УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
январь–февраль 2021 Том 21 № 1 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
January–February 2021 Vol. 21 No 1 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2021-21-1-92-101

Goodpoint: unsupervised learning of key point detection and description

Anatoly V. Belikov[1], Alexey S. Potapov[2], Artem V. Yashchenko[3]✉

[1,2,3] SingularityLab, Saint Petersburg, 198152, Russian Federation
[3] ITMO University, Saint Petersburg, 197101, Russian Federation

[1] awbelikov@gmail.com, http://orcid.org/0000-0002-9081-642X
[2] pas.aicv@gmail.com, https://orcid.org/0000-0001-6013-8843
[3] yashenkoxciv@gmail.com✉, https://orcid.org/0000-0001-7292-2301

Abstract

Subject of Research. The paper presents the study of algorithms for key point detection and description, widely used in computer vision. Typically, the corner detector acts as a key point detector, including neural key point detectors. For some types of images obtained in medicine, the application of such detectors is problematic due to the small number of detected key points. The paper considers a problem of a neural network key point detector training on unlabeled images. **Method.** We proposed the definition of key points not depending on specific visual features. A method was considered for training of a neural network model meant for detecting and describing key points on unlabeled data. The application of homographic image transformation was basic to the method. The neural network model was trained to detect the same key points on pairs of noisy images related to a homographic transformation. Only positive examples were used for detector training, just points correctly matched with features produced by the neural network model for key point description. **Main Results.** The unsupervised learning algorithm is used to train the neural network model. For the ease of comparison, the proposed model has a similar architecture and the same number of parameters as the supervised model. Model evaluation is performed on the three different datasets: natural images, synthetic images, and retinal photographs. The proposed model shows similar results to the supervised model on the natural images and better results on retinal photographs. Improvement of results is demonstrated after additional training of the proposed model on images from the target domain. This is an advantage over a model trained on a labeled dataset. For comparison, the harmonic average of such metrics is used as: the accuracy and the depth of matching by descriptors, reproducibility of key points and image coverage. **Practical Relevance.** The proposed algorithm makes it possible to train the neural network key point detector together with the feature extraction model on images from the target domain without costly dataset labeling and reduce labor costs for the development of the system that uses the detector.

Keywords

unsupervised learning, deep learning, key points detection, local features

For citation: Belikov A.V., Potapov A.S., Yashchenko A.V. Goodpoint: unsupervised learning of key point detection and description. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2021, vol. 21, no. 1, pp. 92–101. doi: 10.17586/2226-1494-2021-21-1-92-101

УДК 004.272 004.032.26

Хорошая точка: обучение без учителя обнаружению и описанию по ключевым точкам

Анатолий Владимирович Беликов[1], Алексей Сергеевич Потапов[2], Артем Владимирович Ященко[3]✉

[1,2,3] СингуляритиЛаб, Санкт-Петербург, 198152, Российская Федерация
[3] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

[1] awbelikov@gmail.com, http://orcid.org/0000-0002-9081-642X
[2] pas.aicv@gmail.com, https://orcid.org/0000-0001-6013-8843
[3] yashenkoxciv@gmail.com✉, https://orcid.org/0000-0001-7292-2301

Аннотация

Предмет исследования. Алгоритмы выделения и описания ключевых точек широко применяются в компьютерном зрении. Обычно в качестве детектора ключевых точек выступает детектор углов, что относится в том числе и к нейросетевым детекторам. Для некоторых типов изображений, получаемых в том числе в медицине, такие детекторы не подходят из-за малого количества таких ключевых точек. В работе ставится задача обучения нейросетевого детектора ключевых точек на неразмеченных данных. **Метод.** Предложено определение ключевых точек, не зависящее от конкретных визуальных признаков. Рассмотрен способ обучения нейросетевой модели детектирования и описания ключевых точек на неразмеченных данных. В основе метода лежит использование гомографической трансформации изображений. Нейросетевая модель обучается детектировать одни и те же ключевые точки на парах зашумленных изображений, связанных гомографической трансформацией. Для обучения детектора используются только позитивные примеры, а именно только точки, правильно сопоставляемые по признакам, выдаваемым нейросетевой моделью описания ключевых точек. **Основные результаты.** Представленный алгоритм обучения без учителя использован для обучения нейросетевой модели. Для удобства сравнения предложенная модель имеет схожую архитектуру и такое же число параметров, как и модель, обученная с учителем. Проверка моделей выполнена на трех различных наборах данных: с естественными и с синтетическими изображениями, и на фотографиях сетчатки глаза. Предложенная модель показывает схожие результаты с обученной с учителем на естественных изображениях и лучшие — на фотографиях сетчатки глаза. Также демонстрируется улучшение результатов за счет дополнительного обучения рассмотренной модели на изображениях из целевого домена, что является преимуществом относительно модели, обучаемой на размеченных данных. Для сравнения использовалось гармоническое среднее от следующих показателей: точность и полнота сопоставления по дескрипторам, воспроизводимость ключевых точек и покрытие изображения ключевыми точками. **Практическая значимость.** Алгоритм позволяет обучать нейросетевой детектор ключевых точек вместе с моделью описанию ключевых точек на изображениях из целевого домена, при этом не требуя трудозатрат на разметку обучающего набора данных, что позволяет снизить трудозатраты на разработку системы, использующей детектор.

Ключевые слова

обучение без учителя, глубокое обучение, детектирование ключевых точек, локальные признаки

Ссылка для цитирования: Беликов А.В., Потапов А.С., Ященко А.В. Хорошая точка: обучение без учителя обнаружению и описанию по ключевым точкам // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21, № 1. С. 92–101 (на англ. яз.). doi: 10.17586/2226-1494-2021-21-1-92-101

Introduction

Local image features (key point detection and descriptor extraction) form the base of many computer vision applications, most of notably simultaneous localization and mapping (SLAM) and augmented reality. Traditionally, handcrafted local features were used such as Harris corner detector [1], SURF (Speeded-Up Robust Features) [2] and many others, but machine learning methods have shown their usefulness for the task quite early. For example, FAST (Features from Accelerated Segment Test) [3] introduced in 2006 uses decision trees for corner detection. With the improvement of hardware and deep learning theory, it became also possible to learn descriptor extraction and matching, for example, as in SuperGlue [4]. However, typically, supervised learning is used mostly that limits the applicability of the methods to novel domains.

One of machine learning definitions is the ability of a program to improve performance with more information [5]. Deployed feature extraction and image matching methods are to be applied to unlabeled data, and the improvement of their performance naturally supposes unsupervised learning. Although supervised learning demonstrates frequently better performance, unsupervised training of convolutional neural networks for feature generation [6, 7] provides state-of-the-art results. This fact cannot be achieved for the whole key point detection and description of extraction pipeline.

The SuperPoint supervised method [6] features a very simple loss function for descriptors, which minimizes the difference of descriptors of regions that corresponds geometrically, and maximizes the difference otherwise. Since the heatmap for key points is a kind of descriptor too, it hints at the possibility of building good key point detectors in an unsupervised manner by a simple loss function. Corners are the popular type of key points among handcrafted or supervised detectors. Among aforementioned methods, FAST, Harris and SuperPoint detect corners (and also line ends), SURF uses blob detection.

From a practical point of view, an ideal key point detector is the one that optimizes performance of a downstream task (image matching) or even target application (e.g. SLAM), but this measure might be difficult to compute and/or optimize. Instead, we assume that good key points should be distributed more or less evenly throughout the image and have good repeatability between various viewpoints. Good key points should be recognizable and distinguishable with descriptors and be not loo close.

A new unsupervised algorithm for simultaneous training of the key point detector and the descriptor generator is proposed. A single two-headed neural network built up on SuperPoint architecture is used for both tasks. The proposed model can be trained directly on a target domain without the need for performing costly domain adaptation, and it is applicable in situations in which domain adaptation does not work because of large differences between target and source domains. The proposed model achieves competitive performance with SuperPoint when trained on the same dataset, without supervised pre-training, and demonstrates better performance on images with low number of corner-like features. The resulting model is referred hereinafter as GoodPoint.

Related work

SuperPoint [6] introduced a fast convolutional neural network for key point detection and descriptor extraction. Training is split into two stages:

1) supervised training of a detector on synthetic dataset;
2) training of a detector on self-labelled natural images together with unsupervised training of a descriptor.

Our work follows SuperPoint architecture, but simplifies training procedure, removing supervised pre-training and self-labelling from the pipeline. Instead, both heads of the network are trained on natural images from the beginning.

GLAMpoint authors [7] train a key point detector on pairs of images related by a homographic transformation. The method uses non-maximum suppression on heatmaps both on images to extract candidate key points and then uses matching with SURF descriptors [2] to mine positive/negative examples.

Another related research direction is an object key point detection. Authors of [8] propose unsupervised key point-detector learning with conditional image generation. Given a pair of images (x, x') with the same objects, but with a different viewpoint and/or object pose, a training procedure minimizes weighted difference between features extracted from image x' and the reconstruction $\hat{x}' = \Psi(x, K(x'))$ of x'. The reconstruction is produced by a neural network Ψ, given image x and keypoints from x'. The loss function is defined as

$$L = \sum_l \alpha_l \|\Gamma_l(x') - \Gamma_l(\hat{x}')\|_2^2.$$

Here, Γ_l denotes output of a layer l of a pretrained neural network Γ, α_l — scalar weight for the loss computed from the output of layer l of network Γ.

Here, K is a keypoint detector neural network that learns to output k heatmaps

$$K(x') \in \mathbb{R}^{H \times W \times K},$$

where H, W are the height and the width of images x and x'. Each heatmap corresponds to the location of one keypoint and is normalized with softmax function to be a probability distribution.

Authors of [9] reuse the formulation from work [8] restricting to static backgrounds. The main difference from [8] is the introduction of feature transport: features extracted from both image used to generate x':

$$\Phi(x, x') = (1 - H(K(x)))(1 - H(K(x')))\Phi(x) + H(K(x'))\Phi(x').$$

Here, K is a key point detector that outputs k heatmaps. H is a heatmap image containing isotropic Gaussians around each of the key points that are specified by $K(x)$ or $K(x')$. Φ is a feature extraction network. Φ takes background features (that is, features from locations where there are no key points) from both images plus features from x' near target keypoints $K(x')$. The loss is a squared reconstruction error:

$$\|x' - RefineNet(\Phi(x, x'))\|_2^2,$$

where *RefineNet* is a convolutional generative model.

LF-Net [10] advances the results of SuperPoint to the new state-of-the-art on many datasets, though it requires ground truth depth and camera pose information. LF-Net projects score map from source image I_i to target image I_j, applies non-maximum suppression to sample key points and then generates new target score map with Gaussian kernel. The average difference

$$L(S_i, S_j) = |S_i - g(w(S_j))|_2$$

of score maps is minimized.

Here, w is a projection function, S_i, S_j are score maps extracted from images x_i, x_j, and g is a Gaussian kernel application. Descriptors are extracted from key point locations, the difference between a correspondent and non-occluded descriptors is minimized. Also, there is an additional loss for key point scales and orientations.

Architecture overview

The proposed GoodPoint architecture (Fig. 1) is based on SuperPoint architecture and consists of a common VGG backbone followed by two heads: descriptor and detector. The VGG backbone and descriptor heads are left unchanged, except for the activation function. The training procedure, detector head and loss function are different. Activation function used for all layers is leaky ReLU [11]. So, the total number of trainable parameters is the same. The detector is implemented similar to SuperPoint but without dustbin channel. So, the detector head outputs tensor $P \in \mathbb{R}^{H/8 \times W/8 \times 64}$ instead of $P \in \mathbb{R}^{H/8 \times W/8 \times 65}$. This fact does not affect performance and simplifies implementation since all channels are now being treated equally. Softmax is applied along the last axis to ensure that points lie not too close that makes it possible to learn only from positive examples. After the softmax, normalized tensors are reshaped from $\mathbb{R}^{H/8 \times W/8 \times 64}$ to $\mathbb{R}^{H \times W \times 1}$ to form a confidence map. The descriptor head outputs semi-dense tensor $D \in \mathbb{R}^{H/8 \times W/8 \times 256}$ which is interpolated in keypoint locations.

Training

Training is based on homographic warping of images and noise augmentation. The loss is computed on a pair of images: original I and warped with random homography image I_h. Single homography H is used for all images in a mini-batch. Both of the images may be warped, in which case they still are related by single homography H, so equations do not change. The final loss L is a weighted sum of two losses, descriptor loss L_d and detector loss

$$L(P, P_h, D, D_h, H) = \lambda_1 L_d + \lambda_2 L_p,$$

where λ_1, λ_2 are weights. P, P_h are heatmaps for images I and I_h. D, D_h are descriptors for images I and I_h.

After homographic warp, random noise filters are applied independently to images I, and I_h. Details of noise augmentation are provided in **«noise augmentation»** section.

Training of key point detection is inspired by expectation-maximization technique. The network learns to output key points that are easy for it to reproduce. It is trained with target key points computed with the following procedure (Fig. 2): points K found on image I are projected to I_h to

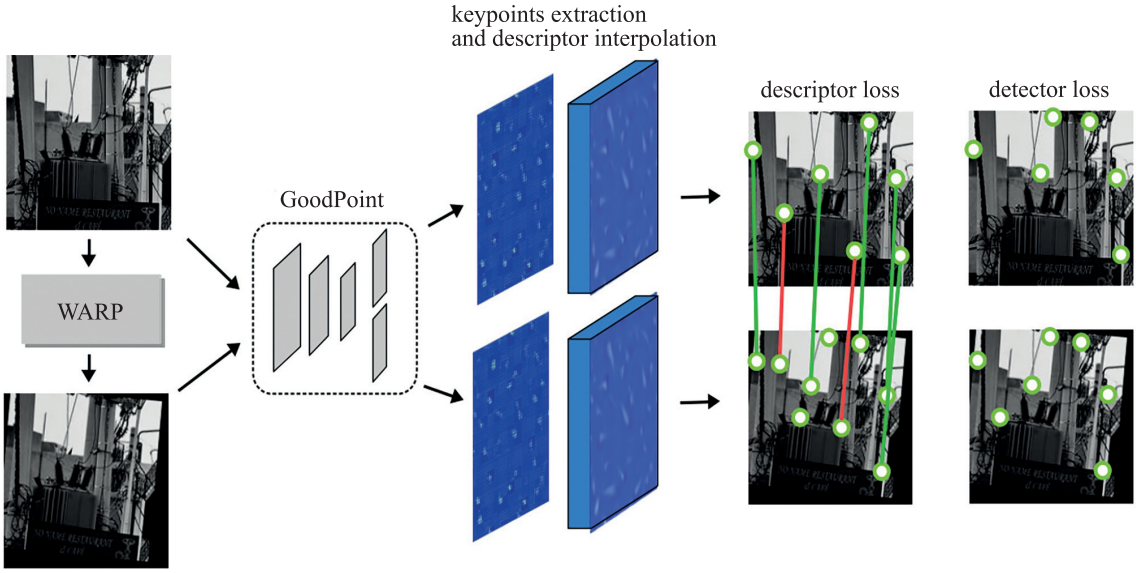

Fig. 1. Unsupervised training overview. First, key points and descriptors are extracted from the original and warped (WARP) images with a two-headed neural network. Descriptors are interpolated in location of key points from semi-dense output. Key points are matched with the descriptors and correctly matched points are used as positive examples for detector training. All interpolated descriptors are used to calculate descriptor loss

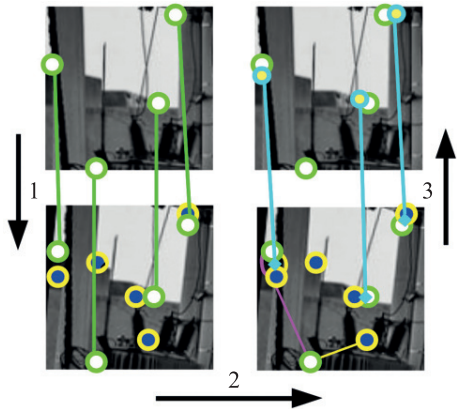

Fig. 2. Key point target estimation

form K_{proj} (1), projected points K_{proj} are matched with K_h by 2D coordinates and by descriptors with the nearest neighbor matcher to form two sets of matches $K_{proj} \to K_h$ (2), pairs of points that are matched by coordinates and by descriptors (namely, pairs are presented in both sets of matches) as the nearest neighbors are used to compute targets. Targets K_h' are projected back to image I (3).

A more detailed description is provided in the next section.

Key points loss

The loss function for key point detector is a sum of negative log-likelihoods of estimated target key point positions for both images plus heatmaps difference:

$$L_p(P, P_h, K, K_h, H) = L_{keypoints} + L_{heatmaps},$$

$$L_{keypoints} = -\frac{1}{2}(\log P[K'] + \log P_h[K_h']), \qquad (1)$$

$$L_{heatmaps} = \lambda_h \frac{1}{N_{mask}} \sum_{(i,j)\in mask}^{height,width} (blr(\widehat{PH}) - blr(P_h I))^2(i,j).$$

$P[*]$ denotes selection of points * from 2D heatmap P. λ_h is a weight for heatmap difference. $blr(\widehat{PH})$ denotes homographic projection of heatmap P, similarly, it is done for image I. Note that bilinear interpolation of P_h is necessary, otherwise the loss will be high even if heatmaps are similar due to $blr(\widehat{PH})$ being much smoother than P or P_h.

The sum iterates over points covered by mask for image I_h. The mask for image I_h is 2D tensor of the same shape as the image, such that for all points $p = (x, y)$ in the mask: $mask[p] = 1$ if projection $\widehat{PH}_{inv} \in I_h$ and 0, otherwise. N_{mask} is the number of nonzero elements of the mask. Tensors of estimated good keypoints positions K' and K_h' are computed using the following steps with the given two heatmaps P and P_h.

Key point arrays K and K_h are extracted from P and P_h with maxpooling of different sizes:

$$K = maxpool_{32\times32}(P),$$
$$K_h = maxpool_{16\times16}(P_h).$$

One key point selection for each region of size equal to 32 × 32 or 16 × 16 follows from the assumption that key points should be distributed more or less evenly throughout an image, but not too close. Max pool function performs max pooling and returns coordinates of key points (x_i, y_i) as an array. That is, K and K_h have $m \times 2$ and $n \times 2$ shapes.

If the projection of points K to image plane K_h is $K_{proj} = \widehat{KH}$, the keypoints projected beyond the boundaries of the image are discarded. D_{proj}, D_h are descriptors of points K_{proj} and K_h, that is, D_{proj} are descriptors extracted from image I of keypoints that stay in bounds when projected on image I_h.

The next step is to match points in I_h with descriptors and with coordinates:

$$dist_{geom}, idx_{geom} = match_{geom}(K_{proj}, K_h),$$
$$idx_{desc} = match_{desc}(D_{proj}, D_h).$$

Here, the function $match_{geom}$ performs the nearest neighbor matching of points K_{proj} to K_h, with Euclidean distance between coordinates as a measure.

$match_{geom}$ returns two vectors with length equal to length of K_{proj}. The first one is the distance from a point in K_{proj} to the nearest point in K_h. The second vector gives an index of the nearest point in K_h. idx_{desc} also gives an index of the nearest point in K_h, but with a distance computed in the space of descriptors. So, idx_{geom} and idx_{desc} are of the same length. Positive examples for keypoints in image I_h computed as mean coordinates of correctly matching points are:

$$K_h' = coords_{mean}(K_{proj}(i), K_h[idx_{geom}(i)])$$

for i, such that $idx_{geom}(i) = idx_{desc}(i)$ and $dist_{geom}(i) < \theta_{dist}$, that is, indices should match, and geometric distance should be less than the threshold. Here, $coords_{mean}(k_1, k_2) = 0.5(k_1 + k_2)$. θ_{dist} is the threshold in pixels for the case that distant points are matched correctly. Then K_h' is projected to image I with inverse homography:

$$H_{inv}K' = K_h'H_{inv}.$$

Thus, we have targets K and K_h' for both images that are needed to compute equation (1).

Descriptor loss

The loss for descriptors consists of three components:

$$L_{desc}(D, D_h, K, K_h, H) = L_{gt} + L_{wrong} + L_{random}.$$

Let element i of vector $g_i = D_{proj}(i)D_h^T(idx_{geom}[i])$, that is, scalar product of descriptors of key points, be matched by their coordinates, not by descriptors. Descriptors are normalized, so scalar product is equal to cosine similarity. L_{gt} maximises similarity of descriptors for each pair of points:

$$L_{gt} = \frac{1}{N_{gt}}\sum_j(1 - g_i).$$

L_{wrong} minimizes similarity of incorrectly matched pairs of descriptors, of points that are reasonably distant from each other

$$L_{wrong} = \frac{1}{N_{wrong}}\sum_j g_j.$$

For such j, that

$$idx_{geom}(j) \neq idx_{desc}(j) \cap dist_{geom}(j) > 7.$$

L_{random} minimizes the difference of randomly sampled descriptors.

$$L_{random} = \frac{1}{N_{random}N_{points}}\sum_{i=0}^{N_{random}}\sum_{j=0}^{N_{points}} D_{proj}(i)sh(D_h(i))^T$$

$sh(D)$ is randomized shuffle of rows of descriptor matrix, such that no pair of $D_{proj}(i)$, $D(i)$ would belong to the nearest neighbors as defined by idx_{geom}.

Implementation details. The model was implemented with pytorch framework. Optimization algorithm used during training is AdamW [12] with initial learning rate of 0.0005, all other parameters are set to default values, particularly weight decay has default of 0.01. The proposed model was trained on the training set images from MS COCO (Microsoft Common Objects in Context) dataset [13]. Each minimatch was composed from random crops of size equal to 256 × 256 px. Weight for heatmap difference was set to 2000. The network was trained with constant learning rate for the first 8 epochs, after the 8th epoch exponential decay of learning rate was used for 10 more epochs.

Noise augmentation. Noise filters are applied in predefined order, sequentially to each image. Each filter is skipped with probability equal to 0.5. Filters used during training are: additive Gaussian, random brightness, additive shade, salt & pepper, motion blur, random contrast scale. After each filter application, the image is checked for validity. The image is considered ruined if its variance is less than 10 % of the original in which case the filter is skipped.

Homographic augmentation. Random homography matrices are generated as a product of simple transformations (Fig. 3). Random shift of points is in the range of ±14 px. Perspective shift of side and/or top, or bottom points are in the range of ±85 px. Random homography augmentation was applied to both I and I_h with random rotation sampled from the range of ±0.08 rad.

Assessing performance. In a two-headed neural network, there is a trade-off between the performance of detector and descriptor networks. Computing a single metric that combines points repeatability and the precision of matching with descriptors is one way to break ties among multiple model variants. The authors of the paper [14] propose the following F1-like metric:

$$F1 = 2 \times (precision(D, K) \times \times repeatability(K)/(precision(D, K) + repeatability(K)),$$

that is, harmonic mean of precision of matching and key points repeatability, which was used for tuning hyper-parameters during training. Here, D are descriptors and K are keypoints. So, for all experiments we compute harmonic mean of all evaluation metrics, which gives a single number for comparison. We also calculate coverage

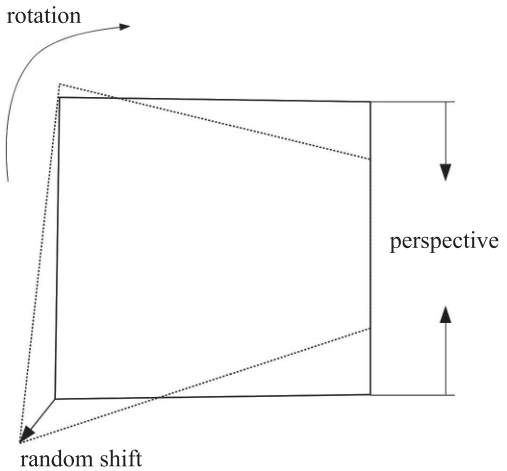

Fig. 3. Random homography. Homography is estimated from random perturbations of rectangle points

Table 1. Test results on AirSim dataset

Dataset	Fantasy village	Village
SuperPoint	Precision: 0.86 Repeatability: 0.57 Coverage: 0.57 Harmonic mean: 0.64	Precision: 0.72 Repeatability: 0.45 Coverage: 0.65 Harmonic mean: 0.58
GoodPoint	Precision: 0.85 Repeatability: 0.55 Coverage: 0.65 Harmonic mean: 0.66	Precision: 0.74 Repeatability: 0.42 Coverage: 0.70 Harmonic mean: 0.58
SuperPoint 5° rotation	Precision: 0.85 Repeatability: 0.54 Coverage: 0.56 Harmonic mean: 0.63	Mean recall: 0.70 Repeatability: 0.42 Coverage: 0.62 Harmonic mean: 0.55
GoodPoint 5° rotation	Precision: 0.85 Repeatability: 0.54 Coverage: 0.63 Harmonic mean: 0.65	Mean recall: 0.70 Repeatability: 0.39 Coverage: 0.67 Harmonic mean: 0.55

additionally to replication ratio and accuracy for all datasets. The methodology was proposed in Irschara et al. [15]. Coverage is a ratio of covered pixels to all pixels in an image, with a pixel considered as covered when it lies within a certain distance from correctly matched key point.

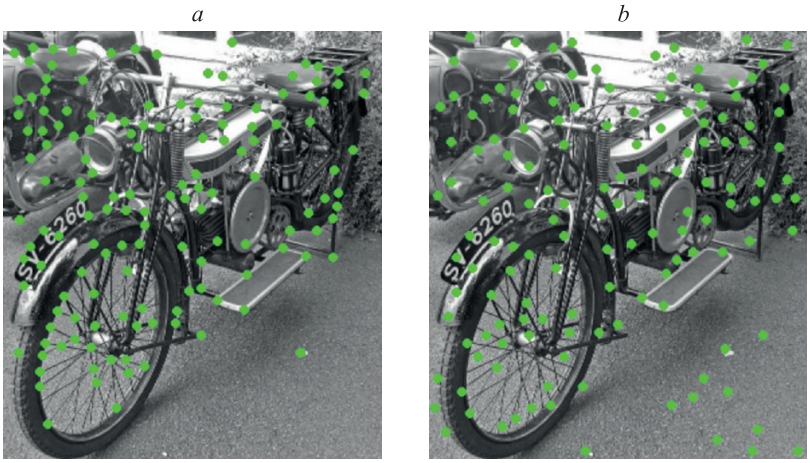

Fig. 4. Comparison of key points: superpoint points (*a*); goodpoint points (*b*). Each image has 143 points. Many of goodpoint points do not correspond to corners due to unsupervised learning, though many points coincide with corners

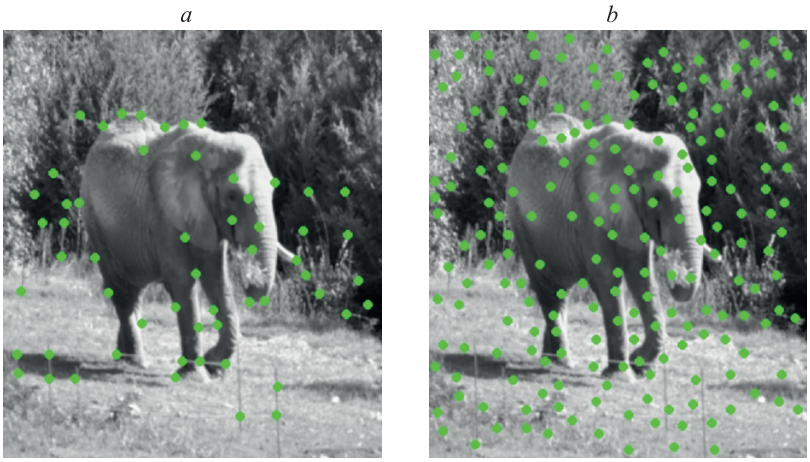

Fig. 5. Comparison of key points: superpoint points (57) (*a*); goodpoint points (201) (*b*). The same thresholds are used as for previous image

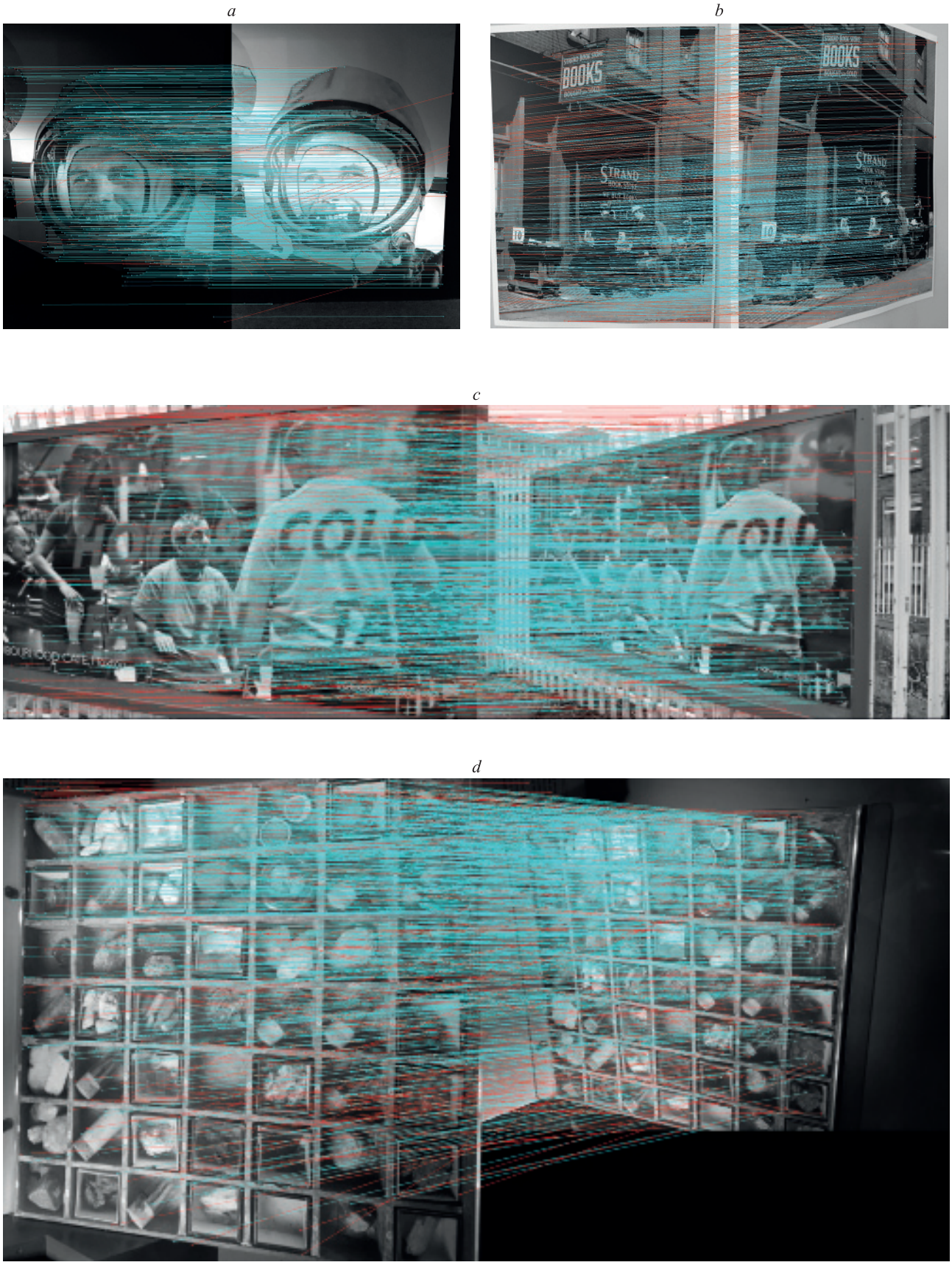

Fig. 6. Points and matches from HPatches dataset: lighting and contrast variability (*a*); small shift of point of view (*b*); change of perspective (*c*); change of perspective (*d*)

Experiments

Figures 4 and 5 show side-by-side comparison of what networks tend to select as key points. The threshold is set so that in the first image the networks detect the same number of key points. It can be seen that the unsupervised model is less biased towards corner features, which may be an advantage or disadvantage depending on scene properties. More example images are available at the project website (https://github.com/singnet/image-matching).

AirSim village dataset. AirSim village dataset was introduced in [14]. It contains two sequences of images made with varying lighting but with the same camera positions, ground truth camera pose and depth information, that may be used for the evaluation of SLAM or related methods, namely, feature extraction and matching. Sequences were made by recording camera motion through a synthetic environment. The test was done on 320 × 240 px resolution. Matching was done with the shift of 5 frames and radius of coverage set to 20 px. Precision and repeatability were calculated as an average of matching in both ways: $I_i \rightarrow I_{i+5}$, and $I_{i+5} \rightarrow I_i$. The model was evaluated with and without roll of 5 degrees. The original dataset was not generated with a roll in camera motion.

The results are presented in Table 1. Threshold for the correct match is 3 px. $\theta_{keypoint} = 0.028$ for GoodPoint, 0.015 for SuperPoint. $\theta_{desc} = 0.8$ for both models. Overall, GoodPoint demonstrates good precision with lower than SuperPoint repeatability of key points.

HPatches. For Hpatches dataset the methodology of LF-net and SuperPoint papers have been used with thresholds for correct match set to 3 and 5 pixels. Coverage radius of 25 px was used. The results are presented in Table 2. The test demonstrates that the models have similar performance, with SuperPoint being more accurate in estimating key points positions, while GoodPoint tends to select more points, thus giving higher coverage, but lower replication ratio.

Fundus Image Registration Dataset. FIRE [16] dataset contains 134 pairs of retinal images with ground truth correspondences for a number of points, which allows for homography estimation. Also the dataset contains two masks, for global and local registration methods. Coverage radius is set to 25 px for this dataset. GoodPoint was tuned on images from FIRE, with the only change in the training pipeline being a different size of the crop window.

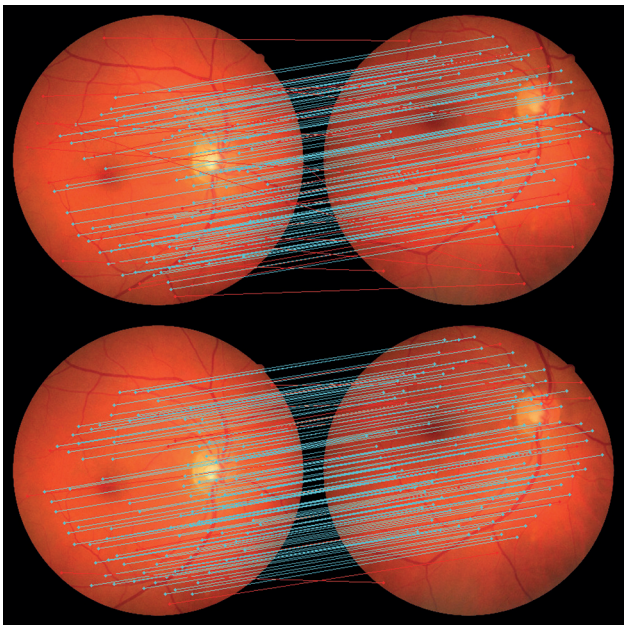

Fig. 7. GoodPoint results on FIRE: before fine-tuning on FIRE dataset (top), after (bottom). The image is equal to 874 × 874 px

Both original (trained on MS COCO) and fine-tuned versions were evaluated. The results are presented in Table 3. GoodPoint demonstrates better coverage than supervised SuperPoint, which shows that unsupervised learning of the key point detector introduced less bias into the model.

There is a trade-off between an accuracy and coverage, and it is possible to achieve the accuracy of GLAMPoint (0.91) on the FIRE dataset with the higher threshold for the key point detector as shown in Table 3. Coverage and replication ratio were not reported in the paper [7].

Conclusion and future work

A novel method for joint training of key points detection and description has been introduced, fully unsupervised, and can be applied to train a model directly on a set of unlabeled images. The method was used to train a convolutional model named GoodPoint. GoodPoint is based

Table 2. GoodPoint and SuperPoint Models on HPatches dataset at θ_{dist} 3 and 5, px

Parameters	Model			
	3		5	
	GP	SP	GP	SP
$\theta_{keypoint}$	0.021	0.015	0.021	0.015
Light Replication	0.48	0.53	0.63	0.63
View Replication	0.33	0.45	0.47	0.55
Light Accuracy	0.69	0.70	0.82	0.80
View Accuracy	0.53	0.64	0.67	0.72
Light Coverage	0.60	0.47	0.64	0.50
View Coverage	0.41	0.42	0.45	0.45
Harmonic mean	0.48	0.52	0.59	0.59

Table 3. Tests on FIRE dataset: original and GoodPoint performance on FIRE with threshold $\theta_{keypoint} = 0.075$

Metric	Models			
	GoodPoint	GoodPoint tuned on FIRE	SuperPoint	GoodPoint with threshold $\theta_{keypoint} = 0.075$
Accuracy	0.78	0.79	0.84	0.91
Coverage	0.66	0.70	0.54	0.21
Replication	0.82	0.82	0.86	0.91
Harmonic mean	0.75	0.75	0.71	0.44

upon SuperPoint architecture. For the ease of comparison, only minor changes were introduced, such as removal of dustbin channel in key point detector, necessary for the proposed training method. As the result, GoodPoint has the same number of layers and parameters as SuperPoint. The trained model was evaluated on diverse datasets and demonstrated a good performance on natural and synthetic images, both rich (HPatches, AirSim village) and poor (FIRE) in corner features. GoodPoint tends to produce dense detections, which corresponds to higher coverage. The results open the way for the following improvements and/or research directions.

Replacement of maxpooling for key point extraction with theoretically sound sampling methods, such as e-greedy sampling, gives the possibility for augmenting local descriptors with global features, in the way it is done in SuperGlue during matching [4].

References

1. Harris C., Stephens M. A combined corner and edge detector. *Proc. of the Alvey Vision Conference*, UK, Manchester, 1988, pp. 23.1–23.6. doi: 10.5244/C.2.23
2. Funayama R., Yanagihara H., Van Gool L., Tuytelaars T., Bay H. Robust interest point detector and descriptor. *Patent US8165401 B2*, 2012.
3. Rosten E., Drummond T. Machine learning for high-speed corner detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 3951, pp. 430–443. doi: 10.1007/11744023_34
4. Sarlin P.E., DeTone D., Malisiewicz T., Rabinovich A. SuperGlue: Learning feature matching with graph neural networks. *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4937–4946. doi: 10.1109/CVPR42600.2020.00499
5. Mitchell T.M. *Machine Learning*. McGraw Hill, 1997, 414 p.
6. DeTone D., Malisiewicz T., Rabinovich A. Superpoint: Self-supervised interest point detection and description. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 337–349. doi: 10.1109/CVPRW.2018.00060
7. Truong P., Apostolopoulos S., Mosinska A., Stucky S., Ciller C., Zanet S.D. GLAMpoints: Greedily learned accurate match points. *Proc. of the IEEE International Conference on Computer Vision*, Korea, Seoul, 2019, pp. 10732–10741. doi: 10.1109/ICCV.2019.01083
8. Jakab T., Gupta A., Bilen H., Vedaldi A. Unsupervised learning of object landmarks through conditional image generation. *Advances in Neural Information Processing Systems*, 2018, pp. 4016–4027.
9. Kulkarni T.D., Gupta A., Ionescu C., Borgeaud S., Reynolds M., Zisserman A., Mnih V. Unsupervised learning of object keypoints for perception and control. *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 10723–10733.
10. Ono Y., Trulls E., Fua P., Yi K.M. LF-Net: learning local features from images. *Advances in Neural Information Processing Systems*, 2018, pp. 6234–6244.
11. Maas A.L., Hannun A.Y., Ng A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. 30th International Conference on Machine Learning*, USA, Atlanta, 2013, pp. 3.
12. Loshchilov I., Hutter F. Decoupled weight decay regularization. Proc. *7th International Conference on Learning Representations (ICLR 2019)*, 2019.
13. Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C.L. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48
14. Yashchenko A.V., Belikov A.V., Peterson M.V., Potapov A.S. Distillation of neural network models for detection and description of image key points. *Scientific and Technical Journal of Information*

Литература

1. Harris C., Stephens M. A combined corner and edge detector // Proc. of the Alvey Vision Conference. UK, Manchester. 1988. P. 23.1–23.6. doi: 10.5244/C.2.23
2. Funayama R., Yanagihara H., Van Gool L., Tuytelaars T., Bay H. Robust interest point detector and descriptor. Patent US8165401 B2. 2012.
3. Rosten E., Drummond T. Machine learning for high-speed corner detection // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006. V. 3951. P. 430–443. doi: 10.1007/11744023_34
4. Sarlin P.E., DeTone D., Malisiewicz T., Rabinovich A. SuperGlue: Learning feature matching with graph neural networks // Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2020. P. 4937–4946. doi: 10.1109/CVPR42600.2020.00499
5. Mitchell T.M. Machine Learning. McGraw Hill, 1997. 414 p.
6. DeTone D., Malisiewicz T., Rabinovich A. Superpoint: Self-supervised interest point detection and description // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2018. P. 337–349. doi: 10.1109/CVPRW.2018.00060
7. Truong P., Apostolopoulos S., Mosinska A., Stucky S., Ciller C., Zanet S.D. GLAMpoints: Greedily learned accurate match points // Proc. of the IEEE International Conference on Computer Vision. Korea, Seoul. 2019. P. 10732–10741. doi: 10.1109/ICCV.2019.01083
8. Jakab T., Gupta A., Bilen H., Vedaldi A. Unsupervised learning of object landmarks through conditional image generation // Advances in Neural Information Processing Systems. 2018. P. 4016–4027.
9. Kulkarni T.D., Gupta A., Ionescu C., Borgeaud S., Reynolds M., Zisserman A., Mnih V. Unsupervised learning of object keypoints for perception and control // Advances in Neural Information Processing Systems. 2019. V. 32. P. 10723–10733.
10. Ono Y., Trulls E., Fua P., Yi K.M. LF-Net: learning local features from images // Advances in Neural Information Processing Systems. 2018. P. 6234–6244.
11. Maas A.L., Hannun A.Y., Ng A.Y. Rectifier nonlinearities improve neural network acoustic models // Proc. 30th International Conference on Machine Learning. USA, Atlanta. 2013. P. 3.
12. Loshchilov I., Hutter F. Decoupled weight decay regularization // Proc. 7th International Conference on Learning Representations (ICLR 2019). 2019.
13. Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C.L. Microsoft COCO: Common objects in context // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2014. V. 8693. P. 740–755. doi: 10.1007/978-3-319-10602-1_48
14. Ященко А.В., Беликов А.В., Петерсон М.В., Потапов А.С. Дистилляция нейросетевых моделей для детектирования и описания ключевых точек изображений // Научно-технический вест-

Technologies, Mechanics and Optics, 2020, vol. 20, no. 3, pp. 402–409. (in Russian). doi: 10.17586/2226-1494-2020-20-3-402-409
15. Irschara A., Zach C., Frahm J.M., Bischof H. From structure-from-motion point clouds to fast location recognition. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2599–2606. doi: 10.1109/CVPRW.2009.5206587
16. Hernandez-Matas C., Zabulis X., Triantafyllou A., Anyfanti P., Douma S., Argyros A.A. FIRE: Fundus image registration dataset. *Journal for Modeling in Ophthalmology*, 2017, vol. 1, no. 4, pp. 16–28.

ник информационных технологий, механики и оптики. 2020. Т. 20. № 3. С. 402–409. doi: 10.17586/2226-1494-2020-20-3-402-409
15. Irschara A., Zach C., Frahm J.M., Bischof H. From structure-from-motion point clouds to fast location recognition // Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2009. P. 2599–2606. doi: 10.1109/CVPRW.2009.5206587
16. Hernandez-Matas C., Zabulis X., Triantafyllou A., Anyfanti P., Douma S., Argyros A.A. FIRE: Fundus image registration dataset // Journal for Modeling in Ophthalmology. 2017. V. 1. N 4. P. 16–28.

Authors

Anatoly V. Belikov — Engineer, SingularityLab, Saint Petersburg, 198152, Russian Federation, SC 57210427029, awbelikov@gmail.com, http://orcid.org/0000-0002-9081-642X
Alexey S. Potapov — D.Sc., Professor, Leading Researcher, SingularityLab, Saint Petersburg, 198152, Russian Federation, SC 7201761961, pas.aicv@gmail.com, https://orcid.org/0000-0001-6013-8843

Artem V. Yashchenko — Postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation; Engineer, SingularityLab, Saint Petersburg, 198152, Russian Federation, SC 5719672261, yashenkoxciv@gmail.com, https://orcid.org/0000-0001-7292-2301

Авторы

Беликов Анатолий Владимирович — инженер, СингуляритиЛаб, Санкт-Петербург, 198152, Российская Федерация, SC 57210427029, awbelikov@gmail.com, http://orcid.org/0000-0002-9081-642X
Потапов Алексей Сергеевич — доктор технических наук, профессор, ведущий научный сотрудник, СингуляритиЛаб, Санкт-Петербург, 198152, Российская Федерация, SC 7201761961, pas.aicv@gmail.com, https://orcid.org/0000-0001-6013-8843
Ященко Артем Владимирович — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация; инженер, СингуляритиЛаб, Санкт-Петербург, 198152, Российская Федерация, SC 5719672261, yashenkoxciv@gmail.com, https://orcid.org/0000-0001-7292-2301

Received 27.10.2020
Approved after reviewing 08.12.2020
Accepted 19.01.2021

Статья поступила в редакцию 27.10.2020
Одобрена после рецензирования 08.12.2020
Принята к печати 19.01.2021