# Progressive Filtering for Feature Matching

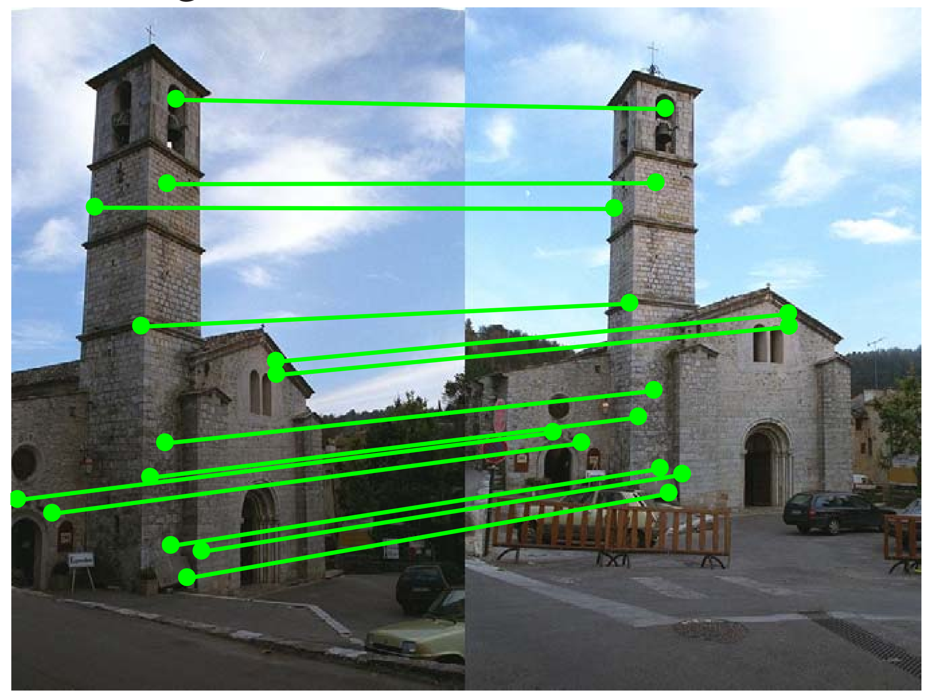
Xingyu Jiang[1], Jiayi Ma[1,*], Jun Chen[2]

[1]Wuhan University, Wuhan, China. [2]China University of Geosciences, Wuhan, China

ICASSP 2019

## Problem Definition and Contribution

**Goal:** Establish correct feature correspondences between two images of the similar or same scene.



**Motivations:**

- Existing parametric matching methods require a pre-defined transformation model and may fail in nonrigid situations, such as RANSAC and its varietas.
- Existing non-parametric matching methods require high computational cost and are easily affected by the noise, outlier and unknown image transformation.

**Key Contributions:**

- A **simple** but **efficient** method for feature matching.
- The **convolution operation** in our method may provide a guide to address the feature matching problem with **deep learning techniques** in future.
- The gridding strategy enables our method to achieve **linear time complexity**, which only requires dozens of milliseconds for thousands of matches.

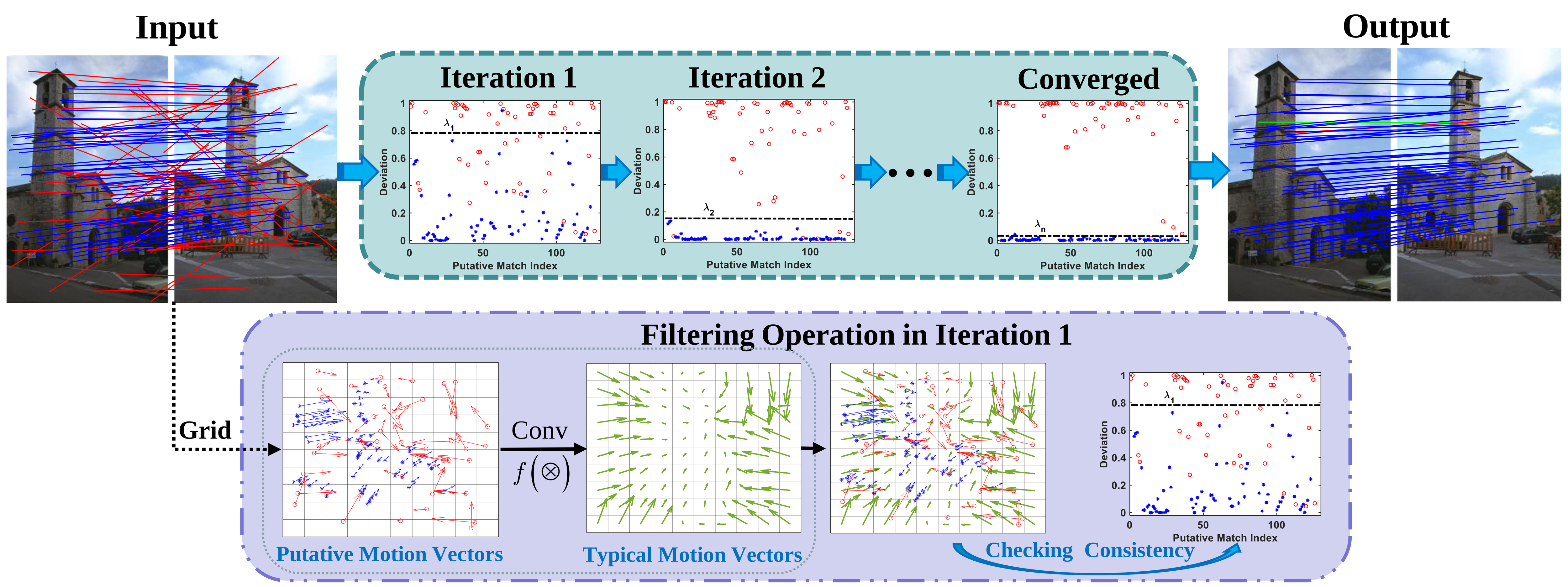
## Problem Formulation

**Main idea:** By assuming that the motion field of correct matches is smooth-and-slow, and considering the false matches as the outliers or noise, we formulate feature matching as a progressive filtering problem.

- Convert the putative match space into motion space and griding:
$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \rightarrow \mathcal{S}' = \{(\mathbf{x}_i, \mathbf{m}_i)\}_{i=1}^N, \quad (1)$$
where $\mathcal{S}$ is putative match set, and the matched feature points $\mathbf{x}_i$ and $\mathbf{y}_i$ are the pixel coordinates of the image pair. $\mathbf{m}_i = \mathbf{y}_i - \mathbf{x}_i$ denotes the motion vector.
- Calculate the average motion vector in each cell:
$$\overline{\mathbf{m}}_{j,k} = \begin{cases} \frac{1}{|\mathcal{C}_{j,k}|} \sum_{i|\mathbf{x}_i \in \mathcal{C}_{j,k}} \mathbf{m}_i, & \text{if } |\mathcal{C}_{j,k}| > 0, \\ \mathbf{0}, & \text{if } |\mathcal{C}_{j,k}| = 0. \end{cases} \quad (2)$$
where $\mathcal{C}_{j,k}$ is the putative set in $(j,k)$-th cell.
- Density estimation to deal with the isolated situation:
$$S(n) = \frac{C(n) - f^{D_0} N}{\sqrt{f^{D_0}(1 - f^{D_0})N}}, \quad (3)$$
where $S(n)$ measures the density degree of cell $n$
- Kernel convolutional filtering operation:
$$f(\otimes) : \widetilde{\mathbf{M}} = \frac{(\mathbf{W} \cdot \overline{\mathbf{M}}) \otimes \kappa}{\mathbf{W} \otimes \kappa + \varepsilon}, \quad (4)$$
where $\widetilde{\mathbf{M}}_{j,k} = \widetilde{\mathbf{m}}_{j,k}$ denotes the *typical motion vector* of cell $(j,k)$, $\mathbf{W}$ is a count matrix with $\mathbf{W}_{j,k} = |\mathcal{C}_{j,k}|$, and $\kappa$ is a Gaussian kernel distance matrix.
- Check motion consistency and identify inlier set $\mathcal{I}^*$.
$$d_i = 1 - \exp\Big\{ - \frac{\|\mathbf{m}_i - \widetilde{\mathbf{m}}_{j,k}\|^2}{\beta^2} \Big\}, \forall i, \mathbf{x}_i \in \mathcal{C}_{j,k}, \quad (5)$$
$$\mathcal{I}^* = \{ i \mid d_i \leq \lambda \}. \quad (6)$$

## Schematic Illustration

PFFM comprises three parts: putative matches griding, convolution operation and consistency checking, and it is implemented based on coarse-to-fine theory with an iterative manner. (Red:mismatches, Blue: correct matches)



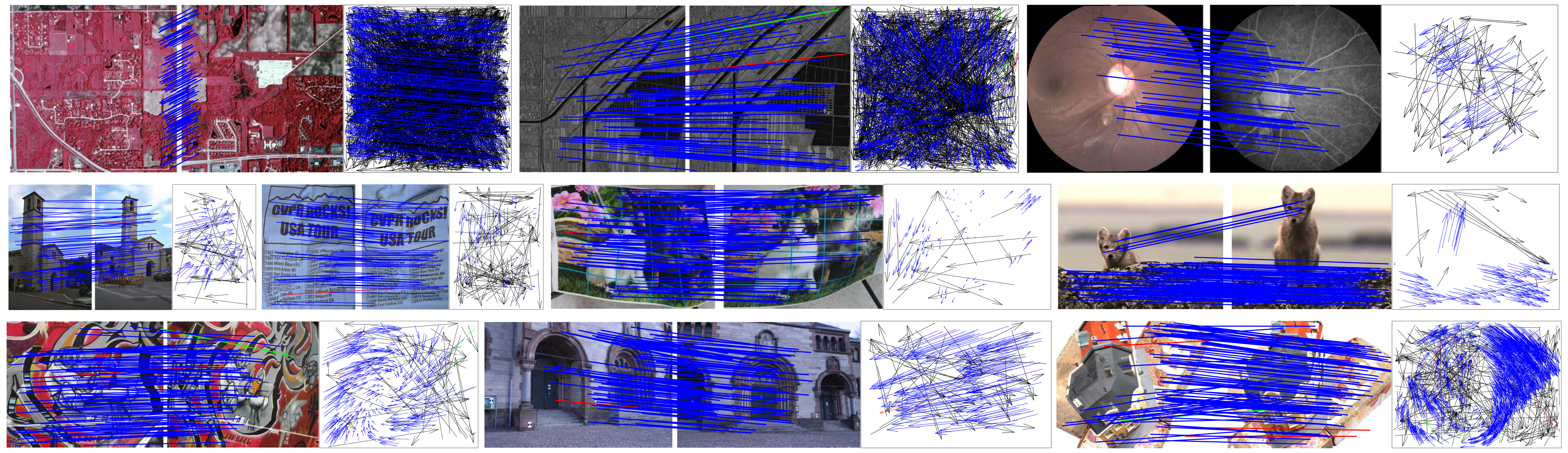

## Experiments & Results

**Dataset:**

Statistics of the introduced datasets

| Image Pair | *RS01* | *RS02* | *Retina* | *Church* | *Tshirt* | *Dogcat* | *Fox* | *Graf* | *Herzjesu* | *House* |
|---|---|---|---|---|---|---|---|---|---|---|
| Type | low-overlapped | scale-changed | nonrigid | wide-baseline | nonrigid | nonrigid | nonrigid | rotation | wide-baseline | rotation |
| Inlier Number | 2152 | 1982 | 101 | 126 | 226 | 113 | 135 | 442 | 184 | 1367 |
| Inlier Rate | 68.48% | 43.09% | 48.51% | 56.35% | 43.81% | 82.30% | 83.70% | 85.52% | 68.48% | 78.49% |

**Qualitative results on real image pairs:**



**Quantitative results of *Precision* (P) and *Recall* (R) on real data:**

| Data | RANSAC | | ICF | | GS | | GMS | | MR-RPM | | LPM | | PFFM (Ours) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P (%) | R(%) | P (%) | R(%) | P (%) | R(%) | P (%) | R(%) | P (%) | R(%) | P (%) | R(%) | P (%) | R(%) |
| *RS01* | **100.0** | **100.0** | 12.74 | **100.0** | **100.0** | 72.37 | 96.43 | 94.74 | 10.59 | **100.0** | 81.04 | 95.61 | **100.0** | **100.0** |
| *RS02* | 96.72 | **100.0** | **100.0** | 82.79 | 99.54 | 75.29 | 83.33 | 71.43 | 44.36 | **100.0** | 96.65 | 87.82 | 99.29 | 98.95 |
| *Retina* | **100.0** | **100.0** | 73.13 | **100.0** | 94.23 | **100.0** | 96.34 | 86.42 | **100.0** | 91.84 | 94.23 | **100.0** | **100.0** | **100.0** |
| *Church* | 95.16 | 83.10 | 93.75 | 63.38 | 95.83 | 97.18 | 86.76 | 83.10 | **100.0** | 80.28 | 82.50 | 92.96 | 98.59 | **98.59** |
| *Tshirt* | 96.39 | 80.81 | 78.26 | 90.91 | 93.06 | 67.68 | 79.21 | 80.81 | 97.98 | 97.98 | 87.76 | 86.87 | **98.02** | **100.0** |
| *Dogcat* | **100.0** | 97.85 | 92.19 | 63.44 | 97.70 | 91.40 | 93.18 | 88.17 | **100.0** | **100.0** | 97.89 | **100.0** | 98.94 | **100.0** |
| *Fox* | 98.10 | 91.15 | 98.57 | 61.06 | **100.0** | 90.27 | 96.49 | 97.35 | 97.12 | 89.38 | 94.87 | 98.23 | **100.0** | **100.0** |
| *Graf* | 99.74 | **100.0** | **100.0** | 14.02 | 99.66 | 77.51 | 98.07 | 93.92 | **100.0** | 96.30 | 98.14 | 97.88 | 99.21 | 99.47 |
| *Herzjesu* | 99.09 | 86.51 | 98.55 | 53.97 | 99.07 | 84.13 | 87.07 | 80.16 | 98.29 | 91.27 | 96.88 | 98.41 | **99.21** | **100.0** |
| *House* | 98.66 | 82.20 | **100.0** | 60.67 | **100.0** | 58.71 | 95.81 | 93.76 | 97.63 | 96.18 | 94.48 | 98.97 | 97.27 | **99.44** |
| **Average** | 98.38 | 92.16 | 84.72 | 69.02 | 97.91 | 81.45 | 91.27 | 86.98 | 84.60 | 94.32 | 92.44 | 95.68 | **99.05** | **99.65** |

**Quantitative results of *F-score* (F) and *Run-time* (T) on real data:**

| Data | RANSAC | | ICF | | GS | | GMS | | MR-RPM | | LPM | | PFFM (Ours) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F | T (ms) | F | T (ms) | F | T (ms) | F | T (ms) | F | T (ms) | F | T (ms) | F | T (ms) |
| *RS01* | **1.00** | 1.12e4 | 0.23 | 4.83e3 | 0.84 | 3.50e3 | 0.96 | **2.04**e0 | 0.19 | 5.15e2 | 0.88 | 2.62e1 | **1.00** | 7.77e1 |
| *RS02* | 0.98 | 1.96e2 | 0.91 | 1.66e1 | 0.86 | 9.67e1 | 0.77 | **1.13**e0 | 0.61 | 4.72e2 | 0.92 | 1.73e0 | **0.99** | 1.25e2 |
| *Retina* | **1.00** | 3.65e2 | 0.84 | 2.67e3 | 0.97 | 6.88e3 | 0.91 | **1.70**e0 | 0.96 | 1.15e1 | 0.97 | 1.22e1 | **1.00** | 6.16e1 |
| *Church* | 0.89 | 1.49e2 | 0.76 | 1.93e1 | 0.97 | 1.00e2 | 0.85 | **0.86**e0 | 0.89 | 1.28e1 | 0.87 | 1.21e0 | **0.99** | 4.40e1 |
| *Tshirt* | 0.88 | 1.00e3 | 0.84 | 4.61e1 | 0.78 | 2.60e2 | 0.80 | **1.02**e0 | 0.98 | 1.89e1 | 0.87 | 5.47e0 | **0.99** | 6.35e1 |
| *Dogcat* | 0.99 | 1.02e1 | 0.75 | 1.66e1 | 0.94 | 5.65e2 | 0.91 | 0.88e0 | **1.00** | 6.53e0 | 0.99 | **0.81**e0 | 0.99 | 4.83e1 |
| *Fox* | 0.94 | 4.58e1 | 0.75 | 1.83e1 | 0.95 | 1.31e3 | 0.97 | 0.84e0 | 0.93 | 8.27e0 | 0.97 | **0.83**e0 | **1.00** | 6.16e1 |
| *Graf* | **1.00** | 1.45e1 | 0.25 | 1.86e2 | 0.87 | 2.42e3 | 0.96 | **1.23**e0 | 0.98 | 2.11e1 | 0.98 | 2.76e0 | 0.99 | 8.67e1 |
| *Herzjesu* | 0.92 | 1.19e2 | 0.70 | 3.07e1 | 0.91 | 2.32e2 | 0.83 | 1.87e0 | 0.95 | 1.49e1 | 0.98 | **1.13**e0 | **1.00** | 6.33e1 |
| *House* | 0.90 | 8.31e1 | 0.76 | 1.91e3 | 0.74 | 2.70e4 | 0.95 | **1.61**e0 | 0.97 | 2.23e2 | 0.97 | 8.96e0 | **0.98** | 1.49e2 |
| **Average** | 0.95 | 1.32e3 | 0.68 | 9.74e2 | 0.88 | 4.24e3 | 0.89 | **1.32**e0 | 0.85 | 1.30e2 | 0.94 | 6.10e0 | **0.99** | 7.80e1 |