segmentation model takes in the input video frame and outputs a semantic segmentation map.

2. An initial homography estimator takes the segmentation map and the overhead rink template as input and outputs the homography needed to align the two together. This homography is then used to warp the overhead template and produce a warped template estimate.
3. Finally, a refinement model takes the segmentation map and warped templates as input and produces a refinement homography to adjust the warped template estimate to be closer to the proper orientation seen in the segmentation map. This process can be iterated to further improve the homography.

The overall pipeline and how the three components interact with each other can be seen in Figure 4.3. However, we still have a lack of labelled training data for non-NHL rinks. To solve this issue, we use domain adaptation, augmentations, and synthetic data to train each component separately. This helps make the system more rink agnostic and helps overcome the lack of data for other domains.

The segmentation module is trained via domain adaptation techniques on both labelled NHL and unlabelled non-NHL data. Augmentations such as logo augmentation are also added, to simulate differences in appearance between rinks and further improve generalizability. The other two modules are trained in a semi-supervised manner using synthetic data, in order to generalize them to different rink types.

4.2.1 Rink Parameterization

The rinks used for synthetic data are developed by randomly altering distances between different features of the rink. The eight distance parameters are:

1. Distance from the left/right sides of the rink template to goal lines.
2. Radius of corner.
3. Distance from the top/bottom edge of rink to offensive zone faceoff circles.
4. Distance from the goal lines to the offensive zone faceoff circles.
5. Distance from the offensive zone faceoff circles to the blue lines.
6. Distance from the blue line to the center line.
7. Distance from the neutral zone faceoff spots to the blue lines.
8. Distance from the top/bottom edge of rink and center faceoff circle.

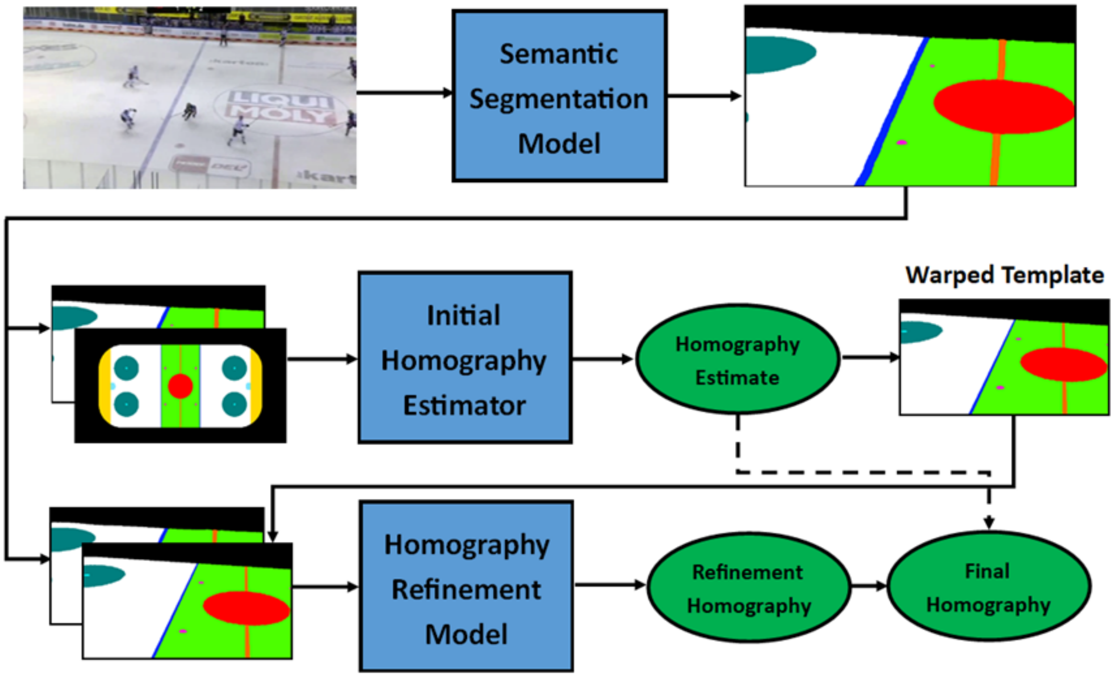

Semantic Segmentation Model
Warped Template
Initial Homography Estimator
Homography Estimate
Homography Refinement Model
Refinement Homography
Final Homography

Figure 4.3: Pipeline of the process during test time, showing the three major components. The inputs to the pipeline are the video frame fed to the segmentation model and the overhead template fed to the initial estimator. The iteration of the refinement model has been omitted for clarity.

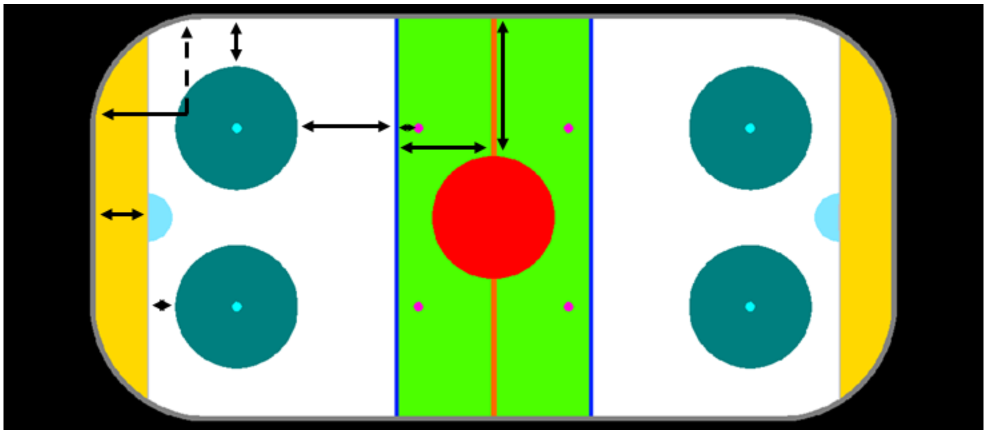


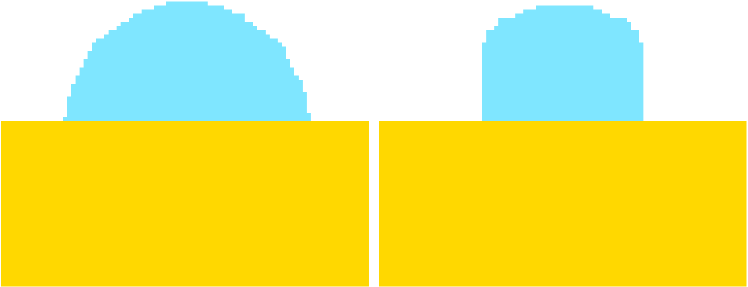
Figure 4.4: The eight distance parameters visualized on overhead rink segmentation template. The central faceoff circle is in red, the neutral zone faceoff spots are pink, and the offensive zones' faceoff circles are in turquoise.

These parameters can be seen in Figure 4.4. These distance parameters, plus the type of goal crease (semi-circle or cropped semicircle, seen in Figure 4.5), form the basis of rink generation, and thus each rink can be defined by eight distance parameters and one goal crease shape parameter. Example rinks can be seen in Figure 4.6.

4.2.2 Semantic Segmentation Module

The semantic segmentation module is designed to identify the various rink features in broadcast video frames, regardless of the type of rink used. Different rinks such as NHL and Olympic rinks can have different structures, and there usually isn't a scaling or direct linear transformation that can warp the rinks to be the same form. These rinks are seen in Figure 4.7. However, although the various features such as faceoff circles and blue lines may differ in size and positioning, they will still exist in all major rinks. This allows them to be used as classes for semantic segmentation regardless of which rink the image was taken from.

In order to improve the model's ability to generalize on all rinks, we used heavy augmentation as well as domain adaptation techniques. On top of general augmentations such as Gaussian noise, color augmentation, shifts, tilts and zooms, we added copy-paste augmentation and logo augmentation.



(a) Semi-circle goal crease shape

(b) Cropped semi-circle goal crease shape

Figure 4.5: Two goal crease shapes

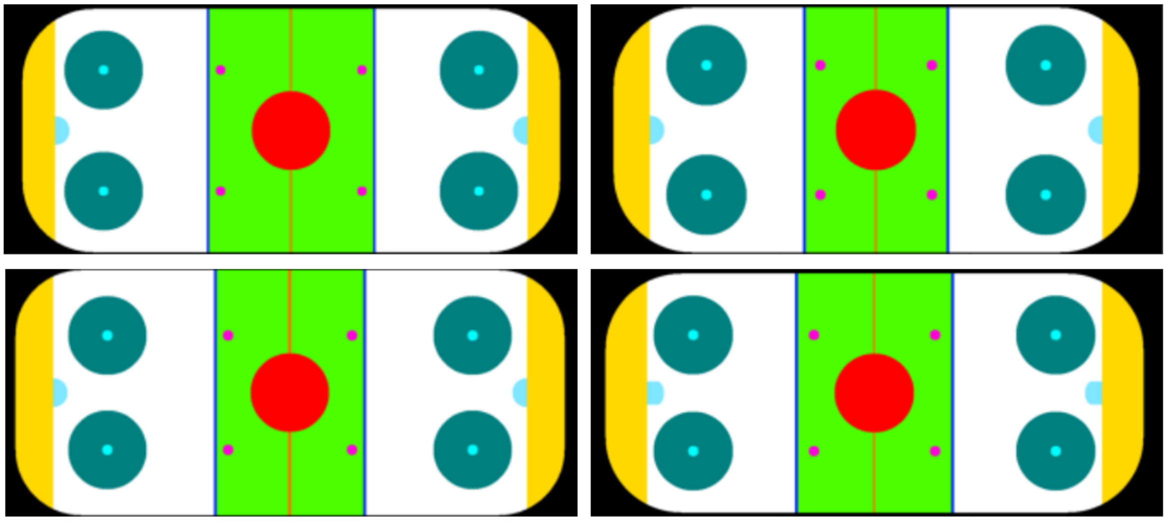


Figure 4.6: Examples of randomly generated rinks. The feature types and rough positions were kept constant, while the sizes, scales, and more precise positioning was varied each time. Some differences include goal crease shape, wider rinks having more space between faceoff circles and edges, and the blue lines and goal lines being in shifted locations. For example, the two bottom rinks are wider than the two top ones, and also have a larger distance from goal line to the edges of the rink (wider yellow region).

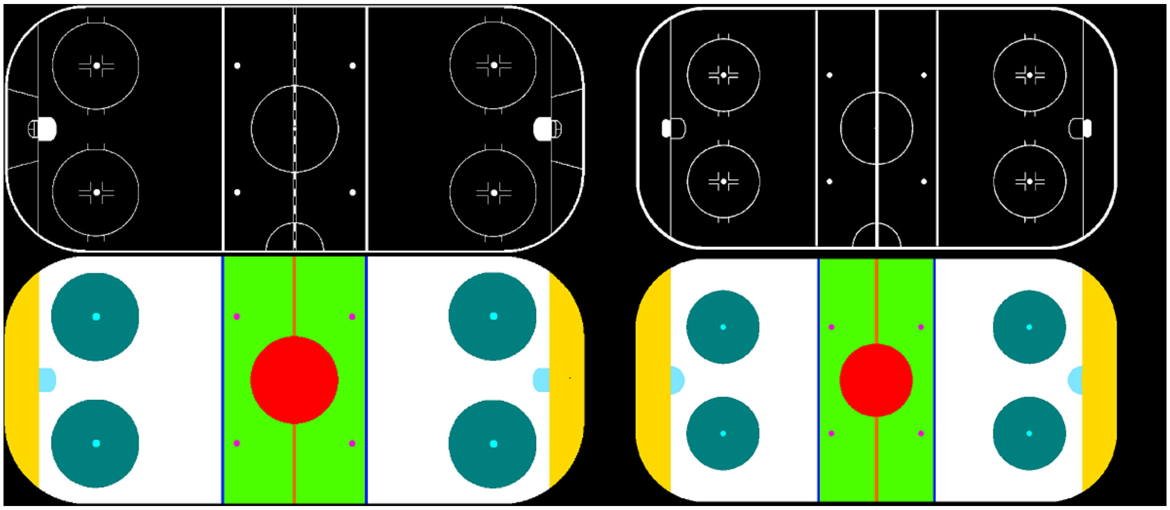


Figure 4.7: The line and segmentation overhead templates used for NHL (left) and Olympic (right) rinks. Note that in reality, both rinks are the same lengthwise, and the Olympic rinks are wider than the NHL rinks. They were both scaled to fit the same template space for this analysis, while maintaining their length to width ratios.

Augmentations

Copy-paste augmentation is based on the work of Ghiasi *et al.* [12]. However, their copy-paste system involved pasting instances from one image onto the other to improve the instance segmentation of items in different scenarios. In our case, we copy-paste players from other images to simulate the natural occlusion of rink features. This is used to improve the model's ability to segment rink features even when they are occluded.

Logo augmentation is designed to simulate the random advertising and text that may appear on different rinks. Randomized text, rectangles, and circle fillings are added in areas with space that may have logos in some rinks. This is done to teach the network to ignore the effects of such advertising. Examples can be seen in Fig. 4.8.

Domain Adaptation

We also use domain adaptation to improve the model's performance on Olympic rinks, where we do not have any ground truth segmentation training data. In particular, we adopt some methods described in MIC [18], to improve the model's ability to learn the

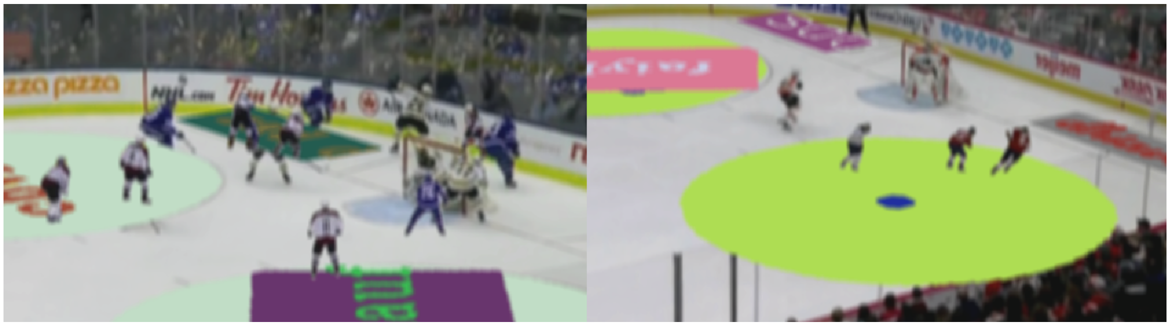


Figure 4.8: Examples of logo augmentation, which sometimes added text, rectangles, and circle fillings in order to augment the existing dataset further.

context between different components in the target domain. This would allow us to use unlabelled non-NHL data during our training.

We primarily add the exponential moving average (EMA) teacher-student and input masking behavior to our pipeline, as described by Hoyer et al. [18]. The EMA teacher-student approach has been shown to improve results for semi-supervised training [40, 16], and in domain adaptation self-training. In this case, the target domain of non-NHL rinks is unlabelled and pseudo-labels generated by the teacher are used instead. So during training, we have a student model that learns via loss functions, and a teacher model whose weights are altered over time based on the EMA of the student's weights over time. When training on the unlabelled target domain, the teacher has access to the unmasked image, and produces a pseudo-label.

The student, however, only has access to the masked input and produces a segmentation mask which is compared against the pseudo-label with a segmentation loss. This loss is weighted by the confidence weighting of the pseudo-label (as pseudo-labels may not be precise), and used to update the student model. The teacher's weights are then updated in turn via the EMA equation, as seen in equation 4.1, where t denotes timestep, Φ denotes teacher weights, Θ denotes student weights, and α is a smoothing factor [40]. The usage of MIC is seen in Figure 4.9.

$$\Phi_{t+1} \leftarrow \alpha\Phi_t + (1-\alpha)\Theta_t \tag{4.1}$$

We used a DeepLabV3+ model [6] from the Segmentation Models PyTorch library[19] as the segmentation model in this case. We also used focal loss for the segmentation loss [23], and AdamW optimizer. It was trained for 180 epochs, with a step learning rate

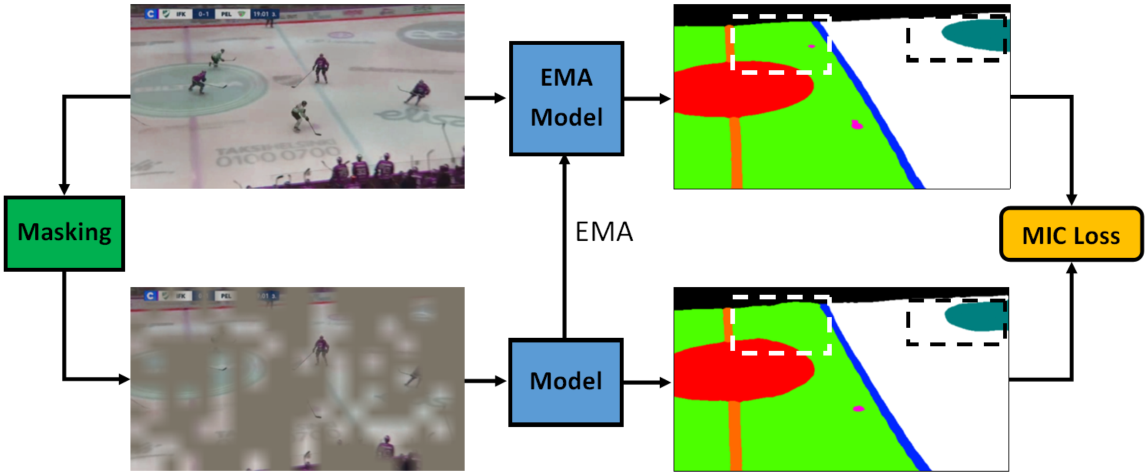


Figure 4.9: Pipeline of MIC method from Hoyer et al. [18]. It promotes the model to learn contextual clues as it needs to identify the hidden areas based on information from other non-hidden areas. The dashed rectangles highlight some areas that the model needs to correct. These areas were masked.

scheduler than reduced the learning rate by 10× every 70 epochs. Segformer models were also tested, but did not provide noticeable improvement over the DeepLabV3+ models in our cases.

4.2.3 Homography Estimator Module

The homography estimator module consists of a Resnet18-based regressor that estimates the normalized homography matrix, in a similar manner as Zhou and Li [48]. During the inference time, it takes the segmentation map output of the first module alongside an overhead template of the rink as input. It then produces an estimate of the homography needed to warp the overhead template to be aligned with the segmentation map (which makes it also aligned with the actual input frame if the segmentation map is accurate).

However, during training, we use synthetic data because we only had labelled training data for NHL rinks. To generalize effectively on all rink setups and sizes, we use synthetic rink generation to simulate different rink setups. This is done by altering various distances in the overhead template, such as the distance between faceoff circles and the goal line,

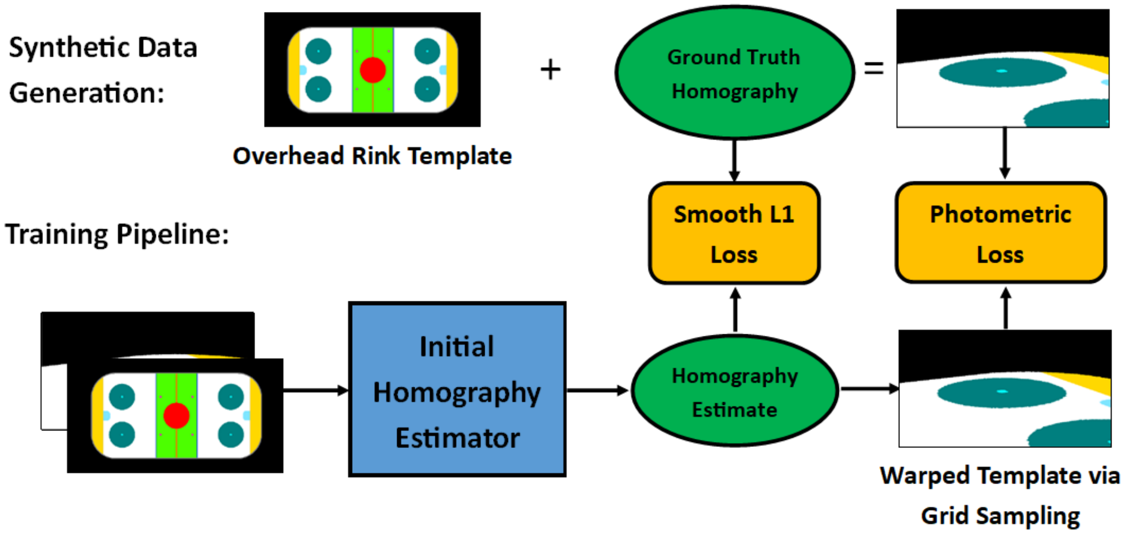


Figure 4.10: Data generation and training process of the homography estimation module. The overhead template and warped template are used as input, and the resulting homography and warped output template are compared with the ground truth. Note the normalization and unnormalization of the homography is omitted in this image for clarity.

or the distance between blue lines and the center line. Examples of these can be seen in Fig. 4.6, and the process was described in Section 4.2.1.

During data generation, we choose from common pre-defined rinks such as NHL or Olympic rinks, or create our own randomly generated rink to serve as the initial overhead rink template. This would thus improve its accuracy on a wide variety of rinks, as the model would be trained on a wide variety of templates.

The next step in data generation involves acquiring a homography to warp the overhead rink to create a warped template. To do this, we use a ground truth homography matrix from the NHL dataset, and augment it with slight perturbations, zooms, and flips. The resulting warped template simulates what a segmentation mask input would look like, and is used as the synthetic data. This process can be seen in Fig. 4.10. We use ground truth homographies from the NHL training set to represent the range of homographies that correspond to broadcast video. The augmentations applied to the homography matrix help cover this expected range. It also covers potential differences in homography ranges that may occur when we use different templates, as the rink sizes can differ in those cases.

During training, the overhead rink template and warped template are fed as input to the initial homography estimator, which estimates the normalized homography. This homography is then used to warp the overhead template to produce a warped template output. This is done using grid-sampling, which preserves the gradient flow and allows the loss to be propagated back to the estimator model. The normalized homography estimate is compared with the ground truth homography via smoothed L1-loss, while the warped template output is compared with the original warped template via L1 loss. This process can be seen in Fig. 4.10. The model was trained for 180 epochs, with AdamW optimizer and a step learning rate scheduler than reduced the learning rate by 10× every 75 epochs.

Note that the model only outputs a homography, so it cannot directly produce a copy of the warped template that was given as input. Thus, it needs to learn the homography required to warp the overhead template to the warped template input. During test time, we take the homography estimate and use it for the next module in the pipeline.

4.2.4 Refinement Module

The final module in the pipeline is the refinement model. During test time, its input consists of the segmentation mask from the first module alongside a warped template using the homography estimate from the second module. During training however, we once again leverage the use of synthetic data and semi-supervised learning to improve the model's performance on multiple rink types.

For training data generation, we follow a similar scheme as Shi et al. [35], where we take an existing ground truth homography and image, and augment them before feeding them into the model for training. This augmentation step involves selecting four random points in a rectangle on the image and perturbing them by a small amount. The perturbation amount for x and y is randomly selected from a uniform distribution. The previous and new positions of these points can then be used to produce a homography matrix, which is used to warp the image. The warped image and original image are then sent to the model during training, and it tries to calculate the homography needed to perform this warping process. In our case however, rather than using the video frame directly, our image consists of the overhead template warped by an existing ground truth homography. This homography is augmented before use, and is used to represent examples of rink orientations as viewed by the camera. This process is visualized in Fig. 4.11.

The refinement model is a Resnet18-based regressor, and uses the four-point approach to estimate homography, where it estimates the locations of four points from one image in the image space of the other. These sets of points can then be converted into a refinement

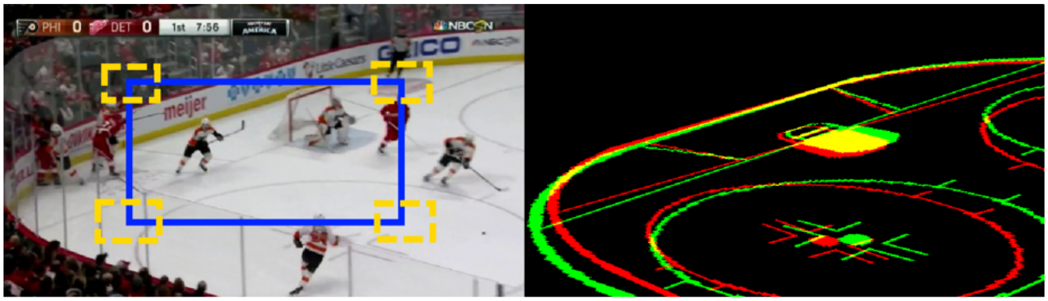


Figure 4.11: Data generation for refinement. The blue rectangle represents an example initial four corners, and the yellow rectangle represent the possible perturbations for this example. The shift in homography can be seen on the right, with green being the original rink position and red being the perturbed version.

homography via DLT [14], and will represent the warp needed to align the two input images. An example of pre-refinement inputs and a resulting refinement can be seen in Fig. 4.12. The model was trained for 180 epochs, with an AdamW optimizer, smooth-L1 loss, and a step learning rate scheduler than reduced the learning rate by 10× every 75 epochs.

During test time, the refinement process can be iterated to further improve the homography refinement. The refinement homography can be combined with the initial estimate to produce a better estimate. This estimate is then used to warp the overhead template to produce a better warped template, which is fed back as input alongside the segmentation map. The refinement model performs this warp estimation process repeatedly, improving the alignment each time. In practice however, the alignment is only improved for the first few times, as small misalignments may not be aligned properly. Thus, we restrict at test time the number of iterations to three, as we found not much improvement beyond that. This process is visualized in Fig. 4.13.

4.3 Results and Discussion

We first present experimental results for each component in the pipeline, and then describe the results for the overall pipeline.

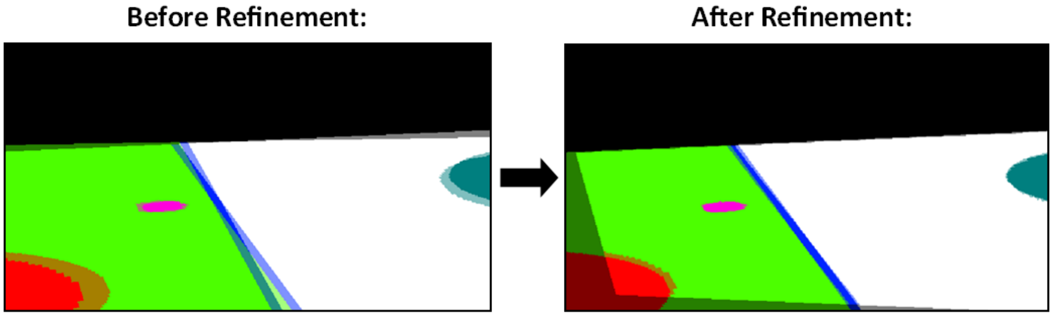


Figure 4.12: Example of refinement. The left side shows the two input images overlaid on each other, and the right side shows the alignment that can occur after the refinement matrix is calculated.

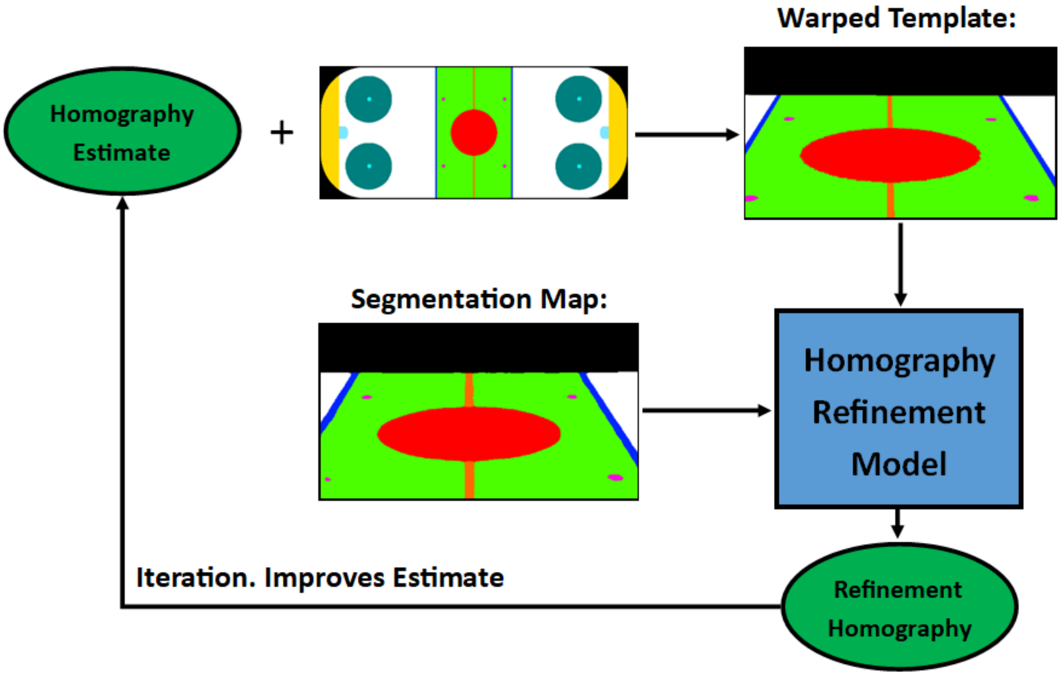


Figure 4.13: Refinement iteration during testing. The resulting refinement matrix can be combined with the initial homography estimate to create a better warped template, which is fed into the refinement model again.

Table 4.1: Overall Intersection over Union (IOU) results from the validation set of NHL games. Although the augmentation and domain adaptation (DA) improvements did not affect the overall numbers much, they produce qualitative improvements on the target dataset results.

Model	Overall IOU
NHL-Only Model	78.3%
Model with Augmentations	78.6%
Model with Augmentations and DA	78.5%

4.3.1 Segmentation Module

The segmentation module was trained to predict 11 different classes of pixels from input images, including general areas such as background and defensive zones to more specific features such as center line and goals.

The copy-paste and logo augmentation had a minor effect on the overall results for the source domain, as seen in Table 4.1. Qualitatively however, these augmentations were able to improve the model's ability to identify parts occluded by players. Some features such as faceoff spots can be occluded completely by players, for example. Therefore, these augmentations help the model to learn to segment such features even when they are occluded by people, as the model would be trained on more examples of such cases.

We perform a sanity check by comparing an NHL-only trained model with the domain adaptation trained model, to ensure the accuracy did not drop on a validation set of NHL rinks. This can be seen in Table 4.1.

Results on NHL rinks can be seen in Table 4.2. Accuracy for these segmentations are measured via intersection over union (IOU), a common metric for this type of task. For the source domain validation set, we can see how the results are good for classes that cover areas, but have more errors in classes that represent lines or spots. This is partially because lines and spots are more likely to be obscured by players or the boards at the bottom of the rink, and any small deviation in prediction can cause a large IOU drop.

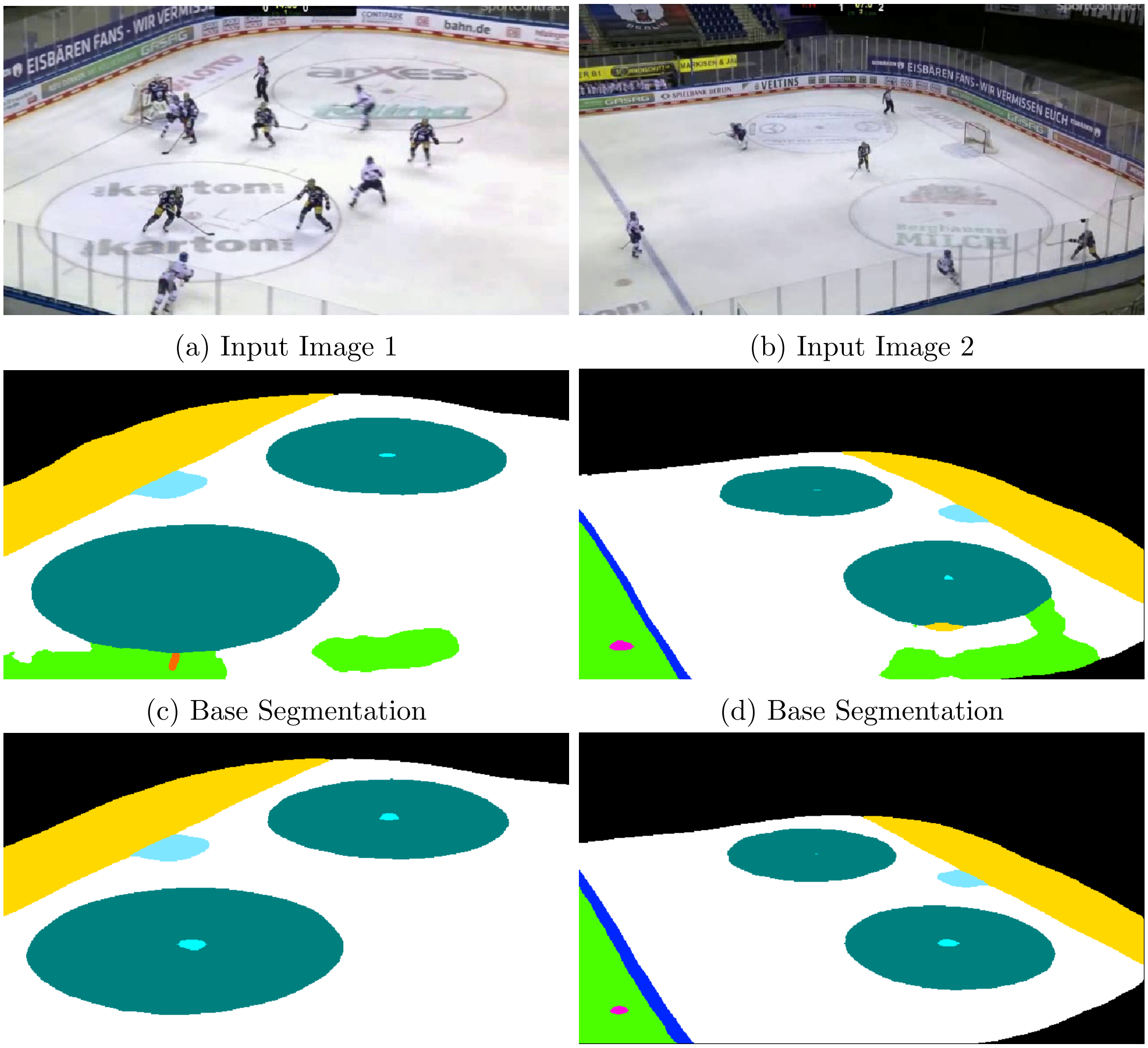
The domain adaptation model resulted in higher results on the target domain of non-NHL rinks, when compared with the NHL-only trained model. It was tested on various unlabelled target domain data, such as Olympic/European rinks. This shows that even with heavy augmentation, the changes between NHL and non-NHL rinks can still be quite large, resulting in a domain gap that needs to be bridged in another way.

Table 4.2: Intersection over Union (IOU) results for each class for segmentation models trained on source domain vs both domains. These results are the validation results from a set of held-out data on other NHL rinks and matches (source domain).

Class	Single-Domain	Domain Adaptation
Background	97.1%	97.0%
Behind Goal	87.0%	86.2%
Blue lines	45.4%	51.3%
Center Faceoff Circle	95.2%	95.0%
Center Line	62.3%	60.6%
Outer Faceoff Circles	94.4%	94.4%
Outer Faceoff Spots	61.6%	61.0%
Goal Creases	81.8%	81.5%
Neutral Zone	94.9%	94.4%
Inner Faceoff Spots	46.8%	47.2%
Defense Zones	94.8%	94.9%
Overall Average	78.3%	78.5%

Although no labels, and thus no quantitative results, are available, qualitative analysis can still be done, where the predicted segmentation map is compared with the original image, to see if the components line up. Examples of this can be seen in Fig. 4.14, using results on the Olympic and European validation set, which has different Olympic-style rinks not seen by any model during training. In it, we see in (c) and (d) that the base segmentation model misclassifies portions of the offensive zones (white) as green (which represents the neutral zone). However, using the improved segmentation model that incorporates our new augmentations and domain adaptation, these errors were fixed, as the model is trained to work in more varied situations.

The results of the model trained with domain adaptation and our copy-paste and logo augmentations were noticeably better. In particular, cases of major misclassified regions and missing regions that were present in the predictions from the base segmentation model were fixed in the domain adaptation trained model. This system is scalable such that it can work on full game videos as well as individual frames.



(a) Input Image 1

(b) Input Image 2

(c) Base Segmentation

(d) Base Segmentation

(e) Improved Segmentation

(f) Improved Segmentation

Figure 4.14: Examples of Olympic style rink images and corresponding predicted segmentation maps from a baseline model and an improved model with logo augmentation and DA. Domain adaptation and logo augmentation have improved the generalization capabilities of the model, allowing it to segment this new rink better than the no DA model.

Table 4.3: Intersection over Union (IOU) results using ground truth homographies from the validation set of NHL games. Different stages in the pipelines are compared, such as Initial Estimator Model (IEM), Refinement Model (RM), and Iterative Refinement (IR). The multi-rink model with iterative refinement achieves similar accuracy as the NHL-Only pipeline on our data. However, it has the added benefit of working on non-NHL rinks as well.

Pipeline	IOU_{part}
NHL-Only Baseline IEM	96.0%
Rink-Agnostic IEM	94.4%
NHL-Only Baseline IEM + RM + IR [35]	96.9%
Rink-Agnostic IEM + RM	96.7%
Rink-Agnostic IEM + RM + IR	96.9%

4.3.2 Homography Estimator Module

The homography estimation module is designed to roughly estimate the homography needed to warp the rink template onto an input image (or vice-versa by inverting the homography matrix to warp in the opposite direction). To compare homography results for homography estimation, we use IOU_{part}, where only the portion of the rink template that would have been in the image is considered. We use IOU_{part} because the ground truth data collected was primarily done with just the visible portion in mind, and thus the ground truth for $\text{IOU}_{\text{whole}}$ may not have been accurate. The image is warped using both the predicted homography and the ground truth homography, and the resulting intersection and union are calculated.

As with the semantic segmentation model, we compared a model trained solely on the source domain NHL rink with another model trained on multiple rink types and randomly generated rinks. This helps determine the viability of a rink-agnostic homography estimator. The source-domain trained initial estimator model performed 1.6% better than the multi-rink trained model on the NHL validation dataset, as seen in Table 4.3. However, the multi-rink trained model is still competitive and has the added benefit of working for multiple types of rinks, whereas the NHL-only model results sometimes had inaccuracies. The refinement module, later on, is used to improve the accuracy of the warps.

4.3.3 Refinement Module

The refinement model is the last component of the pipeline. It is designed to determine small homography differences between the segmentation mask output of the first model and the warped template created using the homography estimate of the second model. The refinement model must accurately calculate the homography needed to align the two inputs, and is trained on multiple fixed rinks and randomly generated rinks.

The refinement process results on the synthetic data used in validation have an accuracy of approximately 98% IOU_{part}.

4.3.4 Overall Pipeline

The results of the overall pipeline were analyzed to determine how well this system works on both NHL and non-NHL data. We use a model based on Shi et al. [35] as the baseline, which was replicated because the source code, original model, and data were unavailable to the public. Using the source domain NHL validation set, the results of our pipeline are roughly on par with that of the baseline, as seen in Table 4.3.

The visual results for this approach on the NHL validation data can be seen in Fig. 4.15. As seen in the images, the pipeline can warp the template to be closely aligned to the markings on the rink.

Adding iterations to refinement improves the accuracy by a small amount as the refinement module can make further adjustments to the estimated homography after applying the previous refinement to the estimate. We compared single round refinement vs iterative refinement and saw the overall results on the NHL data improved by a small amount with iterations, as seen in Table 4.4. Further adding iterations to refinement beyond three iterations did not increase the result by a meaningful amount.

On Olympic data, the resulting warps are usually close, but there are sometimes qualitative issues where the warped template is misaligned. One potential cause for this involves problems during the segmentation stage, where the segmentation maps are not fully accurate. Sometimes, regions such as faceoff circles or the bottom edge of the rink may not be segmented accurately and can be shifted from their true location, as seen in Fig. 4.16. In the top right image, the white region (representing the offensive zone) is uneven. Ideally it should form a curve that matches the bottom of the boards, which hide the bottom part of the rink from view of the camera. In the bottom right location, the marked faceoff spot is barely visible. Ideally it should be larger, like the other faceoff spots in the image.

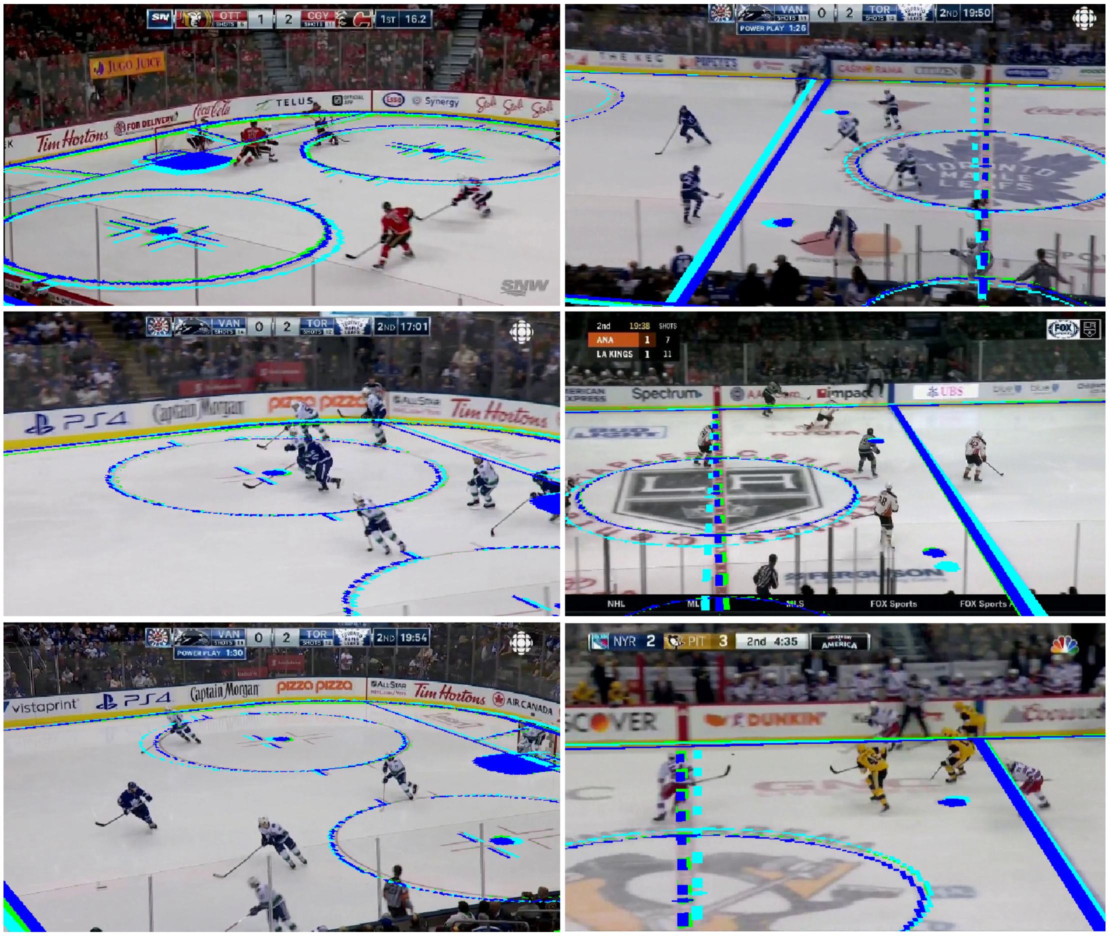


Figure 4.15: Example results on NHL validation data. Green is ground truth, light blue is initial homography estimate, and dark blue is the final homography after refinement. Note that sometimes the green may be covered completely by the blue at points due to how the results are visualized.

Overall results on two types of non-NHL rinks (Olympic 2014 and Berlin Mercedes-Benz Arena rink) can be seen in Table 4.5. Here we see that the NHL-only model results are not as good, even after scaling is provided to make the template closer to the NHL template. The rink-agnostic model is more robust to arena template changes as it performs better than the baseline with these non-NHL rinks. It is more robust to rink appearance differences as well, since the performance of the NHL-model dropped dramatically on the

Table 4.4: Intersection over Union (IOU) results using ground truth homographies from the validation set of NHL games. The first and second iterations of the refinement module have the largest effect, while later iterations do not have much effect. We stop at three iterations as results do not change much after.

Pipeline With Different Refinement Iterations	IOU_{part}
No Refinement	94.4%
Refinement with 1 Iteration	96.7%
Refinement with 2 Iteration	96.9%
Refinement with 3 Iteration	96.9%

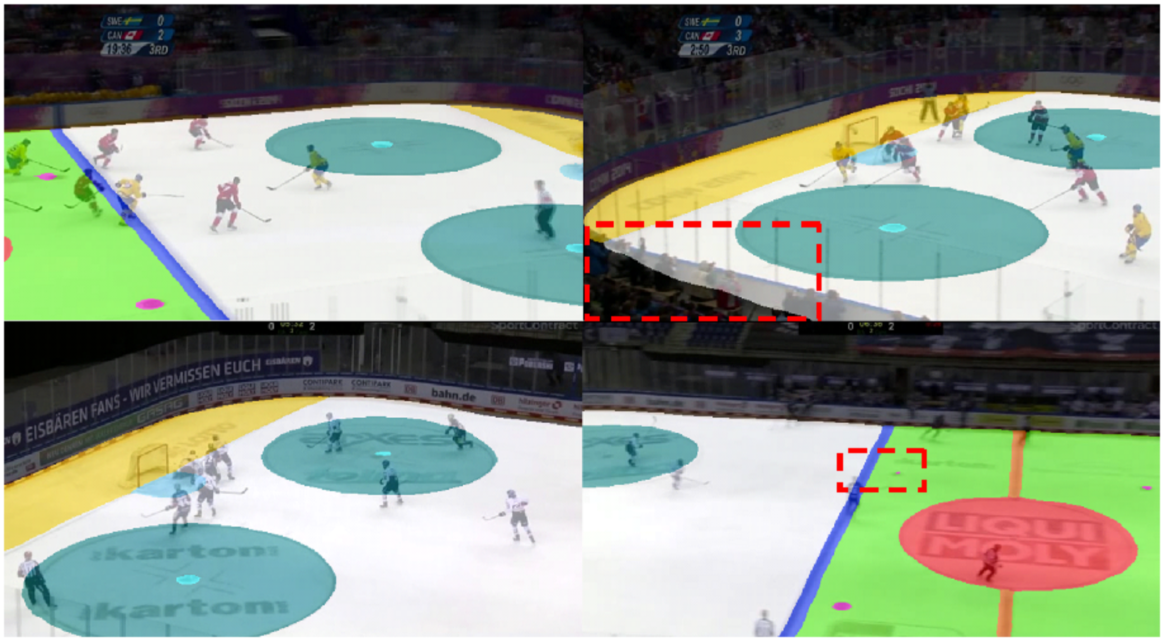


Figure 4.16: Examples of segmentation on Olympic rinks. The left two are examples where the segmentation is accurate, whereas the right two shows some more obvious defects. These include missing spots or over/underflowing edges (marked in red boxes).

Table 4.5: $\mathrm{IOU}_{\mathrm{part}}$ results on two non-NHL rinks. The rink-agnostic model outperforms the NHL-only baseline, especially in the Berlin arena where there are more differences in rink appearance.

Pipeline	Berlin Arena	Olympic 2014 Arena
NHL-only Baseline	87.7%	96.2%
Rink-Agnostic Model	**96.5%**	**97.3%**

Berlin arena, whereas the rink-agnostic model is not affected much.

Examples of the refinement model predictions can be seen in Fig. 4.17. The left side shows some examples of accurate rink registration, while the right side shows examples of misalignments. The top row also has the segmentation overlaid on top, highlighting the slight issues with the segmentation mentioned before.

Using batch size 64 on a single NVIDIA RTX 2080 Ti GPU, the inference time takes roughly 90s to estimate the homography per minute of gameplay footage at 30 fps. Further parallelization via larger batch sizes and more GPUs may improve inference time.

4.4 Conclusion

This chapter presents a novel approach to sports rink registration, by using a three part pipeline that is generalized to work on multiple rink types, despite only having labelled data for NHL rinks. The models are able to learn how to process different rink types and overcome a lack of labelled training data. This is done by using domain adaptation and augmentation techniques in the segmentation module, along with synthetic data and self-supervised methods in the homography and refinement modules. By doing this, we do not need additional labelled training data for other rink types, thus greatly saving annotation time and effort. This also produces a single model capable of working on multiple rink types.

Results show that the current pipeline is competitive with results obtained by supervised NHL trained models, while also having the ability to estimate homography for non-NHL rink types as well, demonstrating great potential. Some improvements in segmentation and handling of segmentation inaccuracies can be made to further improve the robustness and accuracy of the pipeline.

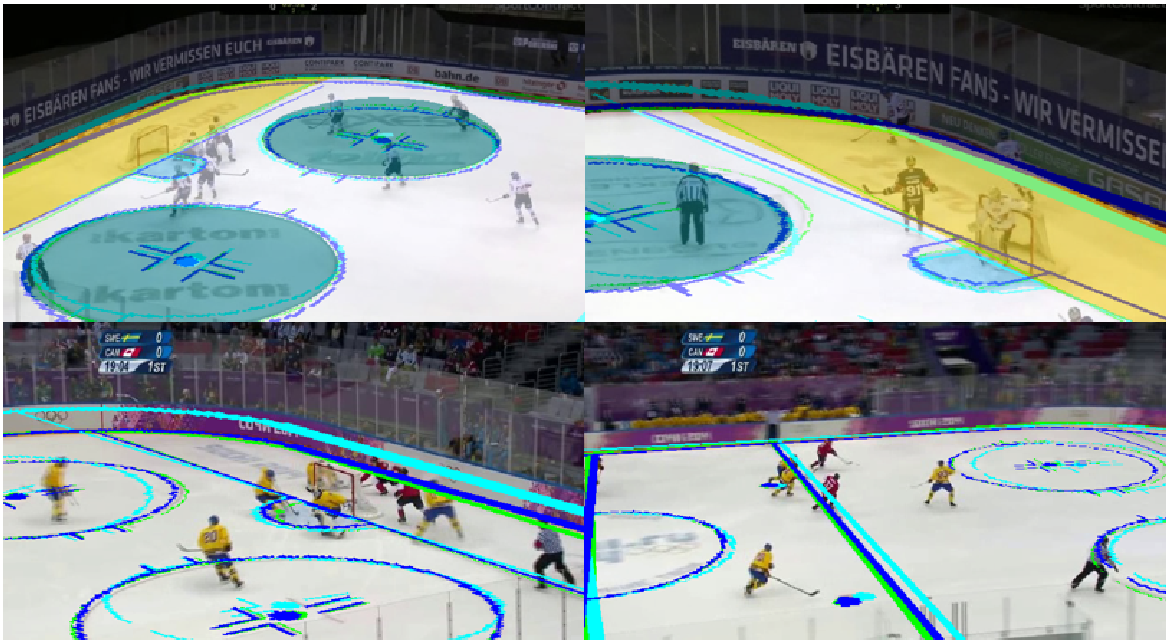

EISBÄREN FANS - WIR VERMISSEN EUCH
bahn.de
kartom
EISBÄREN FANS - WIR VERMISSEN
SWE 0
CAN 0
19:04 1ST
SWE 0
CAN 0
19:07 1ST

Figure 4.17: Example results on Olympic validation data. Light blue is initial homography estimate, dark blue is the final homography after refinement, and green indicates ground truth. These cases show that although the alignment can be quite close usually, sometimes the alignment can still be off even after refinement for the Olympic rinks, likely due to the difficulty in segmenting these rinks.

Chapter 5

Rink Generation

In the previous chapter, one of the inputs to the system is the overhead template of the rink. Normally, the dimensions and feature positions for this can be acquired by a diagram of the rink, measuring the features on the ice, or from an overhead image of the entire rink. However, if these options are unavailable, we may need to acquire this information from video of the arena instead.

This chapter will describe a semi-automated method of estimating the dimensions of the rink from a series of broadcast video frames showing the rink. These frames need to be close together in terms of rink orientation and cover over a quarter of the rink when combined. The method proposed uses segmentation, homography, and image stitching to produce a rough overhead template. The dimensions of rink features that we need can be accessed from this rough overhead template.

5.1 Methodology

The process of constructing the overhead template from the images is done in four phases:

1. Segmentation: All frames used for this process are first converted into segmentation maps.
2. Calculating warp-to-overhead homography: The homography warp needed to go from an image of the center of the rink to the overhead view is calculated. This forms the centerpiece of the rough template, and is adjusted such that lines are straight.

3. Image Stitching - growing the rough overhead template: The homography warps are calculated between close pairs of images (such as successive frames in video). These homography matrices are combined with the warp-to-overhead homography from the previous step to warp each frame to the overhead view.
4. Forming Complete Overhead Template: Dimensions from the rough overhead template are calculated, and used to create a full overhead rink with those parameters.

5.1.1 Segmentation

At the start of this process, all frames involved are converted into segmentation maps. This helps the later phases ignore foreground objects such as players, non-essential features such as audience, and items such as ads or scores overlaid on the screen. It is also useful as later steps would need to analyze dimensions and distances of rink features, which are more easily accessible if the images are converted into segmentation maps.

The model used is the same segmentation model in Chapter 4, and thus has the same benefits and downsides as the one in that chapter.

5.1.2 Warp to Overhead

First, a “centerpiece” frame is selected, that shows a wide view of the center of the rink. The centerpiece is then warped to overhead view by a homography estimated from a Resnet-18 based regressor.

This overhead warping regressor is trained on randomly generated rinks to warp them to the overhead view, by using the 4-point method of homography estimation. This goal is slightly different from the models in the previous chapter. The previous chapter’s model had the overhead template known as an input, and thus had some additional information not present in this case, as our ultimate goal here is to develop the overhead template. The model in this case is trained to warp centerpieces of random rinks to the overhead view while the template is unknown. It used synthetic rinks during this process, so that it would be trained to warp different types of rinks. This warping process is illustrated in Figure 5.1.

Once the warped centerpiece is known, the average distances from the center to the top edge, bottom edge, and blue lines are calculated, to generate a straightened “guide” of the center piece. This is done by calculating the average distances from the center to the blue

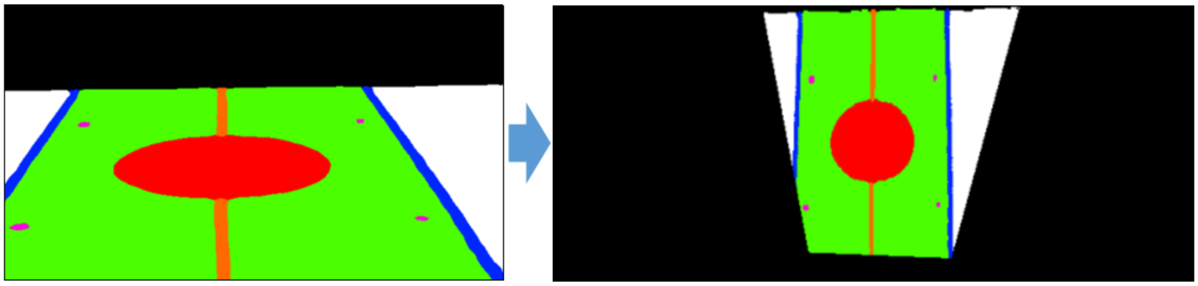
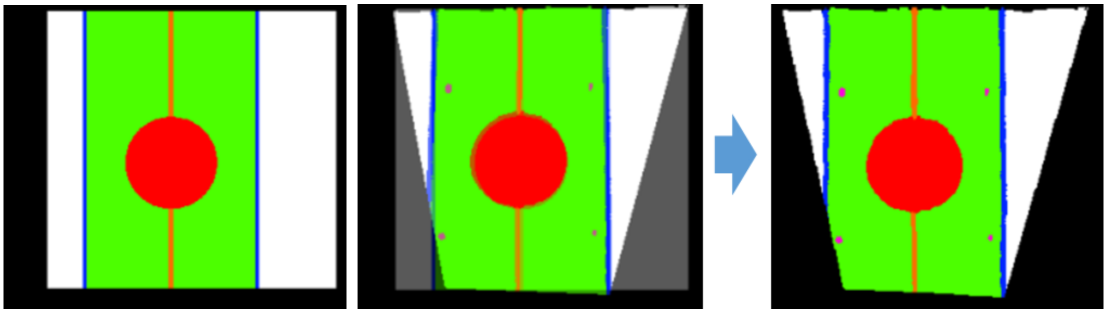


Figure 5.1: Example of warping a “centerpiece” segmentation map to the overhead view



(a) Straightening guide

(b) Straightening Process

Figure 5.2: Straightening process to ensure the lines in the centerpiece are vertical. The right image shows the overlap of the centerpiece and straightening guide, and direct homography optimization is used to align them together.

lines, along with the distance from the center to the top edge and bottom edge. An example of the straightening guide and the straightening process can be seen in Figure 5.2. The warped centerpiece is then straightened by calculating the homography between the centerpiece and straightening guide using image alignment via direct homography optimization [39] [2].

5.1.3 Image Stitching

Once we obtain the homography needed to go from centerpiece to overhead, we can now calculate the warp needed to go from other frames to the centerpiece. This is done via direct homography optimization, because this is done only once per rink type, making speed less of a concern. We calculate the homography warp between close pairs of frames,

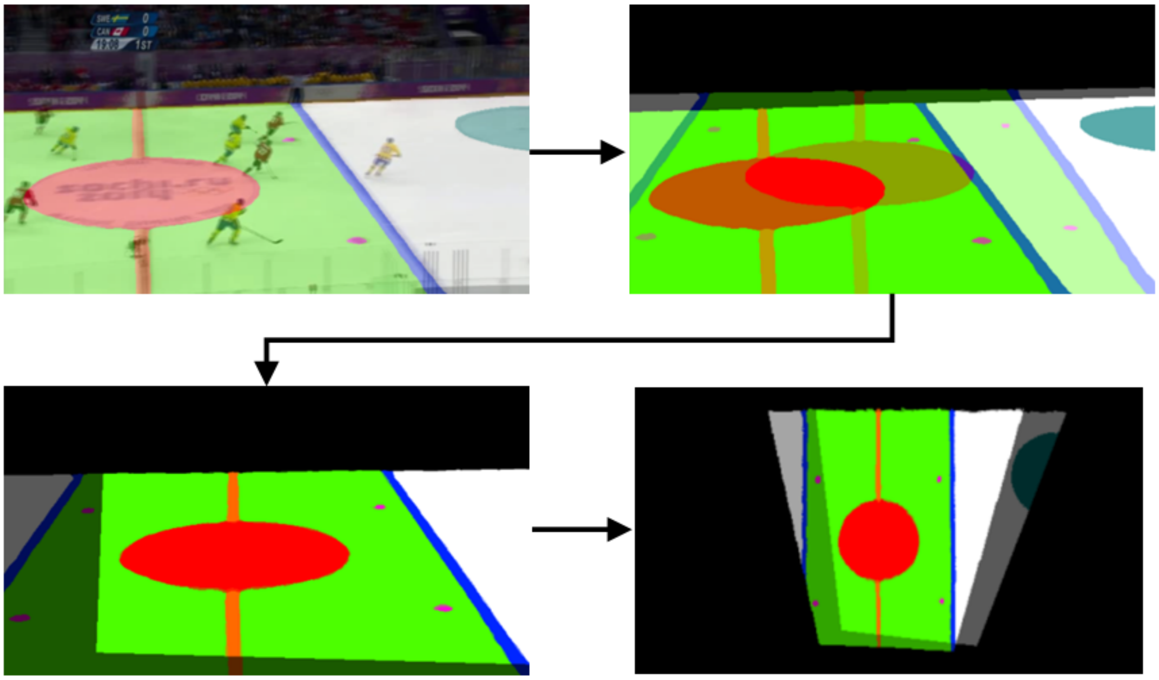


Figure 5.3: Example of segmentation, comparison, and homography warp estimation. Once the homography matrix is calculated between pairs of frames, it is combined with the overhead warping matrix to warp the new frame onto the overhead.

and then combine these warps together. When used in conjunction with the homography for warping the centerpiece to overhead, this lets us calculate the homography to go from each frame to overhead, growing the template as seen in Figure 5.3. The first image shows the segmentation of the current frame in the series. The second image shows the previous segmentation in the series overlayed on the current frame. The third image shows how alignment can be used to warp the current frame onto the previous frame. The last image shows how this alignment homography can be combined with the warp-to-overhead homography, in order to grow the rough overhead template. The portion on the right of the overlap is newly added.

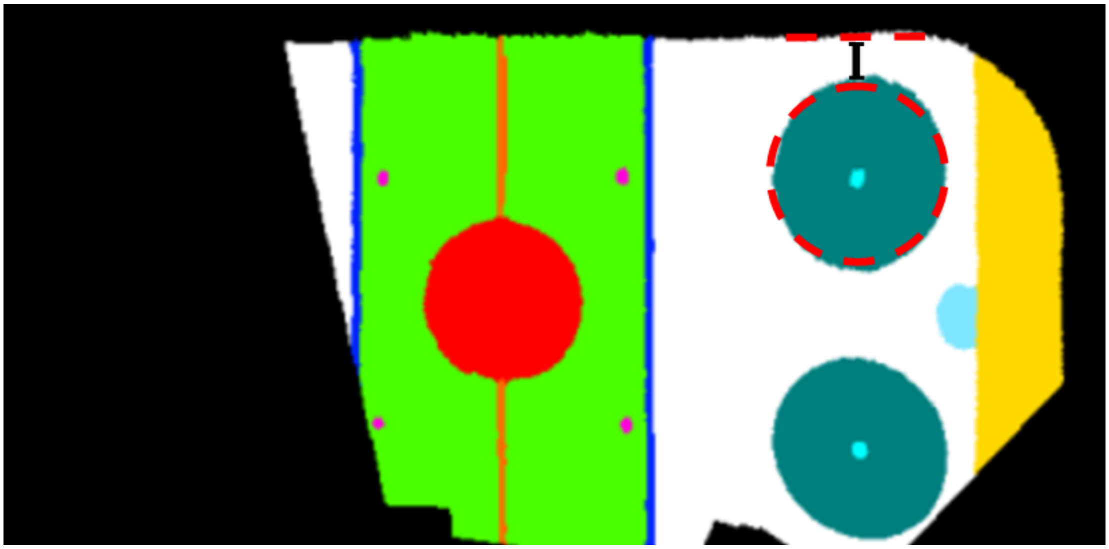


Figure 5.4: Calculating the distance between faceoff circle and rink top edge on a rough template. Averages are used to calculate the y-position of the edge (shown as a red dashed line), and the radius of the faceoff circle (shown as a red dashed circle).

5.1.4 Forming Complete Overhead Template

Now that the rough overhead template is obtained, the parameters needed for rink generation can be extracted. The parameters and rink generation scheme are the same as described in section 4.2.1.

In order to calculate the 8 distance parameters, we average distances between features in the rough template. An example can be seen in Figure 5.4. Here, we wish to calculate the distance between faceoff circle and the top of rink. The average value of the top edge pixels in the vicinity are used as the top edge value, represented by the red dashed line. The average distance to the edge of the rough faceoff circle is used as the radius, with the estimated circle shown in red. The distance between the two features is then calculated and represented as a black line between the features in this example.

The rink is supposed to be symmetric both horizontally and vertically, so only a quarter of the rough rink needs to be visible in order to start estimating distances.

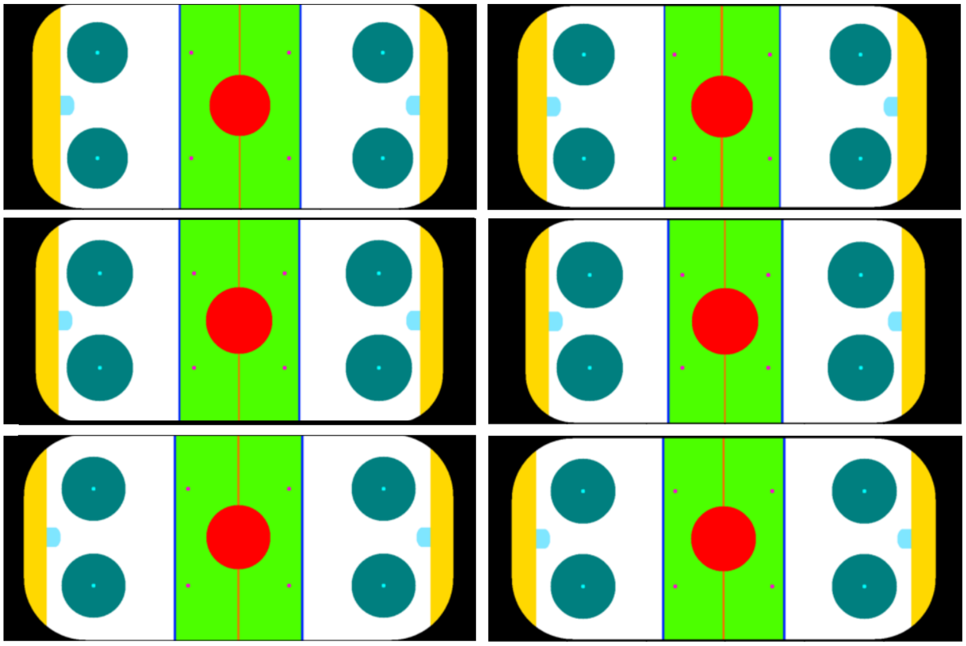
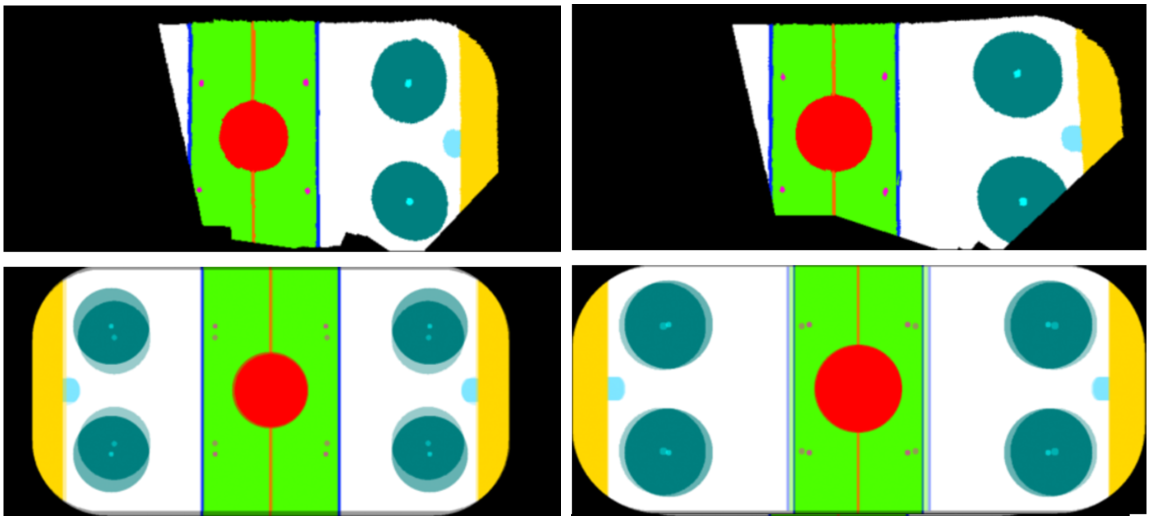


Figure 5.5: Rink generation on a series of synthetic rinks. The left side are the re-generated rinks, whereas the right side are the original rinks. Usually the re-generated rink is close to the original in these cases.

5.2 Results

This system was tested on both synthetic and real rinks. For synthetic rinks, the images consisted of warped versions of the rink template, bypassing the segmentation stage as the warped rink templates are already segmented. When the system is run in this case, the results are generally good, as the system can re-generate the original rinks used, as seen in Figure 5.5.

When tested on real rinks, the results were mixed, depending on the quality of the segmentation and image stitching. Issues in segmentation can cause the homography direct optimization to be off slightly, causing errors during image stitching. An example of reconstruction from a series of frames of real rinks can be seen in Figure 5.6. The Olympic 2014 rink reconstructed version seems to have calculated the distance between faceoff circles and top of edge to be smaller than the true value. This results in the turquoise faceoff circles being closer to the edge compared to the true location. For the NHL rink, the horizontal


(a) Olympic 2014

(b) NHL

Figure 5.6: Reconstruction of two rinks. Rough templates are shown above, and a comparison of the reconstructed version vs the ground truth are seen below.

location of various features seems to be closer to the middle compared to the true location, resulting in faceoff circles and blue lines being shifted closer to the center than the true values. Nevertheless, these results are promising, and will improve if the segmentation and alignment issues are reduced.

5.3 Conclusion

Rink generation from a series of frames may be needed in cases where the rink template is unavailable. The rink template provides the fixed reference template that broadcast video frames are warped onto, making it an important part of rink registration. Issues with the template also have negative effects when trying to align it with rink features during homography estimation. Thus, it is important to have a template available that is accurate. This rink generation method provides a way to estimate rink parameters and is a novel way to generate a rink template from select broadcast frames. It is able to re-create synthetic rinks, and provides a suitable estimate for real rinks as long as the segmentation is clean.

Issues to be addressed mostly involve improving the segmentation model to be more robust to different rink types. Replacing the direct homography estimation during the image stitching portion with a homography warp estimator deep learning model could also improve robustness against segmentation inaccuracies, but may not be as accurate in clean cases. Ideally, the best case is still obtaining the rink parameters directly, such as from a diagram or image of the overhead rink.

Chapter 6

Conclusion

Rink localization is an important aspect of hockey analytics, as it allows for all video frames to be warped onto the same reference template. In conjunction with player tracking, it allows for the location of everything in the video to be identified. This enables further analysis such as velocity calculations, direction of movement, and scoring and passing opportunities. With this information and further analysis, teams can make better informed decisions during hiring and improve player performance, which is all vital in such a competitive sport.

This thesis presents various improvements made for automatic rink localization from video. First, robustness improvements were implemented on an existing homography estimation method, and a temporal setup was also tested using those models.

Then, a novel rink-agnostic setup was developed, with the goal of providing homography estimation for all rinks as long as the template was available, regardless of rink type or appearance. This system would work without the need for additional ground truth training data for other rink types and does not need any additional rink-specific training for non-NHL rinks. This was achieved thanks to the use of domain adaptation, augmentations, synthetic data, and self-supervised learning.

Finally, a novel rink generator was proposed, in order to estimate the rink template in cases where it is unavailable. All three systems rely on broadcast video information, which is readily available for rinks without the need to install additional technology. This reduces the time and cost associated with the analysis, making it more equitable and more available to smaller groups. With the robust rink registration models described in this thesis, location information can be more readily obtained in a wide variety of matches and rinks. This allows for further analysis to be done, even on non-NHL rinks.

6.1 Directions for Future Research

Further research can be made to improve the accuracy of the models used in these systems. For example, the rink-agnostic homography estimator and the rink generator both rely on segmentation, so improving its robustness would allow for more accurate estimations, especially on rinks with differing appearance to standard NHL rinks. This may be done by testing new state-of-the-art domain adaptation techniques, to help train this model on more varied rinks.

Smoothing techniques and additional temporal information can also be applied to the homography estimation systems developed in this thesis. This would take advantage of the fact that the input is video, and can help reduce the effect of singular “bad” frames with blurs, motion, or other imperfections.

Improved image stitching can also be used for the rink generator. This can help make the rough rink generation step more robust, as currently it is susceptible to issues with segmentation. By having a more robust image stitching system, it may be possible to ignore some minor segmentation issues.

References

[1] International Ice Hockey Federation ice rink guide. https://www.iihf.com/en/static/5890/iihf-ice-rink-guide.

[2] Kornia Authors. Image alignment by homography optimization. https://kornia-tutorials.readthedocs.io/en/latest/_nbs/homography.html.

[3] Jianhui Chen and James J Little. Sports camera calibration via synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[7] Yen-Jui Chu, Jheng-Wei Su, Kai-Wen Hsiao, Chi-Yu Lien, Shu-Ho Fan, Min-Chun Hu, Ruen-Rone Lee, Chih-Yuan Yao, and Hung-Kuo Chu. Sports field registration via keypoints-aware label condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3523–3530, 2022.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] Martin Formánek. Nokia Arena, Tampere. https://www.eurohockey.com/arena/2154-nokia-arena-tampere.html.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[12] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[13] Christina Gough. NHL league revenue 2005-2021. https://www.statista.com/statistics/193468/total-league-revenue-of-the-nhl-since-2006/, Aug 2022.

[14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[15] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5212–5220, 2017.

[16] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9924–9935, 2022.

[17] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 372–391. Springer, 2022.

[18] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. MIC: Masked image consistency for context-enhanced domain adaptation. *arXiv preprint arXiv:2212.01322*, 2022.

[19] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[21] Wei Jiang, Juan Camilo Gamboa Higuera, Baptiste Angles, Weiwei Sun, Mehrsan Javan, and Kwang Moo Yi. Optimizing through learned errors for accurate sports field registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 201–210, 2020.

[22] Matthew CH Lee, Ozan Oktay, Andreas Schuh, Michiel Schaap, and Ben Glocker. Image-and-spatial transformer networks for structure-guided image registration. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pages 337–345. Springer, 2019.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015.

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[28] Alison Lukan. Beyond the box score - an intro to hockey analytics. https://www.nhl.com/kraken/news/beyond-box-score-intro-to-hockey-analytics/c-335471754, Sep 2022.

[29] Xiaohan Nie, Shixing Chen, and Raffay Hamid. A robust and efficient framework for sports-field registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1936–1944, 2021.

[30] Research and Markets. Global sports market forecast to 2032: Sector is expected to reach $623.63 billion in 2027 at a cagr of 5%. https://www.globenewswire.com/en/news-release/2023/05/03/2660537/28124/en/Global-Sports-Market-Forecast-to-2032-Sector-is-Expected-to-Reach-623-63-Billion-in-2027-at-a-CAGR-of-5.html, May 2023.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[32] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[33] Long Sha, Jennifer Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly. End-to-end camera calibration for broadcast videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13627–13636, 2020.

[34] Jia Cheng Shang, Mehrnaz Fani, David A Clausi, and Mohammad Javad Shafiee. Improved hockey rink localization via augmentation and temporal frame analysis. *Journal of Computational Vision and Imaging Systems*, 8(1):45–47, 2022.

[35] Feng Shi, Paul Marchwica, Juan Camilo Gamboa Higuera, Michael Jamieson, Mehrsan Javan, and Parthipan Siva. Self-supervised shape alignment for sports field registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 287–296, 2022.

[36] Matthew Sinclair, Andreas Schuh, Karl Hahn, Kersten Petersen, Ying Bai, James Batten, Michiel Schaap, and Ben Glocker. Atlas-ISTN: joint segmentation, registration and atlas construction with image-and-spatial transformer networks. *Medical Image Analysis*, 78:102383, 2022.

[37] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[38] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Nature, 2022.

[39] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007.

[40] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30, 2017.

[41] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021.

[42] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[44] Evan Weiner. Not every 200 foot by 85 foot NHL rink is the same. https://www.nhl.com/news/not-every-200-foot-by-85-foot-nhl-rink-is-the-same/c-501626, Oct 2009.

[45] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[46] Neng Zhang and Ebroul Izquierdo. A high accuracy camera calibration method for sport videos. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2021.

[47] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[48] Qiang Zhou and Xin Li. STN-homography: Direct estimation of homography parameters for image pairs. *Applied Sciences*, 9(23):5187, 2019.