Insertion of 2D Graphics into a Scene Captured by a Stationary Camera

Son Hai Nguyen*
Supervised by: Adam Herout†

Faculty of Information Technology
Brno University of Technology
Brno / Czechia

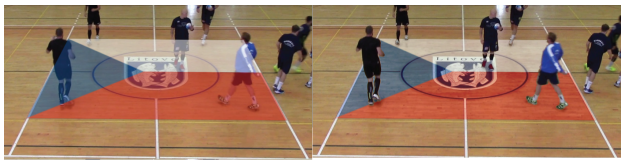
Abstract

Augmented reality visualizes additional information in real-world environment. Main goal is achieving natural looking of the inserted 2D graphics in a scene captured by a stationary camera with possibility of real time processing. Although several methods tackled foreground segmentation problem, many of them are not robust enough on diverse datasets. Modified background subtraction algorithm ViBe yields best visual results, but because of the nature of binary mask, edges of the segmented objects are coarse. In order to smooth edges, Global Sampling Matting is performed, this refinement greatly increased the perceptual quality of segmentation. Considering that the shadows are not classified by ViBe, artifacts were occurring after insertion of segmented objects on top of the graphics. This was solved by the proposed shadow segmentation, which was achieved by comparing the differences between brightness and gradients of the background model and the current frame. To remove plastic look of the inserted graphics, texture propagation has been proposed, that considers the local and mean brightness of the background. Segmentation algorithms and image matting algorithms are tested on various datasets. Resulted pipeline is demonstrated on a dataset of videos (sports and other).

Keywords: Augmented Reality, Computer Vision, Image Processing

1 Introduction

In recent years augmented reality has been broadly used in various applications such as video games, designing or sport broadcasting. Existing algorithms [5, 14] addressing this problem have been mainly focused on stitching graphics to the ground due to moving camera. Natural look of the inserted graphics is not discussed at all. Thus inserted graphics look too plastic. Therefore this paper is focused on natural look of the inserted virtual 2D graphics, resulting in graphics that give the impression, that it has been painted on the surface of the ground (Figure 1).



(a) (b)

Figure 1: (b) **Example** of the result obtained from the proposed pipeline. (a) Result without texture propagation and foreground segmentation.

With increasing resolution and color accuracy, many works are focused towards better accuracy, unfortunately at the expense of processing time. Thus, these methods can not be used in the proposed pipeline, since it aims to be possible of real time processing. Another criterion is keeping the integrity of the segmented objects, which is more important than overall precision. Due to this fact, used algorithms can not be chosen only by metrics.

Proposed pipeline should result in more natural looking of virtual graphics, that can be used in sport broadcasting, visualizing sponsors or virtual advertisement.

In this paper, we propose flexible and robust system for natural looking graphics insertion using ViBe [2], which is then refined by shadow removal (Section 3.1.2) and smoothed by Global Sampling Matting method. Eventually graphics is refined using texture propagation described in Section 3.3.2. In Section 4 proposed pipeline is demonstrated on various datasets.

2 Related Work

In the past several years, numerous foreground segmentation and image matting methods has been proposed. Nonetheless combinations with image matting algorithms were rarely tested.

*xnguye16@stud.fit.vutbr.cz
†herout@fit.vutbr.cz

2.1 Segmentation

Since a majority of sports are played by humans, human segmentation can be adopted. Pose machines algorithms [19, 4] use multi-stage Convolutional Neural Network (CNN) to generate confidence map of joints locations, better precision can be achieved by using more stages, at the expense of speed. Afterwards joints with high connection confidence are connected. To address joints ambiguity in crowds of people, Cao et al. [4] uses part affinity fields, which are predicted using CNN as well. Nonetheless such approach can be used only to refine the segmentation mask, since it generates only skeleton of persons.

Although with DensePose [15] release, pose machine can be utilized as standalone foreground segmenter. It is based on DensePose-RCNN architecture which uses ResNet50 [11] as region of interest (ROI) detector. Although it was trained on the large DensePose-COCO [15] dataset, it fails to detect body parts with a skin color similar to the background, see Figure 10.

DensePose classify every body part, however such information is not required, segmentation mask is sufficient, thus CNN-based semantic segmentation such as DeepLabv3+ [6] can be used. It is based on the encoder-decoder architecture. The decoder is fairly simple, the encoder part is on the other hand more complex and flexible. Any deep CNN could be used as an encoder part. Output from atrous spatial pyramid pooling is added to the decoder input.

Another approach is background subtraction, which models the background and then compare it with the current frame to detect changes. FgSegNet [13] is CNN-based background subtraction method, which utilizes autoencoder architecture with triplet of CNNs as the encoder. Although it has best performance in ChangeDetection challenge, it needs to be retrained on each category, which makes it impractical in situations, where camera need to be often repositioned.

ViBe [2], MoG [18] and SubSENSE [17] are classical landmark algorithms. Unlike MoG which models background pixels using multiple gaussians, ViBe as well as SubSENSE model background using multiple samples. Nonetheless all background subtraction have to handle sudden movement of static object, previously classified as background, this phenomenon (referred to as a ghost) leaves behind static hole in the background, that will be misclassified as foreground.

ViBe use the L2 norm to compute the color distance between the background samples $\mathbf{M_i}$ and the frame $\mathbf{I}$, pixels with the color distance larger then the threshold T_D, over T_{SC} samples are marked as a foreground in the segmentation mask $\mathbf{S}$ (Equation 1). ViBe+ [7] propose refinement of the segmentation mask using morphological operations, and usage of different function computing color distance, however our implementation of proposed color distance does not yield similar results as in Van Droogenbroeck et al. [7], so only refinement using the morphological operations is used. Classical ViBe refined by morphological operations is addressed as **ViBe#**.

SubSENSE [17] is similar to ViBe [2], yet it uses L1 norm to the compute color distance and LBSP [3] features to detect camouflaged objects, nonetheless it does not exploit the LBSP features to classify a shadow. Only MoG from mentioned algorithms separate shadow.

$$\mathbf{S} = \begin{cases} foreground, & (\sum_{i=1}^{n}[||\mathbf{M_i} - \mathbf{I}|| > T_D]) >= T_{SC} \\ background, & \text{otherwise.} \end{cases} \tag{1}$$

ViBe authors set T_D, T_{SC}, n to $T_D = 20$, $T_{SC} = 2$, $n = 20$. Colors are in the range $[0, 255]$.

Chroma Key computes the color distance as well, nonetheless is does not model a background. Color distance is computed from the keyed color $\mathbf{K_i}$, and colors similar to the keyed color are classified as a background. It can be expressed as a sum of differences over all the channels (Equation 2).

$$\mathbf{S} = \begin{cases} foreground, & (\sum_{i=1}^{3}|K_i - \mathbf{I_i}|) > T \\ background, & \text{otherwise.} \end{cases} \tag{2}$$

$\mathbf{I}$ denotes input frame.

2.2 Alpha Matting

Alpha matting refers to the problem of softly and accurately extracting the foreground from an image [10]. Specifically the algorithm determine an alpha mask α in order to create a composition $\mathbf{C}$ of the foreground image $\mathbf{F}$ and the background image $\mathbf{B}$:

$$\mathbf{C} = \mathbf{F}\alpha + \mathbf{B}(1 - \alpha) \tag{3}$$

The input of the alpha matting algorithms is trimap, which specifies foreground (white color), background (black color) and unkown pixels (gray color).

Methods like Closed Form [12] which can be categorized as propagation-based method, yields results better than the sampling-based methods, however only the sampling-based and the CNN-based methods can be optimized to real time performance. Well known sampling based methods are Global Sampling [10] and Shared Matting [9]. Sampling methods create samples set from known pixels and then compute alpha matte [10].

Global sampling matting [10] create samples from the border with an unknown area. Random samples are added to the sample set as well. For every tested sample, the cost function is computed to determine fitness of the sample in the matting equation (Equation 3). The best pair of sample (F, B) is found using the SamplePatch algorithm.

In the bechmark conducted by Erofeev [8], DeepMatting has the best performance [20], is uses an autoencoder network. The encoder is consisted of layers from VGG-16 [16] architecture.

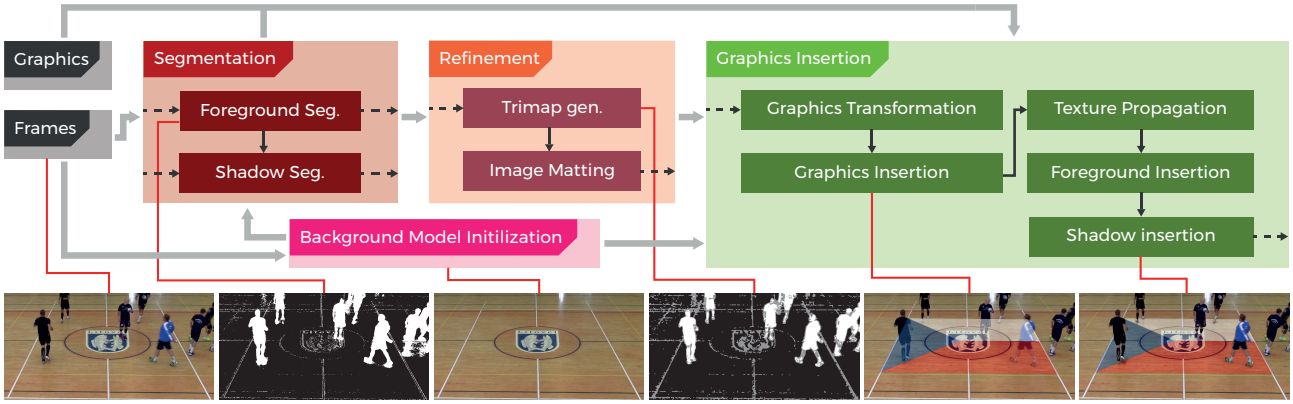


Figure 2: Pipeline is divided into three major tasks, segmentation, segmentation refinement, graphics insertions. Segmentation is performed by **ViBe** [2], then the segmentation mask is refined using open and close morphological transformations, followed by shadow removal from the segmentation mask (described in Section 3.1.2). Next trimap is generated from the edges of the segmented objects as specified in Section 3.2.1. Afterwards **Global Sampling Matting** [10] uses the trimap and the input frame, to smooth the segmentation mask. Inserted graphics is transformed to match the position and the shape defined by a user. Then the graphics is added to the input frame. **Texture is propagated** from the modeled background to the graphics, see Section 3.3.2. The remaining task is to bring foreground objects with their shadows on top of the inserted graphics.

3 Proposed pipeline

The pipeline is divided into three major tasks, coarse segmentation, segmentation refinement, graphics insertion (Figure 2). The pipeline accepts an input frame and an inserted graphics as an input.

The coarse segmentation step segments the foreground objects, that will be brought on top of the inserted graphics. After coarse segmentation is executed, refinement of the segmentation mask is performed. Then the inserted graphics, foreground objects and the current frame are composed.

3.1 Coarse Segmentation

Considering that this work aims on sport broadcasting, human segmentation can be obtained by a pose estimation [19, 4] in the combination with superpixelization [1]. However the result shown in Figure 3 is not optimal, thus this approach is not tested furthermore.

Input image (b) Output

Figure 3: Human segmentation using Convolutional Pose Machines (CPM) [19] with SLIC [1]. (b) Red area denotes human pose estimated by CPM, white area stands for resultant segmentation.

(a) First frame (b) Input frame (c) (d)

Figure 4: Ghosts can appear in the foreground mask created by the background subtraction algorithms. If these methods are initialized simply with the first frame, multiple ghosts can occur, which leads to a misclassified background area as can be be seen in (d). (a) First frame of the JNZP dataset. (b) Frame number #21. (c) Segmentation mask produced by ViBe initialized with a computed background, see Section3.1. (d) Mask from ViBe with the first frame initialization. Red area denotes a ghost merged with a foreground object.

3.1.1 Segmentation

As mentioned in Section 1, preservation of the integrity of the segmented objects is crucial for the perceptual accuracy. As can be seen in Figure 10 ViBe preserve objects integrity, although a lot of noise is classified as a foreground. Fortunately, the misclassified noise forms only small blobs, which can be removed by applying the same morphological operations as in Van Droogenbroeck et al. [7] (Open morphology with kernel size 4x4, followed by Close morphology with kernel size 3x3). **ViBe#** refers to ViBe refined by morphological operations. **ViBe#** yielded overall the best results as shown in Table 1 and Figure 10,

thus it is used in the proposed pipeline.

3.1.2 Shadow detection

As mentioned in Section 2.1, ViBe [2] does not distinguish the foreground objects from their shadows. Thus a simple shadow detector is proposed. It compares a segmented objects color $\mathbf{I_i}$ and texture (Equation 5) with the modeled background $\mathbf{M_i}$, where index i denotes channel. Color distance is measured only from A and B components in the LAB color space as can be seen in Equation 4. $\mathbf{O}$ is the resulted shadow intensity. Result from this method can be seen in Figure 5.

(a) (b) (c)

Figure 5: Shadow comparison on the PETS 2009 dataset. (a) Without shadow segmentation artifacts are present. (b) Result with the proposed shadow segmentation, see Section 3.1.2. (c) Smoothed out shadow using Gaussian blur with the kernel size $5x5$ and $\sigma = 5$. Shadow intensity was three times increased to enhance visibility in figure.

$$\Delta\mathbf{D} = \sqrt{\sum_{i\in(A,B)} (\mathbf{I_i} - \mathbf{M_i})^2} \tag{4}$$

where $\mathbf{A}, \mathbf{B}$ denote image components of the LAB color space.

$$\Delta\mathbf{G} = ||\mathbf{G_I} - \mathbf{G_M}|| \tag{5}$$

where $\mathbf{G_I}$ and $\mathbf{G_M}$ are gradients of the grayscaled frame and background. Gradients were computed using Sobel operator.

$$\mathbf{O} = \begin{cases} \mathbf{L_{bg}} - \mathbf{L_{frame}}, & \Delta\mathbf{D} < T_C \wedge \Delta\mathbf{G} < T_G \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

thresholds T_C, T_G were experimentally set to $T_C = 8$ and $T_G = 60$.

3.1.3 Background model initialization

ViBe is also greatly sensitive to the background model initialization, nonetheless it can be exploited to increase the short term performance by initializing the background model with a background analogous to the groundtruth background resulting in the ghosts absence in the beginning of the processing. Such initialization can be accomplished by computing a median of a few hundred frames

(a) LITOVEL_01 (b) ALFHEIM (c) PETS09 (d) JNZP

Figure 6: Background computed by median of the first 300 frames. On very dynamic scenes 300 frames is enough to estimate the background, however when foreground objects do not move enough, some parts of the foreground objects can be classified as background, see highlighted red area.

(6). As can be seen in Figure 6 300 hundred frames proved sufficient for the dynamic datasets. Due to slow decay rate of the foreground pixels, ghosts disappearance would take longer than with the mentioned background model initialization, see Figure 4.

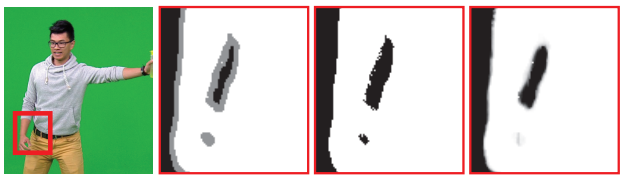
Mode-based background modeling [21] is faster than median-based, as can seen in evaluation in Zheng et al. [21]. Mode needs fewer frames than a median to produce an accurate background model. However the chosen method does not matter, since the background initialization is supervised by a user, and as can be seen in Figure 6 median produces sufficient results.

3.2 Refinement

Coarse segmentation provides only a binary mask, which causes sharp edges of the foreground objects, as can be seen in Figure 7. Therefore smoothing of the segmentation mask is performed.

3.2.1 Trimap generation

After shadow is removed from the foreground mask, trimap is automatically generated by applying the Sobel operator, edges with magnitude lower than 10 are removed, threshold $T_E = 10$ was empirically set. Afterwards a dilatation (with kernel size $3x3$) is used to thicken the unknown area in the trimap. Example of a generated trimap can be seen in Figure 7.



(a) Input frame (b) Trimap (c) ViBe (d) DeepMatting

Figure 7: Comparison of binary mask produced by Vibe [2], and alpha mask created by DeepMatting [20]. (a) Red area refers to the zoomed region. (b) Trimap generated from edges, described in Section 3.2.1. (c) As can be seen, edges of the segmented object are coarse. (d) Smoothed segmentation mask using DeepMatting.

3.2.2 Alpha Matting

In order to the increase perceptual quality of the segmentation, alpha matting algorithms are used. Alpha mattings algorithms take a trimap as an input and produce an alpha mask α for the segmented objects. Global Sampling Matting (GSM) [10] and Closed Form Matting (CFM) [12] are well known matting algorithms. CFM [12] in comparison with GSM [10], is more prone to outliers as can be seen in Figure 8, despite that CFM [12] results is more accurate in VideoMatting challenge [8]. Unfortunately CFM [12] can not be accelerated to real time processing, so GSM [10] is used in the proposed pipeline.

CNN-based method DeepMatting [20] yields the best results in the evaluation conducted by Erofeev et al. [8]. It can achieve real time processing, however only on the more powerful GPU than the authors of this paper have used, as mentioned in Section 4.

Figure 8: Alpha matting algorithms comparison evaluated on the PETS 2009 dataset. CFM [12] is more prone to outliers than GSM [10].

3.3 Graphics insertion

Firstly, coordinates of the inserted graphics $\mathbf{x}$ are transformed using the homography $\mathbf{H}$ to the user defined coordinates $\mathbf{x}'$ using Equation 7.

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \tag{7}$$

3.3.1 Composing

After the graphics transformation, it is added to the frame, followed by the foreground objects and the shadows. In order to add shadows, graphics must be converted to the LAB color space, then shadows are added to the L component of the inserted graphics.

3.3.2 Texture Propagation

With a plain insertion of the graphics, background texture is suppressed. Naive texture propagation can be achieved by lowering the opacity of the graphics, however it lowers the visibility of the graphics, see Figure 9. Proposed solution is to propagate the texture of the background considering local and mean brightness of the background, see Algorithm 1.

Algorithm 1 Texture propagation algorithm

procedure PROPAGATETEXTURE
  $G \leftarrow graphics$
  $Bg \leftarrow modeled\ background$
  $GL, GA, GB \leftarrow LAB(G)$
  $AvgBg \leftarrow AverageColor(Bg)$
  $AvgBgL \leftarrow ComponentL(LAB(AvgBg))$
  $BgL \leftarrow ComponentL(LAB(Bg))$
  $DiffL \leftarrow BgL - AvgBgL$
  $GL \leftarrow Clamp(GL + DiffL, 0, 255)$
  $G \leftarrow Merge(GL, GA, GB)$

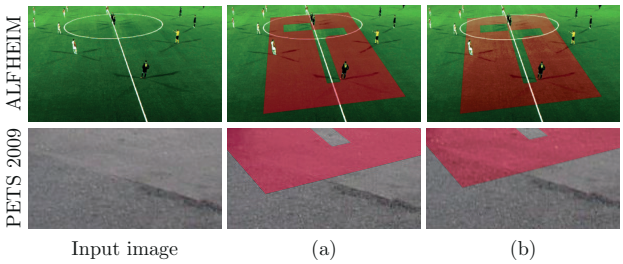


Figure 9: Texture propagation enhances natural appearance of the inserted graphics. (a) Naive texture propagation using lowering opacity of the graphics. (b) Proposed texture propagation algorithm using Algorithm 1.

4 Results

As can be seen in Figure 10 DensePose [15] and DeepLabv3+ [6] fails to segment persons, that are "camouflaged" or small. Suprisingly DeepLabv3+ completely fails in the greenscreen dataset (JZNP). SubSENSE [17] performs quite well, however it sometimes remove details, see Figure 10, LITOVEL_01 column. ViBe is producing more false positives than false negatives, however such result is more desired since the false positives can be removed in the refinement step (Section 3.2). As previously stated the false positives can be removed in the refinement step, unfortunately some matting algorithms are not robust enough to handle correctly false positives. Due to this fact small blobs of false positives are rather removed by morphological operations. As can be seen in Figure 10 (ViBe#) a majority of the false positives is removed.

Although DeepMatting [20] and Closed Form Matting [12] have best results in VideoMatting benchmark [8], DeepMatting [20] processes *320x320* image for *36ms* (on *Nvidia GTX 1070*), thus it not capable of *25fps* processing with the combination of the algorithms used in the pipeline. Only sample-based methods [10, 9] have the potential to achieve desired speed. As can be seen in He et al. [10] Global Sampling Matting (GSM) slightly outperforms Shared Matting [9], therefore GSM [10] is used in the proposed pipeline.

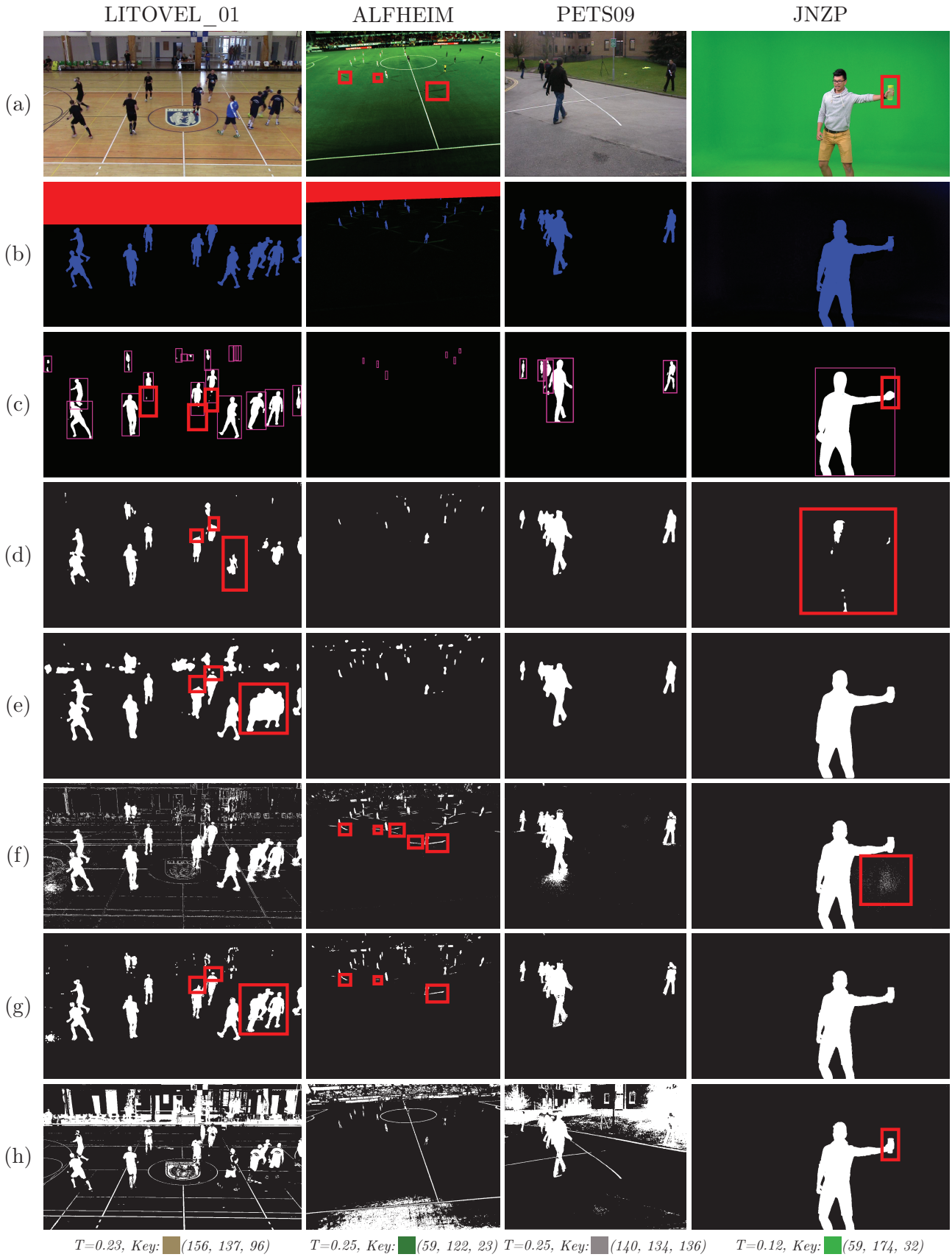


Figure 10: Segmentation evaluation over various datasets. (a) Shows the input frames, (b) shows groundtruths in the ssf format (blue channel = foreground mask, green channel = shadows mask, red = Outside region of interest). Results correspond to the following methods (c)**DensePose**, pink areas are detected ROIs, (d) **DeepLabv3+** [6], (e) **SubSENSE** [17], (f) **ViBe** [2], (g) **Vibe#**, see Section 2.1, (h) **Chroma-key**, specified by Equation 2, the bottom parameters refers to the key color and the threshold which is used for a colors in the range $\langle 0, 1 \rangle$.

Table 1: Comparison of segmentation methods based on RMSE metric. None that RMSE metric is computed only from a single frame.

Dataset	**DensePose** [15]	**DeepLabv3+** [6]	**SubSENSE** [17]	**ViBe** [2]	**ViBe#**	**Chroma-key**
PETS09	31.9649	19.1179	**18.9051**	28.9328	20.1093	113.7187
LITOVEL_01	39.3016	47.2691	54.1864	61.8984	**35.2635**	107.0621
ALFHEIM	19.0938	**15.6315**	20.1774	29.0520	20.4717	94.9830
JZNP	26.8841	68.8796	25.2947	22.6819	19.1658	**15.0399**
Average RMSE	29.3111	37.7245	29.6409	35.6413	**23.7526**	82.7009

(a) VIRAT (b) JNZP (c) LITOVEL_01 (d) PETS09 (e) ALFHEIM

Figure 11: Evaluation performed on VIRAT, JNZP, LITOVEL_01, PETS 2009 and ALFHEIM datasets. First row corresponds to the processed frames. As can be seen on (a), hard shadows are classified as the foreground, however due to fix threshold. However background color nor texture are propagated, thus it does not change the visual appearance of the resulted image. Although it can spotted, that the proposed pipeline still struggles with the colors similar to the background. (b) The pipeline is tested on green screen as well. (c), (e) show usage of the pipeline on the sport datasets.

The described pipeline is written in Python 3 using OpenCV and NumPy packages. Current state of the pipeline is far from real time processing (currently 5s on the *1280x720* frame using *i7-7500U processor*), it does not take any advantage of GPU nor faster programming language. However ViBe has been accelerated on the *Nvidia GTX 1070 GPU* and it can process 180 frames (with resolution *1280x720*) per second.

5 Conclusions

In this paper, experiments were carried out to determine the best combination of existing algorithms to handle the outlined task. Pose Machines combined with superpixelization were found unsuitable for dense human segmentation.

Utilization of ViBe has drastically increased accuracy of the segmentation. Exploiting the background initialization (Section 3.1.3) removes majority of ghosts .

As mentioned in Section 4 the system is not real time, although the real time implementations of the used algorithms are possible. Shadow segmentation method has been proposed, which has similar performance as MoG, although segmentation is failing if the shadow is close to the black, see Figure 11.

Although texture propagation of the background increases the natural look in the resulted image, conversion to the LAB color space is rather complex thus experiments with HSV color space will be conducted.

Despite the results of the suggested pipeline appears relatively satisfying, motion blur and color similarity between the background and the foreground are still challenging for the system.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[2] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.

[3] G. Bilodeau, J. Jodoin, and N. Saunier. Change detection in feature space using local binary similarity

patterns. In *2013 International Conference on Computer and Robot Vision*, pages 106–112, May 2013.

[4] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, July 2017.

[5] R. Cavallaro, M. Hybinette, M. White, and T. Balch. Augmenting live broadcast sports with 3d tracking information. *IEEE MultiMedia*, 18(4):38–47, April 2011.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[7] M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 32–37, June 2012.

[8] Mikhail Erofeev, Yury Gitman, Dmitriy Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 99.1–99.12. BMVA Press, September 2015.

[9] Eduardo S. L. Gastal and Manuel M. Oliveira. Shared sampling for real-time alpha matting. *Computer Graphics Forum*, 29(2):575–584, May 2010. Proceedings of Eurographics.

[10] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. In *CVPR 2011*, pages 2049–2056, June 2011.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[12] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 61–68, June 2006.

[13] Long Ang Lim and Hacer Yalim Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *CoRR*, abs/1801.02225, 2018.

[14] S. Monji-Azad, S. Kasaei, and A. Eftekhari-Moghadam. An efficient augmented reality method for sports scene visualization from single moving camera. In *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, pages 1064–1069, May 2014.

[15] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[17] P. St-Charles, G. Bilodeau, and R. Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, Jan 2015.

[18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252 Vol. 2, June 1999.

[19] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, June 2016.

[20] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, July 2017.

[21] Jianyang Zheng, Y Wang, Nancy L. Nihan, and Mark E. Hallenbeck. Extracting roadway background image: Mode-based approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1944:82–88, 01 2006.