Shadow Detection and Segmentation on Satellite Images: a Survey

Bin Lei
CETC Les Information System Co., Ltd
China
40220720@qq.com

Wei Wan
CETC Les Information System Co., Ltd
China
1271130252@qq.com

Qing Bu
CETC Les Information System Co., Ltd
China
39020765@qq.com

Stanislav V. Sholtanyuk
The Department of Computer Applications and Systems
Belarusian State University
Minsk, Belarus
ssholtanyuk@bsu.by
ORCID: 0000-0003-0266-7135

***Abstract*—Shadow detection and segmentation are widely used in many computer vision and image processing applications. Shadows on various types of images can provide both positive and negative traits so a researcher can retrieve some useful information or, on the contrary, must get rid of or mitigate some predicaments. In satellite imagery, the problem of shadow detection is of special importance as far as shadows can give useful insights into objects, landscapes, and dynamics of a captured scene, as well as pose some obscurity about objects of a researcher's interest. This survey paper provides a comprehensive exploration of the state-of-the-art techniques and methodologies in the domain of shadow detection and segmentation within satellite imagery. We give descriptions and analysis for ten method and algorithm categories. We also compare them based on the selected aspects: accuracy, complexity, robustness, ability to work with different types of images, and data processing requirements.**

***Keywords*—computer vision, image processing, data processing, satellite images, shadow detection, segmentation, accuracy, robustness**

I. INTRODUCTION

In the realm of remote sensing and image analysis, the utilization of satellite imagery has become indispensable for a multitude of applications, ranging from environmental monitoring to urban planning and disaster management. These high-resolution satellite images provide valuable insights into our dynamic world from above, allowing us to monitor and analyze various aspects of the Earth's surface.

Shadows on satellite images have become one of the key issues in image analysis and computer vision. They impact both positively and negatively on the satellite images processing. Some useful traits of shadows feature enhancing feature visibility and exposing additional structural details of depicted objects; providing additional information on object orientation and shape; aiding energy and solar studies, as well as post-disaster assessment. Of course, shadows also present some challenges, featuring obscuring the features of interest in satellite images like land landscape, object shape and size, etc.; as well as temporal variability (Fig. 1). Those issues lead to necessity of using extra computational efforts and implementing new algorithms.

Recognizing and effectively managing shadows have thus become pivotal in harnessing the full potential of satellite imagery. The detection and segmentation of shadows present a complex yet crucial task, demanding innovative approaches and methodologies that can enhance

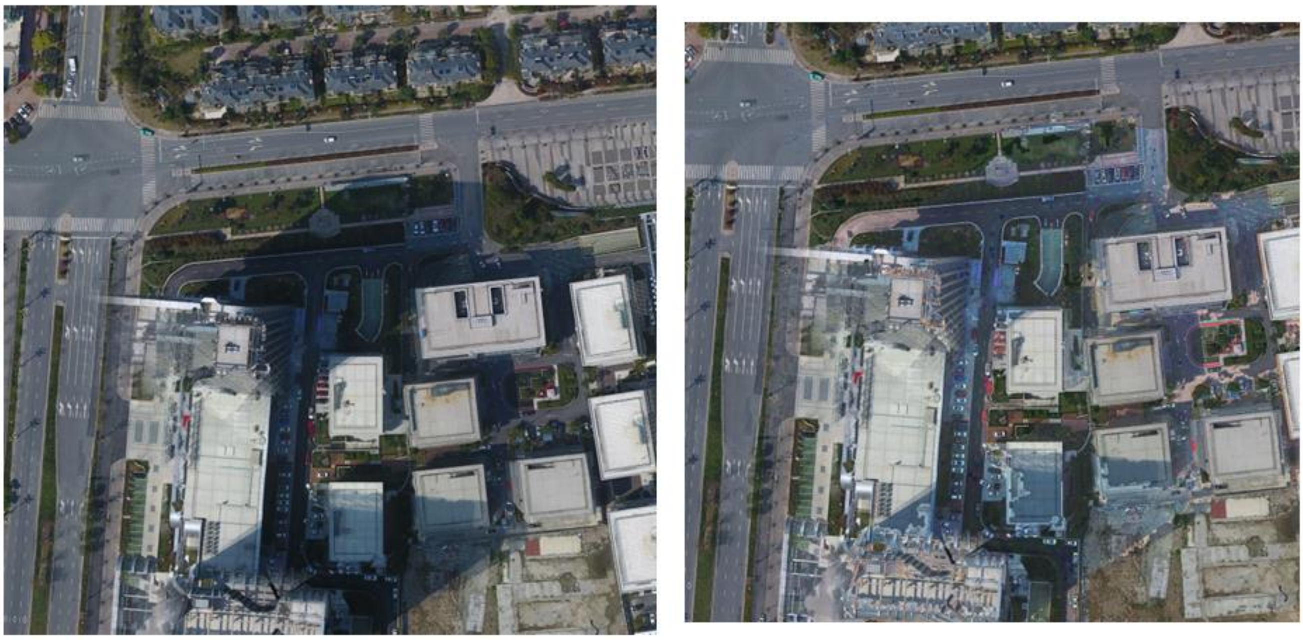


Fig. 1 An image before and after shadow correction

the accuracy and reliability of image-based assessments. Besides, the amount of satellite images has been enlarging fast, so there is no way all of them can be processed by a few universal algorithms. It is one of the reasons of the shadow detection and segmentation algorithms diversity.

II. Definitions Formalizing and Taxonomy

A. Key Definitions

In terms of geometrical optics, shadow is a part of a surface which is blocked from one or multiple light sources ((1) on Fig. 2), partially or completely. However, that area isn't commonly black (even if no light source illuminates it) because of ambient light. Shadow size and shape depend on size and shape of an object (2) blocking light. That is why we can talk about a shadow of an object. There are two types of shadow: self-shadow (3), which is a part of an object surface, and cast shadow (4 and 5), which is a part of some surface nearby the object. Lit area (6) is a part of a surface where light ray from at least one light source fall directly.

Depending on whether an area blocked completely or partially, a cast shadow can be divided into two parts called umbra (4) and penumbra (5) respectively. In satellite imagery, it is common for both to appear, because the Sun isn't a point light source.

In this paper, we considered shadow detection and segmentation algorithms for cast shadows only, no matter whether they work with umbras and penumbras or just with a shadow as a homogeneous object.

B. Taxonomy

Nowadays the broad of shadow detection and segmentation algorithms have been developing which serves various purposes and are used to solve various problems. Their usability depends on several conditions:

- observation domain (indoor, outdoor scenes, as well as observing from a bird's eye view),
- purpose (e.g. urban modeling, landscape researches, traffic monitoring, etc.),
- objects being segmented, and their features,
- camera properties (static/moving, resolution, angle view, position, etc.).

Also, those algorithms can be divided into several categories according to used approaches. Some of them are listed in the next section.

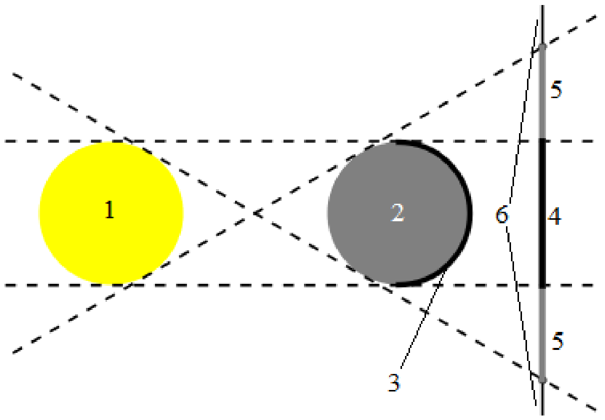


Fig. 2 Types of shadows: (1) lightsource, (2) an object, (3) self-shadow, (4) umbra, (5) penumbra, and (6) lit areas

III. Shadow Detection and Segmentation Algorithms

A. Image Difference

Using image difference is one of the most common approaches for shadow detection and processing on satellite images. It is based on comparison of two captures, one of them being captured before shadow appeared (Image 1), the other one being captured after the shadow appeared (Image 2). This approach is widely used in other computer vision applications [1].

Image difference-based shadow detection algorithm features the next steps:

1) Image normalizing. Before the processing, images are normalized by fetching them to the same scale and contrast. Images can be tramsformed into the same color space, gamma-corrected, or their brightness and contrast can be lineary corrected.

2) Image subtraction. After the normalization, Image 1 is subtracted from Image 2. The resultant difference contains the information about the shadow and its changing.

3) Threshold filtration. In order to separate shadows from other objects, threshold filtration is used upon the image difference. As the result, shadow areas become clearer.

4) Noise reduction conducted by various filter methods like median filter or Gaussian filter.

5) Joining shadow areas in order to get larger shadows.

This algorithm gives good results in buildings shadow detection on satellite images. However, multiple factors such as clouds or lighting changes can significantly impact on the result. That's why the algorithm is usually used with other ones.

B. Threshold Algorithm

Thresholding is one of the simplest and basic segmentation technique that separates objects or regions based on a pixel property (like brightness or intensity) values (Fig. 3). In shadow detection, it can be applied to separate shadow areas from lit ones by setting a threshold value. Then, all pixels having intensity less than the threshold, are claimed to be shadow pixels, and all other pixels make up lit area.

The algorithm features the next steps:

1) Converting the image into another color space. LAB is the common choise due to the fact shadow pixels has are less illuminated, hence have lower value of L component than lit pixels [2].

2) Setting up the threshold value for separating shadows

(a) (b) (c)

Fig. 3 Example of shadow detection and removing by a threshold algorithm: (a) the initial image, (b) shadow mask with the shadow region painted in white, (c) the image after shadow removal [2, Fig. 5] from lit areas. This value can be found either empirically or by calculating automatically [1] based on pixels brightness distribution.

3) Threshold processing. For each pixel, it is determined whether it belongs to a shadow or a lit area, depending on the threshold value.

The algorithm implements a simple idea and can be easily implemented into any image processing software. But as with the previous algorithm, there are many side issues here not considered. Also, the setting up the threshold can be a challenging task, especially if an image has a complicated structure or significant noise. Besides, simple thresholding may not be effective in handling variations in shadow intensity caused by factors like shading, reflections, or surface materials.

C. Edge-Based Segmentation

Edge detection algorithms, such as the Canny [3] and Deriche edge detectors [4], can be used to identify abrupt changes in pixel intensity, which often occur at the boundaries of shadows [5, 6]. Edge-based segmentation can help outline shadow areas, although it may produce fragmented results and require post-processing. Sometimes these algorithms implementations can be simplified so they retrieve less accurate but faster results [7].

An edge detector usually does the next steps:

1) Noise removing. In edge detection, it is crucial to get rid of noise from image so it won't be confused with significant image features and objects on it. Firstly, the image is converted to grayscale. Gaussian filter and Infinite impulse response filters are the common choices to perform image smoothing.

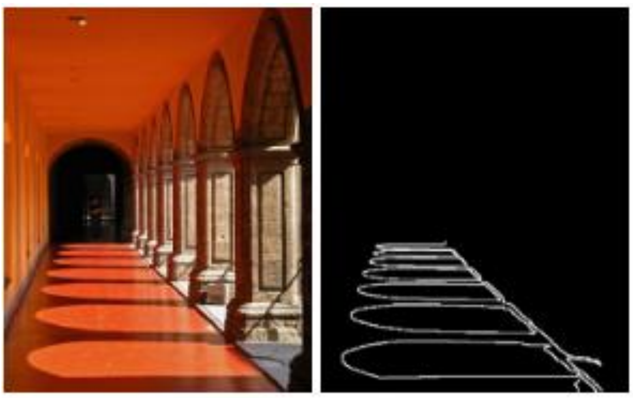
2) Finding the intensity gradients of the image. Those are computed by using an edge detection operator (like Kirsch, Prewitt, Roberts, or Sobel [8, 9]) which returns values of the first derivative in both horizontal and vertical directions for pixels intensities. After that, magnitude and direction of gradients are calculated.

3) In order to thin the edges, local maxima of the magnitude are found. Additionaly, direction are discreted into a small set of basic directions (left, right, up, down, and their in-betweens).

4) To remove spurious edge pixels caused by noise and color variations, all edge pixels must be compared to low and high threshold values for the magnitude. After this operation, pixels are divided into three categories: strong edges (magnitude is greater than the high threshold), weak edges (magnitude lays in between the thresholds), and false edge pixels (magnitude is less than the low threshold).

5) To refine weak edges, their 8-connected neighbour pixels are checked. If there is at least one strong edge pixel among them, the considered weak edge pixel is marked as a strong edge one. One by one, neighbouring weak pixels can be reclassified as strong ones. Remaining weak pixels are suppressed.

An example of shadow edge detecting is given on Fig. 4.



(a) (b)

Fig. 4 Example of using an edge-based algorithm: (a) the initial image, (b) shadow edges sucessfully detected [10, Fig. 10]

Edge-based segmentation can be applied to various images disregarding to their origin and observing camera properties. The obvious flaw is the algorithm doesn't classify received regions into shadows and lit areas. Also, the algorithm doesn't take into consideration image semantics. Hence, this algorithm requires additional actions for shadow detections [11].

D. Clustering Algorithms

Clustering techniques like K-Means or Gaussian Mixture Models [12, 13] can be used to group pixels into clusters based on their properties. In shadow processing, clustering can help distinguish between shadow and lit regions by partitioning pixels into clusters with distinct characteristics [14].

With a number of clusters given, a clustering algorithm groups similar pixels into the same cluster, and different pixels into different clusters. For instance, color difference can be used as a measure of pixels similarity. A few algorithms (e.g. Mean-Shift) don't require pre-defined number of clusters though.

Image clustering by K-Means algorithm conducted as follows:

1) The number k of clusters is initialized.

2) The centers of clusters c_1, c_2, …, c_k, are randomly initialized.

3) For each pixel, its similarity to each of centers c_i is calculated. Pixel p then is attributed to the j^{th} cluster, if the difference between p and c_j is the minimal among differences between p and c_i.

4) For each cluster, a new center is calculated as a mean value of interest between all the pixels within the cluster.

5) Steps 3 and 4 are repeated until clusters stay unchanged.

6) After image is clasterized, for each cluster, it is determined whether it is shadow or lit area. It is done by using the preliminary information about the image objects properties and their colors, which impact possible colors for shadow pixels.

K-Means algorithm is very simple, fast, and easy to implement. Never-the-less, the main drawbacks of K-Means algorithm are necessity to pick number k, sensitivity to initial centers, sensitivity to outliers. Some of them can be mitigated by using other clustering approaches like probabilistic clustering, where pixels are treated as sample values of a continuous function rather than a bunch of points. But such algorithms are more complicated and more difficult to implement.

Overall, the clustering algorithms are generally doing image segmentation (including shadows) well, but perform badly on the images with complex information, leading to the little difference between objects.

E. Growing Regions Method

Region growing is a popular class of segmentation algorithms used in computer vision and image processing to partition an image into coherent regions or objects based on certain similarity criteria. These algorithms operate under the assumption that pixels with similar characteristics or properties should belong to the same region. Among the techniques falling under this category, watershed algorithms [15, 16] are notable for their effectiveness in segmenting objects with distinct boundaries (Fig. 5).

Region growing algorithms typically follow a series of steps to segment an image:

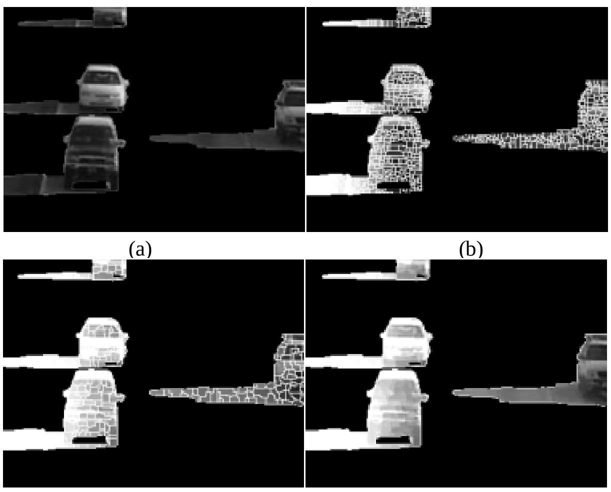
1) Seed Selection. The process begins by selecting one or more seed points within the image. These seed points serve as starting points for the region growing process. For example, in watershed algorithms, multiple seed points are used, typically located on the image's intensity minima. These seed points guide the algorithm in segmenting the image into distinct catchment basins, effectively outlining the objects of interest.

2) Region Initialization. The pixels at the seed points are considered as the initial regions. Each pixel within this region is compared to its neighboring pixels to determine if they meet predefined similarity criteria. Common criteria include similarity in intensity, color, or texture.

3) Pixel Neighbors. For each pixel within the initial region, its neighboring pixels are examined. If a neighboring pixel meets the similarity criteria, it is added to the growing region.

4) Region Expansion. This process continues iteratively, with newly added pixels extending the region. The growing stops when no more neighboring pixels meet the similarity criteria.

Region growing algorithms often produce well-defined region boundaries, making them suitable for tasks where object boundaries need to be accurately delineated. This makes the algorithms suitable for scenarios with multiple, well-separated objects like shadows and lit areas. However, these algorithms are sensitive to seeds selection and homogeneity criteria chosen. Computational complexity and under- and over-segmentation (especially for watershed algorithms) are also significant issues requiring careful consideration. These are the reasons to use some post-processing to merge and refine resulting segments.



(a) (b) (c) (d)

Fig. 5 Watershed segmentation: (a) foreground image, (b) result of edge detection and the segmentation, (c) the segmentation with preliminary median filtering, (d) removing redundant lines to address oversegmentation [17, Fig. 1]

F. Graph-Based Segmentation

Graph-based segmentation represents an image as a graph, where each node represents a pixel, and edges represent relationships between pixels (the stronger connection between pixels, the less the weight of corresponding edge). Algorithms like Minimum Spanning Trees [18] or Graph Cuts [19] can then be applied to segment shadow regions based on pixel relationships and intensity differences. Some approaches like star algorithm can also be used for segmentation and tracking [20].

The minimum spanning tree for an image can be built by using designated algorithms like Kruskal’s, Prim’s, or another one. Then, the tree is clustered by removing some inconsistent edges. Edges consistency must be rigorously defined by a formula, for example, a binary predicate of two vertices incidental with an edge.

One of the clustering algorithms can be described as the sequence of the next steps:

1) Sorting all edges by non-decreasing edge weight.

2) Initializing clustering: each pixel is a separate cluster.

3) For each edge, the following step occurs. Let v_i and v_j denote the vertices incident with the current edge. If v_i and v_j belong to different clusters, and the weight of the edge is less than the distance between the clusters, than they are joined.

Obviously, those algorithms possess some difficulties, like choosing functions for edge weights and cluster distances; lack of the cluster semantics and interpretation; issues with non-local image properties. Besides, sometimes it is nearly impossible to calculate the exact image segmentation graph due to an image properties, as well as algorithms constraints and time consumption. That’s why some fast algorithms can be in use which give a near-to-ideal result.

G. Active Contour Models

Active contour models (“snakes” [21] or “balloons” [22]) are used for boundary-based segmentation (Fig. 6). They are energy-minimizing splines guided by external constraint forces and influenced by image forces so they can catch lines and edges on an image in semi-automatic way [6, 20, 24, 25]. They can be applied to delineate shadow boundaries by iteratively adjusting a contour to fit the shadow's edge. These models, as well as livewire segmentation technique [16], are particularly useful for capturing complex, irregular shadow shapes.

Three components making up the energy of a snake are:

- internal spline energy due to its bending,
- external constant forces conducted by a software user,
- image forces affected by lines and edges on it.

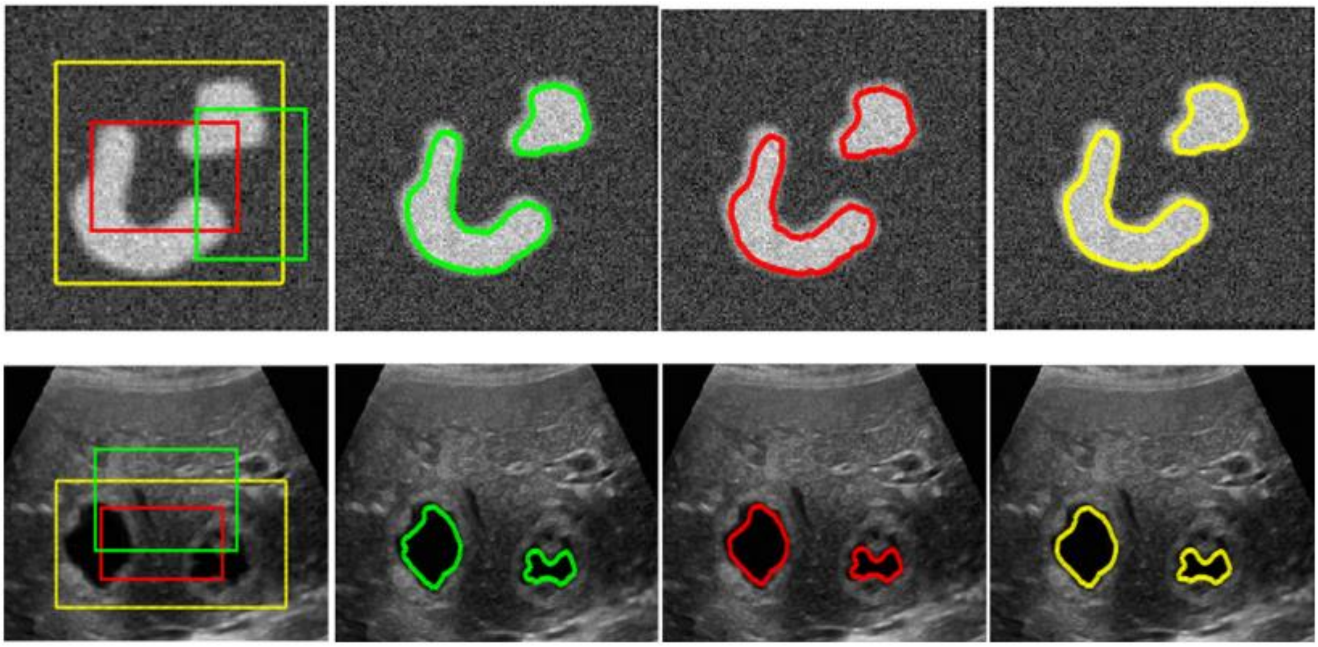


Fig. 6 Using “snake” active contours to detect edges. Different colors correspond to different contours initialization [23, Fig. 7]

The major advantages of the method are ease of interactive control, its self-adapting nature, its relative insensitivity to image noises, as well as the ability to track moving features by “locking on” it. Also, the same snake that finds subjective contours can very effectively find more traditional edges in natural imagery. However, active contour models possess some drawbacks like sensitivity to local minima states, overlooking minute features, its accuracy being dependent on convergence criteria, and rather long computation time when a high accuracy is required.

H. Spectral Analysis

In remote sensing applications, spectral analysis techniques can be used to exploit the different spectral properties of shadow and lit regions in multispectral or hyperspectral imagery [6]. It leverages the varying reflectance properties of different materials and surfaces across different wavelengths of light (spectral bands) to distinguish between shadow and lit areas. These methods are effective for shadow detection in satellite imagery (Fig. 7).

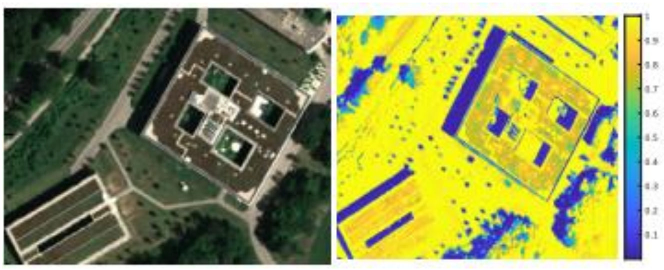
Different spectral analysis approaches can rely on different hardware, especially sensors: cameras, radars, lidars, temperature sensors, Doppler radars [27] etc. Thanks to them, it is possible to retrieve not only visual information (visible spectrum of electromagnetic waves), but also infrared, ultraviolet, and other ranges of electromagnetic spectrum, as well as some auxiliary features like temperature, object heights, radiation, among others. After getting spectral information for an observed area, reflectance values of its surfaces can be compared. Shadow regions usually exhibit lower reflectance values, hence lower energy of electromagnetic waves of various spectra being reflected and captured with the sensor. Also, some widely used indices and characteristics like EVI and NDVI [14] can be applied for shadow detection.

As far as spectral methods rely on both visual and non-visual information, they can detect shadows even if they are not visually presented on an RGB image. The methods are less reliant on subjective visual interpretation, hence the opportunity for accurate shadow detection. Also, beyond shadow detection, spectral analysis allows to figure out the nature of shadow, lit areas, and properties of surfaces they are lying on (e.g. material, height).

The flaws of spectral methods feature their complexity due to sensor properties, as well as computational intensity when processing hyperspectral data. Moreover, to retrieve accurate information, sensors must be properly calibrated, as variations in sensor sensitivity and atmospheric conditions can impact the spectral signature of objects. The fact weather conditions like cloudiness, atmospheric turbulence, and temperature can have a significant impact on sensors which leads to data quality decline, must also be addressed by using other approaches [28].

I. Machine Learning Algorithms

Machine learning approaches is probably the most widely used approach in various computer vision applications. Thanks to ability for recognizing edges, colors, and objects on an image, classifying regions, doing image semantic segmentation, as well as anomalies detection, machine learning established itself as a major tool for shadow detection and related problems. Machine learning algorithms give more accurate results, and work with various data types. Using convolutional neural networks [29] generative adversarial networks (GANs) is a common approach for



(a) (b)

Fig. 7 Example of using hyperspectral images: (a) an initial image, (b) sunlit factor map, computed by spectral Euclidean distances of the reconstruction [26, Fig. 1] shadow detection [30-32].

1) Data preprocessing. It is vital to properly collect and preprocess shadow images for the model training. They are divided into training, validating, and testing sets.

2) GAN training. Using training and validating image sets, the GAN learnes to generate shadows which can be added to an image without a shadow.

3) GAN's discriminator training. Discriminator is a network determinating whether a given shadow image is real or generated by the GAN. This network compares shadow images with images without shadows and finds shadow areas on the formers.

4) Shadow removing. After discriminator finds shadow areas, the shadows are removed. The original and resultant images are fed to the GAN's generator so it can learn to remove shadows using the information from step 2.

GAN based algorithm demonstrates high accuracy of shadow highlighting and removing on building images [33]. But to train GAN to find shadow areas correctly, a large dataset is required which must contain shadow images with objects of different positioning, size, and shape on them. Preparing such a dataset could be a challenging task.

J. 3D-Modeling Algorithm

3D-Modeling plays a crucial role in various computer vision applications, especially in satellite imagery related problems. 3D models, e.g. Digital Elevation Models [34] and 3D city models, provide information about position, height, and shape of objects and terrain. Besides, thanks to the precise nature of such an approach, it is possible to realistically render additional objects of a researcher's interest to complement the model and predict its future behavior.

A method based on urban modeling uses light and shadows mathematical models. It allows calculate the exact position of shadows even if they are not visible on an image. The method uses two assumptions: shadows have the same shape and size disregarding buildings shape and size; and only vertical objects (like buildings) cast shadows.

The algorithm features the next steps:

1) Building the urban model. This can be conducted by computer vision and image processing methods being conducted on one or multiple satellite captures. The model can be used for a building position, size, and orientation determination.

2) Lighting modeling. By using the imformation about geographical position and times of day for the captured scene, the lighting is modeled. It allows find out shadows position and size.

3) Shadow detection by using the information about buildings placement and lighting.

The algorithm is more resistant to image noises and artifacts. But it requires large computational resources to build 3D-models and model lighting, which can be critical in multiple images processing.

IV. Overall Comparison

Comparing shadow detection and segmentation algorithms involves evaluating various aspects of their performance and characteristics. Here are some key aspects you can highlight when comparing these algorithms:

- Accuracy. Some algorithms strive to get perfect result, whereas others focus on other advantages like time consumption and universality.
- Complexity. Methods can vary in computational, algorithmical, hardware complexity, as well as availability as open-source implementations and necessity of custom development.
- Robustness. Robust algorithms successfully evaluate shadow locations on images with different lighting conditions, objects with irregular shapes, shadows of different types, different surfaces, etc. Robustness to algorithm hyperparameters can also be considered [35-37].
- Generalization. Some types of algorithms can be applied to different datasets and scenes, and others are designed to work with certain types of images.
- Pre- and post-processing requirements. To detect shadows on an image, it must be preprocessed first (e.g. noise removal, labeling, channels filtering, atmospheric correction). Besides, some algorithms give noisy or fragmented outputs that need additional processing.

Comparison on these aspects of methods and algorithms listed in Section 3 is given in Table 1.

V. Conclusions

Shadow detection and segmentation is a challenging task which can be addressed by multiple methods and algorithms. They differ in variety of aspects including performance quality, complexity. There is no perfect and universal algorithm which can be used in every single situation because each of them possesses some advantages and disadvantages. The choice of segmentation algorithm depends on the characteristics of the images, the nature of the shadows, and the specific requirements of the shadow processing task. Often, a combination of techniques or post-processing steps may be employed to achieve the most accurate shadow segmentation results.

References

[1] P.L. Rosin, T.J. Ellis, "Image difference threshold strategies and shadow detection," British Machine Vision Conference, 1995, vol. 95.

[2] S. Murali, V.K. Govindan, "Shadow Detection and Removal from a Single Image Using LAB Color Space," Cybernatics and Information Technologies, 2013, vol. 13, iss. 1, pp. 95-103, doi: 10.2478/cait-2013-0009.

[3] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence 6, 1986, pp. 679-698.

[4] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," Int. J. Computer Vision, 1987, vol. 1, pp. 167–187.

[5] R. Ramya, P.S. Babu, "Automatic tuberculosis screening using canny Edge detection method," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2015, pp. 282-285, doi: 10.1109/ECS.2015.7124909.

[6] B. Liu, G. Zhang, J. Gong, X. Zhu, W. Fei, "Semi-automatic typical collieries extraction based on remotely sensed imagery using active contour models," Proc. SPIE 7492, International Symposium on Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining, #749206, 2009, doi: 10.1117/12.838367. [International Symposium

TABLE I. SHADOW DETECTION ALGORITHMS COMPARISON

Name of algorithm	Aspect				
	Accuracy	*Complexity*	*Robustness*	*Generalization*	*Processing requirements*
Image difference and threshold[a]	The algorithms give accurate results in terms of pixels intensity but it doesn't necessarily mean these pixels are genuine shadows	Pretty straightforward algorithms easy to implement in any computer vision system	Multiple factors can significantly impact on the result	The algorithms work the same way with any image given	Some basic preprocessing routines like noise removal are strongly recommended
Edge-based segmentation	Generally, the algorithm gives pretty accurate results. Some errors, however, might take place when an image exibits complicated structure	Some minor complicated calculations are performed. Besides, spurious edges removal and weak edges refinement might be time-consuming	The method is robust as far as an image is preliminarily cleared from noises	A universal algorithm working with various kinds of images	Noise removing is essential for the algorithm. Also, the resulting segments must be classified as shadow/lit areas by another algorithm
Clustering	The accuracy of results can vary depending on quality of image and hyperparameters	An easy-to-implement method which uses rather simple calculations	The method is hyperparameter-dependent and outlier-sensitive	Gives poor and meaningless results for images with complex structure	After the clustering, clusters must be classified as shadow/lit areas by another algorithm
Growing regions method	Shadow and lit areas and their boundaries are calculated rather accurately as far as initial seeds are chosen correctly	An easy-to-implement method which, however, might spend rather much time if an image has complicated structure	The method is unrobust in terms of seed influence on the final result	Like clustering algorithms, the method can give poor results for images with complex structure	Merging and refining resulting regions is oftenly required
Graph-based segmentation	The method can use both accurate and approximate algorithms	The process of building the image graph can be pretty time-consuming	For different images, different edge weight and cluster distances functions might be considered	Generally, the method works the same with different images	The user must firstly determines the functions for joining and dividing clusters, as well as a stop trigger event (like number of iterations or time spent). Also, sometimes some refinement of resulting clustering by auxilary methods is required
Active control models	Not only does the method give rather accurate results, it also can find not-so-obvious edges which leads to improved shadow detection	Computation time might be rather long if high precision is required	The method is relatively insensitive to image noises. However, it is sensitive to minor difference between pixels which not necesserely belong to different areas (shadow/lit)	Different image types might require different hyperparameters values	Some features defining energy-minimizing criteria must be pre-set by the user
Spectral analysis	The methods gives high-accuracy results due to its objective nature. Moreover, the method allowes to find out shadow properties	An extremely compex method relying on a broad set of factors: sensor properties, spectral range, characteristics being calculated and used	Robustness is achived by successfully mitigating noise issues like spectrum changes and atmospheric conditions	Can be applied to broad number of image classes	Sensors must be well-calibrated first, and weather obfuscations must be mitigated
Machine learning	The method can be both accurate and approximate	A complex method requiring high computational capacities	The method is highly dependent on hyperparameters	Can be applied to broad number of image classes	Requires a well-developed dataset to train on
3D-modeling	Gives pretty accurate results as far as the model features precise enough building models, as well as adequate lighting reconstruction	An extremely compex method, especially if a scene is large enough to capture dozens or hundreds of various objects	For a new scene, a new model is required to build	The method is applicable to the majority of open-spaced areas	To build a 3D model of a scene, all or at least the majority of significant objects must be modeled. For this, their placement, size, and shape must be taken into consideration

[a.] The only difference between those algorithms is, threshold algorithm uses a threshold value which must be pre-set depending on various factors

on Spatial Analysis, Spatial-temporal Data Modeling, and Data Mining, 2009, Wuhan, China].

[7] D. Sangeetha, P. Deepa, "An Efficient Hardware Implementation of Canny Edge Detection Algorithm," 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India, 2016, pp. 457-462, doi: 10.1109/VLSID.2016.68.

[8] R. Maini, H. Aggarwal, "Study and comparison of various image edge detection techniques," International journal of image processing (IJIP), 2009, vol. 3, iss. 1: pp. 1-11.

[9] A. Makandar, S. Kaman, R. Biradar, S.B. Javeriya, "Impact of Edge Detection Algorithms on Different Types of Images using PSNR and MSE," LC International Journal of STEM, 2022, vol. 3, iss. 4, pp. 1-11.

[10] L. Shen, T. Wee Chua, K. Leman, “Shadow optimization from structured deep edge detection,” IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2067-2074.

[11] A. Singh, N.S. Kumar, "An Improved Convexity Based Segmentation Algorithm for Heavily Camouflaged Images," IJIGSP, vol.5, no.3, pp.55-63, 2013. doi: 10.5815/ijigsp.2013.03.08.

[12] A.M. Deshpande, M. Gaikwad, S. Patki, A. Rathi, S. Roy, “Shadow Detection from Aerial Imagery with Morphological Preprocessing and Pixel Clustering Methods,” ICTACT Journal on Image & Video Processing, 2021, vol. 11, iss. 3, pp. 2385-2390, doi: 10.21917/ijivp.2021.0340.

[13] N. Mo, R. Zhu, L. Yan, Z. Zhao “Deshadowing of urban airborne imagery based on object-oriented automatic shadow detection and regional matching compensation,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, vol. 11, iss. 2, pp. 585-605, doi: 10.1109/JSTARS.2017.2787116.

[14] M. Aboutalebi, A.F. Torres-Rua, W.P. Kustas, H. Nieto, C. Coopmans, M. McKee, “Assessment of different methods for shadow detection in high-resolution optical imagery and evaluation of shadow impact on calculation of NDVI, and evapotranspiration,” Irrig Sci, 2019, vol. 37, pp. 407–429, doi: 10.1007/s00271-018-0613-9.

[15] S.V. Sholtanyuk, “Finding the optimal segmentation of a crowd image with watershed method,” Information Systems and Technologies, pt. 2, 27–28 October 2022, pp. 217-223. (In Russ.)

[16] F.-P. Zhu, J. Tian, X.-P. Luo, X.-F. Ge, "Medical image series segmentation using watershed transform and active contour model," Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp. 865-870 vol.2, doi: 10.1109/ICMLC.2002.1174506.

[17] J. Gao, J. Dai, P. Zhang, “Region-based moving shadow detection using watershed algorithm,” International Symposium on Computer, Consumer and Control (IS3C), July 2016, pp. 846-849.

[18] H.G. Akçay, S. Aksoy, “Building detection using directional spatial constraints,” IEEE International Geoscience and Remote Sensing Symposium, 2010, pp. 1932-1935.

[19] Q. Shao, C. Xu, Y. Zhou, H. Dong, “Cast shadow detection based on the YCbCr color space and topological cuts,” The Journal of Supercomputing, 2020, no. 76, pp. 3308-3326.

[20] M.F. Esmaile, M.H. Marhaban, R. Mahmud, M.I. Saripan, “Cross-sectional area calculation for arbitrary shape in the image using star algorithm with Green's theorem,” IEEJ Transactions on Electrical and Electronic Engineering, 2013, iss. 8(5), pp. 497-504. doi: 10.1002/tee.21886.

[21] M. Kass, A. Witkin, D. Terzopoulos, “Snakes: Active contour models,” International Journal of Computer Vision, 1988, vol. 1, iss. 4, pp. 321-331, doi:10.1007/BF00133570.

[22] P. Szczypinski, P. Strumillo, "Application of an active contour model for extraction of fuzzy and broken image edges," Machine GRAPHICS & VISION, International Journal, 1996, no. 5(4), pp. 579-594.

[23] L. Fang, X. Pan, Y. Yao, L. Zhang, D. Guo, “A hybrid active contour model for ultrasound image segmentation,” Soft Computing, 2020, no. 24, pp. 18611-18625, doi: 10.1007/s00500-020-05097-y.

[24] O. Shishido, N. Yoshida, O. Umino, “Image processing experiments for computer-based three-dimensional reconstruction of neurones from electron micrographs from serial ultrathin sections,” Journal of Microscopy, 2000, no. 197(3), pp. 224-238. doi: 10.1046/j.1365-2818.2000.00666.x.

[25] P. Makowski, T.S. Sørensen, S.V. Therkildsten, A. Materka, H. Stødkilde-Jørgensen, E.M. Pedersen, "Two-phase active contour method for semiautomatic segmentation of the heart and blood vessels from MRI images for 3D visualization," Computerized Medical Imaging and Graphics, 2002, vol. 26, iss. 1, pp. 9-17, doi: 10.1016/S0895-6111(01)00026-X.

[26] G. Zhang, D. Cerra, R. Müller, “Shadow detection and restoration for hyperspectral images based on nonlinear spectral unmixing,” Remote Sensing, 2020, vol. 12, iss. 23, #3985, doi: 10.3390/rs12233985.

[27] R.E. Mercer, J.L. Barron, A.A. Bruen, D. Cheng, “Fuzzy points: algebra and application,” Pattern Recognition, vol. 35, iss. 5, 2002, pp. 1153-1166, doi: 10.1016/S0031-3203(01)00110-8.

[28] D. Cheng, R.E. Mercer, J.L. Barron, P. Joe, “Tracking severe weather storms in Doppler radar images,” International Journal of Imaging Systems and Technology, 1997, no. 9(4), pp. 201-213. doi: 10.1002/(SICI)1098-1098(1998)9:4<201::AID-IMA3>3.0.CO;2-E.

[29] D. Chai, S. Newsam, H.K. Zhang, Y. Qiu, J. Huang, "Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks," Remote Sensing of Environment, 2019, vol. 225, pp. 307-316, doi: 10.1016/j.rse.2019.03.007.

[30] B. Ding, C. Long, L. Zhang, C. Xiao, “ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal,” Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 10213-10222.

[31] Y. Zhu, X. Fu, C. Cao, X. Wang, Q. Sun, Z.J. Zha, “Single image shadow detection via complementary mechanism,” Proceedings of the 30th ACM International Conference on Multimedia, October 2022, pp. 6717-6726.

[32] O. Naidovich, A. Nedzved, S. Ye, “DSDNet Neural Network for Shadow Detection from Urban Satellite Images,” Pattern Recognition and Information Processing (PRIP'2021), 21–24 Sept. 2021, pp. 191–194.

[33] L. Zhang, C. Long, X. Zhang, C. Xiao, “RIS-GAN: Explore residual and illumination with generative adversarial networks for shadow removal,” In Proceedings of the AAAI Conference on Artificial Intelligence, April 2020, vol. 34, no. 07, pp. 12829-12836.

[34] N. Neckel, N. Fuchs, G. Birnbaum, N. Hutter, A. Jutila, L. Buth,et al. (2023). “Helicopter-borne RGB orthomosaics and photogrammetric digital elevation models from the MOSAiC Expedition,” Scientific Data, 2023, vol. 10, iss. 1, #426.

[35] S. Sholtanyuk, “Comparative Analysis of Neural Networking and Regression Models for Time Series Forecasting,” Pattern Recognit. Image Anal. 2020, vol. 30, pp. 34–42, doi: 10.1134/S1054661820010137.

[36] S.V. Sholtanyuk, “Influence of the Neural Network Hyperparameters on its Numerical Conditioning,” Digital Transformation, vol. 1, pp. 43-50, 2020 (In Russ.)

[37] S. Sholtanyuk, A. Leunikau, “Lightweight Deep Neural Networks for Dense Crowd Counting Estimation,” Pattern Recognition and Information Processing (PRIP'2021), United Institute of Informatics Problems of the National Academy of Sciences of Belarus, pp. 61–64, 2021 [Proceedings of the 15th International Conference, 21–24 Sept. 2021, Minsk, Belarus]