ISSN 0361-7688, Programming and Computer Software, 2018, Vol. 44, No. 4, pp. 258–265. © Pleiades Publishing, Ltd., 2018.
Original Russian Text © S. Arseev, A. Konushin, V. Liutov, 2018, published in Programmirovanie, 2018, Vol. 44, No. 4.

Human Recognition by Appearance and Gait[1]

S. Arseev[a,*], A. Konushin[a,b,], and V. Liutov[a,***]**

[a] *Moscow State University, GSP-1, Moscow, 119991 Russia*
[b] *NRU Higher School of Economics, Moscow, 101000 Russia*
**e-mail: 9413serg@gmail.com*
***e-mail: anton.konushin@graphics.cs.msu.ru*
****e-mail: vladimir.lutov@graphics.cs.msu.ru*

Received January 31, 2018

Abstract—This work is focused on person identification task in video sequences. For this task we propose two complementing solutions, which can be applied in different cases: gait and visual recognition. For gait recognition three kinds of features are used: anthropometric features, based on the length of the skeleton segments; relative distance features, based on relative distances between the skeleton joints; and motion features, based on the movement of a joint between two frames. Two versions of the gait recognition algorithm are presented: the first one uses the depth data alongside with the images while the other one uses only the video sequence. For visual recognition from appearance we propose a deep learning algorithm that returns binary image features. Each algorithm was tested on two datasets. Furthermore, we perform experiments on transfer from one dataset to another to check trained model transferability.

DOI: 10.1134/S0361768818040035

1. INTRODUCTION

In this paper we consider the task of re-identification of a person in a video. The goal is to automatically match people's detections with a list of known people, which is called a gallery. Re-identification is used in video surveillance systems for matching people between different cameras in criminalistics and for search in video archives. Nowadays the main way to identify a person in the video is by face. This method is not applicable in scenarios where a person is not sufficiently visible due to a camera angle, resolution, and also if the face is intentionally closed. In these situations, we can rely only on appearance in general and on movement, for example, human gait.

The aim of the work is to overcome these limitations by developing and improving two complementary methods of human identification in the video: human recognition by gait and recognition by appearance using binary image descriptors.

Recognition by appearance is primarily used to re-identify a person in the tracking algorithms [1]. If a person in the process of movement is overlapped for some time by an element of the scene, the path of the movement is divided into two separate fragments [2]. To unite the fragments into a single path, it is necessary to match later fragment to previous one by appearance. Since it is necessary to solve such tasks constantly during the tacking process, the speed is very important, and the use of binary descriptors is a good way to improve performance.

In some scenarios the person's appearance cannot be used as a reliable feature for the identification because this feature neither separates different people (people can wear similar clothing or totally identical in case of uniform) nor is robust over long time intervals (when a person changes clothes, appearance also changes). In this case gait recognition is used: features are calculated by analyzing a person's movement in a video sequence containing the recording of their gait. This task is related to the biometrics field, i.e. human recognition based on physical or behavioral features. The distinct characteristic of gait recognition is its usability: human gait is a feature based on actions and not on appearance so appearance changes (e.g. clothing changes) have little to no effect on gait.

2. REVIEW OF EXISTING METHODS

Person detections are used as input for the re-identification algorithm. Detections are usually rectangles that contain images of people extracted from static cameras. In the case of gait identification, input is a sequence of rectangles obtained from sequential frames. The output of re-identification is a list of persons in a gallery, sorted by visual or gait similarity.

This task is a special case of the task of human identification. It is traditionally solved in two stages. First stage is the construction of the feature vector or

[1] The article was translated by the authors.

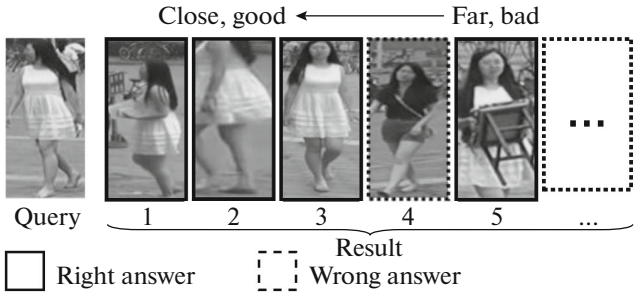

Fig. 1. Example output of the proposed algorithm.

descriptor from image or set of images. The second stage is nearest neighbor search in the gallery, using some similarity metric [3]. An example of the operation is shown in Fig. 1. In this paper we consider mainly the first stage. For the second stage we use linear search in a gallery, but approximate nearest neighbor methods can also be used.

2.1. Gait Recognition

Human gait recognition methods can be divided into two large groups: silhouette-based and structural model-based. Methods that use the second approach are more robust to camera position changes and some factors affecting the silhouette and appearance of a person. The downside of these methods is the high computational cost of model construction which slows down the algorithm. The existing open datasets usually have insufficient number of examples of each person's gait in order to apply common deep learning methods as in face recognition.

1. Silhouette-based methods are mostly derivatives and modifications of the algorithm [4], based on gait energy images or GEI. The gait energy image is produced by averaging the human silhouettes, centered to avoid position shift during the walk, by entire video sequence or gait cycle, i.e. time required to make a single step. Features for classification are extracted from this image. For example, in [5] the gait energy image is first processed with curvelet-transform and then classified using deep learning.

2. Model-based methods rely on human pose model, estimated from person image [6, 7]. Currently, in model-based methods various pose estimation methods are used. Since the introduction of Microsoft Kinect 3D sensors, it is probably the most popular approach [8–10].

2.2. Appearance-Based Recognition

Most reidentification methods build a real-valued descriptor. But if a large person gallery is used, the comparison of real-valued descriptors, even with simple L2 distance metric, can be prohibitively expensive. One of the popular approaches for speeding-up image search is to use binary descriptors, which are much easier to compare using Hamming distance [11]. In this case a real-values feature vector is build first, and then a binary descriptor is built based on it.

2.2.1. Real-valued appearance descriptors. Methods for constructing real-valued appearance descriptors can be divided into three groups:

1. Methods without machine learning show the least accuracy [3, 12].

2. Methods with machine non-deep learning. In this group "Hierarchical Gaussian Descriptor for Human Re-Identification" GOG [13] shows the best accuracy [3].

3. Methods with deep learning show the best results for this task [14, 15] nowadays, as well as for many closely related tasks like moving object detection [16].

2.2.2. Binary appearance descriptors. There are several methods for the binarization of descriptors for re-identification [17]. In contrast to past studies in this paper, we compare several methods at once:

1. Naive binarization, an elemental comparison of the initial descriptor with 0.

2. Selection of the most significant elements of the initial descriptor using the random forest method [18].

3. Conversion into a space with smaller dimensions using the principal component method [19], then binarization.

4. Neural network methods for constructing binary descriptors: the sigmoid algorithm [11] and its DBE modification [20].

The sigmoid algorithm is based on addition of a fully connected layer with sigmoidal activation function to the network, which generates real-valued descriptors. In the DBE algorithm, a sequence of several layers is used instead of one layer:

$$f_{DBE}(x) = \tanh(\mathrm{Re}\,LU(BN(W_{DBE}x + b_{DBE}))),$$

where f_{DBE} is the DBE algorithm, X is the real descriptor, $\tanh(Z)$ is the elemental hyperbolic tangent, $\mathrm{Re}Lu(Z)$ is the elemental $\max(0, z_i)$, BN is the layer of the batch normalization [21], W_{DBE}, b_{DBE} are the optimized weights of the algorithm, the matrix and the vector, respectively.

3. PROPOSED METHODS

3.1. Gait Recognition

The proposed method is based on the method introduced in [10]. Each sequence is pre-processed to produce a pose estimation model for each frame (a skeleton) which is used by the identification algorithm itself. The skeleton produced by Microsoft Kinect sensor is marked as shown on Fig. 2.

The features extracted from the skeleton sequence can be divided into three categories: anthropometric features (AF) i.e. skeleton segments lengths and the

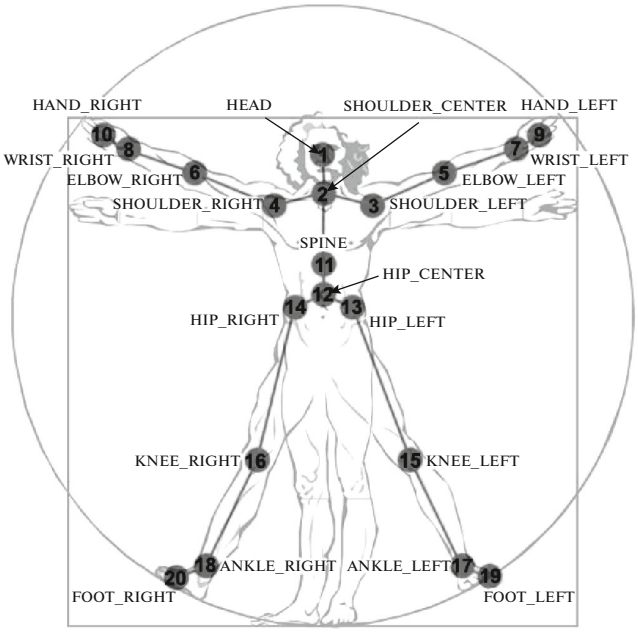

Fig. 2. Kinect skeleton.

person's height; relative distance features (RDF) produced from the coordinate difference between skeleton joints; and motion features (MF) based on the node position shift between two subsequent frames. The classification was performed using K nearest neighbor method and random subspace method for feature selection.

3.1.1. Anthropometric features. Anthropometric features represent skeleton segments lengths and the total height of the person. $Len(a, b)$ is the Euclidean distance between joints a and b.

For the Kinect skeleton:

$$Height = Len(1,2) + Len(2,11) + Len(11,12) + \frac{Len(14,16) + Len(13,15)}{2} + \frac{Len(16,18) + Len(15,17)}{2}$$

19 *Len* values and the *Height* value produce the anthropometric features (AF) vector. Since coordinates of the skeleton nodes are imprecise because of noise and occlusion, each component of the AF vector vas filtered: mean and standard deviation values over the whole sequence were calculated and after that a new mean value was calculated without using frames where this component value deviated from the mean by more than two standard deviations. The resulting vector of these new average values is the final AF vector used for classification.

3.1.2. Relative distance features. Each joint a of the skeleton is described by its coordinates: x_a, y_a and z_a for skeletons with the depth map. Relative distance features represent the distances between skeleton joints.

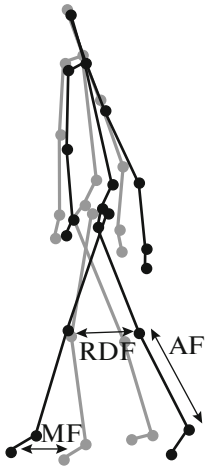

Fig. 3. Three types of gait features.

They are calculated for each axis separately, and the RDF vector consists of all distances represented as $|x_a - x_b|$, where a and b are the indices of the paired joints, such as 3 and 4 or 15 and 16, or a is a joint in the pair and b is the corresponding middle joint (2 for a 3-10 and 12 for 13-20). It also includes features represented as $\left|x_c - \frac{x_a + x_b}{2}\right|$, where a and b are the indices of the paired joints and c is either 1 (head joint) or 11 (spine joint). The mean and standard deviation values of each distance over the sequence produce the final RDF vector.

For three-dimensional Kinect skeletons, the RDF vector consists of 240 elements: there are 8 pairs of joints and each pair produces 10 features for each of the 3-coordinate axis.

3.1.3. Motion features. Motion features are based on the movement speed of skeleton's joints between frames. The shift of the joints between frames is calculated in relative coordinates to ignore the overall movement of the person.

These features are calculated on all axis separately: x, y and z. In the two-dimensional version of the algorithm there are only two axes since there is no depth information.

$$Mx_{a,t} = |x_{a,t} - x_{b,t} - (x_{a,t-1} - x_{b,t-1})|.$$

Here, a is a joint on a limb (joints 3-10 and 13-20) and b is the corresponding middle joint (as in RDF calculation).

The mean (by t) and standard deviation values for each joint and each axis produce the MF vector. For Kinect skeletons, it consists of 96 elements: each of the 16 pairs produces 2 features for each of the 3 axes.

These three feature types are shown on Fig. 3.

3.1.4. Sequence processing. The resulting feature vector, consisting of AF, RDF and MF vectors, is classified using an ensemble of 100 K-nearest neighbor

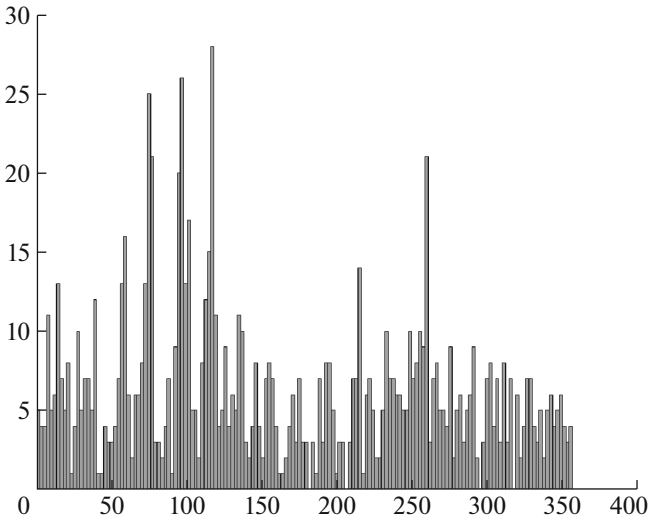

Fig. 4. Gait features usage. X axis represents feature indices in the final vector: first 240 features are RDF, next 20 are AF and the rest are MF. Y axis represents the number of classifiers using this feature.

classifiers with the city block metric. Due to the large size of the feature vector (356 features), each classifier uses a subset of 10 features.

A scoring system is used to assess the quality of the classifier (and, accordingly, the subspace of features, on which it operates): the classifier gives out five closest examples to the probe descriptor, after which it is awarded points, if among these examples there are examples of the same class as the test example: 5 points, if the nearest example is the right class, 4 points for the second closest, and so on up to 1 point for the fifth closest example in case it matches the class with the test. If among the five examples several of them corresponded to the test class, points are awarded for each of them.

A system of penalties is used to avoid loss of effectiveness of the ensemble, due to the similarity of the selected classifiers. When calculating the total number of points, the subspace of the classifier signs is compared with the subspaces of the classifier signs that are in the ensemble at the moment. For matching with any sign, a penalty of 20 points is imposed. This value is selected experimentally. After evaluating every 100 random classifiers, the ensemble is updated by including 100 best classifiers from the considered and those included in the ensemble in the previous step. The final ensemble is formed after evaluating 2.000 random classifiers in this way.

Figure 4 shows the use of features in the ensemble in a three-dimensional version of the algorithm. The diagram shows that almost all features are used in the classifiers, and some of them are used much more often than others. Since the classifiers for the ensemble are randomly generated, the appearance of this

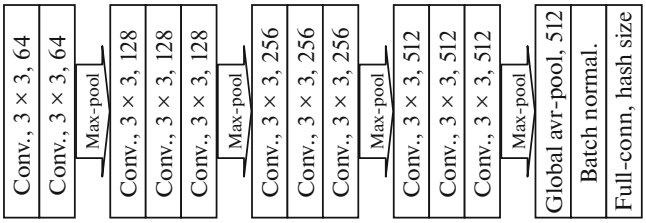

Fig. 5. Scheme of proposed binary visual descriptor construction algorithm.

distribution changes when the ensemble is rebuilt, but the location of the main peaks corresponding to the most significant signs remains unchanged.

The classification results are produced using the weighted vote from the ensemble. Each classifier produces 5 most probable labels, and the class label is assigned a score of 5 for the first place, 4 for the second, and so on until the score of 1 if it was the fifth most probable. Then the total score of each class label is calculated, producing the ranked classification result.

3.2. Appearance-Based Recognition

According to the results of the survey, the VGG16 algorithm was chosen as the base algorithm, which was pretrained on the ImageNet task. It consists of 5 convolution layers with max-pooling and 3 fully connected layers. At the input it receives an image of $224 \times 224 \times 3$ in size, at the output – the probability of belonging of each image to a particular class of 1000 classes. The proposed modification is shown in Fig. 5 and is described in detail below. The source code is available at https://github.com/vslutov/reidentification.

The training data is randomly divided into training and validation samples. The ratio of the number of examples in the training and validation sample is 9/1.

3.2.1. Adaptation to the Task of Re-Identification of a Person. The algorithm uses input images with 128×64 resolution, so only convolution layers from the ImageNet pretrained model VGG are used [22]. In the course of the experiments, it has become clear that when the deepest convolves and the max-pooling layer are removed, the accuracy of the algorithm increases. This may be due to the fact that in this task low-level properties such as color and texture of clothing are more important.

A GlobalAveragePooling layer was added at the end of the network. This is an easy way to get a short feature vector of an input image. Also, a layer of batch normalization was added at the end of the network [21]. This modification increased the accuracy of the algorithm and allowed to carry out naive binarization by simple comparison with zero. The hash length for

such binarization is equal to the number of outputs of the batch normalization layer, that is, 512.

A single fully connected layer with the softmax activation function and the number of neurons equal to the number of classes in the training sample was added at the end of the network – for Market1501 [23] this is 751. This layer is called classifier, and it is used only during training. During the training, an identification error was used as an error function: multiclass cross-entropy of the outputs of the classifying layer and the expected result. The real descriptor is the output of the batch normalization layer, it had the dimension of 512 real numbers.

3.2.2. Neural Network Training. For training a nadam optimizer was used [24]. The size of the batch was 128 images. If during 4 eras the error in the validation sample did not decrease, then the training speed was reduced by 10 times. If during 10 epochs the error in the validation set did not decrease, then the training was completed.

First, the training of all layers was cut off, except for the last fully connected, and trained up to 50 epochs. Then the training of all convolution layers was included and once again up to 50 epochs of learning were conducted.

This method of training showed an accuracy of Rank 1 85%, which is comparable with the best-known methods of solving this task. It may be concluded that in this task low-level signs keep the basic necessary information and further complication of architecture is impractical.

3.2.3. Construction of Binary Outputs. The hash_size parameter is the number of bits in the output binary descriptor. We added one more layer to the network obtained at the previous stage – a fully connected layer with hash_size neurons and a sigmoid activation function. This layer is called binarizing. It was placed after the batch normalization layer, but before the classifying layer. At the same time, the classifying layer was reinitialized and the whole network was trained again according to the scheme proposed above, with the initial approximation set to the weights obtained at the previous stage. The binary descriptor or hash is the result of comparing the outputs of the binarization layer with 0.5, it has the dimension of hash_size bits. By changing this parameter, we checked the hashes with the length of 128, 256 and 512 bits respectively.

The proposed binarization method repeated the sigmoid algorithm from article [11], except that in the proposed method, the modified VGG16 algorithm is used as the basic architecture of the neural network instead of the algorithm from the article.

4. EXPERIMENTS

4.1. Gait Recognition

The TUM GAID [25] dataset and dataset used in [26] (further referred to as Kinect dataset) were chosen for experimental evaluation of the gait recognition algorithm. Rank 1 and Rank 5 percentages of correctly classified sequences were chosen as evaluation metrics. A video sequence counts as correctly classified by Rank k metric if the list of k most probable classes contains the correct class.

Table 1. Dataset comparison

Dataset	CUHK03 [27]	Market1501 [23]
People count	1467	1501
Examples by person	2–10	≈15
Image size	≈160 × 60	128 × 64
Camera count	2	6

Performance on the Kinect dataset was evaluated with 10-fold cross-validation. On the GAID dataset performance was evaluated using the standard procedure for this dataset described in [25]: for each of the 155 classes from the test subset first four examples make up the training set and the rest make up the testing set.

4.2. Appearance-Based Recognition

A collection of Market1501 data was selected for the experimental evaluation of recognition by appearance. A collection of CUHK03 data was selected to evaluate the quality of the results transfer to another dataset [27]. A detailed comparison of these collections is given in Table 1.

1. Test protocol

Two test protocols presented on the selected collection were considered separately [23].

The query quality assessment protocol for one image consists of the following steps:

1. Training of the algorithm for extracting the human image descriptor solely on the training sample.

2. Building of the descriptor database – each picture is matched with a descriptor using the algorithm under test. In the test sample, there are 750 persons and 2 classes with objects that are not people. The images are distributed by classes on a roughly even basis.

3. Building of query descriptors for images from a query sample. A total of 3368 queries. The images from the query sample have not been used before.

4. Calculation of quality according to the metrics described in Section 4.2.2.

The query for multiple images is different in that the queries consist of a set of images: all images from the query_set corresponding to the particular person taken from the specific camera; most often it is 2–6 images. The actions are the same except for clause 3.

Table 2. Gait recognition methods comparison

	Rank 1	Rank 5
Kinect [26]	93.9	100
GAID-N [25]	95.8	98.7
GAID-B [25]	76.5	93.2
GAID-S [25]	87.8	94.5

Table 3. Comparison of real descriptor construction methods, without binarization, single query

Algorithm	Rank 1, %	Rank 5, %	mAP, %
GOG [3]	58.6	79.4	–
Proposed	85.33	99.55	51.42
ResNet + TripletLoss [14]	86.67	93.38	81.07
MobileNet + DML [15]	87.73	–	68.83

Table 4. Comparison of real descriptor construction methods, without binarization, multiple query

Algorithm	Rank 1, %	Rank 5, %	mAP, %
ResNet + TripletLoss [14]	91.75	95.78	87.18
MobileNet + DML [15]	91.66	–	77.14
Proposed	92.91	99.73	62.10

Table 5. Comparison of binarization methods. Represented Rank 1, higher is better

Algorithm	512 bit	256 bit	128 bit
Naive binarization	83.72	–	–
PCA [19]	78.50	75.15	68.34
Random forest [18]	83.72	77.46	65.17
Algorithm DBE [20]	83.90	78.50	71.79
Proposed	83.40	79.15	75.20

5. First, for each image in the query a descriptor is built, then in order to build the query descriptor the descriptors in the set are combined with the help of some algorithm; the average for each element is used in the proposed algorithm.

4.2.2. Quality Metrics. To determine the quality of the algorithm, we used the Rank 1 and Rank 5 metrics. For ranking in these metrics, the distance between real and binary descriptors by metrics L_2 and L_1 respectively was used. The Mean Average Precision (mAP) measure was used for a part of the implementations [28].

The implementation of re-identification algorithms is known on the selected reference data collection. For existing implementations, we used the accuracy described in the original papers. The proposed solution was compared with the following:

1. The best approach for this task at the moment, which does not use GOG neural networks [3].

2. The best approaches for this task at the moment based on ResNet [14] and MobileNet [15] neural networks.

The same test protocol was implemented on CUHK03 to check the portability of the basic algorithm from the Market1501 collection to the CUHK03 collection [27] and back.

5. RESULTS

5.1. Gait Recognition

The algorithm has been evaluated on two datasets: the dataset used in [26] and the TUM GAID dataset [25]. The first one was used to evaluate the three-dimensional version of the dataset and the second one was used to evaluate the two-dimensional version. The results are presented in Table 2.

The "Kinect" line represents the dataset used in [26] and [10], and the following lines represent different subsets of the training set in GAID dataset: GAID-N are normal walking sequences, GAID-B are sequences with backpack (which noticeably affects the silhouette and gait) and GAID-S are sequences where the person is wearing coating shoes. As seen from the table, the algorithm shows good recognition rate both with the depth map and without it. That means that the recognition quality with this method depends more on the skeleton extraction quality than on depth information availability.

5.2. Appearance-Based Recognition

Experimental comparison with the best implementations at the moment (Tables 3, 4) has shown that the proposed method has comparable accuracy to the state-of-the-art in Rank-1 metric with a query from a single image. And with the query from several images, the proposed algorithm has coped better than the known analogs, which demonstrates its applicability in the processing of video sequences where several images are used.

An experimental test (Table 5) showed that the best solution for a 512-bit descriptor is the DBE algorithm, but the naive approach is not much inferior to it. When the descriptor is reduced, the best result is demonstrated by the method of the sigmoid.

The experimental evaluation (Fig. 6) has showed that the algorithm prepared on the data from Market1501 is applicable to other data obtained from an independent source. The accuracy of Rank 1 has decreased by less than 1%. However, the algorithm prepared on the data from CUHK03 loses in accuracy at transition to Market1501, which shows that the reference collection CUHK03 is not sufficiently diverse.

It should be noted that images in different reference collections have different resolution, and people

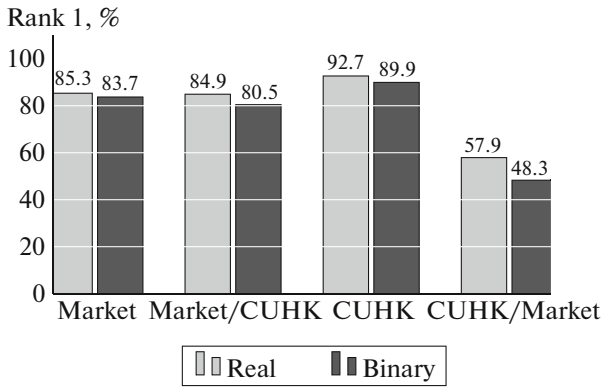

Fig. 6. Transfer to other independent data possibility evaluation.

occupy a different percentage of the frame in them. The question of the dependence of the quality of training on these additional parameters of the reference collection requires additional research in the future. At the moment it cannot be solved due to the small number of collections of sufficient size for training neural network algorithms.

6. CONCLUSION

In this paper we have proposed two complementary methods for re-identification of a person: by appearance and by gait.

Gait recognition algorithm based on [10] uses three types of features for classification: anthropometric features, relative distance features and motion features. The algorithm was evaluated on two datasets and shows good recognition accuracy both with additional information from the depth map and without.

The algorithm for recognizing a person by appearance has coped with the task of searching for several images better than analogues in quality of Rank 1, Rank 5 and mAP on a reference collection Market1501. The binary modification yielded to the basic algorithm in terms of accuracy (83.90% vs. 85.33% in the Rank 1 metric), but the resulting descriptors take 1-2 order less memory. The portability test has shown that the proposed algorithm can be used on data obtained from an independent source.

7. ACKNOWLEDGMENTS

The project was executed with the partial support from the RFBR, grant no. 16-29-09612 OFI-M "Research and development of methods for biometric identification of a person by gait, gestures and constitution in the data of video surveillance".

REFERENCES

1. Kuplyakov, D., Shalnov, E., and Konushin, A., Markov chain Monte Carlo based video tracking algorithm, *Program. Comput. Software*, 2017, vol. 43, no. 4, pp. 224–229.
2. Shalnov, E., Gringauz, A., and Konushin, A., Estimation of the people position in the world coordinate system for video surveillance, *Program. Comput. Software*, 2016, vol. 42, no. 6, pp. 361–366.
3. Srikrishna Karanam, Mengran Gou, Ziyan Wu, et al., A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets, 2016, Preprint arXiv:1605.09653.
4. Man Ju and Bhanu Bir, Individual recognition using gait energy image, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, vol. 28, no. 2, pp. 316–322.
5. Chhatrala Risil and Jadhav Dattatray, V., Gait recognition based on curvelet transform and PCANet, *Pattern Recognit. Image Anal.*, 2017, vol. 27, no. 3, pp. 525–531.
6. Bobick, A.F. and Johnson, A.Y., Gait recognition using static, activity-specific parameters, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, Hawaii, IEEE, 2001, vol. 1, pp. I-423–I-430.
7. Lee, L., Grimson, W., and Eric, L., Gait analysis for recognition and classification, *IEEE Proc. of the Fifth Conf. Automatic Face and Gesture Recognition*, 2002, pp. 155–162.
8. Dimitris Kastaniotis, Ilias Theodorakopoulos, Christos Theoharatos, et al., A framework for gait-based recognition using Kinect, *Pattern Recognit. Lett.*, 2015, vol. 68, pp. 327–335.
9. Rouzbeh Sohrab and Babael Mahdi, Human gait recognition using body measures and joint angles, *Int. J.*, 2015, vol. 6, no. 4, pp. 2305–1493.
10. Yang, K., Dou, Y., Lv, S., et al., Relative distance features for gait recognition with Kinect, *J. Visual Commun. Image Representation*, 2016, vol. 39, pp. 209–217.
11. Lin, K., Yang, H.-F., Hsiao, J.-H., et al., Deep learning of binary hash codes for fast image retrieval, in *IEEE Proc. of the Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 27–35.
12. Zheng, L., Yang, Y., and Hauptmann, A.G., Person reidentification: Past, present and future, 2016, Preprint arXiv:1610.02984.
13. Matsukawa, T., Okabe, T., Suzuki, E., Sato, Y., Hierarchical Gaussian descriptor for person re-identification, in *IEEE Proc. of the Conference on Computer Vision and Pattern Recognition*, IEEE, 2016, pp. 1363–1372.
14. Hermans, A., Beyer, L., and Leibe, B., In defense of the triplet loss for person re-identification, 2017, Preprint arXiv:1703.07737.
15. Zhang, Y., Xiang, T., Hospedales, T.M., et al., Deep Mutual Learning, 2017, Preprint arXiv:1706.00384.
16. Morozov, F. and Konushin, A., Background subtraction using a convolutional neural network, *Proceedings of the 26th International Conference on Computer Graphics and Vision GraphiCon*, 2016, pp. 445–447.
17. Lin, W. and Yang, W., Structured deep hashing with convolutional neural networks for fast person re-identification, 2017, Preprint arXiv:1702.04179.

18. Breiman, L., Random forests, *Mach. Learn.*, 2001, vol. 45, no. 1, pp. 5–32.
19. Halko, N., Martinsson, P.G., and Tropp, J.A., Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.*, 2011, vol. 53, no. 2, pp. 217–288.
20. Liu, L., Rahimpour, A., Taalimi, A., et al., End-to-end binary representation learning via direct binary embedding, 2017, Preprint arXiv:1703.04960.
21. Ioffe, S. and Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.
22. Simonyan, K. and Zisserman, A., Very deep convolutional networks for large-scale image recognition, 2014, Preprint arXiv:1409.1556.
23. Zheng, L., Shen, L., Tian, L., et al., Scalable person re-identification: A benchmark, in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1116–1124.
24. Dozat, T., *Incorporating Nesterov momentum into adam*, 2016. http://cs229.stanford.edu/proj2015/054_report.pdf.
25. Hofmann, M., Geiger, J., Bachmann, S., et al., The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits, *J. Visual Commun. Image Representation*, 2014, vol. 25, no. 1, pp. 195–206.
26. Andersson, V. and Ara'ujo, R., Person identification using anthropometric and gait data from kinect sensor, in *Proceedings of the Twenty-Ninth Association for the Advancement of Artificial Intelligence Conference*, AAAI, 2015, pp. 425–431.
27. Li, W., Zhao, R., Xiao, T., et al., Deepreid: Deep filter pairing neural network for person re-identification, in *IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Columbus, 2014, pp. 152–159.
28. Wikipedia contributors, Information retrieval – Wikipedia, The Free Encyclopedia, 2017. Accessed January 18, 2018. https://en.wikipedia.org/w/index.php?title=Information_retrieval&oldid=815034293.