Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume III, 291-296

Computer Vision Approach for Panorama Creation in Military Observation

Krasimir Slavyanov
Department Computer Systems and Technologies
Vasil Levski National Military University
Shumen, Bulgaria
k.slavyanov@nvu.bg

***Abstract*—In this paper the problem of obtaining a complete picture of monitored military objects is resolved with computer vision tools. The research is focused on the necessity for the correct arrangement of photos obtained in different ways to promptly obtain a comprehensive picture of the situation in an area of monitored military operation. The consecutive stages with functional operations are designed in MATLAB environment. Creation of a dataset with detailed images of specific zone of a military operation is described. Speeded-up robust features point detection is followed by descriptor extraction, transformation calculation and other specific computer vision tools. The functions for alpha blending and automatic black edge cropping are implemented for the final panorama creation.**

Keywords— computer vision, panorama creation, speeded-up robust features, transformation calculation

I. INTRODUCTION

This document provides instructions for panorama creation in military observation. For the military reconnaissance operations, it is commonly popular to have a bunch of pictures derived by drone camera or some hidden stationary camera used to make several captures from one point of view. An approach with multiple viewpoint can also be implemented [1], [2]. To analyze whole picture of the observed military objects with their motion and arrangements approaching any possible future status or activity the panorama view is most valued and can be used easier for whole unit's status information analysis. The lower computational burden and faster processing speed, which can meet the real-time requirements for scene matching like in [3] are also valued in this approach. The speeded-up robust features (SURF) are implemented in this approach. SURF is very popular in different approaches for panorama creation [4] and multi-view image registration like in [5],[6] and can be improved in different ways [7], [8],[9], [10].

The implementation of the spatial orientation approach and the precise local positioning approach like in [11] can also be implemented for calculation of the exact time for the photo capturing. Essential for the military reconnaissance, this approach for panorama creation can be used for other implementations where wider observation picture should be created like for agriculture purposes [12].

II. MATERIALS AND METHODS

To obtain a complete picture of monitored military objects, it is necessary to correctly arrange photos obtained in different ways to promptly obtain a comprehensive picture of the situation in the area of monitored military operation.

Panoramic images allow several photos to be combined into one whole image with a wide viewing angle. In the present scenario, a system for automatic panorama creation using the SURF (Speeded-Up Robust Features) method is implemented, using only classical computer vision techniques, without neural networks.

The first stage in the creation of panorama is the creation of a dataset with detailed images of specific zone of a military operation (status). Images can be obtained from a ground camera, a drone camera, or images generated by artificial intelligence, in which one or more military objects are captured. For the research purpose such a dataset is created and displayed on fig.1.

a) Image 1

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.153

b) Image 2

c) Image 3

d) Image 4

e) Image 5

f) Image 6

Fig. 1. A dataset with 6 images of observed military objects.

On the second stage calculations for each pair of images are performed. The images should be selected at first and then put in alphabetical order to follow their automatically generated by the camera firmware timestamp. The experiment is carried out in the MATLAB environment as follows:

A. User selection of images

An array is created with the filenames and their corresponding path names in the memory directory. The function *uigetfile* is used with the supporting functions for the file type check like *isequal* and *ischar*. Sorting in alphabetical order is the next step here. Variables are created for the full paths, images and their numbers [13].

```
[fileNames, pathName] =
uigetfile({'*.jpg;*.png;*.jpeg;*.bmp'
,
    'Image Files (*.jpg, *.png,
*.jpeg, *.bmp)'}, ...
    'Select images to create a
panorama', ...
    'MultiSelect', 'on');
if isequal(fileNames, 0)
    error('No images selected.');
elseif ischar(fileNames)
    fileNames = {fileNames};
end
```

Sorting in alphabetical order:

```
fileNames = sort(fileNames);
```

Creation of full paths, datastore:

```
fullPaths = fullfile(pathName,
fileNames);
images = imageDatastore(fullPaths);
numImages = numel(images.Files);
```

B. Loading the first image

The following steps on this stage are lading the first image and the detection and matching with the rest of the images from selection. These are carried out with the color to gray transform (function *im2gray*) and creation of variables for the picture size and for the homogeneous transformation matrix (function *projective2d*).

```
I = readimage(images, 1);
grayImage = im2gray(I);
imageSize = size(grayImage);
tforms(numImages) =
projective2d(eye(3));
```

C. Detection and matching with the rest of the images from selection

A simple *For* cycle is used for detection and matching with the rest of the images from selection.

```
for n = 2:numImages
% previous image
I1 = im2gray(readimage(images, n -
1));
I2 = im2gray(readimage(images, n));
%continues…
```

D. SURF features detection and processing

Next step on the second stage is the detection of the sppeded-up robust features of the pictures in grayscale. *detectSURFFeatures* is a MATLAB function for finding the blob features. Description: points = detectSURFFeatures(I) returns a SURFPoints object, points, containing information about SURF features detected in the 2-D grayscale or binary input image I. This and the rest of the MATLAB functions can be found explained in the MATLAB documentation [13]:

```
% … continued cycle:
% SURF features detection
    points1 = detectSURFFeatures(I1);
    points2 = detectSURFFeatures(I2);
[features1, validPoints1] =
extractFeatures(I1, points1);
[features2, validPoints2] =
extractFeatures(I2, points2);
indexPairs = matchFeatures(features2,
features1, 'Unique', true);
matchedPoints2 =
validPoints2(indexPairs(:,1), :);
matchedPoints1 =
validPoints1(indexPairs(:,2), :);
fprintf('Image %d: matches found =
%d\n', n, size(indexPairs,1));
  if size(indexPairs,1) < 10
    warning('Not enough matches
between images %d and %d. Skip.', n-
1, n);
        tforms(n) =
projective2d(tforms(n-1).T);
        continue;
    end
```

E. Calculation of design transformation.

The function *estimateGeometricTransform2D* is used to estimate a 2-D geometric transformation between two images by mapping the inliers in the matched points from one image *matchedPoints1* to the inliers in the matched points from another image *matchedPoints2* [13].

```
tform =
estimateGeometricTransform2D(matchedP
oints2, matchedPoints1, ...
        'projective', 'Confidence',
99.9, 'MaxNumTrials', 2000);

    tforms(n).T = tform.T * tforms(n-
1).T;
```

F. Show the transformation matrix

```
disp(['T matrix of an image',
num2str(n), ':']);
    disp(tforms(n).T);
```

The results here are the matrices of values:

Image 2: matches found = 6721
T matrix of an image2:
0.8710 -0.0085 -0.0000
0.0577 0.9349 0.0000
446.2032 14.0935 1.0000

Image 3: matches found = 6574
T matrix of an image3:
0.7816 -0.0034 -0.0000
0.1300 0.8966 0.0000
828.5522 4.7152 0.9880

Image 4: matches found = 3578
T matrix of an image4:
1.0e+03 *
0.0007 0.0000 -0.0000
0.0002 0.0008 0.0000
1.2649 -0.0351 0.0010

Image 5: matches found = 2381
T matrix of an image5:
1.0e+03 *
0.0005 0.0000 -0.0000
0.0003 0.0008 0.0000
1.7456 -0.0498 0.0009

Image 6: matches found = 1982
T matrix of an image6:
1.0e+03 *
0.0004 0.0000 -0.0000
0.0004 0.0007 0.0000
2.0979 -0.1392 0.0008

G. Match preview

The function *showMatchedFeatures* [13] is used for the preview

```
figure;
    showMatchedFeatures(I2, I1,
matchedPoints2, matchedPoints1,
'montage');
    title(['matches between image ',
num2str(n-1), ' and ', num2str(n)]);
    pause(1); % short break for
review
```

The matched SURF features are displayed on fig. 2.

Centering to the middle image and calculating panorama dimensions are the next steps evaluated for the panorama creation process.

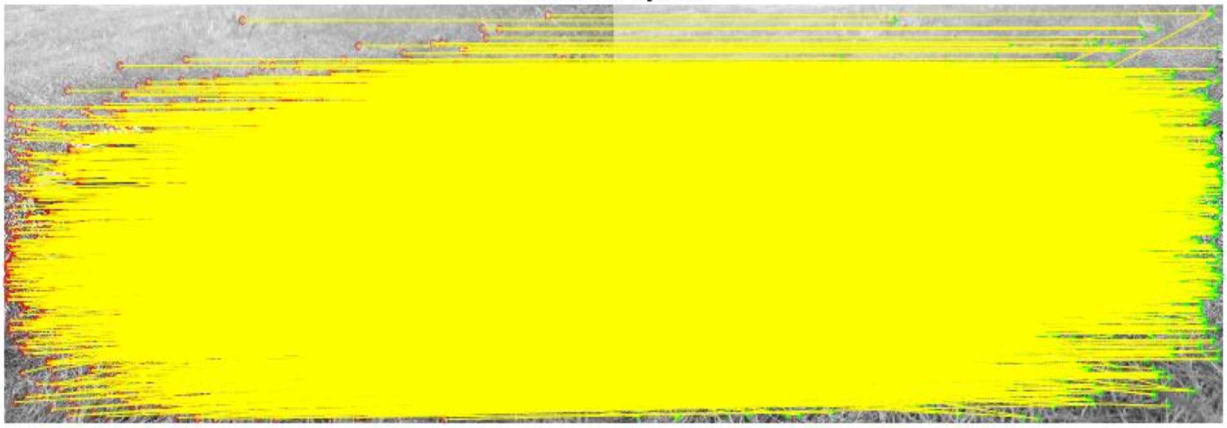

Fig. 2. Matched SURF features between the selected images (shown only between image 1 and image 2)

```
panorama = zeros([height width 3],
'like', I);
blender =
vision.AlphaBlender('Operation',
'Binary mask', ...
    'MaskSource', 'Input port');

for i = 1:numImages

    I = readimage(images, i);
    warpedImage = imwarp(I,
tforms(i), 'OutputView',
panoramaView);
    mask = imwarp(true(size(I,1),
size(I,2)), tforms(i), 'OutputView',
panoramaView);
    panorama = step(blender,
panorama, warpedImage, mask);
    fprintf('Added image %d to the
panorama.\n', i);

end
```

H. Automatic black edge cropping.

The *Imcrop* function used on this stage is an interactive Crop Image tool used to crop the grayscale, truecolor, or binary image displayed in the current figure. imcrop returns the cropped image, Icropped [13].

```
grayPanorama = rgb2gray(panorama);
mask = grayPanorama > 0;

[row, col] = find(mask);

if ~isempty(row) && ~isempty(col)
    minRow = min(row); maxRow =
max(row);
    minCol = min(col); maxCol =
max(col);
    croppedPanorama =
imcrop(panorama, [minCol, minRow,
maxCol-minCol, maxRow-minRow]);
else

    croppedPanorama = panorama; %
fallback

end
```

III. Results and Discussion

For the result showing the MATLAB can easily produce an image of the cropped panorama generated with the code explained above:

```
figure;
imshow(croppedPanorama);
title('Final panorama (cropped and
improved)');
```

For better applicability of the approach, a small but accurate set of captured images of the required quality is required, which must be taken from a properly selected observation point that tracks the movement of objects.

As is described before, even a simple camera can be sufficient for panorama creation. Another implementation of computer vision can be applied afterwards for the military object detection purpose.

The algorithm described can be fruitful for panorama creation even if very detailed pictures are used and as can

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.153

be seen the environment is also very detailed. The Panorama view created is shown of fig. 3.

Fig. 3. Panorama view of the observed military objects.

IV. Conclusions

A common task during reconnaissance missions is to analyze whole picture of the observed military objects with their motion or their predicted arrangements in any possible future status or activity. Thus, the panorama view is most valued and can be used easier for whole unit's status information analysis.

Application creation using this algorithm can furthermore be elaborated where any other previous observation gathering information like predefined object searching or target acquisition can be also implemented for object detection after the panorama creation.

Acknowledgments

This work was supported by the National Science Program "Security and Defence", which has received funding from the Ministry of Education and Science of the Republic of Bulgaria under the grant agreement no. Д01-74/19.05.2022. The material reflects only the author's opinion, and the Ministry of Education and Science is not responsible for the content.

References

[1] G. Ramkumar, T. J. Nagalakshmi, N. Nalini, P. Jagadeesh, P. S. Bharathi and V. Amudha, "An Effective Construction of Multiple Viewpoint based Panorama Design using Digital Image Processing Scheme," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2022, pp. 1-6, https://doi.org/10.1109/ACCAI53970.2022.9752626.

[2] S. Degadwala, U. Chakraborty, P. Kuri, H. Biswas, A. N. Ali and D. Vyas, "Real-Time Panorama and Image Stitching with Surf-Sift Features," 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2021, pp. 1111-1115, https://doi.org/10.1109/ICICT50816.2021.9358586.

[3] Su Juan, Xu Qingsong and Zhu Jinghua, "A scene matching algorithm based on SURF feature," *2010 International Conference on Image Analysis and Signal Processing*, Zhejiang, China, 2010, pp. 434-437, https://ieeexplore.ieee.org/document/5476080 https://doi.org/10.1109/IASP.2010.5476080.

[4] Q. Fu and H. Wang, "A fast image stitching algorithm based on SURF," 2017 IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, China, 2017, pp. 1-4, https://doi.org/10.1109/ICCChina.2017.8330425.

[5] M. S. Patel, N. M. Patel and M. S. Holia, "Feature based multi-view image registration using SURF," 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar, India, 2015, pp. 213-218, https://doi.org/10.1109/ISACC.2015.7377344.

[6] S. Ali and M. Hussain, "Panoramic image construction using feature based registration methods," *2012 15th International Multitopic Conference (INMIC)*, Islamabad, Pakistan, 2012, pp. 209-214, https://doi.org/10.1109/INMIC.2012.6511464.

[7] Cai, P. Wang and Y. -h. Liang, "Fast image stitching based on improved SURF," 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanchang, China, 2016, pp. 411-416, https://doi.org/10.1109/CSCWD.2016.7566024.

[8] X. Li, S. He, X. Gao and J. Zhou, "Research on Container Panorama Image Stitching Method," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 661-664,

https://doi.org/10.1109/CAC.2018.8623589.

[9] L. Juan and G. Oubong, "SURF applied in panorama image stitching," *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, Paris, France, 2010, pp. 495-499, https://doi.org/10.1109/IPTA.2010.5586723.

[10] R. Monali, S. Moonka, A. Priya and S. S. Tripathy, "Effects of noise and relative overlap on image mosaicing using SURF features," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2016, pp. 773-777, https://doi.org/10.1109/RTEICT.2016.7807931.

[11] V. Atanasov, T. Trifonov, "A new model of ultra wideband sensors based interactive system", International Journal on Information Technologies and Security, № 1 (vol. 16), 2024, pp.39-48, ISSN 1313-8251, https://doi.org/10.59035/XYCG3094

[12] M. Iliev and M. Bedzheva, "An Aproach for Application of UAVs for Observation of Processes in Agriculture", 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), 2020, https://doi.org/10.1109/EEAE49144.2020.9279028

[13] https://www.mathworks.com/help/matlab/index.html [Accessed: Apr. 05, 2026].