Supplementary Material for "On the Estimation of Image-matching Uncertainty in Visual Place Recognition"

Mubariz Zaffar
ME, TU Delft
The Netherlands
M.Zaffar@tudelft.nl

Liangliang Nan
ABE, TU Delft
The Netherlands
Liangliang.Nan@tudelft.nl

Julian F. P. Kooij
ME, TU Delft
The Netherlands
J.F.P.Kooij@tudelft.nl

March 29, 2024

6. Supplementary Material

We provide here an ablation of SUE by changing the backbone and the weight function. A probabilistic interpretation of SUE is then presented and later used to perform density compensation for dissimilarly distributed query and reference images. We further provide the precision-recall curves for the remainder five VPR datasets. The complementarity of SUE, STUN, and L2-distance to *GV* is also shown on these datasets. We also show these complementarity plots of other techniques with SUE. Then, we connect the concept of geometric burstiness [40] with SUE. Finally, some qualitative results are shown in the form of correctly/incorrectly matched queries ranked with different types of uncertainty estimates.

6.1. More ablation studies of SUE

We perform two further experiments: changing the backbone feature extractor from STUN [9] to CosPlace [6] to show SUE's generality to other backbones in Fig. 8, and the benefit of using the exponential weighing function (in Equation (2) of the main paper) instead of the uniform weighing, as reported in Table 4.

Weigh.	Pitts.	San.	Stlu.	Eyn.	MSLS	Avg
Uniform	0.81	0.77	0.67	0.77	0.49	0.70
SUE	**0.94**	**0.84**	**0.88**	**0.93**	**0.77**	**0.87**

Table 4. SUE weighs the contribution of the nearest neighbor poses based on the distance in the feature space with an exponentially decaying function. This performs better than uniform weighing of the variance of the reference poses.

6.2. A probabilistic view of SUE

We here present a probabilistic view of SUE, which will help formulate a modified version in Section 6.3 to account for different spatial distributions of queries and references.

Consider $M \in \{1, \cdots, N\}$ as a stochastic 'match' variable that indicates which of the N references is a true reference. So, $M = i$ would mean reference i is the 'true' match for the query. Then $\mathsf{p}(M = i)$ expresses the prior belief that any reference i could be the true reference.

Assuming that some reference i is the true reference, $M = i$, then the observed query feature f_q can be expected to be similar to the reference feature f_i, with some homoscedastic Gaussian noise or variation added to all feature dimensions,

$$\mathsf{p}(f_q|M = i) = \mathbf{N}(f_q|f_{(i)}, \Sigma_f) \tag{4}$$
$$\propto e^{-\lambda \cdot ||f_q - f_{(i)}||_2} \tag{5}$$
$$\propto w_{(i)}. \tag{6}$$

So, the weight term of Equation (3) can be considered as the non-normalized likelihood term. Note that the hyperparameter λ subsumes the noise parameter Σ_f.

Through Bayes' rule, we can express the posterior belief over M given the query feature as

$$\mathsf{p}(M|f_q) = \frac{\mathsf{p}(f_q|M)\mathsf{p}(M)}{\mathsf{p}(f_q)} = \frac{\mathsf{p}(f_q|M)\mathsf{p}(M)}{\sum_j \mathsf{p}(f_q|M = j)\mathsf{p}(M = j)}. \tag{7}$$

With a uniform prior ($\mathsf{p}(M) = 1/N$) that indicates equal probability for all references, we can see that the posterior reduces to $\mathsf{p}(M|f_q) = w_{(i)}/\sum_j w_{(j)}$, since the constant of the prior factors out in the numerator and denominator.

If we now assume that our VPR technique is reasonable, and that the query position should be located at the 'true' reference, then we can express the expected query position, given our belief on the match of each reference, i.e.,

$$\mathbb{E}[p_{(M)}|f_q] = \sum_i \left[\mathsf{p}(M = i|f_q)p_{(i)}\right] \tag{8}$$
$$= \mu_p \tag{9}$$

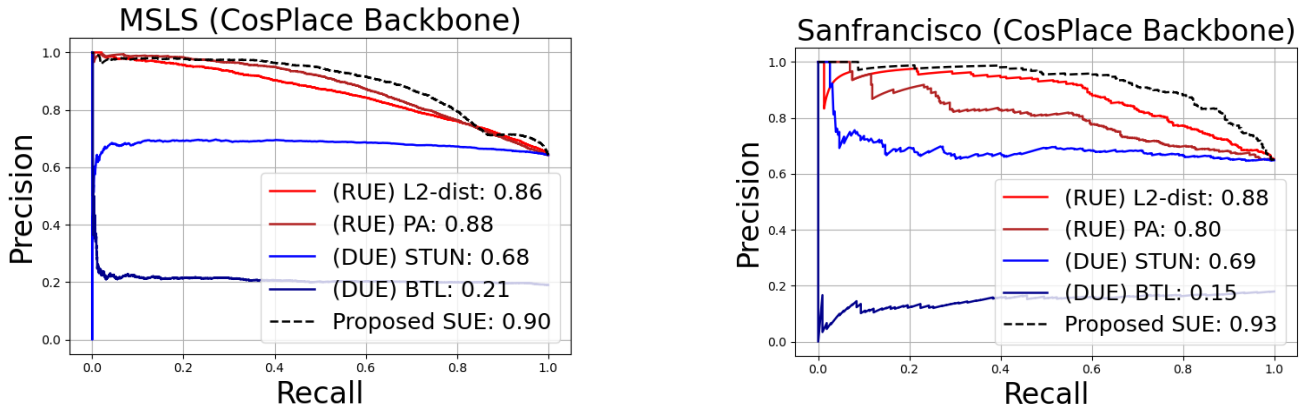


Figure 8. SUE remains SOTA by changing the backbone feature extractor to CosPlace [6] with no retuning of SUE's hyper-parameters. CosPlace is also used as the backbone for L2-distance and PA-score, but it was not possible to change the backbone for BTL and STUN.

Here we recognise Equation (1), assuming the uniform prior $\mathsf{p}(M)$. While we do not necessarily consider this expected pose to be representative of the true query pose (it could be an average location between distant visually-matching areas), it does allow us to compute the expected squared pose distance of the true match to the query,

$$\mathbb{E}\left[||p_{(M)} - \mu_p||_2 \middle| f_q\right] \approx \text{trace}(\Sigma_p) = s_q, \qquad (10)$$

where Σ_p is as defined in Equation (2) for the uniform prior $\mathsf{p}(M)$. In other words, in SUE s_q estimates the expected (squared) distance between the match's pose and the query pose, thus the smaller s_q the higher the chance is that a match selected according to our posterior belief is within an acceptable distance to the true query pose.

Finally, reference $i' = \text{argmax}_i\, \mathsf{p}(M = i|f_q)$ with the highest posterior probability of being the correct match is selected, which based on the likelihood term (and with uniform prior) will be $i' = 1$, i.e. the nearest neighbor in the feature space.

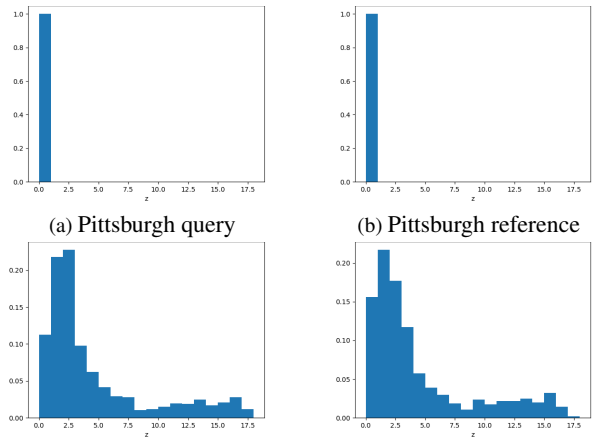
Note that in the above, a uniform prior $\mathsf{p}(M)$ means all references are assumed a-priori equally likely to match the query. In case some areas in the map contain more references than other areas, this also implies a higher prior belief that the query will occur in such a denser sampled area. This 'default' prior is therefore *not* a uniform *spatial* prior over the mapped area, but it assumes that the local spatial density of references in the map is indicative of the probability of a query appearing in such a local region.

6.3. Spatial density compensation for dissimilar query/reference spatial distributions

As explained in SUE's potential limitations of Discussion Section 4.5 and Appendix Section 6.2, the default formulation of SUE assumes that each reference is equally probable to match a query, i.e., a uniform prior $\mathsf{p}(M)$ is assumed. In other words, the query and reference images/poses are expected to be distributed similarly over the map, and the spatial density of the references in an area reflects the assumed prior probability for a query to be located in that area.

To illustrate, consider two perceptually-aliased locations A and B, where location A is represented by 100 images and location B by one image. If a query occurs at A or B, SUE's uncertainty estimate as currently formulated in Equation (2) will be low, since the many references at location A will all agree on low spatial variance, while the contribution of distant references at location B are 100× less. This high confidence could be desired if location A is also 100× more likely to be visited at query-time than location B (i.e. the uniform $\mathsf{p}(M)$ holds, so the spatial density of the references reflects a spatial prior of a query's location). However, this prior could also be undesired if we expect queries at A and B are equally likely to occur, irrespective of the reference density. Ultimately, what is desired depends on the application and data collection procedure.

In case the uniform prior $\mathsf{p}(M)$ over references is undesired, we can substitute it with a different prior in the equations of Section 6.2. Specifically, in Equation (7) the likelihood terms should *not* be multiplied with a constant prior term (which cancelled out in the numerator and denominator). Still, it may be more convenient to express the prior over references in terms of a *spatial prior for the query*. In other words, a reference would be more probable to match if it is in a area where the query is more probable to occur, while a reference would be less probable if there are more other references in the same spatial area. Let $\mathsf{p}_q(p)$ denote the desired spatial prior for the query to be at a pose p, and $\mathsf{p}_r(p)$ denote the spatial density of the references at a pose



(a) Pittsburgh query (b) Pittsburgh reference

(c) Stlucia query (d) Stlucia reference

Figure 9. The density of queries and references is depicted using the distance (z) of each query/ref to its nearest neighbour ($k = 1$) in the pose space. Queries and references in Pittsburgh dataset are highly dense and hence uniformly spatially distributed. The queries and references are non-uniformly (albeit similarly) spatially distributed in the sparser Stlucia dataset.

p, then

$$\mathsf{p}(M = i) \propto \frac{\mathsf{p}_q(p_{(i)})}{\mathsf{p}_r(p_{(i)})}. \tag{11}$$

We will refer to this as *spatial density compensation*. In practice, we can thus compensate SUE for a desired spatial prior by multiplying the reference weight $w_{(i)}$ with a term (proportional to) the desired prior $\mathsf{p}(M)$. Note from Equation (11) that if the spatial distributions of queries and references are assumed equal, we again obtain that $\mathsf{p}(M)$ is uniform, as is the case for the default SUE formulation.

6.4. Validating spatial density compensation

In this section, we test the spatial density compensation concept of adjusting SUE as explained in Section 6.3.

Applying a uniform spatial prior for the query Let's assume the spatial density of query poses is uniform, so all query poses within the map are equally likely, in which case term $\mathsf{p}_q(p)$ becomes a constant (and thus will cancel out when normalizing the weights).

The spatial density of the references $\mathsf{p}_r(p)$ can be estimated from the finite samples of poses in the reference set. We can for instance model the spatial density of references by simply taking the distance $z_{(i)}$ of the reference i to its k-th nearest neighbor in the *pose space*, such that the area $z_{(i)}^2$ is inversely proportional to the local density of the reference i, i.e., $\mathsf{p}_r(p_{(i)}) \propto 1/z_{(i)}^2$. Hyperparameter k regularizes the smoothness of the estimated reference pose density.

Compensation	Pitts.	San.	Stlu.	Eyns.	MSLS
none	0.94	0.84	0.89	0.93	0.76
$k = 1$	0.94	0.84	0.82	0.93	0.76
$k = 3$	0.94	0.84	0.84	0.93	0.77
$k = 10$	0.94	0.81	0.85	0.92	0.77

Table 5. SUE's AUC-PR with reference density compensation.

We can now see that $\mathsf{p}(M = i) \propto z_{(i)}^2$, thus the density compensated SUE for this uniform spatial prior for query poses is obtained by re-weighing Equation (3) with $z_{(i)}^2$, i.e.,

$$w_{(i)} = e^{-\lambda \cdot d_{(i)}} \cdot z_{(i)}^2. \tag{12}$$

Do common datasets have a uniform query distribution? We used the above formulation of spatial density to study the properties in the used VPR datasets. First, we find that most of our datasets *do* have a mostly uniform spatial distribution for both queries and references, except the Stlucia dataset. Fig. 9 illustrates the distribution of distances to the $k = 1$ nearest neighbors for the Pittsburgh and Stlucia datasets. Second, we can conclude that the assumption that references and queries have a similar spatial distribution *does hold* in common VPR dataset, hence SUE's default formulation with uniform reference prior is reasonable.

To properly validate the density compensation concept of Section 6.3, we also create a modified version of the Stlucia data such that queries and reference actually do have a *different* spatial distribution. We greadily subsample the Stlucia queries such that the spatial density of the resampled queries is uniform.

Does assuming a uniform query distribution help? Finally, we test the density compensated SUE of Equation (12) on the VPR datasets for different choices of k, see Table 5.

Since queries and references of datasets other than Stlucia are already uniformly distributed spatially, the table confirms that density compensation does not lead to any major effect on SUE's performance. We also see that for the (unmodified) Stlucia dataset, density compensation actually *hurts* performance because the queries and references are in fact *non*-uniformly and similarly distributed. The default uniform prior assumption of SUE is therefore better suited for Stlucia.

However, if we test density compensated SUE on the modified Stlucia dataset where queries are in fact uniformly spatially distributed while the references are not, then we do observe a benefit over the default SUE as shown in Table 6. In this case, the spatial prior of density compensated SUE does hold, where as the default SUE assumption that queries and references are similarly distributed does not.

z	none	k=1	k=3	k=5	k=8	k=10
$8-9$	0.92	**0.96**	0.96	0.96	0.94	0.94
$10-11$	0.68	**0.76**	0.73	0.7	0.71	0.69

Table 6. SUE's AUC-PR with reference density compensation using different values of k on the Stlucia dataset when the queries are resampled to have a close to uniform spatial density (e.g., $z = 8-9$). Reference density compensation helps SUE when queries are spatially uniformly distributed and references are non-uniformly distributed. Best across the columns is in Bold.

In conclusion, whether spatial density compensation is needed depends on the specifc spatial distributions of the references and queries in a dataset. For the studied VPR benchmark datasets that represent densely collected queries and references, the default assumption of SUE that their spatial distributions are similar holds. Still, in applications where we can expect that queries and references are distributed differently, then additional density compensation can be helpful. The formulation of spatial density compensation can be motivated from a probabilistic view on SUE. Future work can investigate better estimates for query and reference density for non-uniformly distributed data to further improve SUE.

6.5. Precision-Recall curves

In addition to the Precision-Recall curves of the Pittsburgh dataset in Fig. 1, the PR-curves for the remainder five datasets are shown in Fig. 10. SUE outperforms the methods in the *RUE* and *DUE* categories on all datasets. *GV* remains the overall state-of-the-art, albeit at a two to three orders of magnitude higher computational cost.

6.6. Complementing geometric verification

We further show in Fig. 11 the generalization of SVM trained on the Pittsburgh dataset to other datasets. For all these datasets, the relation of our SUE uncertainty with DELF-RANSAC leads to complementarity with queries in the bottom-left of the plot that can be linearly separated.

6.7. SUE combined with other uncertainty estimates

For completeness, we show the combination of other uncertainty estimation methods with SUE in Fig. 12. Most of the queries that can be classified as true- or false-positives by other methods can already be classified using only SUE. We hypothesize that this is because of SUE's similarity to BTL and STUN which also estimate the aleatoric uncertainty, and since SUE already uses the L2-distance and nearest neighbours in its uncertainty estimate.

6.8. Relating SUE to geometric burstiness

Relation: Features that appear in similar configurations across multiple unrelated reference images are referred to as geometric burstiness (GB) [40]. Ideally, such features should not be considered for estimating the image matching confidence using geometric verification (GV). Whether images are related or unrelated is determined using their pose information, i.e., different images that are physically close to each other could be looking at the same place. While the use of pose information of the Top-K retrieved reference images is common between SUE and GB, the latter is evaluated for image re-ranking and the former for VPR. *GB* is implemented on top of *GV* and is more computationally expensive than GV, concretely by an order of K, but gives better uncertainty estimates. For completeness, we implement a version of *GB* inspired by [40] and compare it to SUE. The implementation details are as follows.

Our implementation of GB: We use SIFT features, and perform feature matching in a RANSAC loop between a query and its Top-K retrieved nearest neighbors. Local feature matches $[q_i, r_j^k]$ that satisfy a geometric transform (homographic) are considered inliers, where q_i is the ith query feature and r_j^k is the jth feature in kth nearest neighbour. A query feature q_i contributes to geometric burstiness if it forms part of the inlier set for multiple (say T) retrieved images, and in the most naive case, such $[q_i, r_j^k]$ should be discarded from the inlier count. But similar to [40], we down-weight their contribution by T instead of completely discarding such inliers.

However, Sattler *et al.* [40] further studied that the top retrieved images could come from the same place, and hence query features could *legally* form part of the inlier set for multiple retrieved images. To classify whether a set of reference images represents the same place or not, we use the definition of place from [7] where images that are within 25 meters of each other are considered as the same place. Thus, only inliers from reference images of different (more than 25 meters apart) places are classified as geometric bursts. We use $K = 20$ and for feature matching the same hyperparameters are used as that of SIFT-RANSAC.

Results: We report in Table 7 that adding *GB* on top of SIFT-RANSAC leads to better performance than just using SIFT-RANSAC. Overall, among all uncertainty estimation methods, DELF-RANSAC still performs the best. *GB* is the most computationally expensive among all the uncertainty estimation methods. Note that *GB* could also be added on top of Superpoint-RANSAC and DELF-RANSAC albeit at an even higher computational cost.

We further test if SUE remains complementary to GB, given that both methods use reference poses. Fig. 13 shows that the uncertainty estimates from SUE can also complement *GB*. In conclusion, the several orders of magnitude

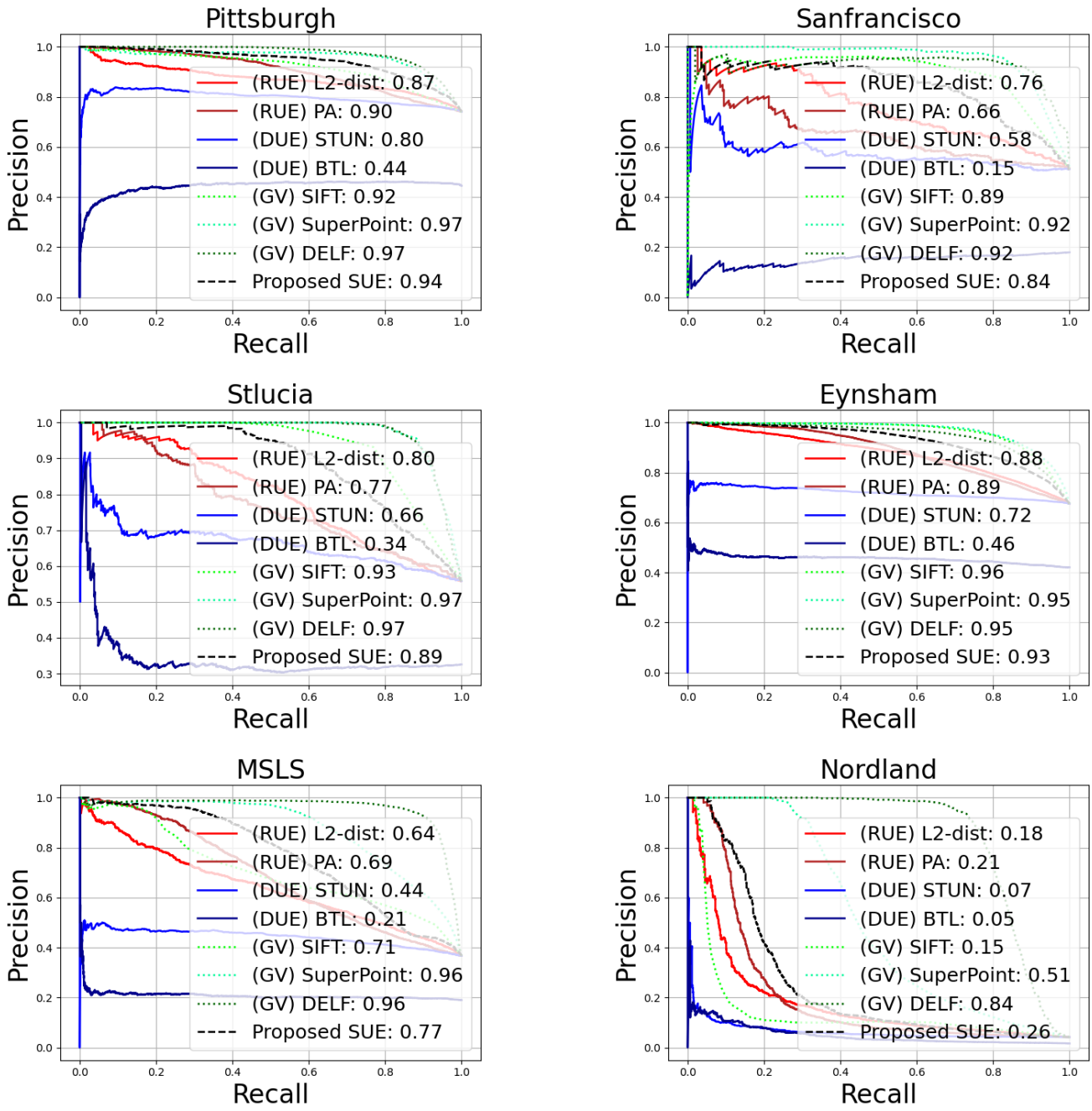


Figure 10. The precision-recall curves on the six datasets using SUE and other baselines. SUE outperforms the existing methods within the efficient category on all datasets. Note how an L2-based retrieval uncertainty outperforms the data-driven aleatoric uncertainty estimated in BTL and STUN.

higher computational needs of *GB* compared to SUE, and their mutual complementarity suggest that SUE is a useful baseline for uncertainty estimation in VPR.

6.9. Qualitative results

We show examples of queries with their corresponding nearest neighbors ranked with the uncertainties computed by the different types of uncertainty estimation methods in Fig. 14. We keep the set of randomly chosen queries the

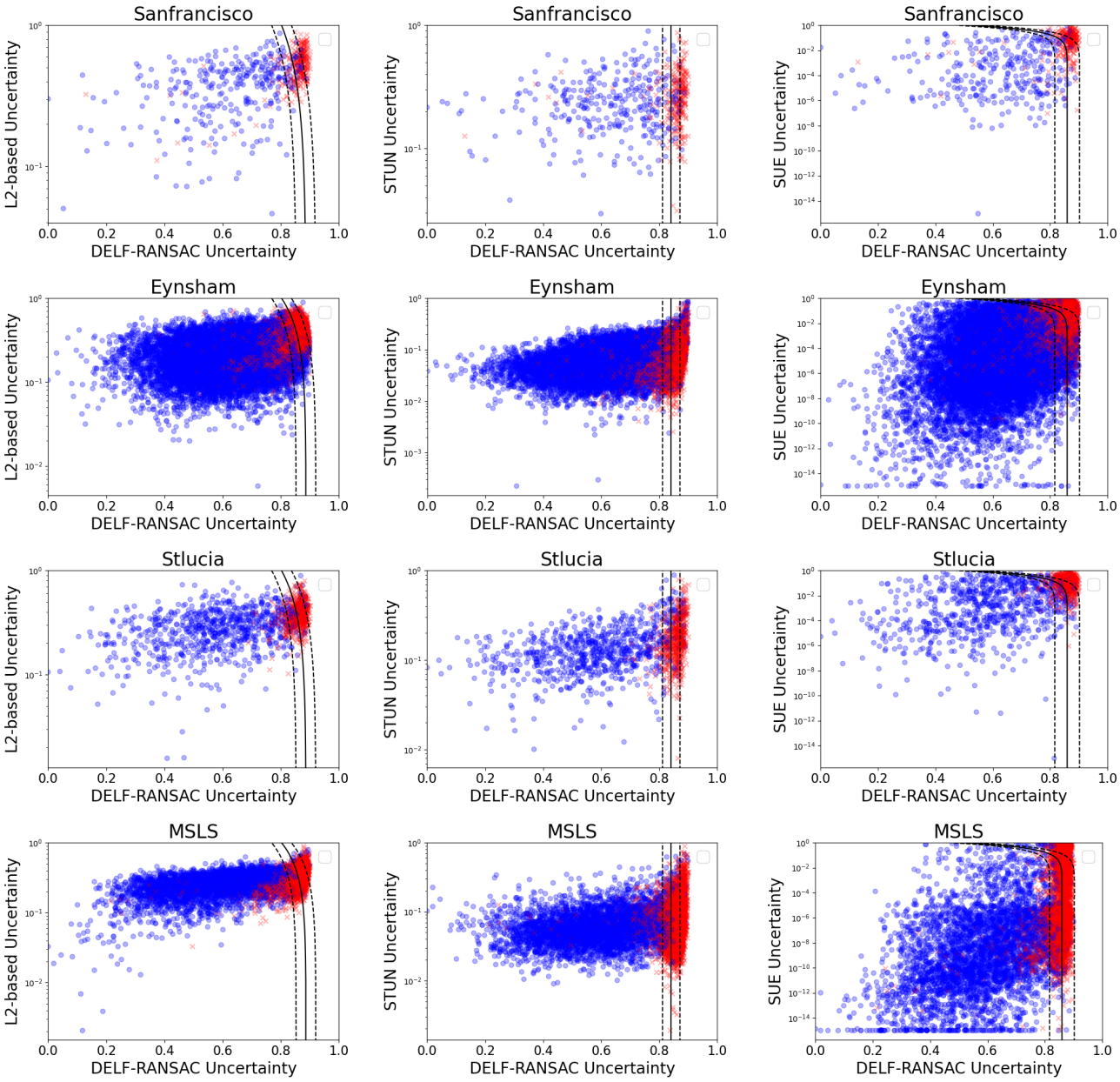


Figure 11. The relation of L2-based uncertainty, STUN, and SUE with geometric verification uncertainty. The SVM boundaries are learned on the Pittsburgh dataset only. Each point represents a query, and the color indicates whether it is a true-positive (Blue) or a false-positive (Red). The linear SVM boundaries are shown as black lines, while the dashed lines are the SVM margins. The combination of SUE with geometric-verification leads to more correctly matched queries in the bottom right (where SUE is certain but *GV* is uncertain) of the plots identifying complementarity. For better visualization, the vertical scale is in log-space, due to which the SVM boundaries appear non-linear to the reader but are linear.

same for all the methods. These examples further indicate what each method is sensitive to for uncertainty estimation.

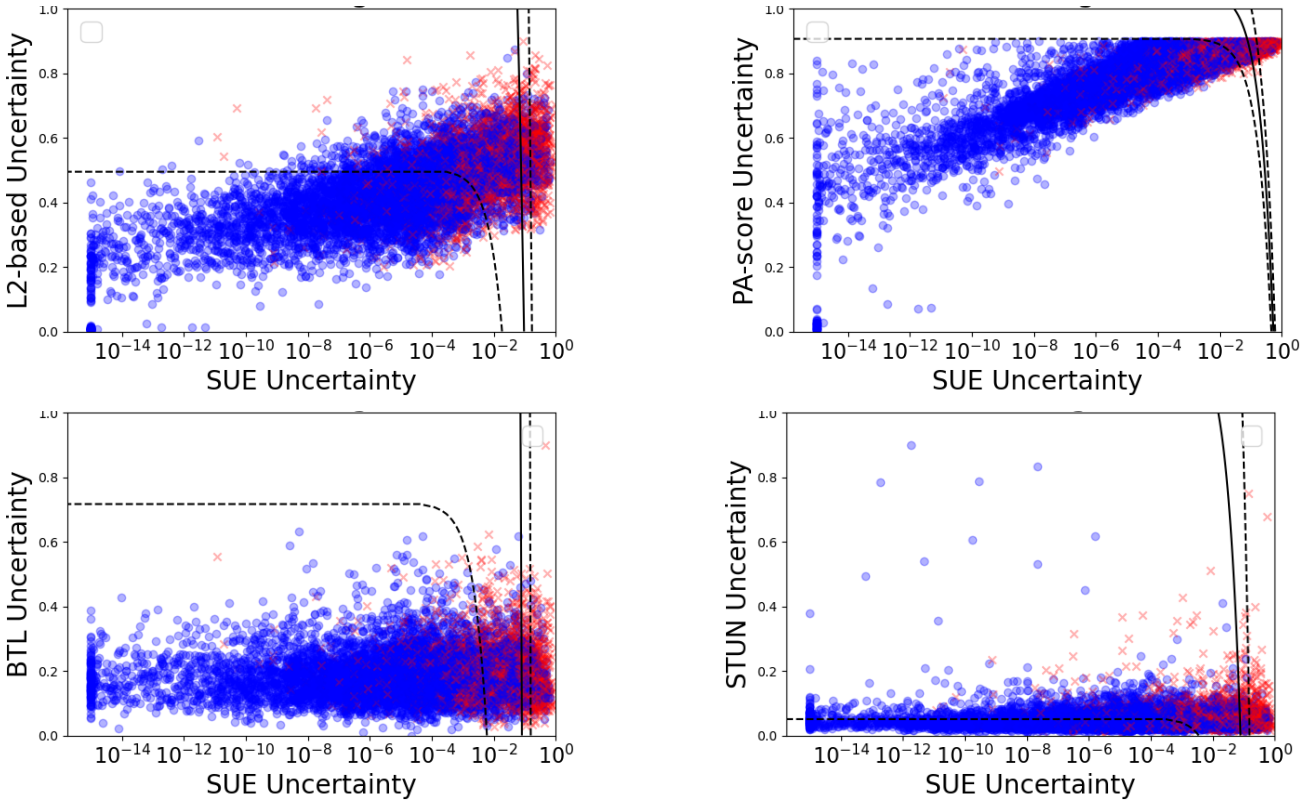


Figure 12. The relation of L2-based, PA-score, BTL, and STUN uncertainties with SUE uncertainty. Each point represents a query, and the color indicates whether it is a true-positive (Blue) or a false-positive (Red). The linear SVM boundaries are shown as black lines, while the dashed lines are the SVM margins. As indicated by the near-vertical decision boundaries, most of the queries that can be classified as true- or false-positives by other methods can also be classified by SUE, and we do not see much complementarity.

	↑ Pitts.	↑ Nord.	↑ MSLS	↓ Time
L2-dist	0.87	0.18	0.64	**0.05**
STUN	0.79	0.05	0.44	0.10
SUE	0.94	0.26	0.77	1.08
SIFT	0.92	0.15	0.70	129
DELF	**0.97**	**0.84**	**0.95**	1587
GB (SIFT)	0.92	0.31	0.87	2709

Table 7. AUR-PR and computation time (msecs) comparison of the methods discussed in the main paper with geometric burstiness [40]. Best across the columns is in Bold. Implementing *GB* on top of SIFT-RANSAC leads to better performance but at several orders of magnitude higher computational cost.

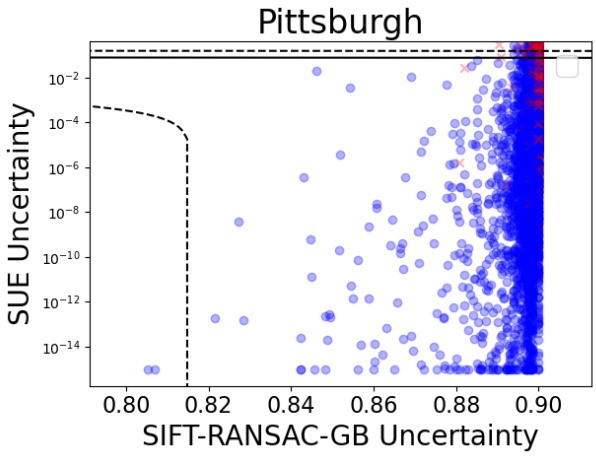


Figure 13. SUE remains complementary to *GB* since many true-positives can be separated from false-positives using SUE uncertainty and not using *GB*. See other such plots in this paper for details on the employed info-graphics.

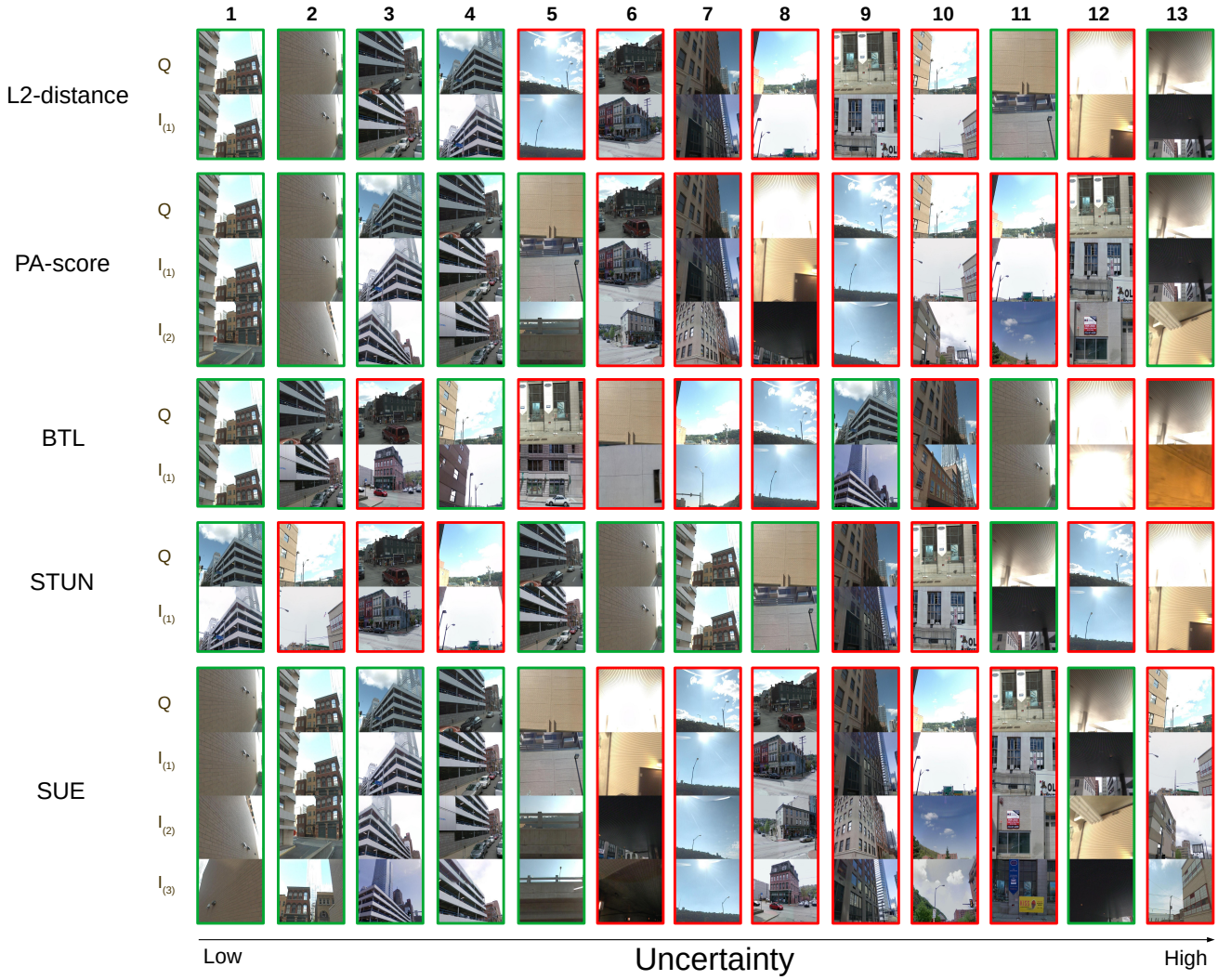


Figure 14. Exemplar matched/mismatched queries are ranked with different types of estimated uncertainties in the Pittsburgh dataset. Note that the set of chosen queries is the same for all types of uncertainty estimation methods. $I_{(n)}$ denotes the nearest neighbor where the subscript n denotes its rank. The number of nearest neighbors shown relates to the corresponding number needed by each method (e.g. PA-score requires two nearest neighbors). The retrieved nearest neighbors for BTL are different than other methods due to the different feature encoder. A good uncertainty estimation method when used for ordering would rank correct matches to the left and incorrect matches to the right of the reader. The query image in column 12 of SUE depicts the failure case of SUE, where the perceptually aliased nearest neighbors are geographically far-apart leading to high uncertainty but the best match is still the correct match.