Automated image registration of cerebral digital subtraction angiography

V.J.W. Hellebrekers

Image registration of cerebral digital subtraction angiography

Finding spatial correspondence between pre-/post intervention DSA series

Thesis report

by

V.J.W. Hellebrekers

to complete Nanobiology Master end-project at the
Delft University of Technology & Erasmus University Rotterdam
to be defended publicly on January 25, 2023 at 15:30

Thesis committee:	
Chair:	Dr. ir. Theo van Walsum
Supervisors:	Dr. I. Smal
External examiner:	Drs. S.A.P. Cornelissen
Place:	Erasmus MC, Rotterdam
Project Duration:	November, 2021 - January, 2023
Student number:	4453840

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Applied sciences · Delft University of Technology
· Erasmus University Rotterdam

Copyright © V.J.W. Hellebrekers, 2023
All rights reserved.

Preface

In thesis I describe the findings of the research I conducting as part of the Nanobiology master end-project, while working with the Biomedical Imaging Group Rotterdam (BIGR), part of the Department of Radiology & Nuclear Medicine of the Erasmus MC.

The thesis is composed of two parts. The first part is the scientific article that describes the main findings for our objective to align cerebral DSA series. I would like to express my gratitude towards Theo van Walsum, Ihor Smal, Matthijs van der Sluijs and Ruisheng Su for their continuous support and feedback throughout the writing process, numerous revisions and the submission process. Formulating the work in such a dense form factor together with the revision process was a unique experience.

The second part of the thesis includes more extensive information regarding the clinical and technical background of the project. Additionally, it contains information on deep-learning, a field which intrigued me during my studies and something I wanted to gain practical experience in. The deep-learning section includes some additional results that were not included in the scientific article. Their added value to the objective of DSA alignment proved limited. Formulating an understanding of these deep-learning methods, and their limitations, became an objective of its own and is therefore included as an extension of the background knowledge on deep-learning.

This thesis is the product of a year long journey with many wonderful people along the way. I would like to thank Theo van Walsum, Ihor Smal, Matthijs van der Sluijs and Ruisheng Su for the weekly meetings. I am grateful for the advice and guidance, for sparking enthusiasm during the covid lockdowns and for giving me the liberty to explore new methods. I also want to thank the researchers and fellow students at BIGR for advice, discussions and inspiration. Finally I want to thank my family and friends for their support, motivation and (for some) enduring proof-reading. Looking back I cannot imagine doing this project without you, and am proud to share this milestone with you.

Vincent Hellebrekers
December 2022

Colophon

This thesis was formatted using the Springer Nature LaTeX template. The Springer Nature style was used to comply to the IPCAI submission standards, and used for part two in order to retain consistent formatting.

Contents

Part I: Scientific article

Automated image registration of cerebral digital subtraction angiography

Vincent Hellebrekers[1], Theo van Walsum[1], Ihor Smal[1], Sandra A. P. Cornelissen[1], Wim H. van Zwam[2], Aad van der Lugt[1], Matthijs van der Sluijs[1] and Ruisheng Su[1*]

[1]Erasmus MC, University Medical Center Rotterdam.
[2]Maastricht University Medical Center.

*Corresponding author(s). E-mail(s): r.su@erasmusmc.nl;

Abstract

Purpose: Our aim is to automatically align Digital Subtraction Angiography (DSA) series, recorded before and after endovascular thrombectomy. Such alignment may enable quantification of procedural success. **Methods:** Firstly, we examine the inherent limitations for image registration, caused by the projective characteristics of DSA imaging, in a representative set of image pairs from thrombectomy procedures. Secondly, we develop and assess various existing image registration methods (SIFT, ORB, elastix). We assess these methods using manually annotated point-correspondences for thrombectomy image pairs. **Results:** Linear transformations that account for scale differences are effective in aligning DSA sequences. Two anatomical landmarks can be reliably identified for registration using a U-net. Point-based registration using SIFT and ORB prove to be most effective for DSA registration and are applicable to recordings for all patient sub-types. Image based techniques are less effective and did not refine the results of the best point-based registration method. **Conclusion:** We developed and assessed an automated image registration approach for cerebral DSA sequences, recorded before and after endovascular thrombectomy. Accurate results were obtained for approximately 85% of our image pairs.

Keywords: Digital subtraction angiography, Ischemic stroke, Endovascular thrombectomy, Image registration

1 Introduction

Stroke is a leading cause of disability and death [1]. In case of a stroke, blood circulation in a region of the brain is compromised. Ischemic stroke is the most common type of stroke, caused by thrombo-embolic occlusion of a brain artery [2]. A minimally invasive procedure known as endovascular thrombectomy (EVT) aims to restore blood flow by mechanical removal of the thrombus using a catheter and stent retriever. In addition, such a procedure allows the intra-arterial administration of clot-dissolving medicine.

Such interventional treatments are guided by fluoroscopy, a low dose X-ray imaging modality. Intermittently, Digital Subtraction Angiography (DSA) is used to visualize the vessels. In DSA imaging, a series of 2D X-ray images is recorded while a contrast medium is injected into the patients' blood vessel through the catheter. A background image (i.e. an image before contrast injection), is digitally subtracted from subsequent X-ray images with contrast medium, resulting in a sequence of images visualizing the contrast medium progressing through the arteries, tissue and veins. Once the procedure is completed (or terminated), the radiologists examine the DSA sequences to grade the procedure using Thrombolysis in Cerebral Infarction (TICI) scoring [3].

Recent studies have made significant advancements in diagnosis for treatment selection and prognosis using pre-procedural information [4]. And, while recent studies have demonstrated image processing strategies to be capable of analyzing DSA sequences, automatically extracting DSA bio-markers [5] and automatic TICI scoring [6] [7], this modality has not been widely used for these purposes. A quantitative comparison of the vessels (or perfusion) before and after the intervention may lead to a better understanding of the result of the intervention, and may also permit prediction of clinical outcome. Such a pre-post image comparison is currently hampered by the lack of an accurate spatial alignment of the sequences obtained before and after the treatment.

Automating such alignment is challenging, as there may be new arteries visualized after a (partially) successful thrombus removal. Additionally, spatial correspondence likely requires a non-linear deformation, even for subsequent frames, as is indicated in previous work [8]. Finally, the orientation of the imaging setup, with respect to the patient, can vary significantly during a procedure, as the ischemic stroke patient will move during the procedure. Additionally, the radiologist changes the orientation intermittently for anterior-posterior (AP) or lateral views.

One previous study [9] compared manually obtained transformations to those computed by a wide range of registration methods on a small dataset from the UCLA stroke center. They conclude image-based methods perform consistent, while point-based methods are more accurate but highly variable.

In this work, we aim to develop and assess an image registration strategy on a large set of images using a quantitative metric. We will investigate which type of transformations is effective in aligning different DSA series. Subsequently, traditional registration methods and a deep learning method are adapted and assessed for automated alignment.

2 Methods

The effects of patient movement and differences in C-arm orientation, inherently present in DSA data, may require additional transformation complexity for effective alignment. Ultimately, it is not apparent what transformation type is suitable to model the projection of 3D motion. We therefore first empirically investigate what transformation type is suited for spatial alignment by fitting different 2D transformations to manually annotated point correspondences.

Subsequently, we assess automatic registration techniques. We first develop a deep learning model to identify two cerebral artery landmarks, which will provide point correspondences for all DSA sequences. For more accurate alignment of sequences pre- and post-EVT images of the same patient, point correspondences from traditional methods, SIFT (Scale-Invariant Feature Transform) [10] and ORB (Oriented FAST and Rotated BRIEF) [11], are used. Finally, we examine image based registration and its potential to further improve solutions from SIFT and ORB.

2.1 Transformation fitting

Linear transformations have few degrees of freedom and can therefore be computed using few point correspondences [12]. Computing a solution using the minimally required number of point correspondences will generally provide an exact fit. Using additional point correspondences does not guarantee a solution, and a least-squares solution is typically used. In some cases, a transformation type may not have the degrees of freedom to align point sets accurately. In such cases, the over-determined linear system benefits from L1 optimization. This will provide a solution that, on average, results in better alignment for a predominant subset. L1 optimization may also be desirable to reduce the effect of outliers when fitting a transformation to automatically detected point correspondences. Most Transformations have closed form L2 solutions, whereas L1 solutions must be approximated numerically.

2.2 Cerebral landmark model

A first approach to automatic alignment is obtained by exploiting common landmarks in both images, and using a simple transformation to align these landmarks. For this, we selected the ICA and M1 (bifurcation) as landmarks, as these vessels are generally visible. Instead of using a standard object detection approach, we opted to use a U-net [13] to compute the probability distributions of the location of the two landmarks (see Figure 3). In this neural network model, the final activation layer is a sigmoid function, thereby enforcing the lower and upper bounds of the probability values. The ground-truth probability distributions are normal distributions located at the manual annotations with a fixed standard deviation of 2.5 pixels. At inference, the final landmark positions are determined by the highest probability (argmax) or expectation (centre of mass). Kullback–Leibler (KL) divergence and Jensen–Shannon (JS) divergence are used as loss functions.

2.3 Automatic point-correspondence detection

Instead of explicit landmark detection, Point-correspondences can also be determined automatically using algorithms such as SIFT and ORB. Both methods can detect points based on local features and output a descriptor for pairing of points between images. A transformation can be fit to these point pairs to obtain a transformation, as described in 2.1. Points extracted by SIFT are local optima of the second derivative, and for the pairing each point is described using a local histogram of gradients. ORB extracts corners using FAST and describes each point using an oriented BRIEF descriptor. Outliers in the pairing are dealt with via a custom implementation of RANSAC [14], where the number of inliers is only evaluated if the scaling factor s and rotation angle θ of a similarity transformation are within reasonable bounds: $\frac{1}{1.5} < s < 1.5$ and $|\theta| < 45°$. Outliers are excluded when computing the definitive transformation.

2.4 Image based registration

Next to point-based approaches, we also investigate traditional image based approaches. These approaches optimize an image-based similarity metric, generally via an (adapted) gradient descent optimizer.

We experiment with three different similarity metrics: Mattes Mutual information [15], Mean Squared Difference and Normalized Cross Correlation. Mattes Mutual information is a very common multimodal similarity metric, whereas the two other metrics are meant for registration of images of the same modality. We compute either a similarity or an affine transformation.

2.5 Combined strategies

The methods discussed in the preceding sections could be improved by combining different approaches. For this, different strategies are used:

1. **landmark detection**: U-net models have been trained using different loss functions. We can combine the output probability distributions (of the different model instances) using the multiplication rule for independent probabilities, $p(x, y \| p_1(x, y), p_2(x, y)...) \propto \prod_i p_i(x, y)$.
2. **point-based registration**: a transformation is fit to automatically identified point-pairs. We combine the point-pairs from different methods before defining the inliers with RANSAC. This increases the number of point-pairs used to compute the transformation. We include every combination of point pairs from SIFT, ORB and the landmarks in our evaluation.
3. **image based registration**: gradient-descent based optimizers are used. Transformation parameters are gradually changed to improve image similarity. Such approaches are very dependent on the initial alignment, but generally converge to a local optimum. We therefore additionally evaluate initialization using the best performing point-based registration method,

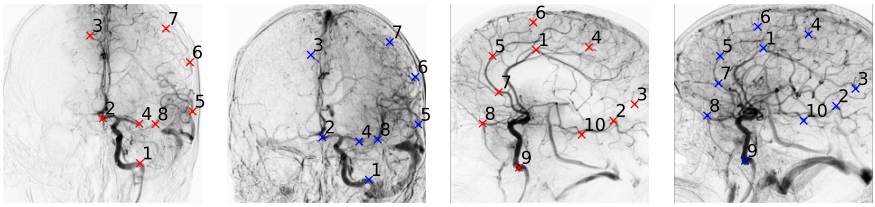

Fig. 1 Manually annotated point-correspondences for pre-/post-EVT (red/blue) minIPs for AP and lateral views (left/right).

thereby assessing the capacity of point-based registration to improve image based registration and vice versa.

3 Data

In this work we use imaging data from the MR CLEAN Registry [16], a registry of consecutive stroke patients treated with EVT in the Netherlands. This registry contains DSA recordings from various imaging systems, which are different in terms of image quality, appearance and recording parameters. Additionally, information on occlusion location, TICI score and use of general anesthesia are available, and used to further analyse the results.

3.1 EVT image registration dataset selection

A selection of pre- and post-EVT sequences from the MR CLEAN Registry is adopted from a previous study [7]. This study selected sequences for automatic procedure evaluation and are therefore particularly of interest. Additionally, this selection readily ensures image quality and recording duration are sufficient for our purpose. Evaluation of the methods will be done using manual annotations. For this, a subset was selected based on the following parameters:

- General anesthesia (GA): the use of GA will likely simplify our task, as reduced patient movement will result in recordings taken from a more consistent perspective.
- Occlusion location: DSA sequences of a proximal ICA occlusion will contain little information compared to more distal occlusions.
- TICI score: the evaluation metric of EVT is based on the perceptible recovery of blood flow and perfusion, indicative of differences in content.

Six patients were included for each combination of these parameters (GA: yes/no, Occlusion location: ICA/M1/M2, TICI score: 0/2b/3); except for one combination (no GA, ICA occlusion, 0 TICI score) as only two such patients are present in the selection of Su et al. [7]. This results in 104 patients to be incorporated in the evaluation.

Corresponding points for AP and lateral pre-/post-EVT images are needed for assessing the impact of the transformation model (see 2.1), and for assessing

automatic registration. Therefore, corresponding points for AP and lateral pre-/post-EVT image pairs of the 104 selected patients were manually annotated. An example is shown seen in Figure 1. Up to ten point-pairs are annotated per image pair, fewer if insufficient corresponding regions are present. The annotations were validated by a second observer. To accurately represent alignment, points are chosen to be well distributed in the field of view, and predominantly positioned on the major cerebral arteries (ICA, ACA, MCA and PCA).

3.2 Cerebral landmark dataset

A deep learning approach was implemented to locate anatomical landmarks in DSA sequences. This approach benefits from a larger dataset than the images annotated for transformation assessment. We therefore use a larger collection of intra-procedural recordings from the MR CLEAN Registry. From the first 1000 patient records, all 3532 lateral sequences and the first 5000 AP sequences (from the first 644 patient records) are included.

Only sequences containing both landmarks were included for training and testing, thereby excluding procedural recordings, recordings with a small field of view or of low quality. This results in 1716 AP sequences and 1472 lateral sequences to be included. Images are oriented in such a way that the AP sequences display the contrasted hemisphere on the right, and lateral sequences display the patient facing to the left.

4 Experiments and Results

4.1 Implementation details

The proposed methods were implemented in Python. We make use of the following frameworks: OpenCV 4.5.4.58, Scikit-image 0.19.3, SciPy 1.7.2, TensorFlow 2.4.1 and elastix 2.0 [17]. Transformation fitting is performed using Skimage estimate transform (L2) and SciPy minimize (L1). A U-net is implemented in TensorFlow. It is composed using two ReLu activation layers per down-sampling, four max-pooling layers and includes drop-out and batch-normalization. The KL divergence loss function can be computed forward and backward, both are used. Additionally, we also compute KL divergence implicitly by converting the output to a normal distribution using its mean and variance. For SIFT and ORB, we use the OpenCV implementations with default parameters. Point matching is performed using brute force for consistency. Lowe's ratio test is applied (best match < 0.75 second best match) using the L2 distance between SIFT descriptors. ORB points are matched using Hamming distance, requiring a bi-directional closest match. Image based registration is done using the elastix framework and makes use of the default parametermap for affine registration, only modifying the similarity metric, transformation type and initialization.

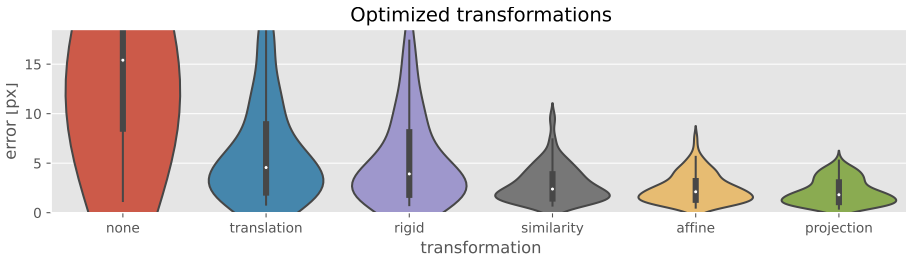

Fig. 2 Average residual error distributions of transformations optimized for annotations of AP MinIP pairs.

4.2 Evaluation metrics

The contents of DSA sequences recorded pre-/postEVT can change significantly. An image similarity measure is therefore not suitable for evaluation. A registration result will therefore be quantitatively assessed by the mean Euclidean distance between the annotated points and the transformed corresponding points from the aligned image. In subsection 4.3 we examine transformations that minimize this metric by directly fitting to the annotated point correspondences. From Figure 2 we conclude registrations to be capable of achieving ≤ 10 pixels average distance error, which will be used as criterion for successful registration.

For the accuracy of landmark detection, we evaluate the Euclidean distance between the predicted coordinate and a manual annotation.

4.3 Intra-patient manual transformation assessment

To assess the impact of additional degrees of freedom on alignment accuracy, global transformations are optimized for manual annotations of pre- and post-EVT recordings. Image pairs with fewer than six point correspondences are excluded to prevent overfitting. The resulting error distributions per transformation type are shown in Figure 2. Transformations that account for scale differences achieve good results, with additional degrees of freedom achieving marginal improvements.

4.4 Landmark detection

For the assessment of the U-net based landmark detection, we performed a three-fold cross-validation. In this cross-validation, the data is randomly split based on patient id, thereby preventing validation and training on images from the same patient. Models were trained using various loss functions and the Adam optimizer until convergence was achieved (supplementary Figure 6, 7, 8 and 9). Weights are stored for the epoch with the best centre-of-mass prediction error on the validation set. The results are shown in Figure 3.

A Student t-test was used to assess to what extent the differences between the results of the models were statistically significant and are presented in

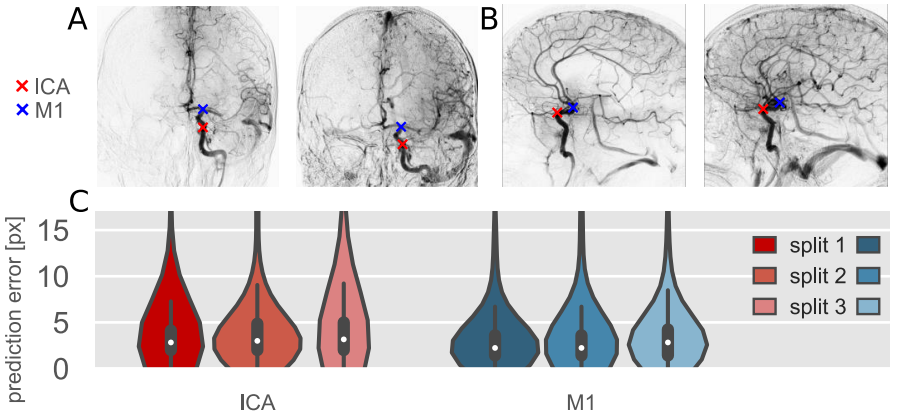

Fig. 3 Landmark model results. ICA and M1 landmark predictions for A: AP and B: lateral minIPs. C: AP landmark prediction error distributions for the three data splits using the best performing configuration.

supplementary Table 4 and 5. Four loss functions (forward and backward KL-divergence, JS-divergence and implicit forward KL-divergence) performed best in landmark detection.

Using these results, two combination models were assembled (using the four best performing models and three of these models, excluding the implicit KL-divergence), evaluated and included in the t-test comparison supplementary Tables 2, 4, 3 and 5. Significant improvements are observed for both combination models, of which the combination model using the three explicit loss functions performs best. The results of the best performing combination model are shown in Figure 3.

4.5 Point-based registration

We also investigated point-correspondences automatically identified using SIFT and ORB. Additionally, we combine these SIFT/ORB based point correspondences with landmarks detected by U-net. For each combination, the best performing global transformations are computed using inliers identified using RANSAC. Both L2 and L1 optimized transformations are examined. The success rate of finding sufficient inliers (≥ 5) for each combination of these methods is appended to Table 1. A complete comparison of all transformation types, norm optimizations and methods, (36 variations in total) using a z-test is included in supplementary Table 1, and 2.

For all methods, the projection transformation performed significantly worse than comparably performing similarity and affine transformations. Additionally, no significant effect could be observed for L1 over L2 optimized solutions. Therefore, the simplest transformation, the similarity transform with L2 optimization, will be used as initialization when combining the best performing point-based registration with image-based registration. Accuracy of

Table 1 Number of solutions (TP+FP) and invalid solutions (FP) found using automatically identified point correspondences for 104 image pairs.

Methods			AP		Lateral	
SIFT	ORB	LM	TP+FP	FP	TP+FP	FP
✓	×	×	58	0	65	2
×	✓	×	101	23	103	21
✓	✓	×	101	19	104	16
✓	×	✓	67	1	76	1
×	✓	✓	102	23	104	19
✓	✓	✓	101	20	103	16

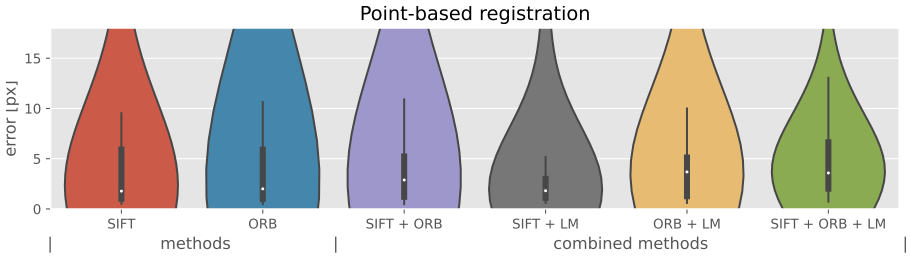

Fig. 4 Registration error for least-squares similarity transformations using automatically identified point correspondences in lateral images. See supplementary Figure 12 for AP results.

the different methods using the similarity transformation and L2 optimization is shown in Figure 4.

4.6 Image based registration

Image based algorithms such as elastix are commonly used in medical image registration. We compared its performance to the best performing point-based registration, by performing a normal registration, and also when initializing the registration with the result of the best-performing point-based registration. All twelve variations (two transformations, three similarity metrics and with/without initialization) and the point-based method are compared using a Z-test in supplementary Tables **??**, **??**. Mattes Mutual information and normalized cross-correlation perform comparably, while mean square difference performs significantly worse. The similarity transformation outperforms the affine transformation. Initialization using the point-based registration method improves the results from elastix (supplementary Figure 13 and 14). However, it does not improve the results from the point-based method.

5 Discussion

In this study, we assessed various transformation models for intra-patient alignment of DSA images during EVT, and we developed and assessed methods for automated alignment of DSA images.

Alignment using manually annotated point-correspondences shows linear transformations, that account for scale differences, to be capable of aligning cerebral DSA images with relatively small errors.

The proposed U-net architecture is effective at identifying two anatomical landmarks by optimizing losses based on the KL-divergence. Training using implicit implementations of the KL divergence is unstable (supplementary Figure 6, 7, 8 and 9). To stabilize training for these loss functions, we included a smoothing kernel before computing the loss, but this proved counterproductive.

Extracting a coordinate from the probability distribution is done using the argmax and centre of mass. The latter is more sensitive to false positives and was therefore chosen for selecting a stable model, while the argmax is better for more accurate results. Interestingly, for the implicit loss functions (whose performance was worse overall), the centre of mass proved more accurate. A closer look at their output distributions showed that, instead of a Gaussian distribution, like the ground truth, a circular uniform distribution was produced. The argmax was therefore extracting a point on the periphery, causing the less accurate results.

Combining models trained with different loss functions significantly improved accuracy to approximately 4 pixels. Two landmarks are insufficient for accurate alignment (supplementary Figure 10 and 11), but do generalize to most cerebral DSA sequences. It may therefore be suited for inter-patient registration.

For more accurate intra-patient alignment, traditional point-based methods were examined. SIFT provides accurate results (60%) and negligible (1%) false solutions. Including the cerebral landmarks improves the number of correct solutions (10%). ORB provides more point correspondences, including more outliers, resulting in more correct (15%) and incorrect solutions (15%). Remarkably, while including anatomical landmarks to the SIFT or ORB point correspondences improves results, the joint set of SIFT and ORB point correspondences proved most accurate, with negligible performance differences when including the cerebral landmarks.

The image based approaches produced results that are less accurate than compensating translation using the cerebral landmarks alone, supplementary Figure 10 and 11. Similarity or affine transformations performed comparably. Mattes mutual information and normalized cross-correlation achieved similar results, while mean squared difference proved significantly worse. The high-frequency signals of angiograms are likely too challenging for gradient-based optimization. A commonly used approach to address this, is the use of a distance map from segmented vessels. Unfortunately, this is likely not applicable to ischemic stroke patients with a proximal occlusion and successful procedural outcome and was therefore not examined. It is known that these approaches, that use a gradient-based optimizer, are sensitive to the initialization, i.e. a proper initialization is a prerequisite for a good registration result. Initializing this method with the result of the best performing point-based method did not

significantly improve the results of the point-based registration. A marginal difference was observed and image based registration methods ($p \approx 0.9$) are therefore not recommended for intra-patient alignment.

We have shown automatic registration methods to achieve high accuracy image alignment for a significant subset (85%) of ischemic stroke patients in our dataset. Most of the remaining patients have one (or even two) DSA series with few visible vessels. Achieving successful alignment for such patients will require additional information to be incorporated, either in the images (i.e. by including unsubtracted X-ray images) or the registration method (i.e. by using a distance map of the background or segmented ICAs).

Image alignment will likely improve (automatic) analysis of procedural DSA series, such as automatic TICI scoring and bio-marker comparison. If such tools are adopted in the clinic, our alignment could aid in repositioning the C-arm such that its orientation w.r.t. the patient is equivalent to that of the pre-EVT DSA.

6 Conclusion

We have investigated approaches to automatically align cerebral DSA series. Transformations that account for differences in scale are capable of aligning cerebral DSA sequences. A deep-learning strategy using the U-net architecture proved capable of identifying cerebral artery landmarks to $4px$ accuracy. Image registration of pre-/post-EVT DSA sequences can be performed using traditional point-based methods with 85% success and comparable performance for various types of stroke patients and procedural outcomes. This will enable further automation of DSA image analysis and procedure evaluation, contributing to outcome prediction and procedural decision making for EVT.

Acknowledgments

The authors acknowledge the support of Q-Maestro (EMCLSH19006) and the CONTRAST consortium (CVON2015-01: CONTRAST).

References

[1] WHO. WHO methods and data sources for country-level causes of death 2000-2019. Global Health Estimates Technical Paper. 2020;.

[2] Campbell BC, De Silva DA, Macleod MR, Coutts SB, Schwamm LH, Davis SM, et al. Ischaemic stroke. Nature Reviews Disease Primers. 2019;5(1):1–22.

[3] Higashida RT, Furlan AJ. Trial design and reporting standards for intra-arterial cerebral thrombolysis for acute ischemic stroke. stroke. 2003;34(8):e109–e137.

[4] Haussen DC, Dehkharghani S, Rangaraju S, Rebello LC, Bouslama M, Grossberg JA, et al. Automated ct perfusion ischemic core volume and noncontrast ct aspects (alberta stroke program early ct score) correlation and clinical outcome prediction in large vessel stroke. Stroke. 2016;47(9):2318–2322.

[5] Scalzo F, Liebeskind DS. Perfusion angiography in acute ischemic stroke. Computational and Mathematical Methods in Medicine. 2016;2016.

[6] Prasetya H, Ramos LA, Epema T, Treurniet KM, Emmer BJ, Van Den Wijngaard IR, et al. qTICI: Quantitative assessment of brain tissue reperfusion on digital subtraction angiograms of acute ischemic stroke patients. International journal of stroke. 2021;16(2):207–216.

[7] Su R, Cornelissen SA, Van Der Sluijs M, Van Es AC, Van Zwam WH, Dippel DW, et al. autoTICI: Automatic brain tissue reperfusion scoring on 2D DSA images of acute ischemic stroke patients. IEEE Transactions on Medical Imaging. 2021;40(9):2380–2391.

[8] Meijering EH, Niessen WJ, Bakker J, Van Der Molen AJ, de Kort GA, Lo RT, et al. Reduction of patient motion artifacts in digital subtraction angiography: evaluation of a fast and fully automatic technique. Radiology. 2001;219(1):288–293.

[9] Tang A, Zhang Z, Scalzo F. Automatic registration of serial cerebral angiography: a comparative review. In: International Symposium on Visual Computing. Springer; 2018. p. 3–14.

[10] Lowe DG. Distinctive image features from scale-invariant keypoints. International journal of computer vision. 2004;60(2):91–110.

[11] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision. Ieee; 2011. p. 2564–2571.

[12] Szeliski R. Computer vision: algorithms and applications. Springer Nature; 2022.

[13] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. p. 234–241.

[14] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. 1981;24(6):381–395.

[15] Mattes D, Haynor DR, Vesselle H, Lewellyn TK, Eubank W. Nonrigid multimodality image registration. Proc SPIE Medical Imaging. 2001;4322.

[16] Jansen IG, Mulder MJ, Goldhoorn RJB. Endovascular treatment for acute ischaemic stroke in routine clinical practice: prospective, observational cohort study (MR CLEAN Registry). bmj. 2018;360.

[17] Klein S, Staring M, Murphy K, Viergever MA, Pluim JP. Elastix: a toolbox for intensity-based medical image registration. IEEE transactions on medical imaging. 2009;29(1):196–205.

Part II: Background information

1 Clinical background

This chapter introduces the clinical application of DSA imaging, focusing on its role in stroke treatment. It elaborates on the generation of the images and refers to applications of digital analysis of DSA images.

1.1 Ischemic stroke

Stroke is a leading cause of disability and death globally [1]. Increasing incidence caused by an aging population further emphasizes the need to improve stroke treatment. Endovascular thrombectomy (EVT) has become the predominant treatment for ischemic stroke patients during the last decade [2].

During a stroke, critical access to oxygen is compromised such that a region of the brain cannot maintain homeostasis. Cerebral hypoxia (lack of oxygen in the brain) leads to unrecoverable loss of tissue resulting in disability or death. Early signs resulting from cerebral hypoxia, such as speech difficulty and loss of control over facial muscles, indicate a stroke.

In the hospital, the cause of the stroke is identified, either being hemorrhagic (29% globally), caused by bleeding of a cerebral artery, or ischemic (71% globally) [3], caused by occlusion of a cerebral artery. While the extensive anatomy of the cerebral arteries is beyond the scope of this thesis, it should be noted that the internal carotid artery (ICA) and middle cerebral artery (MCA) are of particular importance. The ICA and MCA are the predominant locations for an occlusion to occur. As the MCA traverses through multiple cerebral regions, its segments are further specified M1, M2 etc. (from proximal to distal more distal segments).

1.2 Endovascular thrombectomy

EVT is a minimally invasive procedure that aims to remove the thrombus to restore blood flow in the occluded vessel. A catheter is inserted into the groin and navigated towards the occluded vessel. There, a stent-retriever is used to grab (parts of) the thrombus. This process generally requires multiple attempts and can be aided by locally administering a solvent.

Seventeen Dutch hospitals started collaborating to form a large patient registry, the MR CLEAN Registry [2], combining all EVT patient records. The main purpose being the monitoring of implementation and safety of EVT in the Netherlands, additionally providing the opportunity to improve EVT.

1.3 DSA imaging

EVT is guided by fluoroscopy and evaluated using DSA imaging, which are (low dose) X-ray imaging techniques, recording using a C-arm (Figure 3), further elaborated in 1.4. DSA digitally processes the images to remove the static background while a contrast agent is injected into the patients bloodstream. The first frame of the series, recorded before the injection of the contrast, is digitally subtracted from the subsequent frames. This results in a series of images

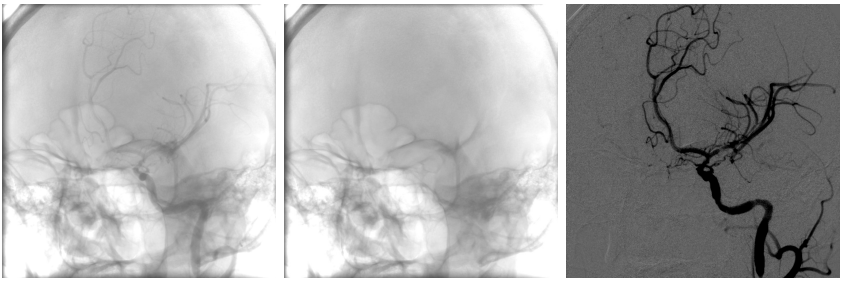

Fig. 1 DSA: Original frame, first frame and subtracted result. Original sequence was obtained from the MR CLEAN registry [2].

showing the progression of contrast through the arteries, tissue and veins. DSA imaging is performed using a higher dose than fluoroscopy to reduce noise, as it is amplified by the use of two images.

DSA is used to evaluate the recovery of blood flow and tissue perfusion. Cerebral DSA imaging is typically performed using anteroposterior (AP) and lateral perspectives, Figure 2.

Once the EVT procedure is completed (or terminated), DSA images from before and after the procedure can be compared to evaluate the recovery of blood flow using a *Thrombolysis in Cerebral Infarction* (TICI) score [4], see Table 1.

This manual scoring is prone to bias and led to methods being developed for automation. To improve such methods, it would be desirable to align the images as is explored in this thesis. Other DSA analysis objectives, such as bio-marker extraction, could also benefit from alignment [5].

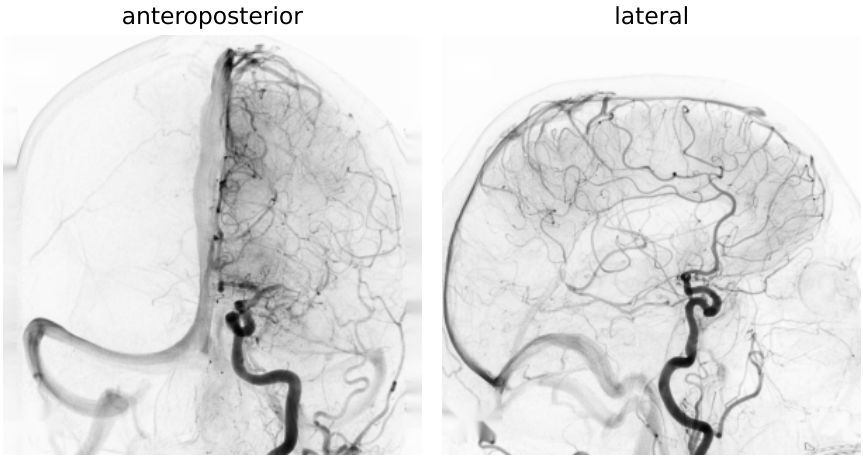

Fig. 2 Common perspectives used in DSA imaging. Original sequences were obtained from the MR CLEAN registry [2].

Table 1 TICI scoring criteria.

0	No improvement in perfusion
1	Contrast penetrates the occlusion site, but fails to restore flow to the arterial bed downstream of the occlusion site.
2	Partial Perfusion. A significant region of the arterial bed downstream of the occlusion site is restored, but blood flow is perceptibly slower than comparable areas.
2a	Partial filling is detected.
2b	Complete filling is detected.
3	Complete perfusion. Perfusion occurs to be completely recovered.

1.4 X-ray imaging

The main components of the C-arm are the X-ray tube, anti-scatter grid and detector. The X-ray tube generates the X-rays by accelerating electrons from its anode to its cathode. At the cathode, most energy is absorbed in the form of heat resulting from inelastic interaction. Elastic interactions ($\approx 1\%$) result in X-rays in the form of characteristic radiation (at cathode specific wavelengths) and Bremsstrahlung (linearly distributed wavelengths). The lower wavelength X-rays, which cannot pass through the body, are absorbed by the cathode and an additional filter to further reduce the radiation received by the patient.

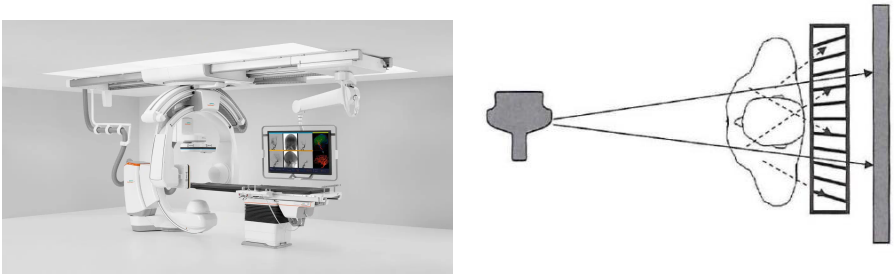

Fig. 3 C-arm [6] and a schematic representation [7] of the main components for X-ray imaging.

The remaining X-rays pass through the patient and are partially absorbed. The absorbance depends on the tissue the photons pass through. Tissue with a higher attenuation coefficient, such as bone, absorbs more photons than soft tissue, Table 2. This results in the contrast in the image. Blood and soft tissue are primarily composed of water and therefore have a similar attenuation coefficient, which unfortunately makes it difficult to distinguish these tissues. Injecting a contrast agent, such as iodine, into the bloodstream increases its attenuation coefficient improving the contrast significantly. Using a contrast agent can cause temporary symptoms such as nausea or headaches, and puts stress on the patients liver.

Table 2 Attenuation coefficients of tissues for medical imaging. Values [8] (Hounsefield units) were converted using $\mu_{air} = 0cm^2/g$, $\mu_{water} = 0.2683cm^2/g$ $[40keV]$ [9].

$\mu[cm^2/g]$	Tissue
$\geq$0.5366	Bone, calcium
0.2951 to 0.4293	Iodinated CT contrast
0.2777	Gray matter
0.2750	White matter
0.2737 to 0.2790	Muscle, soft tissue
0.2683	Water

The anti-scatter grid removes Compton scatter (X-rays whose direction has been altered) to improve image quality. The recorder, most commonly a flat-panel detector, consists of a scintillator and photodiodes (the camera). The scintillator is a material which converts higher wavelength photons, such as X-rays, into lower wavelength photons (optical spectrum). This light is then recorded using a camera to produce a digital image.

2 Technical background

This chapter will introduce the fundamental concepts of image registration followed by a detailed explanation of the algorithms used to automate this process.

2.1 Image transformations

In image registration, transformations are used to warp an image such that its objects are located at the same position as those in a fixed image. Typically, image transformations are categorised as linear or non-linear. Linear transformations model the difference in perspective, while non-linear transformations model local deformation of objects.

2.1.1 Linear transformations

Most scientific image data has an underlying orthographic geometry. This linear geometry, with each of the pixels or voxels having the same dimensions in the image, allows objects to be shifted, rotated, scaled, stretched and sheared (Figure 4) using a linear transformation:

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_{13} \\ T_{23} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}' \tag{1}$$

The projection or perspective transformation, Equation 2, is conventionally also ascribed to the class linear transformations. It completes the modelling of perspective changes of the pin-hole camera model, and even though it is non-linear, in practise its computation is performed using a linear system, either using homogeneous coordinates or by approximation such as the direct linear transform (B).

$$x' = \frac{T_{11}x + T_{12}y + T_{13}}{T_{31}x + T_{32}y + 1} \quad y' = \frac{T_{21}x + T_{22}y + T_{23}}{T_{31}x + T_{32}y + 1} \tag{2}$$

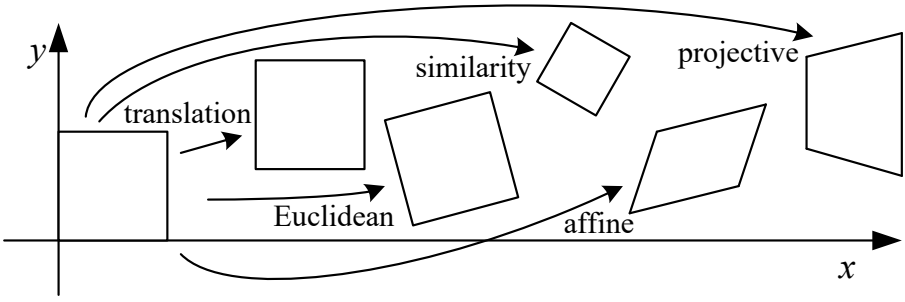

Fig. 4 Linear image transformations. **translation:** pure translation. **Euclidean:** translation & rotation. **similarity:** translation, rotation & scaling. **affine:** linear transformation without constraints, Equation 1. **projective:** pin-hole perspective modelling, Equation 2. Image obtained from [10].

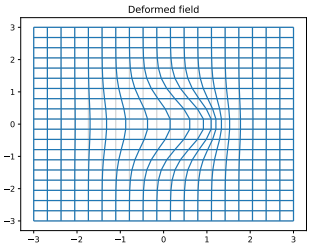

Fig. 5 An example of a non-linear transformation. Image was generated using [11].

2.1.2 Non-linear transformations

Non-linear transformations, such as depicted in Figure 5, are generally described in one of two ways: Using parametric functions, such as splines, where the coefficients β of the polynomials are computed for specified intervals on the grid m, n.

$$\vec{\phi}(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} \beta_{ij}[m(x), n(y)] \begin{pmatrix} (x-m(x))^i (y-n(y))^j \\ (x-m(x))^i (y-n(y))^j \end{pmatrix} \tag{3}$$

Or by a non-parametric deformation field, which defines a vector $\vec{u}(x,y)$ for each pixel x, y in the moving image towards the location in the fixed image.

$$\vec{\phi}(x,y) = \begin{pmatrix} x \\ y \end{pmatrix} + \vec{u}(x,y) \tag{4}$$

A particular concern of the deformation field is its excessive degrees of freedom. In theory, a non-parametric deformation field allows for arbitrary shuffling of pixels, such that neighbours of the original image may end up polar opposite after the transformation. To retain the original structure, penalizing non-diffeomorphic transformations is common. This favours the field to be continuous by having a positive determinant of the Jacobian at each position. The Jacobian is a linear approximation (like the previously introduced linear transformations) and becomes negative if an axis is flipped (causing the mesh to be torn).

2.2 Image interpolation

Pixels in a transformed image may not have come from a position on the original pixel-grid. The solution to this problem requires a re-sampling step, known as interpolation. Interpolation defines a function in the continuous domain to compute values at intermediate coordinates using the pixels in the original image. The most popular functions are *nearest-neighbour*, *bi-linear* and higher-order *spline* functions.

2.3 Point-based registration

Image transformations define point correspondence for all pixels in the image. Inversely, only a few point-correspondences are sufficient to compute linear transformations. Point-based image registration methods aim to find these point-correspondences. Automated methods do this in multiple steps. First, points with favourable properties are detected. The region surrounding each point is used to describe them, allowing them to then be matched to points in a second image. Finally, outliers are removed and a transformation is computed.

An overview of point-based registration, image-based registration (2.4) and spatial transformer networks (3.2.1) can be seen in Figure 6.

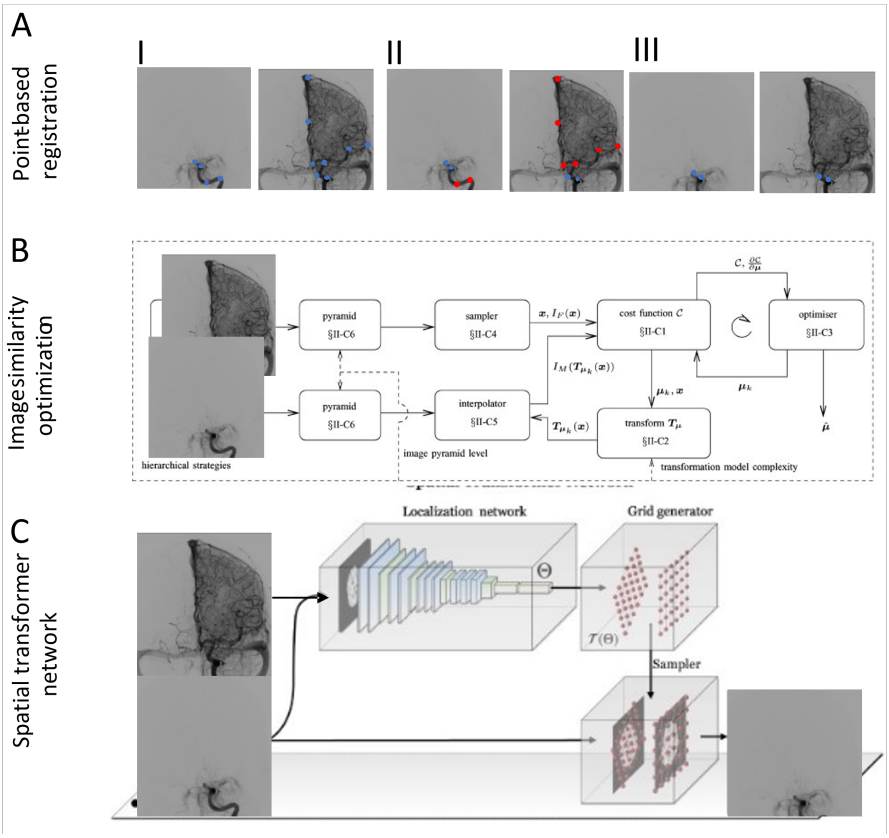

Fig. 6 Image registration methods. A: Point-based image registration. I: point-selection. II: point-matching & outlier detection. III: Image transformation. B: Image-based registration [12]. C: Spatial transformer network [13].

2.3.1 Manual point annotations

Such point-correspondences can be annotated manually. In our work, this approach was used to define a reference standard, and also to determine which transformation model would be sufficient for our application.

Manual annotations are also a way to guarantee favourable conditions that automated methods do not. It is for example beneficial for point-correspondences to be evenly distributed over the objects chosen for alignment. This reduces the sensitivity to small errors when computing the transformation. Additionally, objects can be excluded, or regions of interest can be emphasized with additional annotations.

When performing the manual annotations, it is good practise to use more points than minimally required. This averages out the small errors and produces a compromise for non-linearities. The latter property is particularly important for DSA image registration.

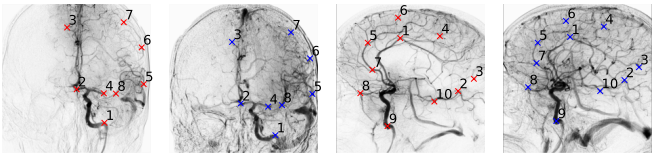

Fig. 7 Manual annotations used in the scientific article.

2.3.2 Scale-invariant feature transform

The scale-invariant feature transform [14] (SIFT) is an automatic point detection method. SIFT extracts local optima of the second order derivative at different image scales. These local optima are, in an ideal sense, the centres of circular structures. These optima therefore do not only extract the points positions, but additionally assigns each point a size. Its size defines the region surrounding the point, which will be used to give it a description. This description is defined by the orientations of the gradients in the surrounding region of the point. These are combined in local histograms and concatenated to form the descriptor.

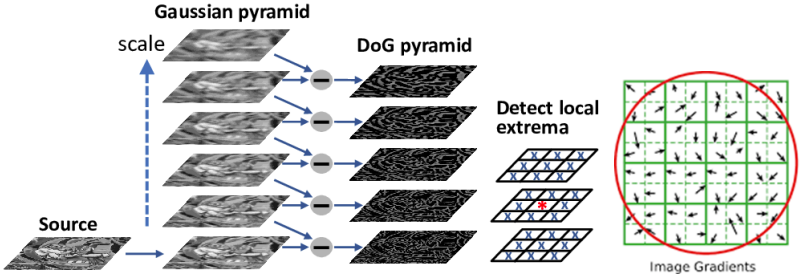

Fig. 8 SIFT candidate points are local optima in space and scale space. A descriptor is composed from the orientation of nearby gradients, images obtained from [15], [14].

2.3.3 Oriented FAST and Rotated BRIEF

Oriented FAST and Rotated BRIEF [16] (ORB) extracts corners using the FAST corner detector. Although corners may often lie on the periphery of an object, they have one big advantage: they are self-similar in scale. They are thereby scale-invariant by nature. The FAST corner detector [17] looks for N consecutive pixels in a circle that all have a higher or lower intensity than the center pixel by a margin t. Then, BRIEF [18] is used to formulate a binary descriptor. After smoothing the local patch, intensity values of a set of pixel pairs are compared ($I_1 \geq I_2$), producing a vector of zeros and ones.

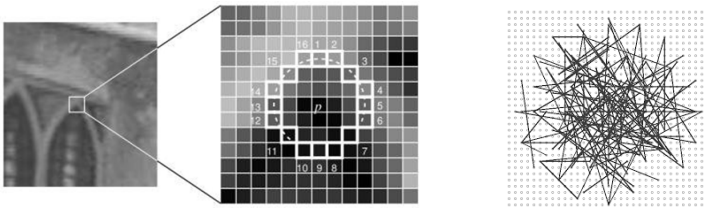

Fig. 9 FAST corner detection and BRIEF descriptor used in ORB. Images obtained from [17], [18].

2.3.4 Point-matching

The points and features computed using SIFT or ORB will have to be matched. This comes down to compare each descriptor from one image to all descriptors from the fixed image, and finding its closest match. For SIFT features the Euclidean distance is computed, and compared to the Euclidean distance to the second closest match. This is referred to as Lowe's ratio test, where the ratio of these distances needs to be higher than the hyper parameter for it to be considered a match. ORB features are binary and the Hamming distance is therefore commonly used instead of Euclidean distance. An additional verification step can be included such that the features are mutually closest.

2.3.5 Homography

Ideally, we have now found the point-correspondences and are ready to compute a transformation. Unfortunately, automated methods also produce invalid correspondences, which proves detrimental to the solutions. An additional outlier detection is therefore employed, using the fact that the majority of points will comply to the transformation. This is known as homography. In our work we use the random sample consensus (RANSAC) [19]. This algorithm starts by selecting a random set of point-correspondences, proficient to compute the transformation. It then evaluates how many of the remaining point-correspondences would be accurately aligned by the transformation. This is iterated and the solution that aligned most points is used to identify the outliers. The remaining points are used to compute the definitive solution.

2.4 Image-based registration

Image-based registration changes the transformation parameters θ in small increments, such that the similarity $\mathcal{S}$ between the transformed moving image $I \circ T(\vec{\theta})$ and the fixed image I_{fixed} improves. This is done using the chain rule and partial differentiation:

$$\frac{d\mathcal{S}(I \circ T(\vec{\theta}), I_{fixed})}{d\theta_i} = \frac{\partial \mathcal{S}(I \circ T(\vec{\theta}), I_{fixed})}{\partial \vec{x}} \cdot \left(\frac{\partial \vec{x}}{\partial \theta_i} \right)_{\theta_{j \neq i}} \tag{5}$$

The first derivative describes how each pixel should move to improve similarity, while the second derivative describes how each transformation parameter should be changed to accomplish this. The iterative process of evaluating the derivative and updating the transformation is automatically done using an optimizer, such as gradient descent.

The initial transformation, similarity function and step size, also known as learning rate, are the hyper parameters that influence the result and computation time.

Similarity metrics
Mean squared difference (MSD) directly compares the intensity values of the images. This metric thereby assumes the intensity values to represent equal measurements.

$$MSD = \sum_{m,n} \|I \circ T[m,n] - I_{fixed}[m,n]\|_2^2 \tag{6}$$

Normalized cross correlation (NCC) assumes intensity values of the two images to be linearly dependent and corrects for difference in variance σ. NCC maximizes if the intensity values are distributed equally in both images.

$$NCC = \frac{1}{\sigma_I \sigma_{fixed}} \sum_{m,n} I \circ T[m,n] \times I_{fixed}[m,n] \tag{7}$$

Mattes mutual information [20] (MMI) is a multi-modal image similarity metric. It assumes that for each intensity value interval of the image, any intensity interval in the fixed image captures the same information. This mutual information is to be optimized.

$$MMI = \sum_{m_1,n_1} \sum_{m_2,n_2} p(I \circ T[m_1,n_1], I_{fixed}[m_2,n_2] \log \left[\frac{p(I \circ T[m_1,n_1], I_{fixed}[m_2,n_2]}{p_T(I \circ T[m_1,n_1] p_{fixed}(I_{ref}[m_2,n_2]} \right] \tag{8}$$

With p, being the joint discrete probability, p_T and p_{ref} the marginal discrete probability function for the intensity intervals in the moving image and fixed image.

3 Deep learning

Image analysis has greatly benefit from advancements in deep learning over the last two decades. Handcrafting algorithms is a time consuming and complex process, while the desired output is often trivial for us humans. Deep learning models are composed of layered functions with parameters that are automatically tuned to approximate the output using examples.

In this chapter, the concepts of *layers* and *model training* will be introduced. This includes the niche *diffeomorphic integration layer* used for the computation of diffeomorphic deformation maps, and different *loss functions* used for training. These concepts are combined to produce deep-learning models for landmark detection and image registration.

3.1 Convolutions and convolutional deep learning models

The convolution operator $*$ is used to combine information of a smaller window within the image I, and is one of the main tools used in traditional image analysis. A 2D discrete convolution with a filter f is formally defined as

$$(I * f)[m, n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I[m - i, n - j] f[i, j] \tag{9}$$

although in practice, the lower and upper bounds are defined by the size of the filter. In Figure 10 a typical example of a classical image analysis algorithm is shown. The filter is convolved with the image we want to analyse. The filtered image contains high values at the positions where the coins are positioned, and noise from other objects. After discarding the lower values of the output below a threshold, only the blobs at the positions of the coins remain. In order to retain the exact positions, we extract the local maxima.

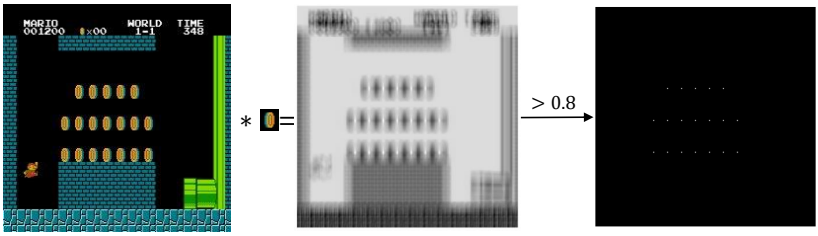

Fig. 10 An image is convolved with a filter, followed by a threshold and local maximum filter.

If the output contains a nonzero pixel, we know that the image contains a coin (*image classification*). Extracting the coordinates of all nonzero values provides the locations of all coins in the image (*object detection*). We can also reconstruct the coins in the image by convolving the output again with the filter (Fig. 11).

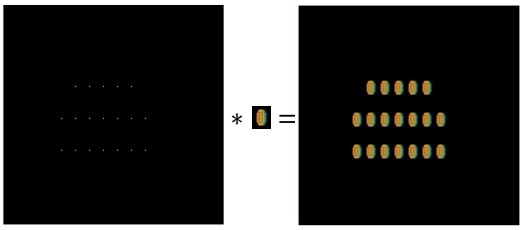

Fig. 11 The output of Fig. 10 is convolved with the filter to reconstruct the coins of the original image.

Convolutions in deep learning were popularized by Krizhevsky et al. in the AlexNet [21] image classification model, which uses a very similar structure to the example algorithm. Typically, a repetitive sequence of a convolution layer (typically including multiple filters) followed by a non-linear function and a local max-pooling layer are used. Five of such convolution-pooling blocks are present in the AlexNet model, figure 12. The filter and threshold values can all be optimized automatically during model training, aiming to reproduce manually annotated data. This is done in a similar manner to the partial differentiation in 2.4, where partial derivatives are chained w.r.t. each layer (going from the output, propagating backwards). This is known as back-propagation and allows each parameter value to be updated to minimize the loss function.

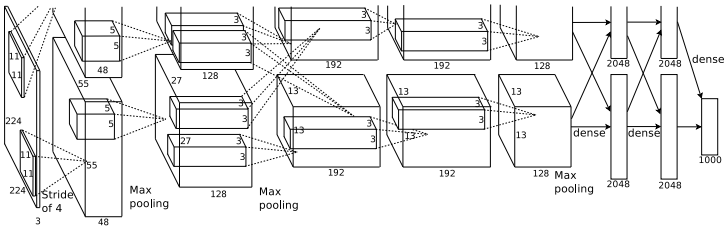

Fig. 12 The AlexNet image classification model [21], composed of five convolution blocks and three fully connected layers.

In image classification, a sequence of these repetitive convolution-activation function-pooling blocks is terminated by flattening the final output into a vector, which can then be analysed with one (or multiple) fully connected layer(s). A fully connected layer is a weighted sum of the input, followed by an activation function.

If we want to compute a property for each pixel in the image using a deep-learning method, such as in image segmentation, *transposed convolution layers* are used. In a similar manner to how the coins could be reconstructed in the example (Fig. 11), these transposed convolutions use trainable filters to provide information to compute an up-sampled output. This can thereby restore the output size, which was reduced by the pooling layers, to that of the original image.

Closing remarks on convolution:
The convolution operator is translation equivariant[1]. This means that its behaviour is spatially consistent, independent of the position of an object in an image. This is the primary reason it is more commonly used in computer vision than fully connected networks. In some special cases, a filter can additionally be decomposed in a rotational variant and invariant component [22] using non-linear functions. Gaussian image derivatives are the most well-known filters that have this property, separating the derivaitve amplitude and angle. The image derivative ∇I can be computed using Gaussian derivative filters $\partial_x G, \partial_y G$. Its amplitude $\|\nabla I\|_2^2$ and angle $\angle \nabla I$ then become

$$\|\nabla I\|_2^2 = (I * \partial_x G)^2 + (I * \partial_y G)^2 \qquad \angle \nabla I = \angle \begin{pmatrix} I * \partial_x G \\ I * \partial_y G \end{pmatrix}$$

Convolutional neural networks (CNNs) do not explicitly have translational equivariance because of its pooling layers. Additionally, CNNs do not use rotational decomposition explicitly, although the 'convolution-activation function' layering shares a lot of resemblance. To retain these favourable properties, it is common practice to use training data that contains rotation and translation variation, such that the trained model will still produce consistent results. This is commonly done by including random transformations of the training data, a process known as data augmentation.

3.2 Model architectures

3.2.1 Spatial transformer network

Spatial transformer networks [23] were introduced by Jaderberg et al., whose primary objective was to improve image classification. They introduced an additional network that predicts a transformation and applies it to the image, such that the subsequent classification network improves. It relies on the property that image resampling is differentiable, as is used during image-based registration. This property is critical to perform model training.

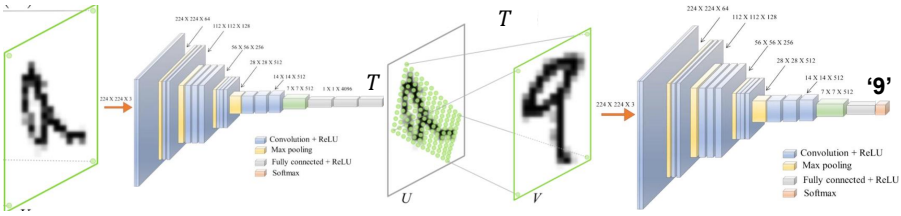

Fig. 13 Spatial transformer network as part of an image classification network. Original images were obtained from [23], [24].

[1] a special case of the commutative property with a shifted Dirac delta function

Jaderberg et al. state that spatial transformer networks could also be used for "co-localisation" of objects of the same class, a qualitative form of image registration. They illustrate this behaviour in the classification of the CUB-200-2011 bird dataset, which consistently extracted the wings and head of the birds before classification. It should be noted that the authors do not specify the network to extract these regions in particular, nor do they guide the network to provide geometrically consistent results. To that end, STNs have been modified and used for image registration. These algorithms either predict the transformation of the input image to a consistent fixed image, or additionally input a (varying) fixed image and optimize for image similarity.

3.2.2 U-net

The U-net algorithm [25] by Ronneberger et al. is one of the most popular convolutional neural network architectures, and is primarily used for image segmentation. It consists of convolution-activation function-pooling layered blocks, followed by transposed convolution-activation function layered blocks. Each up-sampled intermediate is concatenated with the equivalently sized down-sampled block, thereby combining local and global information.

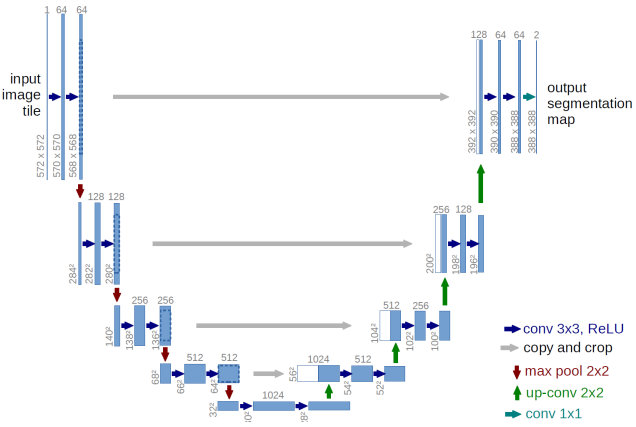

Fig. 14 Original U-net architecture using 2 convolutional layers per down pooling layer, with four total pooling layers. Image obtained from [25].

3.2.3 VoxelMorph

VoxelMorph [26] extents the principle of STNs to non-linear registration using a deformation map. It uses a U-net to predict the deformation of the moving image w.r.t. a fixed image or atlas. To (approximately) enforce diffeomorphic constraints, they include a scaling and squaring integration layer which further improves their results. This architecture produces near real time results and improved accuracy compared to traditional methods, while requiring relatively small training data.

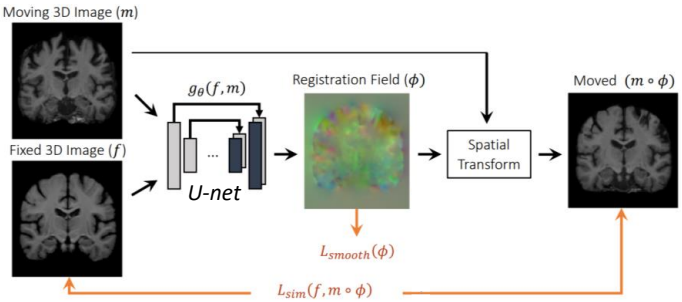

Fig. 15 VoxelMorph default architecture for non-linear image registration. Image was obtained from [26]

Diffeomorphic integration layer

In many deep learning applications, the final activation function is chosen to enforce a constraint. A famous example is the softmax function, which enforces the output probabilities to be positive and sum to one.

As previously introduced, diffeomorphic transformations are desirable in image registration. Unfortunately, enforcing diffeomorphic properties on a deformation field is not a trivial task. Evaluating the determinant of the Jacobian for Equation 4, yields Equation 10. This provides the constraints we wish to enforce.

$$det\left(\begin{bmatrix} 1+\partial_x u_x(x,y) & \partial_y u_x(x,y) \\ \partial_x u_y(x,y) & 1+\partial_y u_y(x,y) \end{bmatrix}\right) > 0 \tag{10}$$

At first glance, one could approach this like a linear programming problem. This would be a valid approach, but the number of potential solutions becomes excessive for practically relevant image sizes. An alternative approach is therefore employed, which integrates deformation over small time steps Δt. This parameter should be chosen to account for the largest deformation gradient.

$$\Delta t^{-1} > \|\nabla u\|_{max}$$

Because this reduces the variable terms to be much smaller than one, it guarantees diffeomorphic properties for a single time step. Iterating a diffeomorphic transformation (as is done during integration) retains the desired properties. Additionally, this method provides a framework to accurately estimate the inverse transformation (simply by replacing $\vec{u}$ by $-\vec{u}$ before integration).

Dalca et al. [26] introduced this approach to its deep-learning model VoxelMorph. The default integration layer uses $\Delta t = 2^{-8} = 256^{-1}$, which is sufficient for most applications. To save on computation time, a recursive approximation is used, known as the scaling and squaring approach [27]. Instead of evaluating the integral at each time step, it re-samples the solution from the previous iteration.

$$u(8\Delta t) = u(\Delta t) + u(\Delta t) + u(2\Delta t) + u(4\Delta t)$$

While the discrete integral approach and the scaling and squaring approximation are very effective, they do not produce perfectly diffeomorphic fields. This is caused by the discrete time step and re-sampling. Nevertheless, these methods are very practical for training and validation of deep learning models. For other applications it may be of interest to examine exact integration.

3.3 Loss functions

Loss functions are used to relate the output of a model to the ground-truth, defining a cost for the difference between them. This cost is then minimized during training using back-propagation. The type of loss function therefore depends on the type of output, and should represent the relative importance that the designer attributes to different errors. In the scientific article we predict the probability distribution of the position of anatomical landmarks using deep learning. The used loss functions are elaborated in 3.3.1. Performing image registration using deep-learning can be done using different loss functions, optimizing image similarity 3.3.2, desirable transformation properties 3.3.3 and adhering to ground-truth information 3.3.4.

3.3.1 Landmark detection

Kullback–Leibler (KL) divergence [28] D_{KL} captures the difference in information between the observed P and ground truth Q probability distribution, particularly punishing large deviations:

$$D_{KL}(P \mid\mid Q) = \sum_{m,n} P(m,n) \times \log\left(\frac{P(m,n)}{Q(m,n)}\right) \tag{11}$$

KL divergence is not symmetric ($D_{KL}(P \mid\mid Q) \neq D_{KL}(Q \mid\mid P)$). Choosing whether P is the observation or ground-truth is therefore relevant, and in practice both are used. They are referred to forward and backward KL divergence respectively. Alternatively the *Jensen–Shannon* (JS) divergence [29] can be used, which is symmetric.

$$JSD = \frac{1}{2} D_{KL}(P \mid\mid Q) + \frac{1}{2} D_{KL}(Q \mid\mid P) \tag{12}$$

The KL-divergence can be implicitly calculated if the observation is expressed as an ideal normal distribution through its mean (μ_Q) and (co-) variance ($\sigma_x, \sigma_y, \rho_{xy}$) [30].

$$D_{KL}(P \mid\mid Q) = \tfrac{1}{2}\left[\log e^{-2}\sigma^{-4}\sigma_x^2\sigma_y^2(1-\rho_{xy}^2 + (\vec{\mu}_P - \vec{\mu}_Q)^T\Sigma^{-1_Q}(\vec{\mu}_P - \vec{\mu}_Q) + \tfrac{\sigma^2(\sigma_x^2+\sigma_y^2)}{\sigma_x^2\sigma_y^2(1-\rho_{xy}^2}\right] \tag{13}$$

While these loss functions are capable of optimizing the output distributions, they do not provide information that is easily interpreted. To complement these methods, such that model building and results become intuitive, an additional

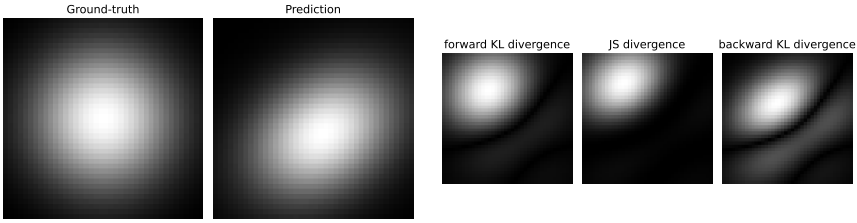

Fig. 16 KL divergence and JS divergence absolute values (before summation over m, n).

metric was included. This is the Euclidean distance between the annotated ground truth coordinate and the coordinate inferred from these probability distributions.

3.3.2 Image similarity

The loss functions for image similarity are equivalent to those in image similarity based registration, see 2.4.

3.3.3 Deformation field regularization

The simplest regularization term for deformation $\vec{u}$ is the gradient amplitude $\|\nabla\vec{u}\|_2^2$, which describes the distance that a pixel will move away from its neighbours.

$$\|\nabla\vec{u}\|_2^2 = \|\partial_x u_x\|_2^2 + \|\partial_y u_y\|_2^2 \tag{14}$$

The Bending energy $\|\nabla^2\vec{u}\|_2^2$, the Frobenius norm of the Hessian matrix, penalizes large gradient changes.

$$\|\nabla^2\vec{u}\|_2^2 = \|\partial_x^2\vec{u}\|_2^2 + \|\partial_y^2\vec{u}\|_2^2 + 2\|\partial_x\partial_y\vec{u}\|_2^2 \tag{15}$$

Additionally, the average displacement $\bar{\vec{u}}$ can be penalized when global transformations have been executed with high accuracy.

$$\|\bar{\vec{u}}\|_2$$

Or, if the linear registration was less accurate, the deformation field can be corrected with a linear transformation T_{cor} before regularization.

$$\vec{u}(\vec{x})_{cor} = \vec{u}(\vec{x}) - T_{cor}(\vec{u}(\vec{x}))\vec{x}$$

3.3.4 Supervised transformation loss

When a similarity metric does not guide our model to a desired solution, we can use manually produced Transformations instead. For this, I use the Euclidean distance between the warped coordinate and the (ground-truth) transformed coordinate.

$$\mathcal{L} = \|\hat{T}\vec{x} - T\vec{x}\|_2 \tag{16}$$

4 Evaluation of deep-learning methods for image registration

Image registration for DSA has proven difficult for gradient descent methods. This may be caused by different factors, such as ineffective similarity metrics, differences in image contents or the sparsity of the image contents. We perform two experiments to evaluate the capabilities of deep-learning strategies and their applicability to DSA.

4.1 Methods

Spatial transformer network
An affine STN implementation is adopted [31]. Input size is increased to $256 \times 256px.$ and three additional convolution-pooling layers are included. Model output and grid sampling are modified to allow for similarity and projection transformations.

VoxelMorph (modified)
We make use of the Tensor Flow implementation of VoxelMorph. The U-net receptive field is extended using three additional encoding and decoding layers, followed by the integration layer (seven steps). Additionally, a linear transformation is extracted. The source image is then re-sampled using the linear and non-linear deformation fields.

The linear transformation is extracted as follows. One could see the deformation field as point-correspondences for each point in the image, and the integration layer as a form of homography. Computing the linear transformation from these point correspondences can be done efficiently using a least-squares solution (B).

$$\vec{x} + \vec{u} \approx T\vec{x} \tag{17}$$

Simultaneously computing the linear and non-linear transformation may benefit both. Linear registration is difficult for non-linear geometries, and the increased degrees of freedom may benefit the search through solution space, while non-linear registration methods do not always incorporate long-distance information, which could benefit from simultaneously upholding a (global) linear transformation.

4.2 Data

MNIST
The Modified National Institute of Standards and Technology (MNIST) dataset[32] is a commonly used example dataset, containing seventy-thousand handwritten digits (zero to nine) stored as $28 \times 28px.$ images. Various different (STN) registration algorithms proved effective when optimizing MSD or NCC. One digit is selected (the eights) and upsampeled to 256×256 to more accurately represent our data.

DSA series pre-/post- EVT
The MR CLEAN registry contains 512 patients with AP lateral pre-/post EVT DSA series. To retain consistent validation metrics for supervised and unsupervised experiments, only series for which 'SIFT+LM' (described in the scientific article) found a solution are included, yielding 353 AP and 404 lateral series. These series will be used with the solutions found using 'SIFT+ORB' as ground truth for validation and supervised training.

4.3 Experiments and results

The STN and modified VoxelMorph are evaluated for tasks with increasing complexity.

MNIST image registration
In the first experiment, networks have been trained to compute the three linear transformations, trained on random image pairs (default), to optimize NCC. To increase complexity, random similarity transformations[2] are applied to the source image (augmented source) or both images (augmented) before computation. Table 3 displays the NCC values of both networks. To provide context, Figure 17 shows validation examples of alignment with respective NCC values.

Table 3 Validation NCC for upsampled MNIST eights ($256 \times 256px$) aligned using STN || VoxelMorph.

NCC	**default**	**augmented source**	**augmented**
(none)	0.50	0.39	0.30
similarity	0.63 || 0.68	0.59 || 0.53	0.54 || 0.35
affine	0.70 || 0.74	0.65 || 0.66	0.59 || 0.57
projection	0.76 || 0.76	0.70 || 0.65	0.62 || 0.43

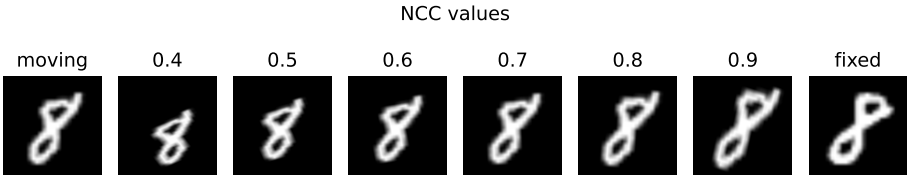

Fig. 17 NCC values for alignment examples of two different eights.

One of the remarkable properties of VoxelMorph is its ability to generalize image registration to different objects. We apply the trained models to a different digit to see whether this phenomenon extents to STNs and the modified VoxelMorph implementation in Figure 18.

[2] rotation $\|\theta\| \leq 45°$, scaling $1.5^{-1} \leq s \leq 1.5$, translation $\|\Delta x\| \leq 0.2 \times 256$

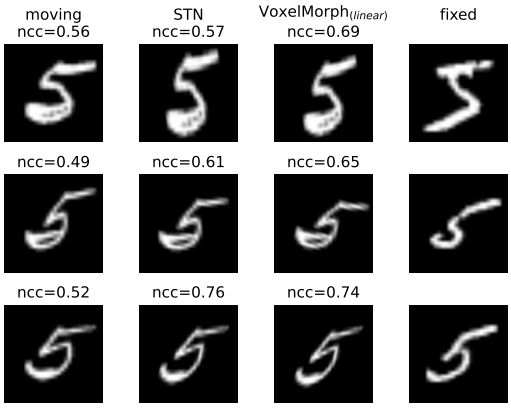

Fig. 18 Alignment of fives using the STN and custom Voxel Morph trained on eights.

pre-/post-EVT DSA image registration

The second experiment will evaluate the networks on DSA data. As similarity losses proved less effective (elastix, scientific article), supervised training is performed using solutions from point-based image registration and Equation 16. Training and validation curves are shown in Figure 19

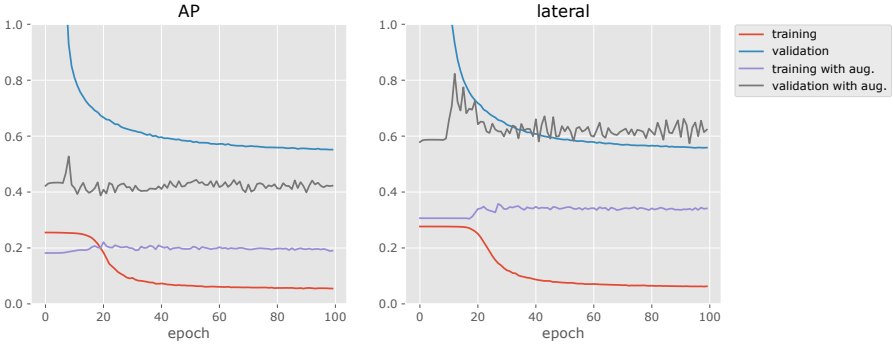

Fig. 19 Aug.: augmentation. Training and validation loss, Equation 16, of supervised STN image registration using solutions from SIFT+ORB.

4.4 Discussion

For default MNIST data, STNs are shown to improve alignment of random image pairings. Adopting VoxelMorph for linear alignment proved effective and comparable, although performance decreased for more difficult transformations. Surprisingly, both methods indicate capabilities of generalized image registration, Figure 18.

For DSA image registration, supervised image registration was performed to circumvent training using image-similarity based learning. While the model proved capable of learning transformation variability for augmented training

data, no generalization is observed for validation data. While we can only speculate, possible causes include, but are not limited to: insufficient training data, large data variability, sparsity of mutual structures and the general sparsity of information in DSA images.

4.5 Conclusion

Spatial Transformer Networks are capable of improving alignment for elementary data. Nevertheless, one-shot transformation computation does not seem feasible for variable targets. Incorporating a recursive component would be a recommended improvement to address this limitation.

Combining linear and non-linear image registration in VoxelMorph proved functional. Its longer computation time and comparable performance does not justify its use for linear image registration, or should be limited to fine-tuning in more extensive frameworks that include non-linear alignment.

Automated DSA image registration using deep-learning remains unsolved.

Appendices

A Supplementary data

A.1 Data pre-processing

DSA images can contain embedded text (including shadow effect), bounding boxes and border artefacts. Examples are shown in Figure 1. Figures 2, 3 show the masks for the artefacts produced by the two most common capturing devices in the MR CLEAN registry, together with the resulting in-painting using Open CV. Figure 4 shows the border identification result used by both methods.

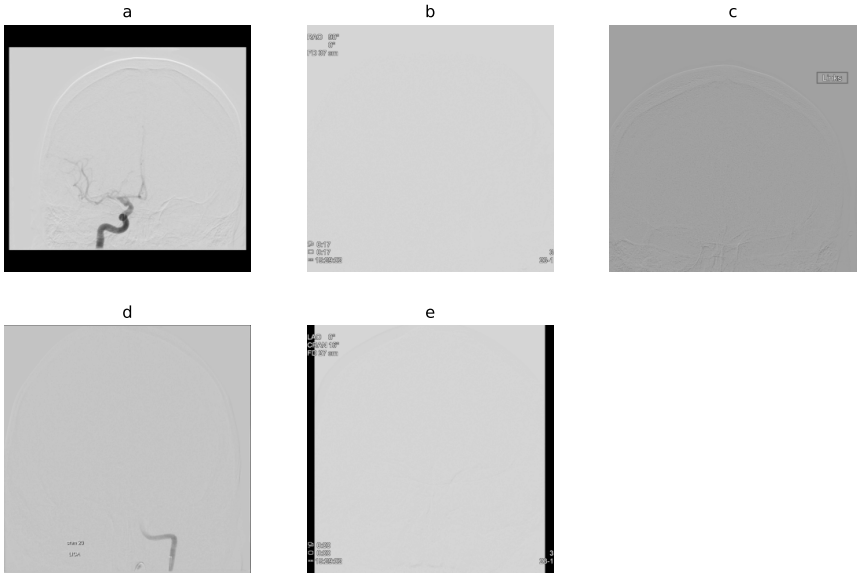

Fig. 1 Common artefacts in DSA sequences in the Mr Clean registry. a: border artefacts. b: embedded overlays. c: embedded text with bounding box (Allura Xper device) d: embedded text (Axiom artis device) e: Combinations of border artefacts and embedded overlays.

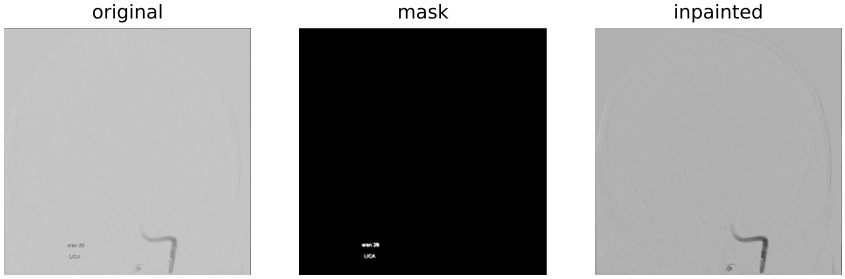

Fig. 2 Axiom Artis pre-processing. Text is automatically identified, masked and inpainted using Open CV. Borders are removed if identified.

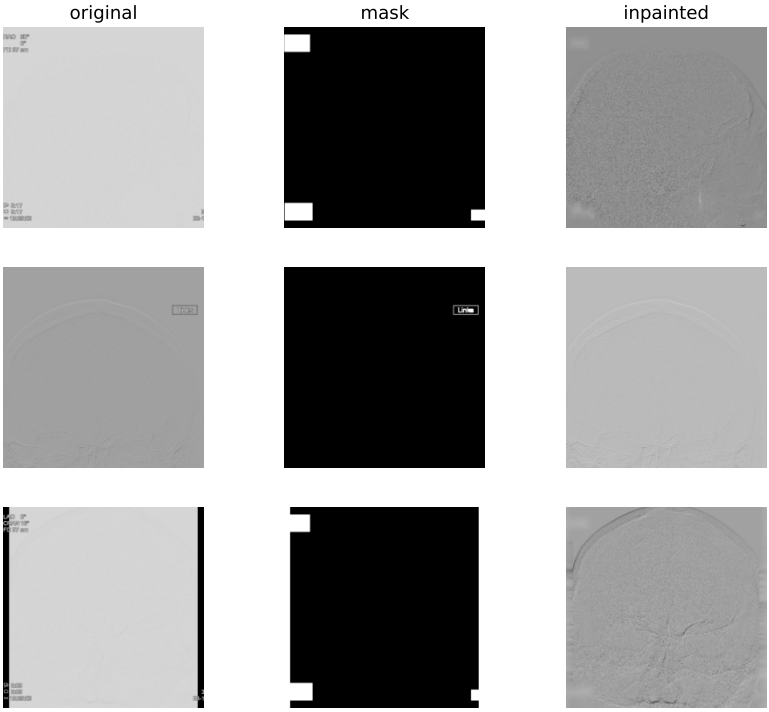

Fig. 3 Allura Xper pre-processing. The standard overlay and text with bounding box are automatically identified, masked and inpainted using Open CV.

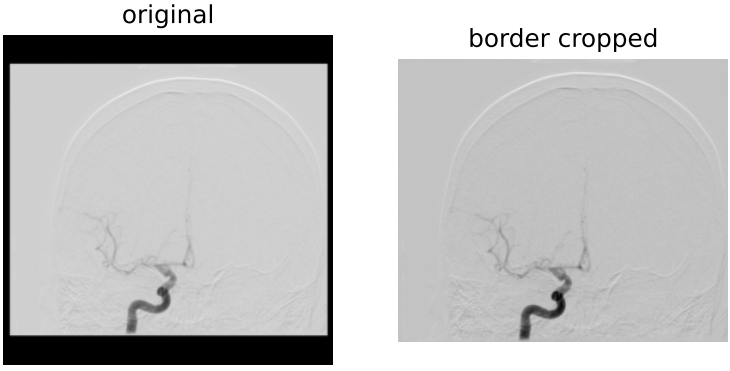

Fig. 4 Border removal. Border artefact in original image is automatically identified and cropped (or masked in the proceeding Figures).

A.2 Optimized transformations (lateral)

The experiment in which the transformations were optimized to manual annotations, (4.3) was repeated for lateral MinIPs. The residual error per transformation type is shown in Figure 5.

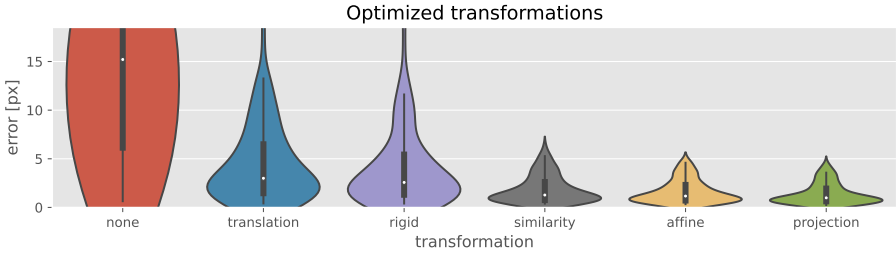

Fig. 5 Average alignment error for transformations computed using manual annotations from lateral MinIP pairs. The violin represents the data distribution and inside, the median (white), inter-quartile range (gray, thick) and inter-adjacent value range (gray, thin) are indicated.

A.3 Landmark model performance

The accuracy of the landmark model (2.2), the Euclidean distance between predicted and annotated coordinate, is evaluated for each loss function. Argmax and centre of mass are used to infer the coordinate from the predicted probability distributions. The prediction error over the three fold validation is summarised separately for the AP and lateral models in tables 2, 3 respectively.

Table 2 The prediction error over the three fold validation for AP minIps.

centre of mass average prediction error [*pixels*] (mean + s.d.)

Loss	mean	25^{th} ICA	mean ICA	75^{th} ICA	25^{th} M1	mean M1	75^{th} M1
KL_{fw}	6.6 ± 0.7	1.9 ± 0.3	7.0 ± 0.7	6.9 ± 2.6	1.7 ± 0.3	6.2 ± 0.8	5.8 ± 1.7
KL_{bw}	5.9 ± 0.4	1.8 ± 0.3	6.1 ± 0.6	5.7 ± 1.6	1.6 ± 0.3	5.7 ± 0.4	5.1 ± 1.0
KL_{fw} N	5.8 ± 1.0	1.9 ± 0.5	5.6 ± 0.8	5.7 ± 1.2	2.1 ± 0.8	6.0 ± 1.3	6.6 ± 2.0
KL_{bw} N	6.4 ± 0.6	2.1 ± 0.5	5.9 ± 0.4	6.2 ± 1.5	2.9 ± 0.7	6.8 ± 0.8	7.6 ± 1.7
KL_{fw} N_s	8.5 ± 2.2	2.5 ± 0.4	8.0 ± 2.0	7.8 ± 1.7	3.5 ± 1.1	8.9 ± 2.4	10.0 ± 2.6
KL_{bw} N_s	9.9 ± 3.2	3.5 ± 1.0	9.0 ± 2.1	10.5 ± 2.5	5.6 ± 3.0	10.8 ± 4.3	13.8 ± 4.6
JS	6.4 ± 0.7	1.9 ± 0.4	6.9 ± 0.6	6.5 ± 2.0	1.5 ± 0.4	6.0 ± 0.8	5.5 ± 1.8
$p_{comb.1}$	**4.3 ± 0.4**	**1.6 ± 0.1**	**4.6 ± 0.3**	**4.4 ± 0.4**	**1.4 ± 0.2**	**4.1 ± 0.6**	**3.7 ± 0.3**
$p_{comb.2}$	4.9 ± 0.7	1.9 ± 0.3	4.9 ± 0.4	5.0 ± 0.6	1.9 ± 0.5	5.0 ± 1.0	5.7 ± 1.0

argmax prediction error [*pixels*] (mean + s.d.)

Loss	mean	25^{th} ICA	mean ICA	75^{th} ICA	25^{th} M1	mean M1	75^{th} M1
KL_{fw}	5.1 ± 0.4	1.8 ± 0.3	5.3 ± 0.5	5.2 ± 1.4	1.5 ± 0.4	4.8 ± 0.5	4.4 ± 1.3
KL_{bw}	5.1 ± 0.4	1.8 ± 0.3	5.2 ± 0.5	4.9 ± 1.3	1.5 ± 0.4	4.9 ± 1.0	4.3 ± 0.9
KL_{fw} N	11.0 ± 1.2	7.2 ± 1.6	11.7 ± 1.6	13.0 ± 3.0	6.8 ± 0.7	10.4 ± 0.8	12.3 ± 1.5
KL_{bw} N	13.0 ± 1.8	9.1 ± 2.7	13.5 ± 1.5	16.2 ± 3.0	8.9 ± 2.5	12.6 ± 2.3	15.4 ± 4.1
KL_{fw} N_s	12.9 ± 7.2	5.7 ± 4.5	13.2 ± 7.3	13.0 ± 6.8	4.8 ± 2.4	12.5 ± 7.1	11.6 ± 4.1
KL_{bw} N_s	13.4 ± 1.5	5.0 ± 2.9	10.3 ± 2.7	11.6 ± 5.1	11.8 ± 4.4	16.4 ± 5.3	20.3 ± 4.8
JS	6.2 ± 0.2	1.8 ± 0.3	6.6 ± 0.8	5.5 ± 1.1	**1.3 ± 0.2**	5.9 ± 1.2	4.3 ± 1.2
$p_{comb.1}$	**4.6 ± 0.5**	**1.6 ± 0.3**	**5.0 ± 0.6**	**4.5 ± 0.4**	**1.3 ± 0.2**	**4.1 ± 0.5**	**3.8 ± 0.4**
$p_{comb.2}$	11.1 ± 1.4	6.9 ± 1.5	11.8 ± 1.9	12.6 ± 2.7	6.5 ± 0.5	10.4 ± 1.0	11.8 ± 1.0

Table 3 The prediction error over the three fold validation for lateral minIps.

centre of mass average prediction error [*pixels*] (mean + s.d.)

Loss	mean	25^{th} ICA	mean ICA	75^{th} ICA	25^{th} M1	mean M1	75^{th} M1
KL_{fw}	5.5 ± 0.9	1.6 ± 0.3	5.3 ± 0.9	4.3 ± 0.4	1.9 ± 0.5	5.7 ± 0.9	5.4 ± 1.0
KL_{bw}	5.3 ± 1.0	1.4 ± 0.2	5.3 ± 1.2	4.0 ± 0.4	1.9 ± 0.4	5.4 ± 0.8	5.0 ± 0.7
KL_{fw} N	4.8 ± 0.9	1.4 ± 0.2	4.7 ± 1.1	3.8 ± 0.3	1.8 ± 0.4	4.9 ± 0.7	4.9 ± 0.8
KL_{bw} N	5.7 ± 0.6	2.2 ± 0.4	5.6 ± 0.8	5.5 ± 0.7	2.3 ± 0.6	5.7 ± 0.6	6.5 ± 1.2
KL_{fw} N_s	10.7 ± 3.9	3.3 ± 0.2	10.8 ± 3.4	10.5 ± 2.2	3.5 ± 0.3	10.6 ± 4.4	11.3 ± 3.3
KL_{bw} N_s	8.6 ± 2.1	4.6 ± 3.0	9.0 ± 2.2	10.2 ± 4.0	3.9 ± 2.0	8.3 ± 2.1	10.1 ± 4.7
JS	5.2 ± 0.6	1.4 ± 0.3	4.7 ± 0.8	3.7 ± 0.6	1.8 ± 0.4	5.7 ± 0.5	5.2 ± 1.0
$p_{comb.1}$	**3.9 ± 0.4**	**1.3 ± 0.1**	**3.7 ± 0.5**	**3.3 ± 0.1**	**1.6 ± 0.3**	**4.1 ± 0.4**	**4.4 ± 0.2**
$p_{comb.2}$	4.2 ± 0.3	1.4 ± 0.2	4.1 ± 0.4	3.8 ± 0.2	1.7 ± 0.2	4.3 ± 0.3	4.7 ± 0.1

argmax prediction error [*pixels*] (mean + s.d.)

Loss	mean	25^{th} ICA	mean ICA	75^{th} ICA	25^{th} M1	mean M1	75^{th} M1
KL_{fw}	4.3 ± 0.7	1.3 ± 0.2	3.8 ± 0.9	3.4 ± 0.3	**1.7 ± 0.5**	4.8 ± 0.8	4.9 ± 0.7
KL_{bw}	4.6 ± 0.6	1.3 ± 0.2	4.3 ± 0.6	3.5 ± 0.5	1.8 ± 0.6	5.0 ± 0.6	4.7 ± 0.8
KL_{fw} N	10.2 ± 0.6	7.8 ± 0.3	10.6 ± 1.3	11.5 ± 0.1	7.0 ± 0.5	9.8 ± 0.0	11.9 ± 0.8
KL_{bw} N	11.3 ± 0.2	9.6 ± 1.9	12.9 ± 1.2	15.5 ± 1.6	6.7 ± 1.8	9.7 ± 1.2	12.1 ± 2.4
KL_{fw} N_s	26.6 ± 9.5	10.5 ± 2.2	25.2 ± 6.5	24.6 ± 5.4	9.7 ± 2.2	28.0 ± 12.5	30.1 ± 15.2
KL_{bw} N_s	10.5 ± 2.8	6.3 ± 3.6	11.8 ± 2.8	14.1 ± 5.9	3.9 ± 2.4	9.2 ± 2.9	11.1 ± 5.1
JS	4.9 ± 1.1	1.3 ± 0.2	4.3 ± 1.1	3.5 ± 0.5	**1.7 ± 0.5**	5.5 ± 1.2	4.9 ± 1.0
$p_{comb.1}$	**3.9 ± 0.4**	**1.1 ± 0.2**	**3.6 ± 0.5**	**3.2 ± 0.0**	1.8 ± 0.3	**4.2 ± 0.5**	**4.3 ± 0.1**
$p_{comb.2}$	10.0 ± 0.3	7.8 ± 0.3	10.4 ± 0.7	11.4 ± 0.6	6.8 ± 0.5	9.7 ± 0.8	11.6 ± 0.8

A.4 Landmark model t-test comparisons

Student t-tests are used to compare the models using the argmax inference method to establish if models are performing (significantly) better. The null-hypothesis, H_0, is defined such that the average armgax prediction error is smaller for $model_1$ (row) than for $model_2$ (column). Probabilities are computed over the three fold validation. In bold if more than 95% confidence.

Table 4 Students t-test comparing the AP model performance trained with different loss functions.

$model_1$ \ $model_2$	KL_{fw}	KL_{bw}	KL_{fw} N	KL_{bw} N	KL_{fw} N_s	KL_{bw} N_s	JS	$p_{comb.1}$	$p_{comb.2}$
KL_{fw}		0.15	0.21	0.36	0.84	0.88	0.41	0.01	0.04
KL_{bw}	0.85		0.45	0.79	0.91	0.92	0.81	0.01	0.08
KL_{fw} N	0.79	0.55		0.73	0.9	0.92	0.75	0.07	0.19
KL_{bw} N	0.64	0.21	0.27		0.87	0.9	0.55	0.01	0.05
KL_{fw} N_s	0.16	0.09	0.1	0.13		0.69	0.14	0.03	0.05
KL_{bw} N_s	0.12	0.08	0.08	0.1	0.31		0.11	0.04	0.05
JS	0.59	0.19	0.25	0.45	0.86	0.89		0.01	0.04
$p_{comb.1}$	**0.99**	**0.99**	0.93	**0.99**	**0.97**	**0.96**	**0.99**		0.82
$p_{comb.2}$	**0.96**	0.92	0.81	**0.95**	**0.95**	**0.95**	**0.96**	0.18	

Table 5 Students t-test comparing the lateral model performance trained with different loss functions.

$model_1$ \ $model_2$	KL_{fw}	KL_{bw}	KL_{fw} N	KL_{bw} N	KL_{fw} N_s	KL_{bw} N_s	JS	$p_{comb.1}$	$p_{comb.2}$
KL_{fw}		0.43	0.23	0.58	0.93	0.94	0.36	0.04	0.06
KL_{bw}	0.57		0.3	0.65	0.94	0.94	0.45	0.07	0.1
KL_{fw} N	0.77	0.7		0.84	0.95	**0.96**	0.69	0.13	0.21
KL_{bw} N	0.42	0.35	0.16		0.93	0.93	0.25	0.01	0.02
KL_{fw} N_s	0.07	0.06	0.05	0.07		0.27	0.06	0.03	0.04
KL_{bw} N_s	0.06	0.06	0.04	0.07	0.73		0.05	0.02	0.02
JS	0.64	0.55	0.31	0.75	0.94	**0.95**		0.03	0.06
$p_{comb.1}$	**0.96**	0.93	0.87	**0.99**	**0.97**	**0.98**	**0.97**		0.78
$p_{comb.2}$	0.94	0.9	0.79	**0.98**	**0.96**	**0.98**	0.94	0.22	

A.5 Landmark model training curves and metrics

The training and validation loss curves, together with the more intuitive Euclidean distance error, are shown for AP (Figures 6, 7) and lateral (Figures 8, 9) models. The implicit loss functions display unstable behaviour. An attempt was made to improve stability by incorporating a smoothing kernel $\sigma = 0.5$ before computing the loss. This had a negative effect and was not resolved.

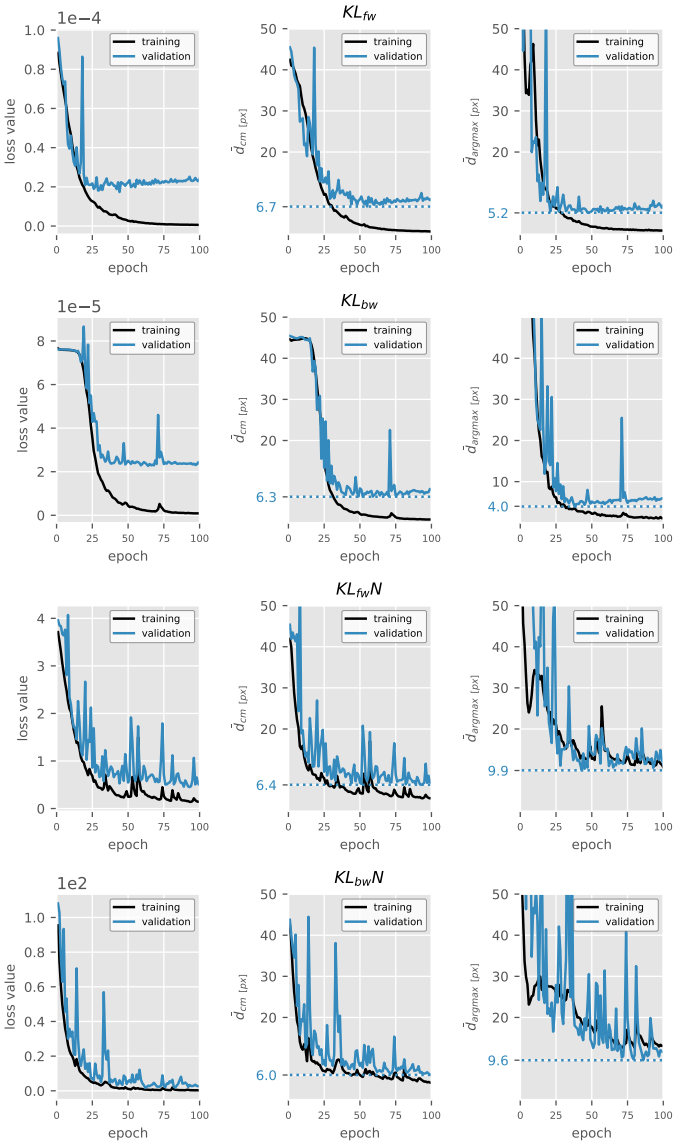

Fig. 6 Loss curves (first column), centre-of-mass error (second column) and argmax error (third column) for loss functions indicated in the centre column and trained on AP images

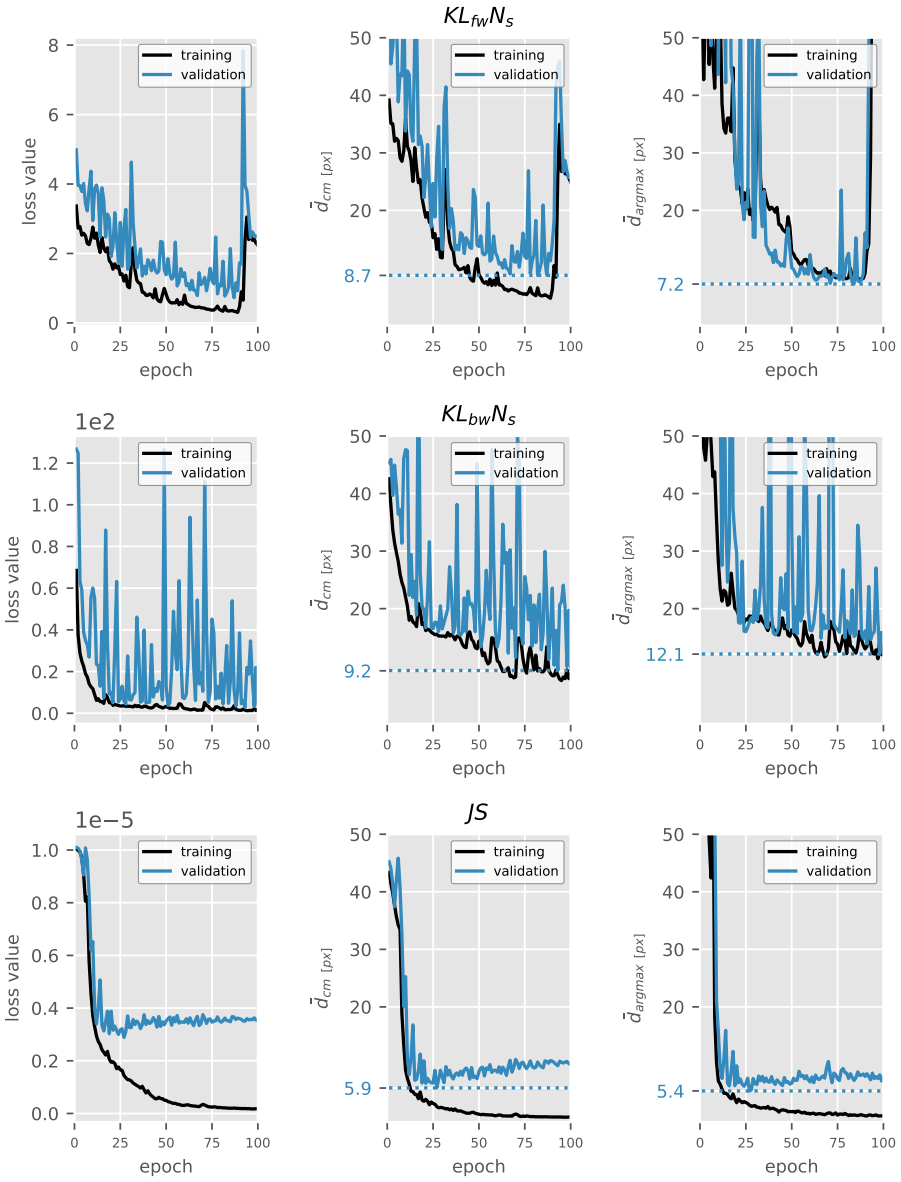

Fig. 7 Loss curves (first column), centre-of-mass error (second column) and argmax error (third column) for loss functions indicated in the centre column and trained on AP images

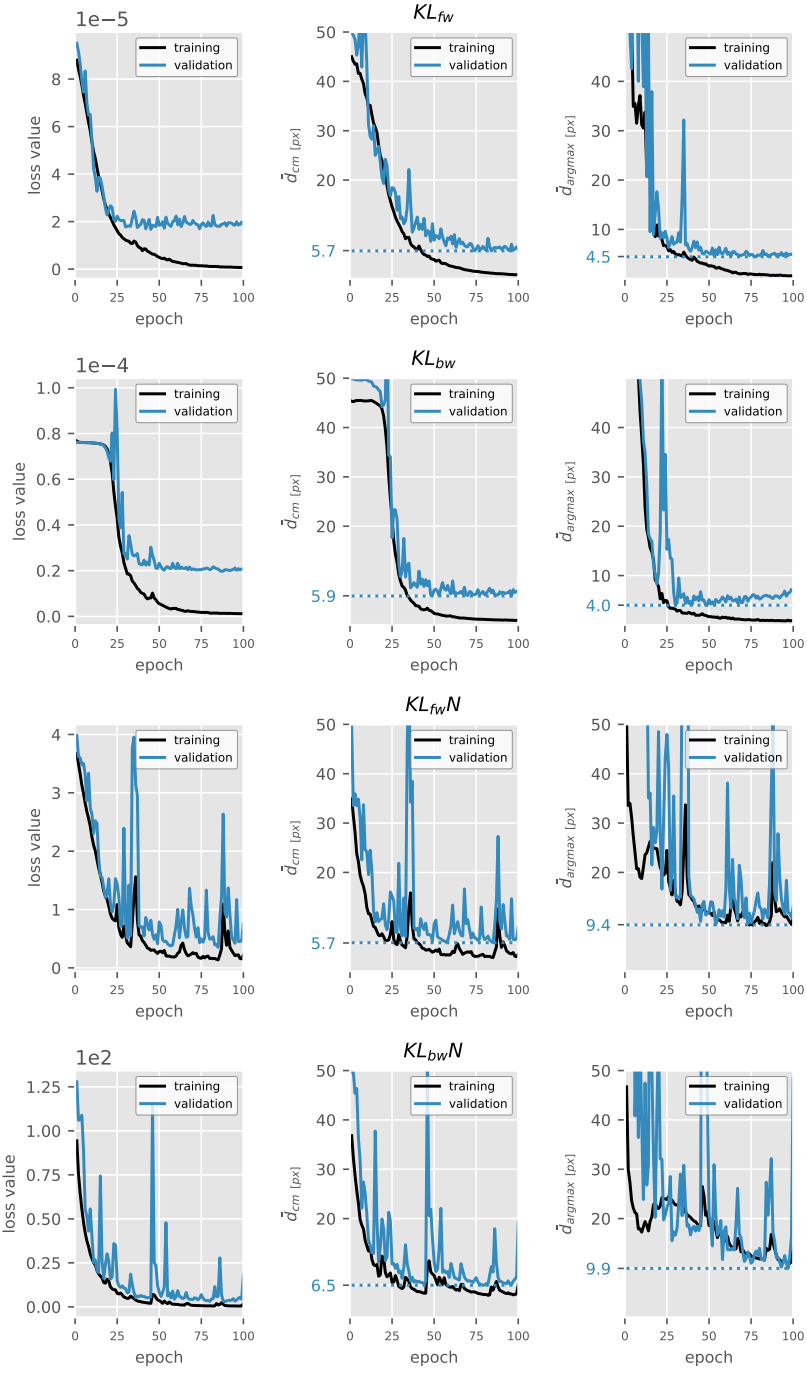

Fig. 8 Loss curves (first column), centre-of-mass error (second column) and argmax error (third column) for loss functions indicated in the centre column and trained on lateral images

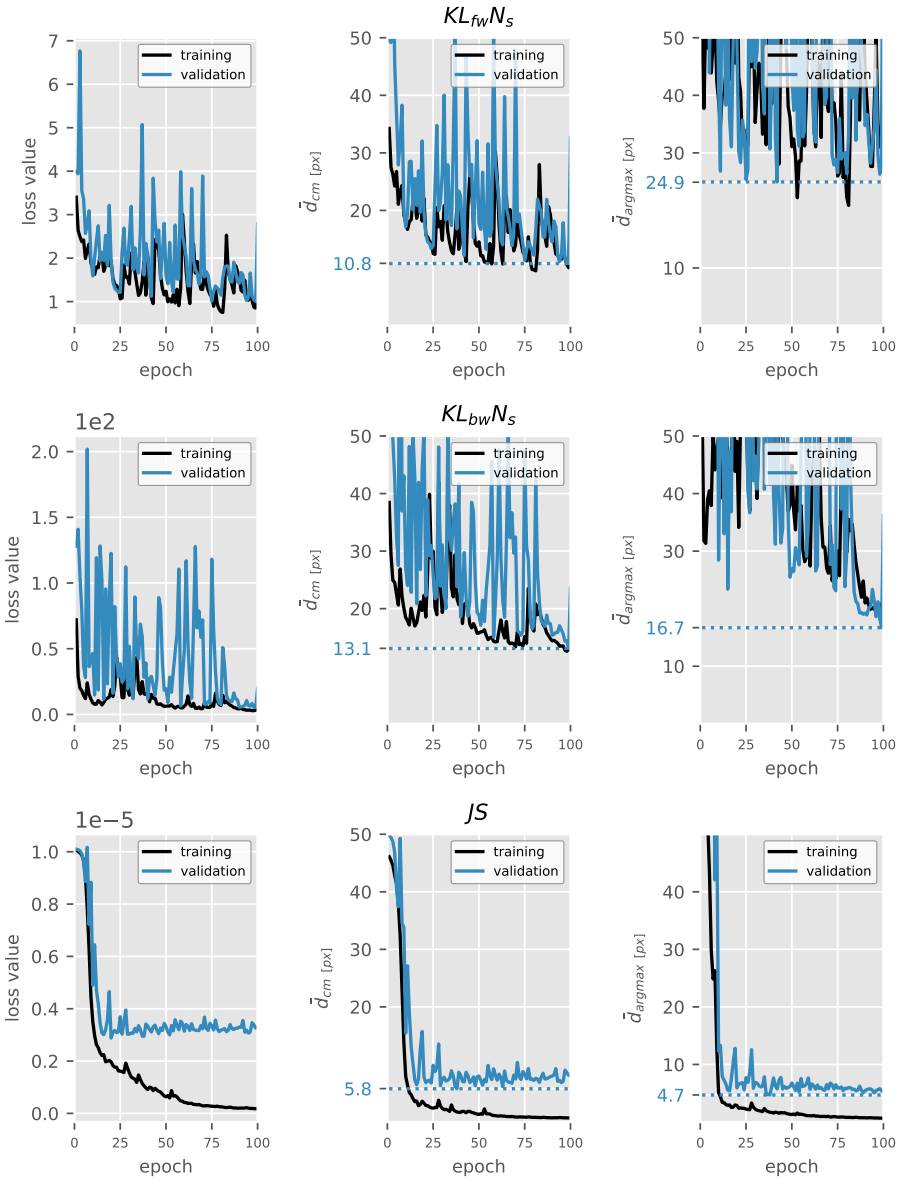

Fig. 9 Loss curves (first column), centre-of-mass error (second column) and argmax error (third column) for loss functions indicated in the centre column and trained on lateral images

A.6 Landmark-based registration

Image registration can be performed using the two cerebral landmarks. Results for registration using AP and lateral cerebral landmarks are shown in Figures 10,11. Translation has previously shown to be insufficient for optimal alignment (4.3). Alternatively, computing a similarity transformation using two close-by points is very sensitive. This is reflected in the results, where the similarity transformation performs worse and translation results are in-line with the optimized translation results.

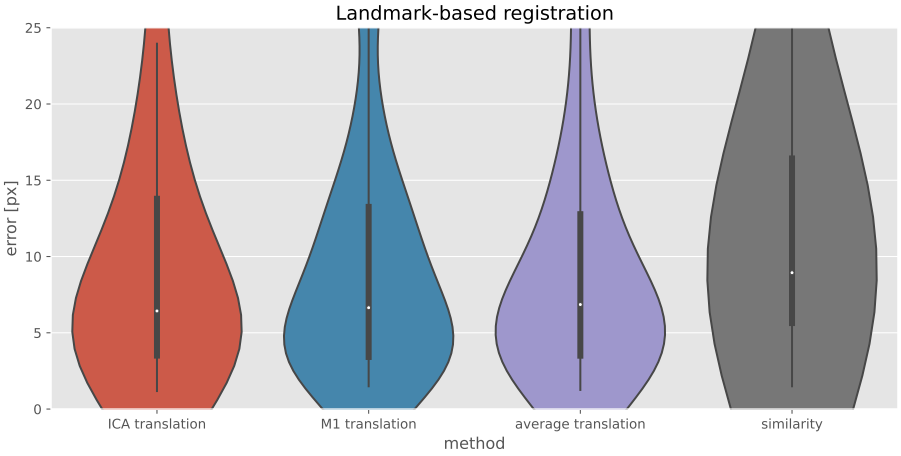

Fig. 10 Registration error using the only the landmark point correspondences for AP DSA MinIPs.

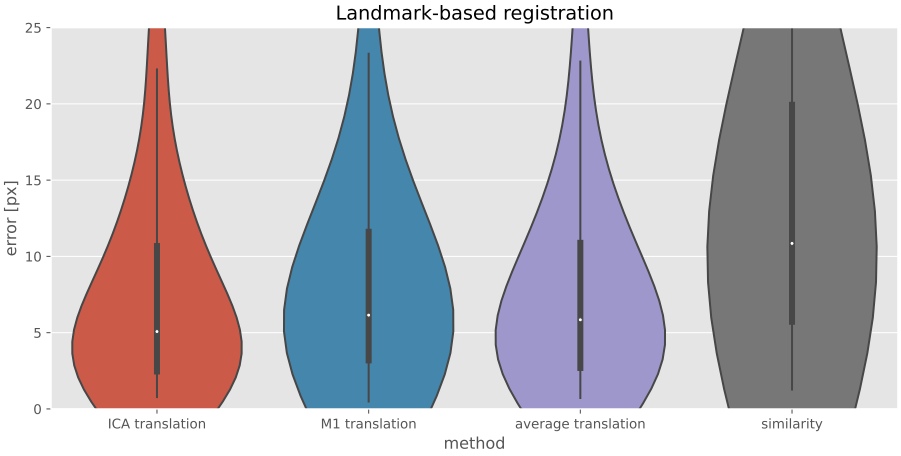

Fig. 11 Registration error using the only the landmark point correspondences for lateral DSA MinIPs.

A.7 Point-based registration z-tests

Table 1. Z-test comparing different methods, norms and transformations for AP images. The probability that the two methods are equivalent. If method (row) has a worse mean, the probability is set to one. The best performing method is highlighted in bold.

Method			SIFT						ORB						SIFT ORB						SIFT LM						ORB LM						SIFT ORB LM					
	Transformation		Similarity		Affine		Projection		Similarity		Affine		Projection		Similarity		Affine		Projection		Similarity		Affine		Projection		Similarity		Affine		Projection		Similarity		Affine		Projection	
		norm	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2	l1
	Similarity	l2	1,00	1,00	0,17	0,16	0,00	0,00	0,30	0,33	0,25	0,26	0,04	0,04	1,00	1,00	1,00	1,00	0,26	0,26	1,00	1,00	0,33	0,34	0,03	0,04	0,30	0,33	0,27	0,29	0,08	0,07	1,00	1,00	1,00	1,00	0,26	0,27
		l1	1,00	1,00	0,17	0,16	0,00	0,00	0,30	0,33	0,25	0,26	0,04	0,04	1,00	1,00	1,00	1,00	0,26	0,26	1,00	1,00	0,33	0,34	0,03	0,04	0,30	0,33	0,27	0,29	0,08	0,07	1,00	1,00	1,00	1,00	0,27	0,27
SIFT	Affine	l2	1,00	1,00	1,00	0,96	0,00	0,00	0,50	0,53	0,42	0,44	0,08	0,08	1,00	1,00	1,00	1,00	0,55	0,56	1,00	1,00	1,00	1,00	0,05	0,05	0,50	0,53	0,46	0,48	0,12	0,11	1,00	1,00	1,00	1,00	0,52	0,55
		l1	1,00	1,00	1,00	1,00	0,00	0,00	0,51	0,54	0,43	0,45	0,08	0,08	1,00	1,00	1,00	1,00	0,56	0,58	1,00	1,00	1,00	1,00	0,05	0,05	0,51	0,54	0,47	0,49	0,12	0,12	1,00	1,00	1,00	1,00	0,54	0,57
	Projection	l2	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	1,00	1,00	1,00	1,00	0,02	0,02	1,00	1,00	0,96	0,95	0,27	0,28	1,00	1,00	1,00	1,00	0,23	0,24	1,00	1,00	1,00	1,00	0,10	0,11	1,00	1,00	1,00	1,00	0,14	0,27	1,00	1,00	1,00	1,00	0,30	0,31
		l1	1,00	1,00	1,00	1,00	0,02	0,02	0,98	1,00	0,94	0,93	0,27	0,28	1,00	1,00	1,00	1,00	0,22	0,24	1,00	1,00	1,00	1,00	0,10	0,10	1,00	1,00	1,00	1,00	0,14	0,27	1,00	1,00	1,00	1,00	0,30	0,31
ORB	Affine	l2	1,00	1,00	1,00	1,00	0,02	0,02	1,00	1,00	1,00	0,98	0,28	0,28	1,00	1,00	1,00	1,00	0,24	0,25	1,00	1,00	1,00	1,00	0,11	0,12	1,00	1,00	1,00	1,00	0,14	0,27	1,00	1,00	1,00	1,00	0,31	0,31
		l1	1,00	1,00	1,00	1,00	0,02	0,02	1,00	1,00	1,00	1,00	0,28	0,28	1,00	1,00	1,00	1,00	0,25	0,26	1,00	1,00	1,00	1,00	0,11	0,12	1,00	1,00	1,00	1,00	0,15	0,27	1,00	1,00	1,00	1,00	0,31	0,31
	Projection	l2	1,00	1,00	1,00	1,00	0,10	0,10	1,00	1,00	1,00	1,00	1,00	0,90	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,78	1,00	1,00	1,00	1,00	1,00	0,84
		l1	1,00	1,00	1,00	1,00	0,10	0,10	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,87	1,00	1,00	1,00	1,00	1,00	0,93
	Similarity	l2	0,72	0,72	0,09	0,09	0,00	0,00	0,41	0,43	0,39	0,38	0,25	0,26	1,00	1,00	1,00	1,00	0,09	0,09	0,49	0,58	0,11	0,12	0,01	0,02	0,56	0,57	0,52	0,50	0,09	0,25	0,89	0,91	0,91	0,91	0,21	0,29
		l1	0,63	0,62	0,07	0,07	0,00	0,00	0,41	0,43	0,39	0,38	0,25	0,26	1,00	1,00	1,00	1,00	0,09	0,09	0,42	0,50	0,09	0,09	0,01	0,02	0,56	0,57	0,52	0,50	0,09	0,25	0,88	0,91	0,90	0,90	0,21	0,29
SIFT ORB	**Affine**	**l2**	**0,56**	**0,56**	**0,05**	**0,05**	**0,00**	**0,00**	**0,34**	**0,35**	**0,32**	**0,31**	**0,24**	**0,25**	**0,89**	**0,89**	**1,00**	**0,98**	**0,07**	**0,08**	**0,34**	**0,42**	**0,06**	**0,06**	**0,01**	**0,02**	**0,48**	**0,49**	**0,44**	**0,42**	**0,08**	**0,25**	**0,78**	**0,81**	**0,81**	**0,80**	**0,20**	**0,29**
		l1	0,53	0,53	0,05	0,05	0,00	0,00	0,36	0,37	0,34	0,33	0,24	0,25	0,91	0,91	1,00	1,00	0,08	0,08	0,33	0,41	0,06	0,06	0,01	0,02	0,49	0,50	0,46	0,44	0,09	0,25	0,80	0,83	0,82	0,82	0,20	0,29
	Projection	l2	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,35	0,35	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,67	0,65	1,00	1,00	1,00	1,00	0,38	0,32	1,00	1,00	1,00	1,00	0,65	0,37
		l1	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,35	0,34	1,00	1,00	1,00	1,00	0,97	1,00	1,00	1,00	1,00	1,00	0,65	0,63	1,00	1,00	1,00	1,00	0,37	0,32	1,00	1,00	1,00	1,00	0,64	0,37
	Similarity	l2	0,91	0,90	0,13	0,13	0,00	0,00	0,19	0,20	0,15	0,16	0,28	0,28	1,00	1,00	1,00	1,00	0,23	0,24	1,00	1,00	0,37	0,38	0,02	0,02	0,19	0,20	0,17	0,18	0,02	0,02	1,00	1,00	1,00	1,00	0,16	0,17
		l1	0,77	0,77	0,10	0,10	0,00	0,00	0,18	0,19	0,15	0,15	0,28	0,28	1,00	1,00	1,00	1,00	0,23	0,23	0,90	1,00	0,31	0,32	0,02	0,02	0,18	0,19	0,16	0,17	0,02	0,02	1,00	1,00	1,00	1,00	0,15	0,16
SIFT LM	Affine	l2	1,00	1,00	0,66	0,62	0,00	0,00	0,27	0,29	0,23	0,23	0,28	0,29	1,00	1,00	1,00	1,00	0,26	0,27	1,00	1,00	1,00	1,00	0,02	0,03	0,27	0,29	0,25	0,26	0,03	0,03	1,00	1,00	1,00	1,00	0,26	0,27
		l1	1,00	1,00	0,64	0,61	0,00	0,00	0,27	0,29	0,23	0,23	0,28	0,29	1,00	1,00	1,00	1,00	0,26	0,26	1,00	1,00	0,99	1,00	0,02	0,03	0,27	0,29	0,25	0,26	0,03	0,03	1,00	1,00	1,00	1,00	0,25	0,27
	Projection	l2	1,00	1,00	1,00	1,00	0,83	0,83	1,00	1,00	1,00	1,00	0,34	0,34	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,87	0,87	1,00	1,00	1,00	1,00	0,34	0,34	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	1,00	1,00	1,00	1,00	0,02	0,02	0,83	0,85	0,80	0,78	0,27	0,27	1,00	1,00	1,00	1,00	0,18	0,20	1,00	1,00	1,00	1,00	0,10	0,11	1,00	1,00	0,94	0,91	0,11	0,26	1,00	1,00	1,00	1,00	0,27	0,31
		l1	1,00	1,00	1,00	1,00	0,02	0,02	0,82	0,84	0,78	0,77	0,27	0,27	1,00	1,00	1,00	1,00	0,18	0,19	1,00	1,00	1,00	1,00	0,10	0,10	0,99	1,00	0,93	0,90	0,11	0,26	1,00	1,00	1,00	1,00	0,27	0,31
ORB LM	Affine	l2	1,00	1,00	1,00	1,00	0,02	0,02	0,88	0,90	0,85	0,83	0,27	0,28	1,00	1,00	1,00	1,00	0,20	0,21	1,00	1,00	1,00	1,00	0,10	0,11	1,00	1,00	1,00	0,97	0,12	0,26	1,00	1,00	1,00	1,00	0,28	0,31
		l1	1,00	1,00	1,00	1,00	0,02	0,02	0,92	0,93	0,88	0,86	0,27	0,28	1,00	1,00	1,00	1,00	0,21	0,22	1,00	1,00	1,00	1,00	0,10	0,11	1,00	1,00	1,00	1,00	0,12	0,26	1,00	1,00	1,00	1,00	0,29	0,31
	Projection	l2	1,00	1,00	1,00	1,00	0,26	0,26	1,00	1,00	1,00	1,00	0,53	0,49	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,61	0,60	1,00	1,00	1,00	1,00	1,00	0,43	1,00	1,00	1,00	1,00	1,00	0,49
		l1	1,00	1,00	1,00	1,00	0,23	0,23	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,59	0,57	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	0,74	0,74	0,09	0,09	0,00	0,00	0,51	0,53	0,49	0,47	0,25	0,26	1,00	1,00	1,00	1,00	0,11	0,12	0,46	0,54	0,10	0,10	0,01	0,02	0,67	0,68	0,63	0,60	0,10	0,25	1,00	1,00	1,00	1,00	0,22	0,30
		l1	0,58	0,58	0,06	0,06	0,00	0,00	0,49	0,51	0,47	0,46	0,25	0,26	1,00	1,00	1,00	1,00	0,10	0,11	0,40	0,47	0,08	0,08	0,01	0,02	0,65	0,66	0,61	0,58	0,10	0,25	0,98	1,00	1,00	0,99	0,22	0,29
SIFT ORB LM	Affine	l2	0,65	0,64	0,06	0,06	0,00	0,00	0,50	0,52	0,48	0,46	0,25	0,26	1,00	1,00	1,00	1,00	0,11	0,11	0,37	0,44	0,07	0,07	0,01	0,02	0,65	0,67	0,61	0,59	0,10	0,25	0,98	1,00	1,00	1,00	0,22	0,29
		l1	0,47	0,47	0,04	0,04	0,00	0,00	0,51	0,52	0,48	0,47	0,25	0,26	1,00	1,00	1,00	1,00	0,11	0,11	0,27	0,33	0,04	0,05	0,01	0,02	0,66	0,67	0,62	0,59	0,10	0,25	0,98	1,00	1,00	1,00	0,22	0,29
	Projection	l2	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,44	0,42	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,08	0,09	1,00	1,00	1,00	1,00	0,74	0,38	1,00	1,00	1,00	1,00	1,00	0,43
		l1	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,07	0,08	1,00	1,00	1,00	1,00	1,00	0,95	1,00	1,00	1,00	1,00	1,00	1,00

Table 2. Z-test comparing different methods, norms and transformations on Lateral images. The probability that the two methods are equivalent. If method (row) has a worse mean, the probability is set to one. The best performing method is highlighted in bold.

Method			SIFT					ORB					SIFT ORB					SIFT LM					ORB LM					SIFT ORB LM				
	Transformation		Similarity		Affine		Projection	Similarity		Affine		Projection	Similarity		Affine		Projection	Similarity		Affine		Projection	Similarity		Affine		Projection	Similarity		Affine		Projection
		norm	l2	l1	l2	l1		l2	l1	l2	l1		l2	l1	l2	l1		l2	l1	l2	l1		l2	l1	l2	l1		l2	l1	l2	l1	
SIFT	Similarity	l2	1,00	1,00	0,47	0,48	0,02	0,02	0,02	0,36	0,65	0,08	0,18	0,00	0,00	1,00	1,00	1,00	1,00	0,11	0,13	1,00	1,00	0,62	0,61	0,01	0,01	0,43	0,78	0,10	0,32	0,00	0,00	1,00	1,00	1,00	1,00	0,07	0,09
		l1	0,79	1,00	0,33	0,34	0,02	0,02	0,23	0,48	0,04	0,10	0,00	0,00	1,00	1,00	1,00	1,00	0,07	0,09	0,86	1,00	0,44	0,44	0,00	0,00	0,28	0,59	0,06	0,20	0,00	0,00	1,00	1,00	0,85	1,00	0,04	0,06
	Affine	l2	1,00	1,00	1,00	1,00	0,02	0,02	0,89	1,00	0,35	0,59	0,00	0,01	1,00	1,00	1,00	1,00	0,32	0,37	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	0,42	0,84	0,01	0,01	1,00	1,00	1,00	1,00	0,21	0,26
		l1	1,00	1,00	0,99	1,00	0,02	0,02	0,88	1,00	0,35	0,58	0,00	0,01	1,00	1,00	1,00	1,00	0,32	0,36	1,00	1,00	1,00	1,00	0,01	0,01	0,98	1,00	0,41	0,82	0,01	0,01	1,00	1,00	1,00	1,00	0,21	0,26
	Projection	l2	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
ORB	Similarity	l2	1,00	1,00	1,00	1,00	0,02	0,02	1,00	1,00	0,86	0,88	0,18	0,05	1,00	1,00	1,00	1,00	0,32	0,35	1,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00	1,00	1,00	0,29	0,19	1,00	1,00	1,00	1,00	0,76	0,63
		l1	1,00	1,00	0,77	0,78	0,02	0,02	0,98	1,00	0,84	0,86	0,18	0,05	1,00	1,00	1,00	1,00	0,32	0,35	1,00	1,00	0,89	0,88	0,00	0,00	1,00	1,00	1,00	1,00	0,29	0,19	1,00	1,00	1,00	1,00	0,74	0,61
	Affine	l2	1,00	1,00	1,00	1,00	0,03	0,03	1,00	1,00	1,00	1,00	0,18	0,06	1,00	1,00	1,00	1,00	0,32	0,38	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,30	0,21	1,00	1,00	1,00	1,00	0,88	0,72
		l1	1,00	1,00	1,00	1,00	0,02	0,03	1,00	1,00	0,98	1,00	0,18	0,05	1,00	1,00	1,00	1,00	0,32	0,37	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,30	0,21	1,00	1,00	1,00	1,00	0,86	0,70
	Projection	l2	1,00	1,00	1,00	1,00	0,09	0,09	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,75	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,08	0,08	1,00	1,00	1,00	1,00	0,44	1,00	1,00	1,00	1,00	1,00	0,47	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,64	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	0,54	0,75	0,18	0,20	0,02	0,02	0,27	0,29	0,20	0,21	0,16	0,03	1,00	1,00	0,92	0,91	0,30	0,22	0,89	0,98	0,25	0,25	0,00	0,00	0,53	0,57	0,31	0,35	0,26	0,10	0,96	1,00	0,92	0,93	0,20	0,21
		l1	**0,32**	**0,48**	**0,09**	**0,10**	**0,02**	**0,02**	**0,26**	**0,27**	**0,19**	**0,20**	**0,16**	**0,03**	**0,97**	**1,00**	**0,89**	**0,88**	**0,30**	**0,22**	**0,65**	**0,73**	**0,15**	**0,15**	**0,00**	**0,00**	**0,51**	**0,54**	**0,30**	**0,33**	**0,26**	**0,10**	**0,93**	**0,97**	**0,89**	**0,90**	**0,19**	**0,20**
SIFT ORB	Affine	l2	0,65	0,86	0,24	0,25	0,02	0,02	0,31	0,33	0,23	0,24	0,16	0,03	1,00	1,00	1,00	0,99	0,31	0,23	1,00	1,00	0,52	0,51	0,00	0,00	0,60	0,63	0,36	0,40	0,27	0,11	1,00	1,00	0,99	1,00	0,22	0,22
		l1	0,52	0,72	0,18	0,19	0,02	0,02	0,32	0,33	0,23	0,24	0,16	0,03	1,00	1,00	1,00	1,00	0,31	0,23	1,00	1,00	0,51	0,50	0,00	0,00	0,60	0,64	0,36	0,40	0,27	0,11	1,00	1,00	1,00	1,00	0,23	0,22
	Projection	l2	1,00	1,00	1,00	1,00	0,03	0,03	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,08	0,08	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,03	0,03	1,00	1,00	1,00	1,00	0,26	0,32	1,00	1,00	1,00	1,00	0,37	1,00	1,00	1,00	1,00	1,00	0,08	0,08	1,00	1,00	1,00	1,00	0,41	0,85	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	0,92	1,00	0,40	0,41	0,02	0,02	0,21	0,39	0,04	0,07	0,05	0,06	1,00	1,00	0,75	0,79	0,02	0,03	1,00	1,00	0,31	0,31	0,00	0,00	0,27	0,53	0,04	0,11	0,09	0,16	1,00	1,00	0,85	1,00	0,03	0,03
		l1	0,78	1,00	0,31	0,33	0,02	0,02	0,18	0,33	0,03	0,06	0,05	0,06	1,00	1,00	0,68	0,71	0,02	0,02	0,91	1,00	0,26	0,26	0,00	0,00	0,22	0,46	0,03	0,09	0,09	0,16	0,97	1,00	0,77	0,97	0,02	0,02
SIFT LM	Affine	l2	1,00	1,00	0,78	0,80	0,02	0,02	0,85	1,00	0,26	0,40	0,07	0,08	1,00	1,00	1,00	1,00	0,06	0,07	1,00	1,00	1,00	0,99	0,00	0,00	0,97	1,00	0,26	0,52	0,11	0,18	1,00	1,00	1,00	1,00	0,10	0,10
		l1	1,00	1,00	0,81	0,83	0,02	0,02	0,85	1,00	0,26	0,41	0,07	0,08	1,00	1,00	1,00	1,00	0,06	0,07	1,00	1,00	1,00	1,00	0,00	0,00	0,97	1,00	0,27	0,53	0,11	0,18	1,00	1,00	1,00	1,00	0,10	0,11
	Projection	l2	1,00	1,00	1,00	1,00	0,09	0,09	1,00	1,00	1,00	1,00	0,85	0,84	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,81	0,60	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,09	0,09	1,00	1,00	1,00	1,00	0,85	0,84	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	1,00	1,00	1,00	1,00	0,81	0,60	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	1,00	1,00	1,00	1,00	0,02	0,02	0,56	0,58	0,44	0,46	0,17	0,04	1,00	1,00	1,00	1,00	0,31	0,27	1,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00	0,68	0,73	0,28	0,14	1,00	1,00	1,00	1,00	0,42	0,37
ORB		l1	1,00	1,00	0,64	0,66	0,02	0,02	0,53	0,55	0,41	0,43	0,17	0,04	1,00	1,00	1,00	1,00	0,31	0,27	1,00	1,00	0,71	0,70	0,00	0,00	0,96	1,00	0,65	0,70	0,28	0,14	1,00	1,00	1,00	1,00	0,40	0,35
	Affine	l2	1,00	1,00	1,00	1,00	0,03	0,03	0,86	0,88	0,71	0,74	0,17	0,05	1,00	1,00	1,00	1,00	0,32	0,32	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	1,00	1,00	0,29	0,17	1,00	1,00	1,00	1,00	0,63	0,52
LM		l1	1,00	1,00	1,00	1,00	0,02	0,02	0,81	0,83	0,67	0,69	0,17	0,04	1,00	1,00	1,00	1,00	0,32	0,31	1,00	1,00	1,00	1,00	0,01	0,01	1,00	1,00	0,95	1,00	0,29	0,17	1,00	1,00	1,00	1,00	0,59	0,50
	Projection	l2	1,00	1,00	1,00	1,00	0,07	0,08	1,00	1,00	1,00	1,00	0,82	1,00	1,00	1,00	1,00	1,00	0,65	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
		l1	1,00	1,00	1,00	1,00	0,07	0,07	1,00	1,00	1,00	1,00	0,28	0,39	1,00	1,00	1,00	1,00	0,38	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,44	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Similarity	l2	0,76	0,98	0,29	0,31	0,02	0,02	0,28	0,30	0,21	0,22	0,16	0,03	1,00	1,00	0,97	0,96	0,31	0,23	0,94	1,00	0,29	0,29	0,00	0,00	0,55	0,59	0,33	0,37	0,27	0,11	1,00	1,00	0,96	0,97	0,21	0,21
SIFT ORB LM		l1	0,45	0,64	0,15	0,16	0,02	0,02	0,27	0,28	0,19	0,21	0,16	0,03	1,00	1,00	0,93	0,92	0,31	0,23	0,67	0,76	0,17	0,17	0,00	0,00	0,52	0,56	0,31	0,35	0,27	0,11	0,96	1,00	0,92	0,93	0,20	0,21
	Affine	l2	0,93	1,00	0,40	0,42	0,02	0,02	0,30	0,32	0,22	0,24	0,16	0,03	1,00	1,00	1,00	1,00	0,31	0,24	1,00	1,00	0,43	0,43	0,00	0,00	0,58	0,62	0,35	0,39	0,27	0,11	1,00	1,00	1,00	1,00	0,23	0,22
		l1	0,59	0,80	0,21	0,22	0,02	0,02	0,30	0,32	0,22	0,23	0,16	0,03	1,00	1,00	1,00	0,99	0,31	0,23	0,94	1,00	0,30	0,30	0,00	0,00	0,57	0,61	0,35	0,38	0,27	0,11	1,00	1,00	0,99	1,00	0,22	0,22
	Projection	l2	1,00	1,00	1,00	1,00	0,03	0,03	1,00	1,00	1,00	1,00	0,18	0,06	1,00	1,00	1,00	1,00	0,33	0,43	1,00	1,00	1,00	1,00	0,03	0,04	1,00	1,00	1,00	1,00	0,31	0,25	1,00	1,00	1,00	1,00	1,00	0,83
		l1	1,00	1,00	1,00	1,00	0,03	0,03	1,00	1,00	1,00	1,00	0,19	0,08	1,00	1,00	1,00	1,00	0,33	0,49	1,00	1,00	1,00	1,00	0,03	0,04	1,00	1,00	1,00	1,00	0,32	0,31	1,00	1,00	1,00	1,00	1,00	1,00

The point-based registration methods are compared using a Z-test. This includes every combination of identified correspondences per method, and the use of l1 and l2 transformation optimization (after outlier detection).

A.8 Point-based registration violin plot (AP)

The accuracy of the point-based registration methods (4.5) has also been evaluated on AP MinIP pairs and is shown in Figure 12.

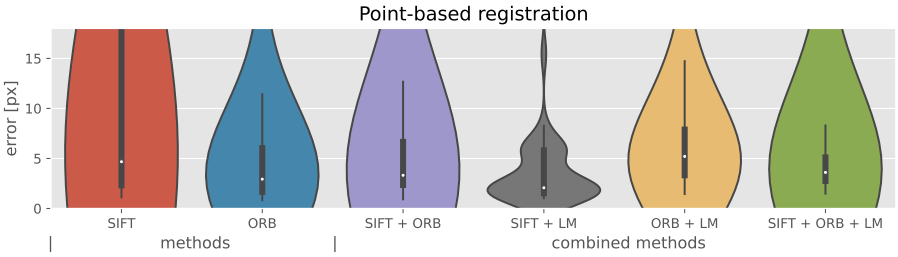

Fig. 12 Registration error (averaged distance between annotated point-correspondences) for least-squares similarity transformations using different subsets of automatically identified point corresondences in AP images.

A.9 Elastix registration Z-tests

The accuracy of the Elastix registration methods, together with the baseline SIFT+ORB similarity transformations, are compared using a Z-test. The best performing Elastix method (over-all worse than the baseline) is highlighted in bold.

Table 3: z-test for elastix method on AP images. If method (row) has a worse mean, the probability is set to one. The best performing elastix method is highlighted in bold.

Metric		MMI				MSD				NCC				SIFT ORB
	Transformation	similarity	affine	similarity (initialized)	affine (initialized)	similarity	affine	similarity (initialized)	affine (initialized)	similarity	affine	similarity (initialized)	affine (initialized)	similarity
MMI	similarity	1,00	0,03	0,56	0,58	0,01	0,29	1,00	0,01	1,00	0,72	0,02	0,97	1,00
	affine	1,00	1,00	1,00	1,00	0,92	1,00	1,00	0,89	1,00	1,00	0,39	1,00	1,00
	similarty (initialized)	1,00	0,08	1,00	1,00	0,05	0,62	1,00	0,06	1,00	1,00	0,04	1,00	1,00
	affine (initialized)	1,00	0,08	0,97	1,00	0,04	0,59	1,00	0,04	1,00	0,98	0,04	1,00	1,00
MSD	similarity	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	1,00	0,42	1,00	1,00
	affine	1,00	0,19	1,00	1,00	0,13	1,00	1,00	0,14	1,00	1,00	0,07	1,00	1,00
	similarty (initialized)	0,56	0,01	0,28	0,28	0,01	0,13	1,00	0,01	1,00	0,48	0,01	0,59	1,00
	affine (initialized)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,50	1,00	1,00
NCC	**similarity**	**0,46**	**0,01**	**0,21**	**0,21**	**0,00**	**0,09**	**0,89**	**0,01**	**1,00**	**0,42**	**0,01**	**0,49**	1,00
	affine	1,00	0,18	1,00	1,00	0,14	0,74	1,00	0,17	1,00	1,00	0,07	1,00	1,00
	similarty (initialized)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	affine (initialized)	1,00	0,04	0,63	0,65	0,02	0,36	1,00	0,03	1,00	0,75	0,02	1,00	1,00
SIFT ORB	similarity	0,33	0,01	0,14	0,17	0,00	0,05	0,75	0,01	0,88	0,39	0,01	0,38	1,00

Table 4: z-test for elastix method on lateral images. If method (row) has a worse mean, the probability is set to one. The best performing elastix method is highlighted in bold.

Metric		MMI				MSD				NCC				SIFT ORB
	Transformation	similarity	affine	similarity (initialized)	affine (initialized)	similarity	affine	similarity (initialized)	affine (initialized)	similarity	affine	similarity (initialized)	affine (initialized)	similarity
MMI	similarity	1,00	0,12	1,00	0,76	0,03	1,00	1,00	0,03	0,66	0,31	0,11	0,24	1,00
	affine	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	1,00	0,57	0,17	1,00	1,00
	similarty (initialized)	**0,59**	**0,08**	**1,00**	**0,40**	**0,01**	**0,73**	**0,74**	**0,02**	**0,44**	**0,29**	**0,10**	**0,18**	1,00
	affine (initialized)	1,00	0,14	1,00	1,00	0,05	1,00	1,00	0,04	0,81	0,33	0,11	0,28	1,00
MSD	similarity	1,00	0,62	1,00	1,00	1,00	1,00	1,00	0,50	1,00	0,47	0,14	0,77	1,00
	affine	0,84	0,10	1,00	0,61	0,02	1,00	0,99	0,02	0,58	0,30	0,11	0,22	1,00
	similarty (initialized)	0,86	0,11	1,00	0,64	0,03	1,00	1,00	0,03	0,59	0,30	0,11	0,22	1,00
	affine (initialized)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,57	0,17	1,00	1,00
	similarity	1,00	0,22	1,00	1,00	0,21	1,00	1,00	0,11	1,00	0,35	0,12	0,36	1,00
NCC	affine	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,33	1,00	1,00
	similarty (initialized)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	affine (initialized)	1,00	0,92	1,00	1,00	1,00	1,00	1,00	0,90	1,00	0,55	0,16	1,00	1,00
SIFT ORB	similarity	0,56	0,08	0,97	0,38	0,01	0,70	0,72	0,02	0,43	0,29	0,10	0,18	1,00

A.10 Elastix registration violin plot

Over-all, the Mattes mutual information similarity function proved most succesfull. Its accuracy for the similarity and affine transformation are compared in Figures 13, 14.

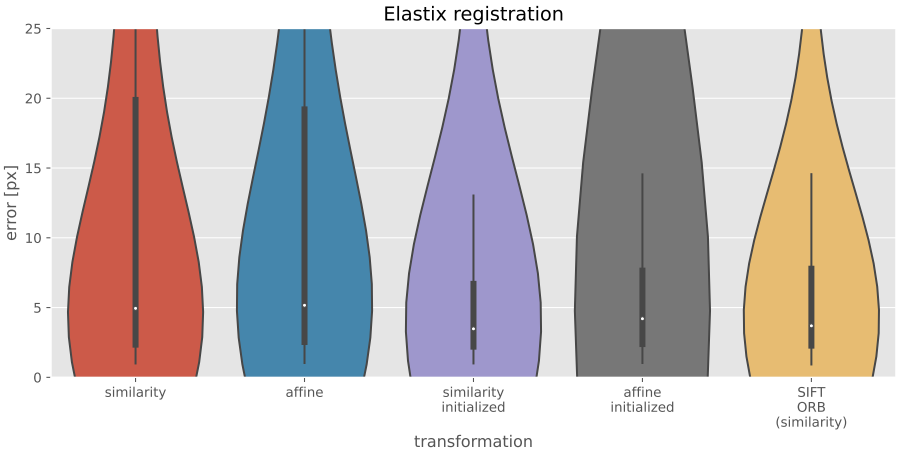

Fig. 13 Results of elastix registration optimizing mattes mutual information for AP images

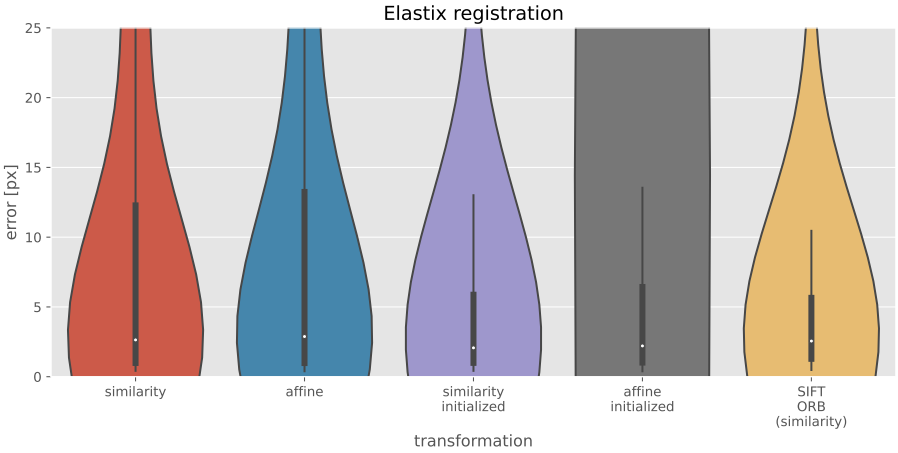

Fig. 14 Results of registration optimizing mattes mutual information for lateral images.

A.11 Automatic registration examples

A set of example images is aligned using SIFT and ORB points to compute the similarity transformation. Results for AP and lateral MinIP pairs can be seen in Figures 15, 16

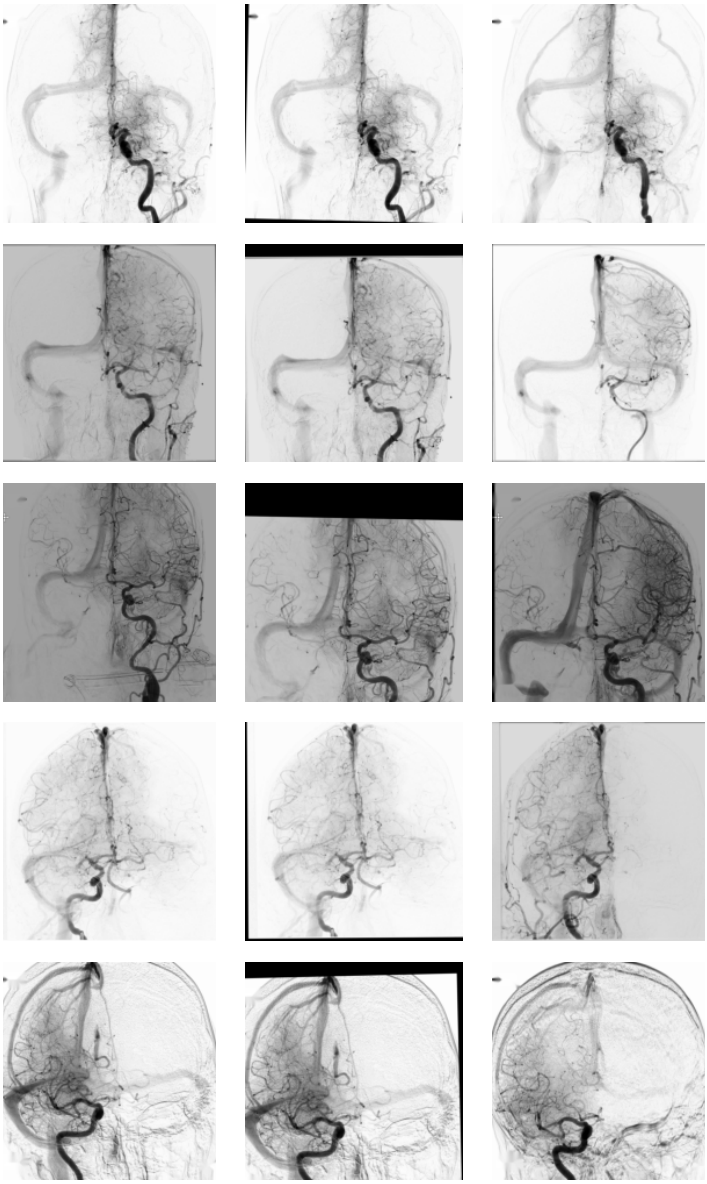

Fig. 15 AP registration examples: left column pre-EVT, middle is pre-EVT aligned to post-EVT in the right column.

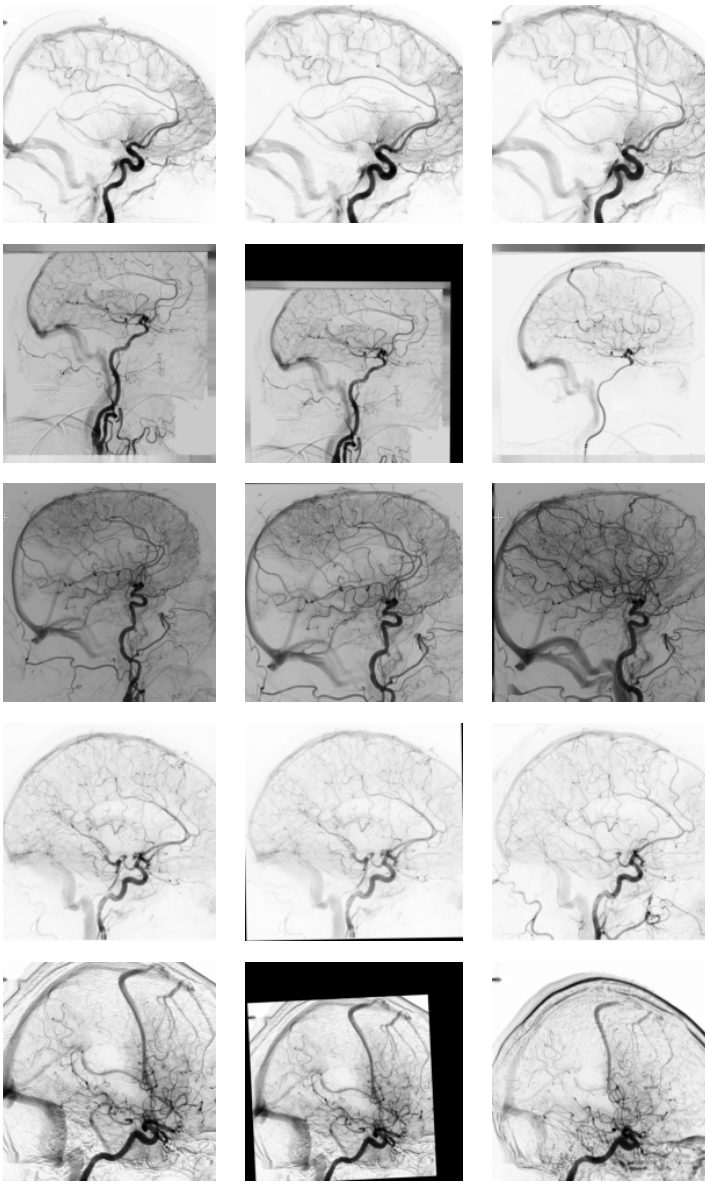

Fig. 16 Lateral registration examples: left column pre-EVT, middle is pre-EVT aligned to post-EVT in the right column.

B Least squares solutions for global transformations

B.1 Translation

For all linear global transformations, translation can separately optimized:

$$\vec{x}_{target} = T\vec{x}_{source} + \vec{b}$$

$$\vec{x}'_{target} = \vec{x}_{target} - \bar{\vec{x}}_{target} \qquad \vec{x}'_{source} = \vec{x}_{source} - \bar{\vec{x}}_{source}$$

$$\min_{T,\vec{b}} \vec{x}_{target} - T\vec{x}_{source} - \vec{b}_2^2 =$$

$$\min_{T,\vec{b}} \vec{x}'_{target} + \bar{\vec{x}}_{target} - T\vec{x}_{source} - T\bar{\vec{x}}_{source} - \vec{b}_2^2 =$$

$$\min_{T} \vec{x}'_{target} - T\vec{x}'_{source}{}_2 + \min_{T,\vec{b}} \bar{\vec{x}}_{target} - T\bar{\vec{x}}_{source} - \vec{b}_2^2$$

Where the second optimization problem is simply solved using the solution from the first optimization problem

$$\min_{T,\vec{b}} \bar{\vec{x}}_{target} - T\bar{\vec{x}}_{source} - \vec{b}_2^2$$

$$\vec{b} = \bar{\vec{x}}_{target} - T\bar{\vec{x}}_{source} \tag{18}$$

Which in the case of a pure translation becomes

$$\vec{b} = \bar{\vec{x}}_{target} - \bar{\vec{x}}_{source} \tag{19}$$

B.2 Rigid and Affine

$$\vec{x}'_{target} = T\vec{x}'_{source}$$

Let the source and target points compose a matrix

$$X'_{target} = [\vec{x}'_{target_1}, \vec{x}'_{target_2}, ...\vec{x}'_{target_N}] \quad X'_{source} = [\vec{x}'_{source_1}, \vec{x}'_{source_2}, ...\vec{x}'_{source_N}]$$

Then the least squares affine transformation is given by:

$$T = X'_{target} X'^{T}_{source} (X'_{source} X'^{T}_{source})^{-1} \tag{20}$$

and the rigid transformation, the closest orthogonal matrix with a positive determinant, is found using a singular value decomposition.

$$T = U\Sigma V^T$$

$$T_{rigid} = U \begin{bmatrix} 1 & 0 \\ 0 & sign(det(UV^T)) \end{bmatrix} V^T \tag{21}$$

Note that, since this is a 2×2 matrix (2D), or 3×3 (3D), an exact singular value decomposition can be computed.

B.3 Similarity

Rewriting the transformation into the following form provides a simple least-squares solution:

$$\begin{bmatrix} x'_{source_1} & -y'_{source_1} \\ y'_{source_1} & x'_{source_1} \\ x'_{source_2} & -y'_{source_2} \\ y'_{source_2} & x'_{source_2} \\ \vdots & \vdots \\ x'_{source_N} & -y'_{source_N} \\ y'_{source_N} & x'_{source_N} \end{bmatrix} \begin{bmatrix} s \times cos(\theta) \\ s \times sin(\theta) \end{bmatrix} = \begin{bmatrix} x'_{source_1} \\ y'_{source_1} \\ x'_{source_2} \\ y'_{source_2} \\ \vdots \\ x'_{source_N} \\ y'_{source_N} \end{bmatrix}$$

Having renamed the *left-hand side* (LHS) matrix to X'_{source} and *right-hand side* (RHS) matrix to X'_{target}, the solution becomes:

$$\begin{bmatrix} s \times cos(\theta) \\ s \times sin(\theta) \end{bmatrix} = (X'^{T}_{source} X'_{source})^{-1} X'^{T}_{source} X'_{target} \tag{22}$$

$$T_{similarity} = \begin{bmatrix} s \times cos(\theta) & -s \times sin(\theta) \\ s \times sin(\theta) & s \times cos(\theta) \end{bmatrix}$$

B.4 Projection

Direct linear transform (DLT)

$$x_{target} = \frac{a_{00}x_{source} + a_{01}y_{source} + a_{02}}{a_{20}x_{source} + a_{21}y_{source} + a_{20}} \quad y_{target} = \frac{a_{10}x_{source} + a_{11}y_{source} + a_{02}}{a_{20}x_{source} + a_{21}y_{source} + a_{20}} \tag{23}$$

Rewriting the equations will produce a linear system

$$(a_{20}x_{source} + a_{21}y_{source} + a_{22})x_{target} = a_{00}x_{source} + a_{01}y_{source} + a_{02}$$

$$(a_{20}x_{source} + a_{21}y_{source} + a_{22})y_{target} = a_{10}x_{source} + a_{11}y_{source} + a_{02}$$

and as long as the LHS matrix below is invertible, this will also provide an exact solution when using four points. When using more than four points, this does not provide a least squares solution to the original problem, but a good approximation:

$$\begin{bmatrix} x_{source_1} & y_{source_1} & 1 & 0 & 0 & 0 & x_{source_1}x_{target_1} & y_{source_1}x_{target_1} \\ 0 & 0 & 0 & x_{source_1} & y_{source_1} & 1 & x_{source_1}y_{target_1} & y_{source_1}y_{target_1} \\ x_{source_2} & y_{source_2} & 1 & 0 & 0 & 0 & x_{source_2}x_{target_2} & y_{source_2}x_{target_2} \\ 0 & 0 & 0 & x_{source_2} & y_{source_2} & 1 & x_{source_2}y_{target_2} & y_{source_2}y_{target_2} \\ \vdots & & & & & & & \\ x_{source_1} & y_{source_N} & 1 & 0 & 0 & 0 & x_{source_N}x_{target_N} & y_{source_N}x_{target_N} \\ 0 & 0 & 0 & x_{source_N} & y_{source_N} & 1 & x_{source_N}y_{target_N} & y_{source_N}y_{target_N} \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{20} \\ a_{21} \end{bmatrix} = \begin{bmatrix} x_{source_1} \\ y_{source_1} \\ x_{source_2} \\ y_{source_2} \\ \vdots \\ x_{source_N} \\ y_{source_N} \end{bmatrix}$$

Renaming the LHS matrix to X, and the RHS vector to $\vec{x}$, the approximation of $\vec{a}$ becomes:

$$\vec{a} = (X^T X)^{-1} X^T \vec{x}$$

Simplification and exact computation of the DLT: The DLT solution uses an 8×8 matrix. One such implementation can be found in Open CV, which uses a numerical solver. An exact solution of an inverse matrix has complexity $\mathcal{O}(n!)$ (with $n = 8$). Firstly we can reduce the system to a 6×6 in a similar manner as Equation 18. Additionally, one can notice $(X^T X)^{-1}$ can be re-written in block form:

$$\begin{bmatrix} X_1 & 0_{2\times 2} & X_2 \\ 0_{2\times 2} & X_1 & X_3 \\ X_2^T & X_3^T & X_4 \end{bmatrix}^{-1} = \begin{bmatrix} X_1^{-1} + X_1^{-1} X_2 X_5^{-1} X_1^{-1} & X_1^{-1} X_2 X_5^{-1} X_2^T X_1^{-1} & -X_1^{-1} X_2 X_5^{-1} \\ X_1^{-1} X_3 X_5^{-1} X_2^T X_1^{-1} & X_1^{-1} + X_1^{-1} X_3 X_5^{-1} X_3^T X_1^{-1} & -X_1^{-1} X_3 X_5^{-1} \\ -X_5^{-1} X_2^T X_1^{-1} & -X_5^{-1} X_3^T X_1^{-1} & X_5^{-1} \end{bmatrix} \quad (24)$$

with

$$X_5^{-1} = (X_4 - X_2^T X_1^{-1} X_2 - -X_3^T X_1^{-1} X_3)^{-1}$$

such that the solution only requires two inverse matrices, both of size 2×2. In our application, one can be pre-computed. Furthermore, as we will elaborate next, we particularly want to know the values of a_{20} and a_{21}.

$$\begin{bmatrix} a_{20} \\ a_{21} \end{bmatrix} = \begin{bmatrix} -X_5^{-1} X_2^T X_1^{-1} & -X_5^{-1} X_3^T X_1^{-1} & X_5^{-1} \end{bmatrix} X^T \vec{x} \quad (25)$$

Exact solutions: While not all transformation parameters have a least-squares solution (a_{20} and a_{21}), most do. For completeness, the solutions are provided on the next page, although in practice it comes down to adapting X_{source} in Equation 20 such that the denominator of Equation 2 is included, with fixed values a_{20} and a_{21}.

Note that theoretically, there are many solutions for a_{20} and a_{21} as its solutions would be the roots of an excessively high-order bi-variate polynomial. Trying to solve this, and evaluating which of the solutions is the global optimum is complicated and numerical. Using the DLT for these two parameters is therefore the better alternative.

Additional constraints The denominator of Equation 2 should not be zero within the field of view (and in practice is never close by). The closest point of that line to the origin (i.e. the centre of the field of view) is:

$$\vec{x} = \begin{pmatrix} \frac{a_{20}}{a_{21}(1+\frac{a_{20}}{a_{21}})^2} \\ \frac{a_{20}^2}{(a_{20}+a_{21})^2} - \frac{1}{a_{21}} \end{pmatrix}$$

Its distance to the origin should therefore be constraint to prevent unstable behaviour; for example by enforcing $\vec{x}^T \vec{x} > \frac{h}{2}^2 + \frac{w}{2}^2$.

$$a_{02} = \alpha \sum_i x_{target_i} - \frac{a_{00}x_{source_i} + a_{01}y_{source_i}}{a_{20}x_{source_i} + a_{21}y_{source_i} + 1} \tag{26}$$

$$a_{12} = \alpha \sum_i y_{target_i} - \frac{a_{10}x_{source_i} + a_{11}y_{source_i}}{a_{20}x_{source_i} + a_{21}y_{source_i} + 1} \tag{27}$$

$$\begin{bmatrix} a_{00} & a_{10} \\ a_{01} & a_{11} \end{bmatrix} = Z^{-1} \begin{bmatrix} \sum_i x_{target_i}(x_{source_i} - \alpha\beta) & \sum_i y_{target_i}(x_{source_i} - \alpha\beta) \\ \sum_i x_{target_i}(y_{source_i} - \alpha\gamma) & \sum_i y_{target_i}(y_{source_i} - \alpha\gamma) \end{bmatrix} \tag{28}$$

$$\alpha = (\sum_j \frac{1}{a_{20}x_{source_i} + a_{21}y_{source_i} + 1})^{-1}$$

$$\beta = \sum_j \frac{x_{source_j}}{a_{20}x_{source_j} + a_{21}y_{source_j} + 1})$$

$$\gamma = \sum_j \frac{y_{source_j}}{a_{20}x_{source_j} + a_{21}y_{source_j} + 1})$$

$$Z = \begin{bmatrix} \sum_i \frac{x_{source_i}(x_{source_i}+\alpha\beta)}{a_{20}x_{source_i}+a_{21}y_{source_i}+1} & \sum_i \frac{y_{source_i}(x_{source_i}+\alpha\beta)}{a_{20}x_{source_i}+a_{21}y_{source_i}+1} \\ \sum_i \frac{x_{source_i}(y_{source_i}+\alpha\gamma)}{a_{20}x_{source_i}+a_{21}y_{source_i}+1} & \sum_i \frac{y_{source_i}(y_{source_i}+\alpha\gamma)}{a_{20}x_{source_i}+a_{21}y_{source_i}+1} \end{bmatrix}$$

References

[1] WHO. WHO methods and data sources for country-level causes of death 2000-2019. Global Health Estimates Technical Paper. 2020;.

[2] Jansen IG, Mulder MJ, Goldhoorn RJB. Endovascular treatment for acute ischaemic stroke in routine clinical practice: prospective, observational cohort study (MR CLEAN Registry). bmj. 2018;360.

[3] Campbell BC, De Silva DA, Macleod MR, Coutts SB, Schwamm LH, Davis SM, et al. Ischaemic stroke. Nature Reviews Disease Primers. 2019;5(1):1–22.

[4] Higashida RT, Furlan AJ. Trial design and reporting standards for intra-arterial cerebral thrombolysis for acute ischemic stroke. stroke. 2003;34(8):e109–e137.

[5] Liebeskind A, Deshpande A, Murakami J, Scalzo F. Automatic Estimation of Arterial Input Function in Digital Subtraction Angiography. In: Bebis G, Boyle R, Parvin B, Koracin D, Ushizima D, Chai S, et al., editors. Advances in Visual Computing. Cham: Springer International Publishing; 2019. p. 393–402.

[6] Siemens.: Artis icono. USA Siemens Medical Solutions. Available from: https://www.siemens-healthineers.com/en-us/angio/artis-icono-topic.

[7] Goorden MC.: Lecture slides Medical Imaging Systems and Signals: conventional X-ray. TU Delft.

[8] Lev MH, Gonzalez RG.: CT Angiography and CT Perfusion Imaging.

[9] Hubbell JH, Seltzer SM.: NIST: X-Ray Mass Attenuation Coefficients.

[10] Szeliski R. Computer vision: algorithms and applications. Springer Nature; 2022.

[11] ImportanceOfBeingErnest.: Generate deformation field. Available from: https://stackoverflow.com/questions/47295473/how-to-plot-using-matplotlib-python-colahs-deformed-grid.

[12] Klein S, Staring M, Murphy K, Viergever MA, Pluim JP. Elastix: a toolbox for intensity-based medical image registration. IEEE transactions on medical imaging. 2009;29(1):196–205.

[13] Tustison NJ, Avants BB, Gee JC.: Learning image-based spatial transformations via convolutional neural networks: A review.

[14] Lowe DG. Distinctive image features from scale-invariant keypoints. International journal of computer vision. 2004;60(2):91–110.

[15] Wang G, Rister B, Cavallaro JR. Workload analysis and efficient OpenCL-based implementation of SIFT algorithm on a smartphone. 2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings. 2013;https://doi.org/10.1109/GlobalSIP.2013.6737002.

[16] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision. Ieee; 2011. p. 2564–2571.

[17] Rosten E, Drummond T. Machine learning for high-speed corner detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006;3951 LNCS. https://doi.org/10.1007/11744023_34.

[18] Calonder M, Lepetit V, Özuysal M, Trzcinski T, Strecha C, Fua P. BRIEF: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012;34. https://doi.org/10.1109/TPAMI.2011.222.

[19] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. 1981;24(6):381–395.

[20] Mattes D, Haynor DR, Vesselle H, Lewellyn TK, Eubank W. Nonrigid multimodality image registration. Proc SPIE Medical Imaging. 2001;4322.

[21] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017;60. https://doi.org/10.1145/3065386.

[22] Freeman WT, Adelson EH. The Design and Use of Steerable Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1991;13. https://doi.org/10.1109/34.93808.

[23] Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial transformer networks. vol. 2015-January; 2015. .

[24] Pandiyan V, Murugan P, Tjahjowidodo T, Caesarendra W, Manyar OM, Then DJH. In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning. Robotics and Computer-Integrated Manufacturing. 2019;57. https://doi.org/10.1016/j.rcim.2019.01.006.

[25] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. p. 234–241.

[26] Dalca AV, Rakic M, Guttag J, Sabuncu MR. Learning conditional deformable templates with convolutional networks. Advances in Neural Information Processing Systems. 2019;32.

[27] Arsigny V, Commowick O, Pennec X, Ayache N.: A Log-Euclidean Framework for Statistics on Diffeomorphisms.

[28] Kullback S, Leibler RA. On Information and Sufficiency. The Annals of Mathematical Statistics. 1951;22. https://doi.org/10.1214/aoms/1177729694.

[29] Lin J. Divergence Measures Based on the Shannon Entropy. IEEE Transactions on Information Theory. 1991;37. https://doi.org/10.1109/18.61115.

[30] Gupta R.: KL-divergence between two Gaussian Distributions. Available from: https://mr-easy.github.io/2020-04-16-kl-divergence-between-2-gaussian-distributions.

[31] : GitHub - kevinzakka/spatial-transformer-network: A Tensorflow implementation of Spatial Transformer Networks — github.com. [Accessed 12-Jan-2023]. https://github.com/kevinzakka/spatial-transformer-network.

[32] Deng L. The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine. 2012;29. https://doi.org/10.1109/MSP.2012.2211477.