

Efficient Implementation of Regional Mutual Information for the Registration of Road Images

Johann Lembach

Ronny Stricker

Horst-Michael Gross

*Neuroinformatics and Cognitive Robotics Lab
Technische Universität Ilmenau*

Ilmenau, Germany

{johann-uwe.lembach; ronny.stricker; horst-michael.gross}
@tu-ilmenau.de

Abstract—Motivated by recent advancements in the detection and classification of road distress, we aim to align road images from different years to facilitate automated change detection. We present a variable and efficient variant of the Regional Mutual Information (RMI) similarity metric to speed up the registration process while keeping the alignment robust and precise. We propose several modifications to the sampling process of RMI that allow an efficient implementation based on the resulting Hankel-type structures. We analyze the theoretical performance gains and the practical application of the proposed method in the road image domain. The achieved speed-ups significantly reduce the optimization duration while retaining the robust nature of RMI as a similarity measure.

Index Terms—regional mutual information, optimization, image alignment, road distress analysis

I. INTRODUCTION

Within the ASFaLT¹ project, deep learning techniques are used to detect and classify road distress on standardized road images. Motivated by recent advancements in the automated detection and classification of distress on road images [8], [26], we apply image registration to facilitate the automatic analysis of such images recorded over multiple years. To perform change detection and continuously assess the state of the road, robust alignment of images from different years is necessary. As the measurement vehicles cannot follow the exact same path in different years, the sequences typically exhibit misalignments across the years and need to be roughly aligned using GPS as a first step. Using the framework depicted in Fig. 1, we use two candidate images from a later year in a way that, when stitched together, they contain the corresponding road segment seen in the reference image and can be aligned to the reference image using an affine transformation, excluding the shear parameter.

This work has received funding from the Austrian Research Promotion Agency (FFG) as part of the ASFaLT project under project no. 869514.

¹ASFaLT: Automatisierte Schadstellenerkennung für unterschiedliche Fahrbahnbeläge mittels Deep Learning Techniken (Automated distress detection for varying road surfaces using deep learning techniques.)

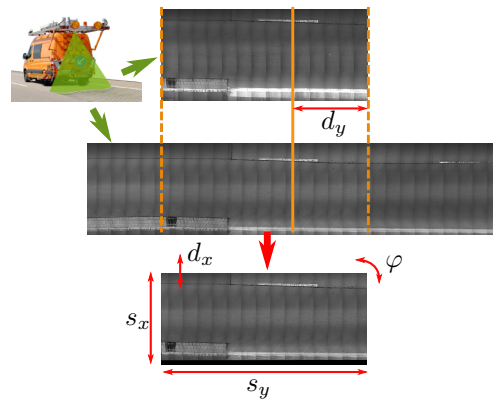


Fig. 1: The registration process can be described by applying five affine parameters to stitched candidate images: two translations d_x, d_y , two scale factors s_x, s_y and a rotation φ .

We present an adapted version of the regional mutual information (RMI) [22] similarity measure used for optimization based registration in medical imaging [4] and other fields like remote sensing [3] or aerial surveillance [1].

As non-differential optimization approaches generally have a high number of iterations, we need to make the proposed approach as efficient as possible.

Our main contributions are threefold:

- Extend RMI by subsampling methods
- Derive efficient implementation of subsampled RMI method using matrix shape and memoization
- Apply and evaluate RMI-based registration to road images

In the following sections, we will first introduce approaches to image registration. We will introduce mutual information (MI) as the basis, followed by regional mutual information (RMI) in II. Originating from an in-depth analysis of the RMI sampling process, we will present an efficient sampling method and implementation of RMI in III. An analysis of the proposed adaptation and its implementation follows in IV. In V, we will evaluate the method on a dataset of road surface

images.

II. PRIOR WORK

Image registration is an essential preprocessing step in many applications of medical imaging, remote sensing, change detection, and others. Its goal is to find a transformation between two images that transforms the floating image such that it aligns with the reference image.

Approaches to perform this task are generally divided into **feature-based** and **intensity-based** approaches. Both types have been studied extensively in the literature of different fields of computer vision [16], [33]. In feature-based approaches, one uses visual correspondences between two images, to directly compute a transformation [33]. In intensity-based methods, similarity metrics are calculated from pixel intensities, to find the optimal transformation iteratively [2], [16].

We decided to follow a mutual information-based approach as “intensity-based techniques [are] now forming the basis of the vast majority of registration approaches” and the “use of mutual information as a similarity measure has played a prominent role in this process” [28]. Non-differentiable optimization methods are generally used as, in most cases, it is infeasible to compute the full gradients of the cost function. For our tests, we selected Differential Evolution [25] as it offers a reasonable trade-off between speed and consistency, although it is not the focus of the research here.

The part of the process with the greatest research interest is the design of robust cost functions or similarity measures that lend themselves to iterative optimization. Of the similarity measures discussed in the literature, an ongoing debate is on information-theoretic measures, especially mutual information-based measures [18].

In the next section we will introduce mutual information as a similarity measure as well as a few of its many variants.

A. Mutual information

Initially, Viola and Wells, and Collignon et al. independently introduced the information theory based measure of mutual information (MI) as a criterion for multi-modal image registration in [5], [29] and subsequent papers [14], [15]. An overview and thorough introduction of mutual information-based image registration is given in [18].

The mutual information (MI) $I(X, Y)$ between two random variables X, Y with density functions $p_X(x)$, $p_Y(y)$ and joint density function $p_{XY}(x, y)$ is defined as

$$I(X, Y) = h(X) + h(Y) - h(X, Y) \quad (1)$$

using the entropies $h(X) = -\sum_x p_X(x) \log p_X(x)$, $h(Y)$ correspondingly, and conditional entropy $h(X, Y) = -\sum_{x,y} p_{XY}(x, y) \log p_{XY}(x, y)$ [15].

As an information-theoretic similarity measure, the idea is that with increasing overlap/similarity knowing something about the first image decreases the uncertainty about the second image. This corresponds to the conditional entropy term decreasing while the individual entropies stay constant. Hence

the goal is to maximize MI during a registration process. MI is primarily adopted and used in the field of medical imaging [24], [27], but also adapted by a number of researchers for various other tasks [3], [7], [9], [19], [23].

Although there is a variety of approaches that try to incorporate some kind of localized features, like [12], [13], [17], [24], [30], neither MI nor most variants use regional information.

B. Regional mutual information

Building on Rueckert’s second-order extension to MI [21], both Kybic [11] and Russakoff et al. [22] introduced versions of regional mutual information (RMI) that sample a window around every pixel to create a sample matrix that is subsequently used to estimate high-dimensional entropy. Both steps are introduced below.

1) *Extending MI to higher dimensions:* To incorporate regional information Kybic and Russakoff create a high-dimensional sampling matrix from the pixels of the image. At each pixel, a rectangular region with radius r is sampled and vectorized into d^2 dimensional column vectors $\mathbf{s}^{(i)} = [s_0^{(i)}, s_1^{(i)}, \dots, s_{d^2-1}^{(i)}]^T$ where $d = 2r + 1$. As visualized in Fig. 2, the vectors are concatenated horizontally to create the sampling matrix $\mathbf{S}_X = [\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots] \in \mathbb{R}^{d^2 \times N}$ of the reference image and $\mathbf{S}_Y \in \mathbb{R}^{d^2 \times N}$ of the floating image. Considering the edge cases on both sides the number of samples $N = (H - 2r) \cdot (W - 2r)$ with W and H being the width and height of the images. The two resulting ma-

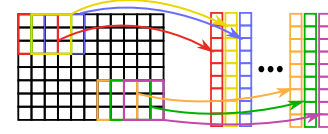


Fig. 2: Sampling matrix: vectorized regions are stacked to create the sampling matrix.

trices are concatenated vertically to form a sampling matrix $\mathbf{S} = [\mathbf{S}_X^T \ \mathbf{S}_Y^T]^T \in \mathbb{R}^{2d^2 \times N}$. With regards to creating \mathbf{S} , Kybic and Russakoff coincide. It is at the step of entropy estimation that they diverge from each other.

2) *Entropy estimation:* Whereas Kybic [11] uses the Kozachenko-Leonenko estimator [10] based on nearest-neighbors, Russakoff [22] uses a number of simplifying assumptions to estimate the entropy of the high-dimensional sample matrix.

In practice he subtracts the mean along each row of \mathbf{S} from \mathbf{S} so as to obtain a matrix $\tilde{\mathbf{S}}$ with points centered at the origin before computing the covariance matrix as

$$\mathbf{C} = \frac{1}{N} \tilde{\mathbf{S}} \tilde{\mathbf{S}}^T = \frac{1}{N} \begin{bmatrix} \tilde{\mathbf{S}}_X \\ \tilde{\mathbf{S}}_Y \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{S}}_X^T & \tilde{\mathbf{S}}_Y^T \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{XX} & \mathbf{C}_{XY} \\ \mathbf{C}_{XY} & \mathbf{C}_{YY} \end{bmatrix} \quad (2)$$

Using $\mathbf{C}_{XX}, \mathbf{C}_{YY} \in \mathbb{R}^{d^2 \times d^2}$ and $\mathbf{C} \in \mathbb{R}^{2d^2 \times 2d^2}$ he can estimate the entropies. Using the entropy of a normally distributed set of points in \mathbb{R}^n .

$$H_g(\Sigma_n) = \log \left((2\pi e)^{\frac{n}{2}} |\Sigma_n|^{\frac{1}{2}} \right) \quad (3)$$

where Σ_n is the covariance matrix of a n dimensional signal matrix and $|\cdot|$ is the determinant [6]. The joint entropy $H_g(\mathbf{C})$ is estimated from the entire covariance matrix \mathbf{C} . The marginal entropies $H_g(\mathbf{C}_{XX})$ and $H_g(\mathbf{C}_{YY})$ are computed from the top-left and bottom-right parts of the covariance matrix corresponding to the respective images.

The regional mutual information is finally computed as $I_r(X, Y) = H_g(\mathbf{C}_{XX}) + H_g(\mathbf{C}_{YY}) - H_g(\mathbf{C})$. A number of simplifications are available in the literature. [31] simplifies RMI by using the average of the window as the single value to compute a mutual information. [4], [20] use the principal component analysis to reduce the computational burden of RMI for multi-modal image registration.

III. EFFICIENT IMPLEMENTATION

We found that a large part of the complexity of RMI goes into creating the sample matrix, and hence propose a variable sampling method that lends itself to an efficient implementation.

A. Variable region extraction

In most papers using RMI, the regions are sampled densely in two ways, i.e., at **every pixel location** and **every pixel in the region** is used. The only parameter to vary the sampling process is the radius r of the region, leading to an increase in the number of rows of the sample-matrix (and a decrease in columns because of edge-case handling). Russakoff already analyzed the radius as a trade-off between the independence across samples and the “curse of dimensionality” for the computation of the covariance matrix [22].

A variant that uses a dilation factor to reduce the amount of necessary memory has been used as a loss for semantic segmentation in [32]. We propose a similar idea, but further increase the efficiency and variability while keeping the performance as a robust similarity measure high.

Introducing the option of reducing the interval at which the regions are sampled, and the step at which the pixels are sampled within a region can reduce the size of the sample matrix. We suggest using the sample step s_s and radius step r_s additionally to the radius r as parameters for RMI. Fig. 3

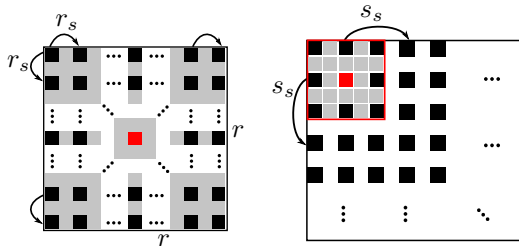


Fig. 3: Left: Subsampling within a region relating parameters r and r_s . Right: Example with regions ($r = 2, r_s = 2$) sampled at sample step $s_s = 4$.

illustrates both parts of the subsampling. For a given radius r , the radius step r_s specifies the stride with which pixels are sampled from a region, giving us $d = \frac{2r}{r_s} + 1$ as the effective

diameter. This parameter allows us to use large radii, while at the same time keeping the dimension of the sample matrix low.

The parameter of the sample step s_s allows us to sample only every s_s 'th region in an image. This reduces the number of samples in the sample matrix by a factor of s_s^2 leaving us with $N = \frac{(W-2r)(H-2r)}{s_s^2}$.

Depending on the settings, the sparse sampling dramatically reduces the computational burden, without a significant influence on the overall performance of RMI, which we will analyze in sections IV and V.

B. Exploitation of Hankel-type matrix structure

We can further reduce the computational burden by using the Hankel-type matrix structure that results from the sparse sampling. Due to the repetitive nature of the sampling, there are inherent, self-repeating patterns within the sample matrix. For the following observations we suppose that the indices of the pixels of an image are (m, n) for row $m \in 0, 1, \dots, H-1$ and column $n \in 0, 1, \dots, W-1$.

Looking only at the indices of the pixels that make up the sample matrix \mathbf{S}_X with respect to the input image (top part of \mathbf{S}) for the simple case $r = r_s = s_s = 1$ we get

$$\mathbf{S}_X^{(1,1,1)} = \begin{bmatrix} (0,0) & (0,1) & \dots & (1,0) & (1,1) \\ (0,1) & (0,2) & \dots & (1,1) & (1,2) & \dots \\ (0,2) & (0,3) & \dots & (1,2) & (1,3) & \dots \\ (1,0) & (1,1) & \dots & (2,0) & (2,1) \\ (1,1) & (1,2) & \dots & (2,1) & (2,2) & \dots \\ (1,2) & (1,3) & \dots & (2,2) & (2,3) & \dots \\ (2,0) & (2,1) & \dots & (3,0) & (3,1) \\ (2,1) & (2,2) & \dots & (3,1) & (3,2) & \dots \\ (2,2) & (2,3) & \dots & (3,2) & (3,3) \end{bmatrix} \quad (4)$$

The highlighted blocks contain the same pixels from the first row of the original image and are arranged in a Hankel-type pattern. A Hankel matrix is a square matrix where ascending diagonals contain the same value.

We can write each block \mathbf{i} according to the row i it originated from and thus we can re-write \mathbf{S}_X in terms of such block matrices.

$$\mathbf{i} = \begin{bmatrix} (i,0) & (i,1) & (i,2) & \dots \\ (i,1) & (i,2) & (i,3) & \dots \\ (i,2) & (i,3) & (i,4) & \dots \end{bmatrix} \rightarrow \mathbf{S}_X^{(1,1,1)} = \begin{bmatrix} 0 & 1 & 2 & \dots \\ 1 & 2 & 3 & \dots \\ 2 & 3 & 4 & \dots \end{bmatrix} \quad (5)$$

This is best summarized as sub-matrices that are Hankel-type matrices and that themselves are arranged as Hankel-type blocks. Depending on the combination of r, r_s and s_s the respective layout of the blocks varies. For an image $\mathbf{X} \in \mathbb{R}^{H \times W}$ this can be condensed as

$$\mathbf{S}_X^{(r,r_s,s_s)} = \begin{bmatrix} 0r_s + 0s_s & 0r_s + 1s_s & \dots & H - 2r - 1 \\ 1r_s + 0s_s & 1r_s + 1s_s & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 2r + 0s_s & 2r + 1s_s & \dots & H - 1 \end{bmatrix} \quad (6)$$

with

$$\mathbf{i} = \begin{bmatrix} (i, 0r_s + 0s_s) & (i, 0r_s + 1s_s) & \dots & (i, W - 2r - 1) \\ (i, 1r_s + 0s_s) & (i, 1r_s + 1s_s) & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ (i, 2r + 0s_s) & (i, 2r + 1s_s) & \dots & (i, W - 1) \end{bmatrix} \quad (7)$$

It can be seen that $\mathbf{i} \in \mathbb{R}^{d \times \frac{W-2r}{s_s}}$ and as $\mathbf{S}_{X^{(r, r_s, s_s)}}$ contains $d \times \frac{H-2r}{s_s}$ such matrices we get that $\mathbf{S}_{X^{(r, r_s, s_s)}} \in \mathbb{R}^{d^2 \times \frac{(W-2r)(H-2r)}{s_s^2}}$, where $d = \frac{2r}{r_s} + 1$.

Ending the submatrices at $W - 1$ and $H - 1$ constitutes an ideal case. In practice the last indices might deviate depending on the size of the image.

C. Memoization of submatrices

The above findings greatly simplify the construction of the sample matrix by precomputing the needed submatrices only once. It can be seen that for most combinations of settings, it suffices to only compute a portion of all possible submatrices. Generally, we only have to compute every n 'th submatrix where $n = \min(r_s, s_s)$. As the smaller of the two values will determine the step at which submatrices are used in the sample matrix, it suffices to compute only those. The submatrices can be precomputed and stored in, e.g., a hash-table. This way, they can be efficiently accessed and stacked into the final sample matrix.

Following we will refer to this variant as Memoized Sampling RMI (MS-RMI).

There are a few limitations to how the parameters can be chosen. The radius step has to be smaller than the radius, i.e., $r_s \leq r$, and both should be even to ensure an uneven effective diameter d . Additionally, r_s should be an integer factor of r such that the region lies over a central pixel.

IV. ANALYSIS OF EFFICIENT RMI

A. Effect of subsampling

As a first evaluation, we will compare the “full” sampling to our sparse subsampling with regular matrix computation (i.e. without memoization). For a set radius, we compute the sample matrix iterating over all pixels and selecting all pixels in the area, and compare that to different settings of r_s and s_s for the same radius. The left part of table I shows the cumulative duration of the sampling process over 500 iterations for a test image (400×200 pixels). We performed the experiments with two different radii and 3 to 4 different settings for r_s and s_s , doubling the parameters for each configuration. All experiments are run on a 8-Core Intel i7-2600 CPU (16GB RAM) using Python 3.7.7 with Scipy and Numba where possible.

The results, roughly follow the expected theoretic gain for changes of s_s , where a doubling leads to approximately a fourth of the number of samples in the sample matrix. This means only a fourth of the samples have to be accesses and stored in the sample matrix. As table I shows, this translates nicely to a reduction of the sampling duration by a similar

factor.

The speedups achieved by increasing the radius step are visible, though not as extensive. The decreased dimensionality has a larger influence on the following computations in RMI. Comparable settings between different radii, seem to give an edge to the larger radius. Comparable in this sense means that the settings result in roughly the same dimensions for the sample matrix (e.g. $r=4, r_s=1$ compared to $r=8, r_s=2$). It is usually a little faster to use a larger radius with larger r_s , even though d is equal but N is slightly smaller due to edge cases.

TABLE I: Duration (in [s]) for 500 iterations of basic sampling (RMI) and memoized sampling (MS-RMI) for random test images.

Radius r	Radius step r_s	RMI				MS-RMI			
		Sample step s_s				Sample step s_s			
		1	2	4	8	1	2	4	8
2	1	54.06	12.22	3.20	0.82	7.01	2.16	0.68	0.54
	2	47.25	11.84	2.63	0.64	2.95	0.56	0.29	0.15
4	1	128.60	24.27	4.77	0.88	40.57	5.42	1.49	0.58
	2	61.3	13.50	3.18	0.91	6.70	1.73	0.47	0.26
	4	44.22	9.35	2.61	0.69	2.57	0.59	0.16	0.10
8	1	417.22	89.36	17.48	4.97	106.17	31.73	4.86	1.49
	2	104.62	21.67	4.37	0.99	29.19	4.14	1.14	0.38
	4	49.04	12.60	4.49	1.04	5.79	1.20	0.29	0.15
	8	38.42	8.68	2.00	0.46	2.28	0.47	0.14	0.06

B. Effect of memoization

Next, we will analyze the performance gains introduced by pre-computing and memoizing submatrices. We showed that only a portion of submatrices need to be extracted and can be repeatedly used in the construction of the final sample matrix. We compare the duration of the matrix construction between RMI and MS-RMI. Comparing the two sides in table I, we can see clear speed gains for random images. Table II additionally shows the factors of speed gained for varying sizes of real images for a fixed radius $r = 8$ and step size $s_s = 16$. Interestingly the factors increase for larger images and increasing radius step. This nicely shows the potential of the memoized sampling.

Naturally, there is a trade-off between the complexity we achieve and the error we introduce by subsampling. There are many ways to analyze this trade-off. We will use the

TABLE II: Average duration (in [s]) of matrix creation for real images with varying size for set radius $r = 8$ and step size $s_s = 16$.

Width	Radius step r_s	Method		Factor \approx
		RMI	MS-RMI	
512	2	0.00341	0.00218	1.564
	4	0.00231	0.00078	2.961
	8	0.00187	0.00033	5.666
1024	2	0.01573	0.00799	1.968
	4	0.00984	0.00263	3.741
	8	0.00773	0.00114	6.780
1536	2	0.03782	0.01673	2.260
	4	0.02334	0.00577	4.045
	8	0.01710	0.00213	8.028

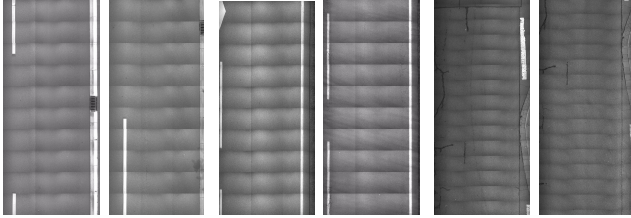
application in the road image domain to judge the overall performance and robustness as a similarity measure.

V. APPLICATION TO THE ROAD DOMAIN

To analyze the performance in the road domain, we need to introduce the data available as well as the metrics used to evaluate different settings of MS-RMI.

A. Image data

As mentioned above, our domain is 10m surface images that are regularly captured using standardized measurement vehicles. Depending on the vehicle, the images have different resolutions and exhibit various artifacts. To cover a wide range of possible applications, our dataset for testing the approach on real data comprises of data from different vehicles and different road types. We have a dataset of 200 labeled images from three different sub-domains, highway (60), federal (96), and inner-city (44) roads. An example for each road type is given in Fig. 4. Federal and inner city roads generally contain more unambiguous areas making the task somewhat easier. To correct for the different capture devices, we normalize the pixels of the road to achieve similar brightness levels across all images. To achieve this, we select only pixels that are likely to lie on the surface (using the peak of the histogram) to estimate the mean and standard deviation, and use those values to normalize the entire image.



(a) Highway (b) Federal road (c) Inner city

Fig. 4: Example images for different types of road domains.

B. Evaluation of registration quality

As suggested by Legg [12] we compute the registration error as the average distance between grid points transformed with the ground truth and transformed with the found transformation parameters. All the images are labeled using manual correspondences. Whereas Legg only uses the four corner points, we generate a grid of points and average across the individual distances $d_{\tau,i} = |T(\mathbf{p}_i) - \tau(\mathbf{p}_i)|$ between the points \mathbf{p}_i transformed with the labeled $T(\cdot)$ and found transform $\tau(\cdot)$ giving us $D_\tau = \frac{1}{n} \sum_i d_{\tau,i}$. To account for different scaling of the images, we always scale the parameters and grid points as if the images had a width of 1000 pixels, to keep results across experiments comparable.

In addition, we compute a performance measure based on the number transformations with a registration error below a threshold $n_t = |\{\tau | D_\tau \leq t\}|$. We sum these for a series of thresholds $t \in \{0, 5, 10, \dots, 50\}$ and divide by the number of total images in the sequence S to get a performance score $r = \frac{1}{S} \sum_t n_t$. We selected the above thresholds as they are

the ones that can be considered “good” in terms of visual acceptance. The performance score s lies between 1 and 0, with larger values corresponding to better performance.

C. Results

As a baseline, we performed regular RMI with $r = 2$, as Russakoff et al. “found that even a radius as low as $r = 2$ works quite well in practice” [22]. We achieved $s = 0.489$ which indicates that RMI itself is a robust similarity measure for our scenario, compared, e.g., to $s = 0.378$ which we achieved with mean squared error as a similarity measure (both with $W = 512$). It can be assumed that the score improves for larger radii, however the time needed for the optimization would increase dramatically as shown in the previous section. This is why we evaluated a set of parameters ($r \in \{2, 4, 8, 12, 16\}$, $r_s \in \{2, 4, 8\}$, $s_s \in \{4, 8, 16, 24, 32\}$) that offer a reasonable trade-off between speed and performance. We achieved such speeds that a lot of the combinations could also be tested on images with twice the size ($W = 1024$). The results are presented in Fig. 5 with s_s increasing along the x-axis and r_s for lower figures. Generally we achieved very good scores for $r \geq 4$ with a tendency for larger radii to perform better. The trade-off between speed and performance

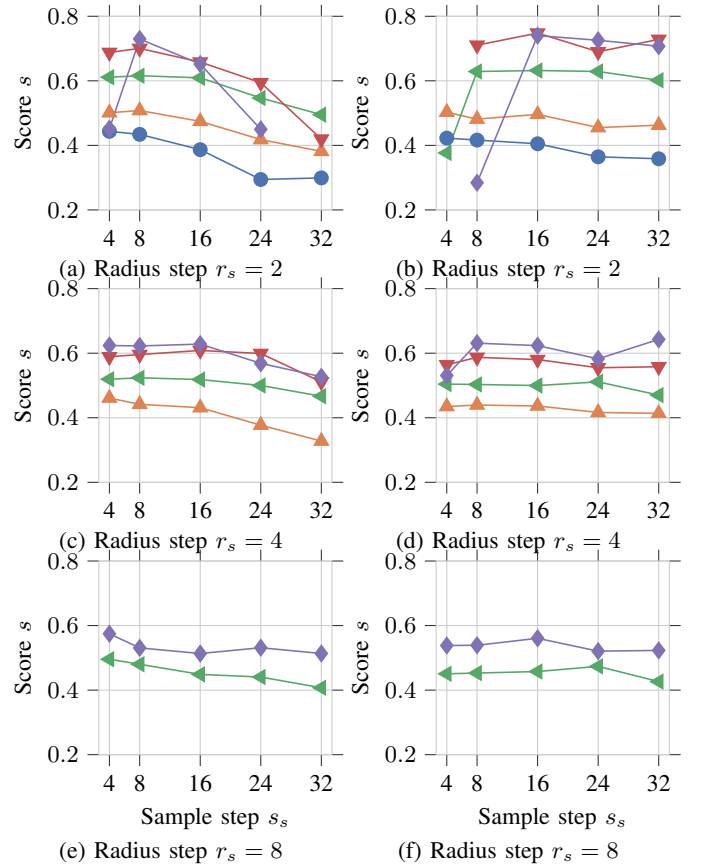


Fig. 5: Average performance s (averaged over all images) for different settings of s_s and r_s with $W = 512$ (Figs. 5a, 5c and 5e) and $W = 1024$ (Figs. 5b, 5d and 5f) pixels.

Radius: ● = 2, ▲ = 4, ◀ = 8, ▶ = 12, ◆ = 16

when increasing the radius step is quite obvious, with larger r_s performing worse than when more pixels of a region are used. This is what is expected, but interestingly for most radii, larger sampling steps do not have a very large (negative) influence. This means that we can easily increase the speed without sacrificing performance. This is especially true for larger r_s and the larger images. Hence, careful choices must be made with regard to the necessary accuracy of an application. For some of the smaller sample steps, the set time limit stopped the optimization, which is why not all results are available. One other case that did not work is $r = 16, r_s = 2, s_s = 32$ where we fall victim to the curse of dimensionality. Compared to the baseline, we achieve significantly better results at lower computational costs for a number of larger radii (up to $s = 0.730$ for $W = 512$) and at similar cost for the larger images (up to $s = 0.748$ for $W = 1024$).

VI. CONCLUSION

Using a variable sampling process and the resulting Hankel-type structure of the sampling matrix, we introduced Memoized Sampling RMI as an efficient, yet robust implementation of RMI. We analyzed its performance with respect to the original implementation by Russakoff et al. [22]. Our proposed algorithm can speed up the computations drastically, while containing the accuracy and robustness in a practical scenario. Aligning road images, we achieved higher registration scores ($s = 0.748$) for a number of configurations while reducing the computational cost compared to the original RMI method ($s = 0.489$). The approach is a general adaptation and could also be combined with different image features. Hence it offers great variability, while speeding up the computation of a similarity measure.

REFERENCES

- [1] Mojtaba Asgarizadeh and Hossein Pourghassem. A robust object tracking synthetic structure using regional mutual information and edge correlation-based tracking algorithm in aerial surveillance application. *Signal, Image and Video Processing*, 9(1):175–189, 2015.
- [2] Lisa G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.
- [3] H. M. Chen, M. K. Arora, and P. K. Varshney. Mutual information-based image registration for remote sensing data. *IJRS*, 24(18):3701–3706, 2003.
- [4] Yen Wei Chen and Chen Lun Lin. PCA based regional mutual information for robust medical image registration. *LNCS*, 6677(3):355–362, 2011.
- [5] A Collignon, F Maes, D Delaere, D Vandermeulen, P Suetens, and G Marchal. Automated multi-modality image registration based on information theory. *IPMI*, 3(6):263–274, 1995.
- [6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. 1991.
- [7] Yunfei Du, Haifang Zhou, Panfeng Wang, Xuejun Yang, and Hengzhu Liu. A parallel mutual information based image registration algorithm for applications in remote sensing. *LNCS*, 4330:464–473, 2006.
- [8] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst Michael Gross. How to get pavement distress detection ready for deep learning? A systematic approach. *IJCNN*, pages 2039–2047, 2017.
- [9] Reham Gharbia, Sara A. Ahmed, and Abdoul ella Hassanien. Remote sensing image registration based on particle swarm optimization and mutual information. In *INDIA*, volume 340, 2015.
- [10] L. F. Kozachenko and N. N. Leonenko. Sample Estimate of the Entropy of a Random Vector. *Probl. Peredachi Inf.*, 23(2):9–16 (in Russian), 1987.
- [11] Jan Kybic. High-dimensional mutual information estimation for image registration. *ICIP*, 3:1779–1782, 2004.
- [12] Philip A. Legg, Paul L. Rosin, David Marshall, and James E. Morgan. A robust solution to multi-modal image registration by combining mutual information with multi-scale derivatives. *LNCS*, 5761(1):616–623, 2009.
- [13] Xiaoping Liu, Shuli Chen, Li Zhuo, Jun Li, and Kangning Huang. Multi-sensor image registration by combining local self-similarity matching and mutual information. *Frontiers of Earth Science*, 12(4):779–790, 2018.
- [14] F. Maes, D. Loeckx, D. Vandermeulen, and P. Suetens. Image registration using mutual information. *Proceedings of the IEEE*, 91(10):1699–1722, 2003.
- [15] Frederik Maes, André Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality Image Registration by Maximization of Mutual Information. *IEEE T-MI*, 16(2):187–198, 1997.
- [16] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [17] Josien P.W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Image Registration by Maximization of Combined Mutual Information and Gradient Information. *IEEE T-MI*, 9(8):809–814, 2000.
- [18] Josien P.W. Pluim, J. B.A. Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE T-MI*, 22(8):986–1004, 2003.
- [19] Chunmei Qing, Jianmin Jiang, and Zhijing Yang. Normalized Co-Occurrence Mutual Information for Facial Pose Detection Inside Videos. *TCSVT*, 20(12):1898–1902, 2010.
- [20] Parminder Singh Reel, Laurence S. Dooley, and Patrick Wong. Efficient image registration using fast principal component analysis. *ICIP*, pages 1661–1664, 2012.
- [21] D. Rueckert, M. J. Clarkson, D. L. G. Hill, and D. J. Hawkes. Non-rigid registration using higher-order mutual information. In *Medical Imaging 2000: Image Processing*, volume 3979, pages 438–447, 2000.
- [22] Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, and Calvin R. Maurer. Image similarity using mutual information of regions. *LNCS*, 3023:596–607, 2004.
- [23] Kumar Shaurya Shankar and Nathan Michael. Robust direct visual odometry using mutual information. *SSRR*, pages 9–14, 2016.
- [24] Marius Staring, Stefan Klein, Max A. Viergever, Josien P.W. Pluim, and Uulke A. van der Heide. Registration of cervical MRI using multifeature mutual information. *IEEE T-MI*, 28(9):1412–1421, 2009.
- [25] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.
- [26] Ronny Stricker, Markus Eisenbach, Maximilian Sesselmann, Klaus Debes, and Horst Michael Gross. Improving Visual Road Condition Assessment by Extensive Experiments on the Extended GAPs Dataset. *IJCNN*, 2019-July(July):1–8, 2019.
- [27] C. Studholme, D. L.G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999.
- [28] Max A. Viergever, J. B. Antoine Maintz, Stefan Klein, Keelin Murphy, Marius Staring, and Josien P.W. Pluim. A survey of medical image registration – under review. *Medical Image Analysis*, 33(2016):140–144, 2016.
- [29] Paul Viola and William M. Wells. Alignment by Maximization of Mutual Information. In *ICCV*, pages 16–23, 1995.
- [30] Jonghye Woo, Maureen Stone, and Jerry L. Prince. Multimodal registration via mutual information incorporating geometric and spatial context. *IEEE Transactions on Image Processing*, 24(2):757–759, 2015.
- [31] Chunlan Yang, Tianzi Jiang, Jianzhe Wang, and Lian Zheng. A neighborhood incorporated method in image registration. *LNCS*, 4091 LNCS:244–251, 2006.
- [32] Shuai Zhao, Yang Wang, Zheng Yang, and Deng Cai. Region Mutual Information Loss for Semantic Segmentation. In *NeurIPS*, volume 33, pages 1 – 11, 2019.
- [33] Barbara Zitová and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21:977–1000, 2003.