PCSAC: Robust Conditional Multi-Model Fitting

Wenyu Yin 23020230157341[1], Guangyao Li 36920230157049[1], Xintao Xu 23020230157340[1], Siping Zhuang 23020231154267[1], Suzhou Que 36920231153230[1]

[1]School of Information Science and Engineering, Xiamen University, China

Abstract

We utilize a deep learning architecture to learn model fitting problems in computer vision. Robust optimization by extending the classic RANSAC algorithm. We learn search strategies from data. Based on the previously detected model, a neural network is applied to guide the RANSAC estimator to different subsets of all measurements, in order to find model instances. Applications include finding multiple vanishing points in artificial scenes, fitting planes to building images, or estimating multiple rigid motions within the same sequence.

Introduction

Humans primarily perceive the external world through hearing and vision, with over 80% of information obtained through vision. With the development of science and social progress, computers have become an indispensable tool in human daily life and work. In order to endow computers with the perceptual function of human vision and enable them to process visual information, an emerging discipline - computer vision has received widespread attention. In the past few decades, a large number of computer vision based products have also emerged in our daily lives. For example, automatic recognition of license plates,Face detection and beautification with digital cameras, autonomous driving of cars, panoramic image synthesis, and more.

How to enable computers to extract effective information from images is crucial for computer vision. In most cases, this effective information can be represented through parameter models (Wong 2013; Pham 2014).Model fitting refers to estimating appropriate model parameters from a set of observation data, which can be used to estimate parameter models. It can be seen that model fitting plays an important role in computer vision. The current model fitting methods can be used in many computer vision applications. For example, 3D reconstruction(Se and Jasiobedzki 2006; Izadi et al. 2011), image stitching(Gao, Kim, and Brown 2011; Zaragoza et al. 2013), motion segmentation(Crocco, Rubino, and Del Bue 2016; Schonberger and Frahm 2016), object recognition(Yan et al. 2015; Kong et al. 2016) , and so on. With the development and intersection of research

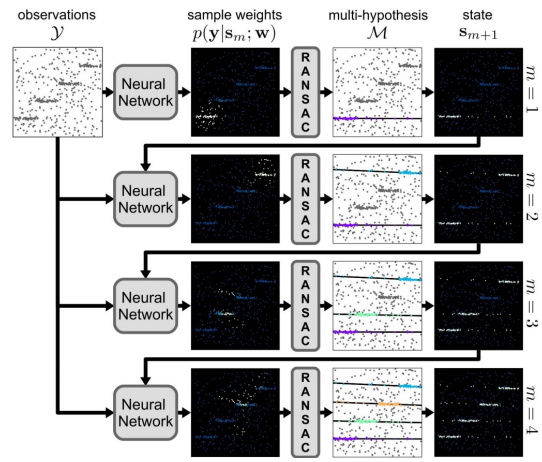

Figure 1: Multi-Hypothesis Generation: a neural network predicts sampling weights p for all observations conditioned on a state s. A RANSAC-like sampling process uses these weights to select a model hypothesis and appends it to the current multi-instance hypothesis M. The state s is updated based on M and fed into the neural network repeatedly.

fields such as machine learning, image processing, and robust statistics, model fitting(Farina et al. 2023; Churchill et al. 2023) has made significant progress.

Model fitting has a long history of development in the field of statistics. With the development of interdisciplinary fields, model fitting also involves various theories and technologies such as image processing, machine learning, and artificial intelligence. Model fitting is a fundamental discipline in the field of computer vision, providing reliable research basis for other tasks in this field. Currently, many universities and research institutions both domestically and internationally are paying increasing attention to computational vision, and model fitting is also playing an increasingly important role.

In recent decades, many experts and scholars at home and abroad have conducted in-depth research on model fitting theory and methods, and proposed many fitting algorithms. However, the existing fitting algorithms are far from meeting the needs of practical engineering. Currently, the main difficulties faced by model fitting algorithms include sensitivity to imbalanced data and interference from high outliers.

Despite its simplicity and robustness to outliers, Random Sample Consensus (RANSAC)(Fischler and Bolles 1981) remains an important method for robust optimization and an important component of many state-of-the-art visual stimuli(Mur-Artal and Tardós 2017; Brachmann and Rother 2018). RANSAC allows for accurate estimation of model parameters from a set of observations, some of which are outliers. For this purpose, RANSAC iteratively selects the random subset Ns of observations, known as the minimum set, to create model assumptions. Assumptions are sorted based on their consistency with all observations, and the highest ranked assumption is returned as the final estimate.

Modern, state-of-the-art methods solve multi-model fitting simultaneously instead, by using clustering or optimisation techniques to assign data points to models or an outlier class(Barath and Matas 2018b, 2019) . In our work, we revisit the idea of sequential processing, but combine it with recent advances in Deep Learning robust estimators(Yi et al. 2018; Brachmann and Rother 2019). Sequential processing easily lends itself to conditional sampling approaches, and with this we are able to achieve state-of-the-art results despite supposedly being conceptually inferior to simultaneous approaches.

The main inspiration of our work stems from the work of Brachmann and Rother(Brachmann and Rother 2019) , where they train a neural network-PointNet to enhance the sample efficiency of a RANSAC estimator for single model estimation. In contrast, we investigate multi-model fitting by letting the neural network update sampling weights conditioned on models it has already found. This allows the neural network to not only suppress outliers, but also inliers of all but the current model of interest. Since our new RANSAC variant samples model hypotheses based on conditional probabilities, we name it PointNet Conditional Sample Consensus or PCSAC proves to be powerful and achieves top performance for several applications.

To summarise, our main contributions are as follows:

- A robust multi model fitting method based on deep learning. It is based on a neural network that sequentially updates the conditional sampling probability of the hypothesis selection process.
- Choosing the inlier count itself as training objective facilitates self-supervised learning of our work.
- We have achieved vanishing point estimation and multi model homogeneous estimation, which exceeds competitive robust estimation.

Related work

Robust model fitting is a key problem in Computer Vision, which has been studied extensively in the past. RANSAC is arguably the most commonly implemented approach.While effective in the single instance case, RANSAC cannot estimate multiple model instances apparent in the data. Sequential RANSAC(Vincent and Laganiére 2001) fits multiple models sequentially by applying RANSAC, removing inliers of the selected hypothesis, and repeating until a stopping criterion is reached.Another group of methods utilises preference analysis(Zhang and Kǒsecká 2005) which assumes that observations explainable by the same model instance have similar distributions of residuals(Magri and Fusiello 2016, 2019) . In order to better deal with intersecting models, RansaCov(Magri and Fusiello 2016) formulates multi-model fitting as a set coverage problem.Several works propose improved sampling schemes to increase the likelihood of generating accurate model hypotheses from all-inlier minimal sets(Torr, Nasuto, and Bishop 2002; Barath and Matas 2018a) in the single-instance case.

Deep learning has been applied in the past to fit single parameter models by directly predicting model parameters from images(Kendall, Grimes, and Cipolla 2015; DeTone, Malisiewicz, and Rabinovich 2016), replacing robust estimators(Sun et al. 2020) or enhancing robust estimators. Later, deep learning model fitting methods were often inspired by point cloud registration, constructing a model framework that directly takes the point cloud as input and outputs the entire input class label or each point segment/-part label of each input point. Yi et al. trained a neural network to classify a set of sparse correspondences between internal and external values. Their network is inspired by PointNet(Qi et al. 2017) and independently processes each corresponding relationship through a series of multi-layer perceptrons (MLPs). Normalize injection of global context through the use of instances and batch processing between layers(Ulyanov, Vedaldi, and Lempitsky 2016) . The NGRANSAC network predicts weights to guide RANSAC sampling, rather than using internal class labels, using the angle error between the estimated relative attitude and the actual ground attitude as task loss. Notably, Brachmann and Rother(Brachmann and Rother 2019) train a neural network to enhance the sample efficiency of RANSAC by assigning sampling weights to each data point, effectively suppressing outliers. Few works, such as the conditional sampling based on residual sorting by Chin et al.(Chin, Yu, and Suter 2011) , or the guided hyperedge sampling of Purkait et al.(Purkait et al. 2016) , consider the case of multiple instances. In contrast to these handcafted methods, we present the PointNet-based conditional sampling approach.

Method

Given a set of noisy observations y contaminated by outliers, we seek to fit M instances of a geometric model h apparent in the data. We denote the set of all model instances as M. PCSAC estimates M via three nested loops.First, We generate a single model instance h via RANSACbased sampling, guided by a neural network. Second, We repeat single model instance generation while conditionally updating sampling weights. Multiple single model hypotheses compound to a multi-hypothesis M. Third, We repeat steps 1 and 2 to sample multiple multi-hypotheses M independently. We choose the best multi-hypothesis as the final multi-model estimate M.We discuss these conceptional levels more formally below.

Single model sampling

We estimate parameters of a single model, e.g. one VP, from a minimal set of C observations, e.g. two line segments, using a minimal solver f_s. As in RANSAC, we compute a hypothesis pool H via random sampling of S minimal sets. We choose the best hypothesis h based on a single instance scoring function g_s. Typically, gs is realised as inlier counting via a residual function r(y, h) and a threshold t.

Generate Multiple Model Assumptions

We repeat single model instance sampling M times to generate a full multi-hypothesis M, e.g. a complete set of vanishing points for an image. Particularly, we select M model instances h_m from their respective hypothesis pools H_m. Applied sequentially, previously chosen hypotheses can be factored into the scoring function g_s when selecting h_m:

$$h_m = \arg\max_{h \in H_m} g_s(h, y, h_m - 1) \tag{1}$$

Sampling Multiple Model Assumption

We repeat the previous process P times to generate a pool of multi-hypotheses P. We select the best multi-hypothesis according to a multi-instance scoring function g_m:

$$M = \arg\max_{M \in P} g_m(m, y) \tag{2}$$

where g_m measures the joint inlier count of all hypotheses in M, and where the m in gm stands for multi-instance.

Conditional Sampling

RANSAC samples minimal sets uniformly from Y. For large amounts of outliers in Y, the number of samples S required to sample an outlier-free minimal set with reasonable probability grows exponentially large. RANSAC instead sample observations according to a categorical distribution y $\sim$ p(y ; w) parametrised by a neural network w. The neural network biases sampling towards outlier-free minimal sets which generate accurate hypotheses h. While this approach is effective in the presence of outliers, it is not suitable for dealing with pseudooutliers posed by multiple model instances. Sequential RANSAC conditions the sampling on previously selected hypotheses, i.e. y $\sim$ p(y ; h_1..., h_{m-1}), by removing observations already deemed as inliers from Y after each hypothesis selection. While being able to reduce pseudo-outliers for subsequent instances, this approach can neither deal with pseudo-outliers in the first sampling step, nor with gross outliers in general. Instead, we parametrise the conditional distribution by a neural network w conditioned on a state s: y $\sim$p(y ; s ; w).

The state vector s_m at instance sampling step m encodes information about previously sampled hypotheses in a meaningful way. We use the inlier scores of all observations w.r.t. all previously selected hypotheses as the state s_m. We define the state entry $s_{m,i}$of observation y_i as:

$$s_m, i = \max_{j \in [1,m)} g_y(y_i, h_j) \tag{3}$$

with g_y gauging if y is an inlier of model h. We sample multi-instance hypothesis pools independently:

$$p(P; w) = \prod_{i=0}^{n} p(M, w), \tag{4}$$

while conditioning multi-hypotheses on the state s:

$$p(M; w) = \prod_{m=1}^{M} p(H; s_m; w), \tag{5}$$

with

$$p(H; s; w) = \prod_{s=1}^{S} p(h; s; w), \tag{6}$$

with

$$p(H; s; w) = \prod_{c=1}^{C} p(y; s; w). \tag{7}$$

Note that we do not update state s while sampling single instance hypotheses pools H, but only within sampling of multi-hypotheses M. We provide details of scoring functions g_y, g_m and g_s in the appendix.

Optimize network

Neural network parameters w shall be optimised in order to increase chances of sampling outlier- and pseudo-outlierfree minimal sets which result in accurate, complete and duplicate-free multi-instance estimates M. we minimise the expectation of a task loss $\mathcal{L}$(M) which measures the quality of an estimate:

$$\mathcal{L}(w) = \mathcal{E}(P; w)[\mathcal{L}(M)], \tag{8}$$

In order to update the network parameters w, we approximate the gradients of the expected task loss:

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \mathcal{E}(P)[\mathcal{L}(M)\frac{\partial \log p(P; w)}{\partial w}], \tag{9}$$

by drawing K samples $P_k \sim$ p(M; w):

$$\frac{\partial \mathcal{L}(w)}{\partial w} \approx \frac{1}{K}\sum_{k=1}^{K}[\mathcal{L}(M_k)\frac{\partial \log p(P_k; w)}{\partial w}]. \tag{10}$$

As we can infer from Eq. 9, neither the loss $\mathcal{L}$, nor the sampling procedure for ˆM need be differentiable. we subtract the mean loss from $\mathcal{L}$ to reduce variance.

Supervised Training

If ground truth models M^{gt} = h_1^{gt},...,h_G^{gt} are available, we can utilise a task-specific loss $\mathcal{L}_s$(h, h^{gt} measuring the error between a single ground truth model m and an estimate h. For example, $\mathcal{L}_s$ may measure the angle between an estimated and a true vanishing direction. First, however, we need to find an assigment between M^{gt} and M. We compute a cost matrix C, with C_{ij} = $\mathcal{L}_s(h_i$, h_j^{gt}, and define the multi-instance loss as the minimal cost of an assignment obtained via the Hungarian method f_H:($\mathcal{L}$(M, M^{gt}= $f_H(C_{1:min(M,G)})$. Note that we only consider at most G model estimates h which have been selected first, regardless of how many estimates M were generated, i.e. this loss encourages early selection of good model hypotheses, but does not penalise bad hypotheses later on.

Self-supervised Training

In absence of ground-truth labels, we can train PCNSAC in a self-supervised fashion by replacing the task loss with another quality measure. We aim to maximise the average joint inlier counts of the selected model hypotheses:

$$g_{ci}(h_m, y) = \frac{1}{|y|} \sum_{i=1}^{|y|} \max_{j \in [1,m)} g_i(y_i, h_j). \tag{11}$$

We then define our self-supervised loss as:

$$\mathcal{L}_{self}(M) = -\frac{1}{M} \sum_{m=1}^{M} g_{ci}(h_m, y). \tag{12}$$

Eq. 11 monotonically increases w.r.t. m, and has its minimum when the models in M induce the largest possible minimally overlapping inlier sets descending in size.

Instance Selection

In some scenarios, the number of instances M needs to be determined as well but is not known beforehand, e.g. for uniquely assigning observations to model instances. For such cases, we consider the subset of instances $M_{1:q}$ up to the q-th model instance h_q which increases the joint inlier count by at least Θ. Note that the inlier threshold Θ for calculating the joint inlier count at this point may be chosen differently from the inlier threshold t during hypothesis sampling. For example, in our experiments for homography estimation, we use a $\Theta > t$ in order to strike a balance between under- and oversegmentation.

Experiments

For conditional sampling weight prediction, we implement a neural network based on PointNet. We provide implementation and training details, as well as more detailed experimental results, in the appendix.

Line Fitting

We apply PCSAC to the task of fitting multiple lines to a set of noisy points with outliers. For training, we generated a synthetic dataset: each scene consists of randomly placed lines with points uniformly sampled along them and perturbed by Gaussian noise, and uniformly sampled outliers. After training PCSAC on this dataset in a supervised fashion, we applied it to the synthetic dataset. Fig. 2 shows how PCSAC sequentially focuses on different parts of the scene, depending on which model hypotheses have already been chosen, in order to increase the likelihood of sampling outlier-free non-redundant hypotheses. Notably, the network learns to focus on junctions rather than individual lines for selecting the first instances. The RANSAC-based single-instance hypothesis sampling makes sure that PCSAC still selects an individual line.

Two-view Plane Segmentation

Given feature point correspondences from two images showing different views of the same scene, we estimate multiple homographies H conforming to different 3D planes in

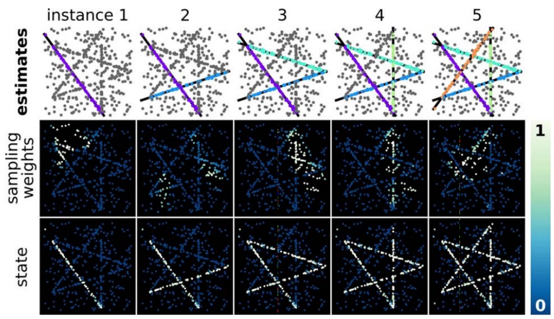

Figure 2: Line fitting result for the star5 scene.We show the generation of the multi-hypothesis M eventually selected by PCSAC.Top: Original points with estimated line instances at each instance selection step. Middle: Sampling weights at each instance step. Bottom: State s generated from the selected model instances.

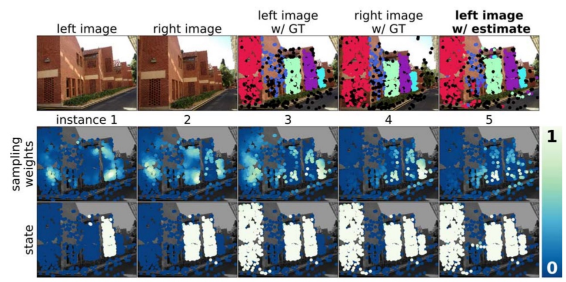

Figure 3: Homography fitting result for the AdelaideRMF unihouse scene. Top: Left and right image, feature points with ground truth labels, and feature points with labels predicted by PCSAC-S. Middle: Sampling weights of feature points at each instance step. Bottom:State s generated from the selected model instances.

the scene. As no sufficiently large labelled datasets exist for this task, we train our approach self-supervised (PCSAC-S) using SIFT feature correspondences extracted from the structure-from-motion scenes. Evaluation is performed on the AdelaideRMF homography estimation dataset and adheres to the protocol, i.e. we report the average misclassification error (ME) and its standard deviation over all scenes for five runs using identical parameters. We compare against the robust estimators Progressive-X, Multi-X, PEARL, MCT, RPA, T-Linkage, RansaCov and Sequeantial RANSAC.

Vanishing Point Estimation

A vanishing point $v \propto Kd$ arises as the projection of a direction vector d in 3D onto an image plane using camera parameters K. Parallel lines, i.e. with the same direction d, hence converge in v after projection. If v is known, the corresponding direction d can be inferred via inversion: $d \propto K^{-1}v$. Fig. 4 shows VPs therefore provide information about the 3D structure of a scene from a single image. While two corresponding lines are sufficient to estimate a VP, re-

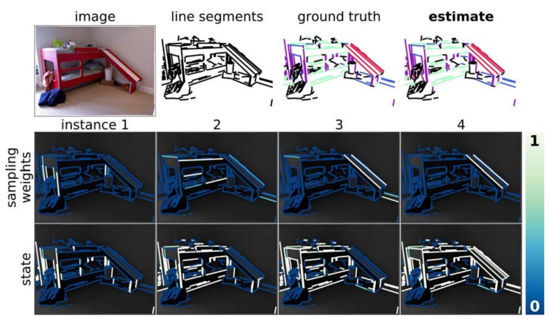

Figure 4: VP fitting result for a scene from the NYU-VP test set.Top: Original image, extracted line segments, assignment to ground truth VPs, and assignment to VPs predicted by PCSAC.Middle: Sampling weights of line segments at each instance step.Bottom: State s generated from the selected model instances.

alworld scenes generally contain multiple VP instances. We apply PCSAC to the task of VP detection and evaluate it on NYU-VP and YUD+ datasets, as well as on YUD. We compare against several other robust estimators, and also against task-specific state-of-the art VP detectors. We train PCSAC on the training set of NYU-VP in a supervised fashion and evaluate on the test sets of NYU-VP, YUD+ and YUD using the same parameters. YUD and YUD+ were neither used for training nor parameter tuning. Notably, NYU-VP only depicts indoor scenes, while YUD also contains outdoor scenes.

Results

We recomputed results for MCT using the code provided by the authors. For Sequential RANSAC, we used our own implementation. Other results were carried over. PCSAC-S outperforms state-of-the-art Progressive-X, yielding a significantly lower average ME with a marginally higher standard deviation. Notably, Sequential RANSAC performs favourably on this task as well. Fig. 3 shows a qualitative result for PCSAC-S.

Ablation Study

We perform ablation experiments in order to highlight the effectiveness of several methodological choices. PCSAC with EM refinement consistently performs best on both vanishing point and homography estimation. If we disable EM refinement, accuracy drops measurably, yet remains on par with state-of-the-art. On NYU-VP we can observe that the self-supervised trained PCSAC-S achieves state-of-the-art performance, but is still surpassed by PCSAC trained in a supervised fashion. Training PCSAC-S without inlier masking regularisation reduces accuracy measurably, while training only with IMR and disabling the self-supervised loss produces poor results. Switching to unconditional sampling for PCSAC (NYU-VP) or PCSAC-S (AdelaideRMF) comes with a significant drop in performance, and is akin to incorporating vanilla NG-RANSAC into Sequential RANSAC.

Conclusion

We have presented PCSAC, the learning-based robust estimator for detecting multiple parametric models in the presence of noise and outliers. A neural network learns to guide model hypothesis selection to different subsets of the data, finding model instances sequentially. We have applied PCSAC to multi-homography estimation, achieving state-of-the-art accuracy for both tasks.

References

Barath, D.; and Matas, J. 2018a. Graph-cut RANSAC. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6733–6741.

Barath, D.; and Matas, J. 2018b. Multi-class model fitting by energy minimization and mode-seeking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 221–236.

Barath, D.; and Matas, J. 2019. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3780–3788.

Brachmann, E.; and Rother, C. 2018. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4654–4662.

Brachmann, E.; and Rother, C. 2019. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4322–4331.

Chin, T.-J.; Yu, J.; and Suter, D. 2011. Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4): 625–638.

Churchill, V.; Manns, S.; Chen, Z.; and Xiu, D. 2023. Robust modeling of unknown dynamical systems via ensemble averaged learning. *Journal of Computational Physics*, 474: 111842.

Crocco, M.; Rubino, C.; and Del Bue, A. 2016. Structure from motion with objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4141–4149.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2016. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*.

Farina, M.; Magri, L.; Menapace, W.; Ricci, E.; Golyanik, V.; and Arrigoni, F. 2023. Quantum Multi-Model Fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13640–13649.

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

Gao, J.; Kim, S. J.; and Brown, M. S. 2011. Constructing image panoramas using dual-homography warping. In *CVPR 2011*, 49–56. IEEE.

Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. 2011. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 559–568.

Kendall, A.; Grimes, M.; and Cipolla, R. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, 2938–2946.

Kong, T.; Yao, A.; Chen, Y.; and Sun, F. 2016. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 845–853.

Magri, L.; and Fusiello, A. 2016. Multiple model fitting as a set coverage problem. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3318–3326.

Magri, L.; and Fusiello, A. 2019. Fitting multiple heterogeneous models by multi-class cascaded t-linkage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7460–7468.

Mur-Artal, R.; and Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5): 1255–1262.

Pham, T. T. 2014. *Robust estimation in computer vision: optimisation methods and applications.* Ph.D. thesis.

Purkait, P.; Chin, T.-J.; Sadri, A.; and Suter, D. 2016. Clustering with hypergraphs: the case for large hyperedges. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1697–1711.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Se, S.; and Jasiobedzki, P. 2006. Photo-realistic 3D model reconstruction. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 3076–3082. IEEE.

Sun, W.; Jiang, W.; Trulls, E.; Tagliasacchi, A.; and Yi, K. M. 2020. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11286–11295.

Torr, P. H.; Nasuto, S. J.; and Bishop, J. M. 2002. Napsac: High noise, high dimensional robust estimation-it's in the bag. In *British Machine Vision Conference (BMVC)*, volume 2, 3.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Vincent, E.; and Laganiére, R. 2001. Detecting planar homographies in an image pair. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, 182–187. IEEE.

Wong, H. S. 2013. *A preference analysis approach to robust geometric model fitting in computer vision.* Ph.D. thesis.

Yan, J.; Yu, Y.; Zhu, X.; Lei, Z.; and Li, S. Z. 2015. Object detection by labeling superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5107–5116.

Yi, K. M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2666–2674.

Zaragoza, J.; Chin, T.-J.; Brown, M. S.; and Suter, D. 2013. As-projective-as-possible image stitching with moving DLT. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2339–2346.

Zhang, W.; and Kŏsecká, J. 2005. Nonparametric estimation of multiple structures with outliers. In *International Worksop on Dynamical Vision*, 60–74. Springer.