A review of silhouette extraction algorithms for use within visual hull pipelines

ASCENSO, Guido <http://orcid.org/0000-0002-9050-9892>, YAP, Moi Hoon, ALLEN, Thomas, CHOPPIN, Simon S. <http://orcid.org/0000-0003-4910-9149> and PAYTON, Carl <http://orcid.org/0000-0001-8896-9753>

Published version

ASCENSO, Guido, YAP, Moi Hoon, ALLEN, Thomas, CHOPPIN, Simon S. and PAYTON, Carl (2020). A review of silhouette extraction algorithms for use within visual hull pipelines. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 1-22.

A Review of Silhouette Extraction Algorithms for use within Visual Hull Pipelines

Guido Ascenso[a]*, Moi Hoon Yap[b], Thomas Allen[c], Simon S. Choppin[d], Carl Payton[a]

[a]Department of Sport and Exercise Sciences, Manchester Metropolitan University, Manchester, UK; [b]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK; [c]Department of Engineering, Manchester Metropolitan University, Manchester, UK; [d]Centre for Sports Engineering Research, Sheffield Hallam University, Sheffield, UK

Email of the *corresponding author: guido.ascenso@stu.mmu.ac.uk

A Review of Silhouette Extraction Algorithms for use within Visual Hull Pipelines

Markerless motion capture would permit the study of human biomechanics in environments where marker-based systems are impractical, e.g. outdoors or underwater. The visual hull tool may enable such data to be recorded, but it requires the accurate detection of the silhouette of the object in multiple camera views. This paper reviews the top-performing algorithms available to date for silhouette extraction, with the visual hull in mind as the downstream application; the rationale is that higher-quality silhouettes would lead to higher-quality visual hulls, and consequently better measurement of movement. This paper is the first attempt in the literature to compare silhouette extraction algorithms that belong to different fields of Computer Vision, namely background subtraction, semantic segmentation, and multi-view segmentation. It was found that several algorithms exist that would be substantial improvements over the silhouette extraction algorithms traditionally used in visual hull pipelines. In particular, FgSegNet v2 (a background subtraction algorithm), DeepLabv3+ JFT (a semantic segmentation algorithm), and Djelouah 2013 (a multi-view segmentation algorithm) are the most accurate and promising methods for the extraction of silhouettes from 2D images to date, and could seamlessly be integrated within a visual hull pipeline for studies of human movement or biomechanics.

Keywords: markerless motion capture; biomechanics; silhouette extraction; background subtraction; semantic segmentation; multi-view segmentation.

Introduction

Accurate 3D kinematic data are needed by biomechanists to study and understand the movement patterns of humans (1). Marker-based systems, such as Vicon, are considered the gold standard for motion capture due to their accuracy (3)(2). The movement of markers attached to the skin of participants is used to infer the

underlying relative movement between two adjacent segments that form a joint (e.g. ankle), with the goal of precisely defining the movement of the joint (4). However, skin movement relative to the underlying bone is a primary factor limiting the resolution of detailed joint movement using marker-based systems (5)(6)(7)(8)(9). Also, fixing markers to a participant can be time consuming, especially if upwards of 30 markers are to be used and if the activity to be performed causes frequent detachment of markers due to profuse sweat or the performance of highly dynamic movements (6).

Because of such limitations, researchers in biomechanics have long been trying to develop markerless motion capture tools (10)(11) or to adapt existing ones to their needs (12). One such tool is the visual hull, a shape-from-silhouette method first developed by Laurentini in 1994 (13) which uses 2D images to reconstruct the object of interest. In biomechanics, the visual hull has been used to study the biomechanical differences between three types of tennis serves (14)(15), to perform gait analysis (16), to study the biomechanics of the arm during front crawl swimming (12), and to analyse the movement pattern of gymnasts (17). The first step in the visual hull pipeline is to separate the object of interest ("foreground") from the rest of the image ("background"), a process referred to as "silhouette extraction" in this paper[1]. To compute a visual hull, a silhouette needs to be extracted from each camera view (of which there may be several) at each frame of a possibly long video (13). Consequently, an automatic method for accurate silhouette extraction is necessary (18). The extent to which the accuracy of the silhouettes

[1] A silhouette is defined as the outer contour of an object. However, most algorithms discussed in this paper give as an output a mask of the object (i.e. a silhouette and the area enclosed by it). This is not an issue for the purposes of shape-from-silhouette tools, as a silhouette or an object mask will behave similarly when used as inputs for the visual hull reconstruction.

influences the accuracy of the reconstructed visual hull has not been investigated yet. However, Grauman et al. (19) and Gall et al. (20) have suggested that a small segmentation error in even just one camera view could have a significant effect on the reconstructed visual hull. Though the authors did not quantify this "significant effect", it is reasonable to assume that higher-quality silhouettes would produce higher-quality visual hulls, thus making silhouette accuracy a critical bottleneck in the reconstruction of a visual hull. When constructing a visual hull, a 3D point is labelled as part of the visual hull if and only if its projection lies within the silhouette on all the camera views; therefore, a view having errors in its silhouette could spoil the quality of the entire visual hull (21). Several authors who have applied the visual hull do not mention the silhouette extraction method used in their studies (22)(23), while others have used basic silhouette extraction methods (16)(24)(25)(26). The main reason such basic methods for silhouette extraction have been used for the visual hull in previous publications is simply because the studies mentioned above were published before the advanced methods of silhouette extraction available today had been developed. Nowadays, however, there exist several methods that can rival the silhouette segmentation accuracy of humans (27) and which take as little as a hundred milliseconds to extract a silhouette from a large high-quality image (28). It is our hope that, by presenting a detailed review of the methods available today for accurate silhouette extraction, future researchers in biomechanics will be able to make a more informed decision with regards to which silhouette extraction method to choose when using the visual hull as a tool for markerless motion capture.

The Computer Vision literature offers several algorithms that enable the extraction of accurate silhouettes. These algorithms fall into three categories: background subtraction, semantic segmentation, and multi-view segmentation[2]. Each category of algorithms has specific datasets used to test the algorithms and metrics used to evaluate their performance. In other words, algorithms that belong to different categories are tested on different datasets, with different metrics. This is because each category of algorithms is typically devoted to a specific task for which specialised datasets have been developed and for which the evaluation criteria are task-specific. For example, background subtraction algorithms are often used for surveillance-camera applications (30)(31)(32), while semantic segmentation algorithms not only detect the silhouette of the object in the image, but also label it as belonging to one of several possible classes (car, human, cat, dog, etc.) (33). For the purposes of the visual hull, any algorithm that extracts an accurate two-dimensional silhouette from an image would be applicable (13). Nevertheless, the diversity of datasets and metrics used to evaluate methods that belong to different categories makes it difficult to identify the optimal method for the task at hand. Therefore, the main goal of this paper is to provide a reference for researchers looking for a silhouette extraction method to use in their shape-from-silhouette pipeline, focusing on applications in biomechanics of human motion.

Within each category of silhouette extraction methods there are hundreds of algorithms. A detailed analysis of all of them would be prohibitively long and is therefore beyond the scope of this paper. Instead, what this review paper endeavours

[2] Methods that rely on depth data (RGB-D) will not be covered here, as they cannot be easily applied to the visual hull, which is the main focus of this review paper. For a detailed review of RGB-D methods, please refer to (29).

to do is to give an outline of each category of algorithms in terms of what their intended application is, what the most popular and best-performing methods within the category are, what datasets they are tested on, and what metrics are used to evaluate their performance. The choice to limit our analysis to a select few algorithms is reinforced by the fact that modern silhouette extraction methods significantly outperform traditional ones under all metrics, as we discuss in later sections. Consequently, only the algorithms that achieve the best results on the datasets of their respective category will be discussed here. Review papers whose scope is limited to listing and discussing all the algorithms, old and new, present within an individual category of algorithms already exist; for more details on a particular category, the reader is invited to read the corresponding reviews, which will be highlighted throughout this paper.

Given the importance of silhouette accuracy for visual hull accuracy (19), it would be expected that previous research that used the visual hull for biomechanical applications would have used highly refined silhouette extraction methods; this is not the case. Vlasic et al. (24) and Furukawa et al. (25) justified the use of a rudimentary silhouette extraction algorithm in their visual hull pipeline by stating that in a controlled laboratory setting it is possible to artificially manipulate the scene so that in each camera view the object would appear against a monotone, dark background, thus making the task of silhouette extraction trivial. While their reasoning is certainly valid, it confines their algorithm to the highly controlled setup they adopted and it renders its application within biomechanics virtually impossible outside of a laboratory, since the background can scarcely be controlled in most

settings that require the motion capture of humans (12). Mündermann et al. (22) performed a study regarded as a guideline for the number of cameras required to construct a sufficiently accurate visual hull for biomechanical applications; the method they used for silhouette extraction, however, was not mentioned. Similarly, Nobuhara et al. [41] developed a post-processing tool to refine the visual hull, but did not mention the silhouette extraction method used to obtain the initial visual hull. Ceseracciu et al. (12) used a basic Gaussian Mixture Model (34) to extract the silhouettes of underwater swimmers. However, as we discuss in Section 3.2, this method is too inaccurate to be used in a swimming pool, which presents one of the most challenging backgrounds imaginable: constant motion at the water-air surface, possible colour camouflage of the person against the wall of the swimming pool, frequently and randomly changing lighting intensity, and bubbles present around the edges of the silhouette.

The main contributions of this review paper are the following: 1) we present an overview of methods for silhouette extraction that belong to different categories of algorithms. Although reviews that focus on individual categories already exist in the literature, to the best of our knowledge this is the first paper to compare methods, datasets, and metrics across different categories; 2) having discussed the different methods, we highlight the ones that would make the strongest candidates for use within a visual hull pipeline. Both points constitute a novelty in the literature and could help researchers interested in the visual hull in making a more informed decision regarding what silhouette extraction algorithm to use.

Background Subtraction

Overview

Background subtraction algorithms seek to separate the moving objects (foreground) present in an image from the static background (35)(36). Their main application is within intelligent video surveillance tasks like the automatic tracking of objects within a scene or their recognition (i.e. assigning them a class label), both of which are typically applied to videos recorded from surveillance cameras placed on roads (37), at airports (38), or within buildings (30)(31)(39). Background subtraction algorithms are developed to meet the specific challenges of this field of computer vision, such as gradual changes in the intensity of the lighting of the scene, insertion in the scene of new background objects (a man carries a bag and then leaves it on the floor: should the bag be treated as background or as foreground?), and dynamic background objects such as waving trees or water rippling in a lake (35)(40). Furthermore, the algorithms need to be able to model any generic object, the shape and size of which may vary considerably: from an airplane, to a person, to a bike, to a cat. Also, because background subtraction is typically only the first step within a complex Computer Vision pipeline (41), its computation should happen in real time or close to it, meaning that researchers often have to balance a trade-off between speed of execution and accuracy (35). If the objective is to use a background subtraction algorithm within a markerless motion capture system, the computational complexity problem is simplified considerably, mainly because the object that the algorithm needs to be able to segment is not generic: all humans have similar shapes and sizes, and using this *a priori* knowledge

to introduce bias into the model would mean that a simpler algorithm could be used (42); this, in turn, would translate into faster computation. Secondly, biomechanical analyses are not constrained to happen in real-time like most video surveillance tasks; this means that speed of execution could be sacrificed in favour of accuracy, which is of paramount importance in biomechanics (16).

Traditional[3] methods for background subtraction that rely on hand-crafted features are still widely used because they are computationally affordable and easy to implement (35). However, the field of background subtraction, like many others in Computer Vision, has been revolutionised by the advent of deep learning (43)(44); a glance at the leaderboards of the most popular background subtraction benchmarks (27) will reveal that methods that use deep learning occupy all the top leaderboard positions in terms of accuracy. Since researchers looking to implement visual hull algorithms would only be interested in the best performing silhouette extraction methods, this review paper will focus on these more recent methods that use deep learning for background subtraction[4]. Due to its frequent use within visual hull pipelines, the "basic

[3] In this context, the term "traditional" refers to algorithms that do not employ deep learning.

[4] For a thorough review of traditional methods for background subtraction, please refer to the review by Bouwmans et al. (35).

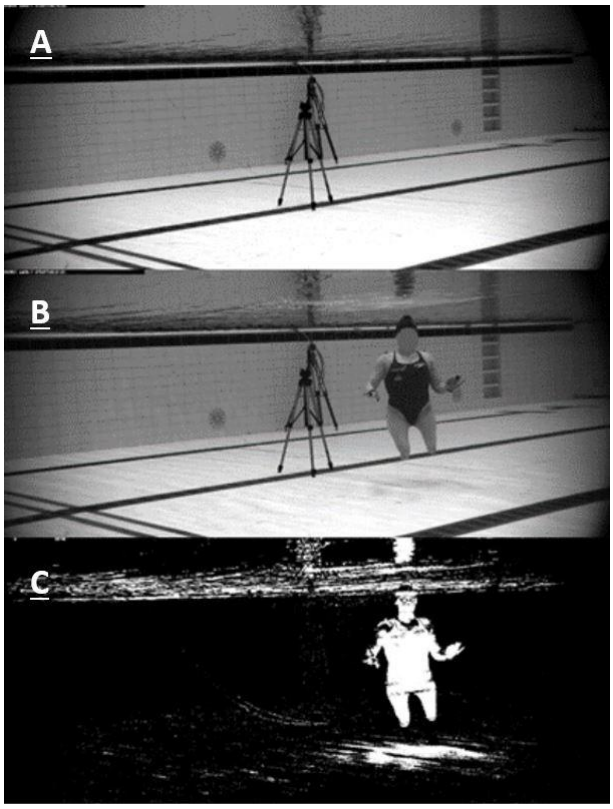

Figure 1. A) background initialisation frame; B) foreground object inserted into the scene; C) result of basic background subtraction. The amount of noise present in the output due to the dynamic background and the appearance of a shadow, coupled with false negatives caused by colour camouflaging, would translate into a noisy, inaccurate visual hull.

background subtraction" algorithm is briefly discussed in this section. The most basic way to perform background subtraction is to take a reference image in which the object of interest does not appear and to subtract its pixel values from the pixel values of the image from which the silhouette is to be extracted. If the difference in intensity between the pixels of the two images is greater than an arbitrary threshold fixed *a priori*, the pixel is labelled as foreground; otherwise, it is labelled as background. Under strictly controlled laboratory conditions where the background is completely stable, the assumption of being able to obtain a clean reference frame unobstructed from the object of interest is easily met. In real-life conditions, however, the reference frame may not be obtainable, or it may be corrupted by the

presence of dynamic objects in the background (e.g. trees or moving water) or by the presence of shadows that appear once the object of interest is inserted in the scene. Furthermore, if the object and the background have similar colours (a phenomenon known as *colour camouflaging*), basic background detection will fail completely or, at best, cause numerous false negatives (35). For these reason, and as shown in Fig. 1, basic background subtraction is not adequate for use within markerless motion capture systems, in which variations of the background are, if not expected, at the very least probable.

State of The Art

Traditional background subtraction algorithms define and update background models, and then classify pixels by using hand-crafted features and simple equations. Conversely, algorithms based on deep learning skip the definition and update of a background model and instead allow the network to learn its own parameters by feeding it several labelled examples of background/foreground pixels. In other words, methods based on deep learning do not compare each pixel to a background model designed by hand; they learn to classify pixels based on examples of previously classified pixels they have seen.

A particular type of deep-learning-based algorithm, Convolutional Neural Networks (CNNs) excel at tasks in Computer Vision(43)(45)(46), in part because they are translation invariant, which means that objects in new examples are recognised even if they appear in a different location than in past examples (47); this is an important feature when dealing with dynamic backgrounds and moving objects (47). Furthermore, the convolution operation can be easily parallelised on a GPU,

granting CNNs exceptional processing time (47). Readers interested in a more detailed of the application of CNNs to computer vision are referred to the review paper by Voulodimos et al. (48).

The following sections present the most accurate CNN-based background subtraction algorithms present in the literature to date. The accuracy and processing time of each algorithm are discussed in detail in Section 2.3.

Cascade MSCNN

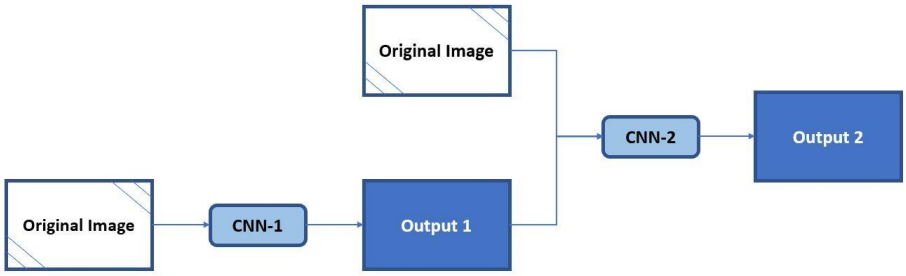

Figure 2. Cascaded structure of Cascade MSCNN.

The Cascade Multi-Scale Convolutional Neural Network algorithm, developed by Wang et al. (47), is scene-specific, meaning that the network has to be trained for each video being analysed. Although this training strategy makes it cumbersome to run Cascade MSCNN on multiple new videos, it enables the algorithm to exploit the high redundancy present in frames that come from the same video, thus reducing the number of training examples required: whereas image classification networks are shown tens of thousands of images during training, Cascade MSCNN converges after 200 examples. The $N = 200$ training examples are selected at random from the

video and they are manually labelled by an experienced user, who labels each pixel as either foreground or background[5].

CNNs were built specifically as a tool for image classification. To adapt them for background subtraction, the intent of which is to label each pixel within an image, many authors use small patches (around 30x30 pixels) centred around the pixel to be classified, which are extracted from the image and fed to the CNN for classification. The CNN gives a label to the entire patch, and that label is attributed to the pixel in the centre of the patch (49). This patch-based approach allows very fine-grained accuracy, but it misses global information that is important to segment large objects (50). Imagine that a large cat (appearing on the image with size 300x300 pixels, for example) were to be segmented from the background. A 30x30-pixels patch at the centre of the cat would not be enough to tell whether those pixels belonged to the cat or not, because context and a point of reference are absent (all pixels in the patch would have similar colour properties). To address this issue, the Cascade MSCNN algorithm adopts a multi-scale approach: it resizes the original image twice, and these 3 images (*size* = 1, *size* = 0.75, and *size* = 0.5 of the original) are fed to the network separately, thus obtaining 3 separate predictions which are later averaged to produce a single output.

Because CNNs process each pixel independently, they often produce isolated false positives and false negatives (47). To address this issue, the authors of Cascade

[5] The 200 frames extracted from each video act as a prior from which the model learns the distribution of the frames present in the video. In other words, the model assumes that the contents of the first 200 frames are representative "enough" (where "enough" cannot be easily defined mathematically) of the contents of all frames present in the video. In cases where this is not true (for example, videos that change scenery significantly, like a camera that starts in forest and during the course of the video ends underwater), the model's ability to generalise from the first 200 frames will be lower. However, this problem can be circumnavigated by selecting the 200 frames so that all "phases" of the video are represented; in the toy example from above, that would mean that some of the 200 frames would come from when the camera was in the forest, and some from when it was in the water.

MSCNN implemented a cascaded CNN model (illustrated in Fig. 2): the first part of the network, CNN-1, makes an initial prediction which is then fed, along with the original input image, to the second part of the network, CNN-2. This cascaded approach ensures that the predicted foreground mask is locally consistent (in other words, the number of isolated false positives and false negatives is reduced) without using postprocessing tools like Conditional Random Fields (CRF), which several authors have used in the past (51)(52) but which slow down training due to its computational complexity (47). Because both CNN-1 and CNN-2 have millions of parameters to train but only 200 images for training, they were pre-trained on a generic dataset (53) for transfer learning purposes. Then, during training, the weights of CNN-1 were fixed, and only the weights of CNN-2 were allowed to learn. The weights of CNN-1 were fixed (instead of having them learn with a low-valued learning rate) so that the information that CNN-1 had learned during pre-training would not be washed away during the second stage of training. This type of approach (which is but one of the possible ways to perform what is called "fine-tuning") assumes great confidence in the fact that the information encoded in the data used for pre-training was highly relevant to the task at hand.

3D-Net

3D-Net was designed to incorporate into the evaluation of an image the information shared with temporally adjacent frames. During training, the network is shown the frame being analysed as well as the 9 frames preceding it[6]. The temporal information present in this sequence of consecutive frames is progressively encoded

[6] Only the ground truth for the frame being analysed is provided to the network during training.

into denser and denser 3D convolutional modules (2D for space, 1D for time, see Fig. 3), until a prediction is made at the end of the last convolutional module. The redundancy present in temporally adjacent frames is leveraged to intentionally introduce bias into the system, much like Cascade MSCNN sought to exploit the redundancy present in multiple frames that came from the same video. Because 3D-Net is trained on the entire dataset being analysed, the overfitting effect of the bias introduced into the system is alleviated, and the network can generalise to new videos without requiring further training. The concept of incorporating temporal information in the pixel classification task of a neural network is similar to the idea of updating the background in traditional algorithms. In this sense, 3D-net is the only deep-learning algorithm that was explicitly designed to adapt to changes in the background. However, videos recorded with a low frequency represent an issue for 3D-net, because if the latency between consecutive frames is too large, the assumption of temporal contiguity between frames, on which the model is based, does not hold.

Noticeably, the authors of 3D-Net do not mention any pre-training or weight initialisation strategies, which are universally recognised as a powerful tool to boost the accuracy of CNNs (43)(54)(55)(56; 57).

BScGAN

BScGAN (Background Subtraction conditional Generative Adversarial Network) is the latest published deep-learning algorithm to date (28) in the field of background subtraction. It is also the first instance of a conditional Generative Adversarial Network (cGAN) (58) being used for background subtraction. A cGAN consists of

two connected networks, called generator and discriminator. The generator learns to produce an output given an input modified by random noise[7], while the discriminator learns to train the generator by comparing the ground truth and the output of the generator. In other words, the challenge of the generator is to produce an output that is as different from random noise and as close to a realistic output as possible; the challenge of the discriminator is to determine if the output of the generator is random noise or if it is a real, non-synthetic output.

In the case of BScGAN, the real input shown to the generator during training consists of two images: one with the foreground object present, one with the foreground object absent (i.e. true background image). The real output shown to the discriminator is a hand-labelled mask of the object in the real input image. During testing, only the generator part of the network is active; therefore, the processing time of BScGAN is faster than that of Cascade MSCNN, since fewer components are active in BScGAN than are active in Cascade MSCNN. The processing time of BScGAN is further reduced by using entire images for testing instead of dividing them into patches.

Internally, the generator of BScGAN has an encoder-decoder[8] structure where both modules have the same architecture (based on U-net (59)) but reversed layer ordering. The internal architecture of the discriminator of BScGAN is a simple series of four convolutional and four downsampling layers.

[7] The distinction between a GAN and a cGAN is that the generator in a GAN is only shown random noise during the first stages of training, whereas the generator of a cGan is trained by using the random noise to modify a real input example.

[8] In an encoder-decoder network, the encoder module gradually reduces the spatial dimension and captures higher semantic information, while the decoder module gradually recovers the spatial information and brings the output back to the original size of the input. This kind of network is explained in more detail in Section 3.2.

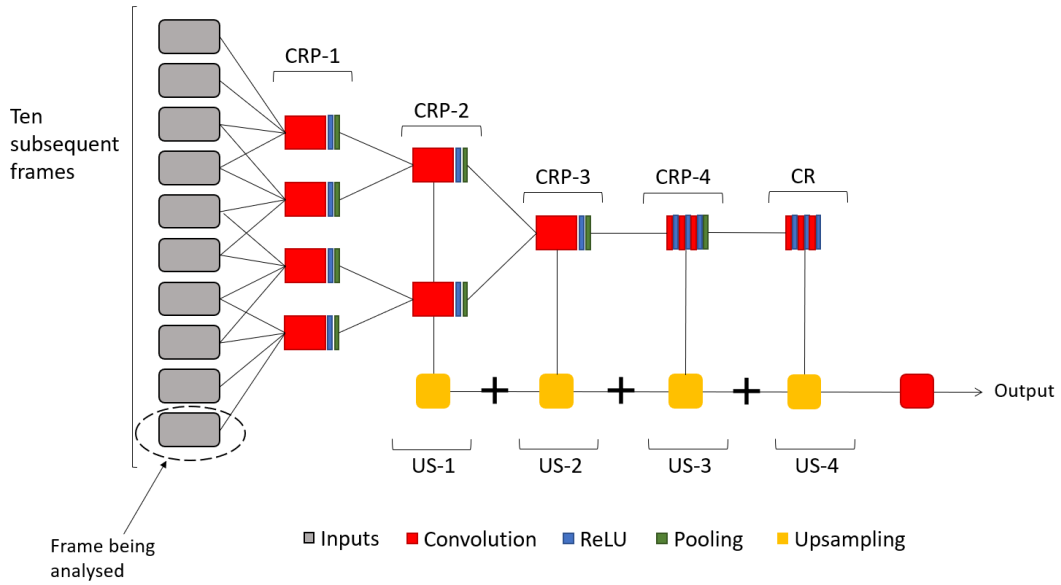

Figure 3. Architecture of 3D-Net. 10 subsequent frames are used for the prediction of just the last one of them. CRP-1 to CRP-3 are 3D convolutional modules, whereas CRP-4 and CR and 2D convolutional modules. The kernel size of each upsampling layer (US-1, US-2, US-3, US-4) is different, granting the network multi-scale spatial resolution. (Image from [49])

FgSegNet

In 2018, Lim and Keles published three papers on three different versions of their background subtraction algorithm, *FgSegNet* (50)(60)(61); to date, these 3 algorithms occupy the top 3 positions of the leaderboard of the ChangeDetection.net background subtraction challenge described in Section 2.3 (27). Similarly to Cascade MSCNN (47), all versions of FgSegNet are scene specific (see Section 2.2.1). The strategy used to select the *N* training frames from a video is identical to that of Cascade MSCNN, but in addition the *N* frames are randomly shuffled to avoid introducing excessive bias into the system by feeding it adjacent frames.

The first version of FgSegNet (50) achieves multi-scale spatial resolution by using as input three copies of the same image, re-scaled using Gaussian filtering (62). The three images are then fed to a triplet of CNNs which share weights to reduce the number of training parameters. The backbone of all three CNNs is taken

from the VGG-16 network (63), an object recognition network that can be adapted to background subtraction by replacing its fully connected layers with convolutional layers[9]. The outputs of the triplet of CNNs are then fed to a decoder which makes the final prediction in the same resolution as the input image.

In the second version of FgSegNet, called FgSegNet S (60), the triplet of CNNs is replaced by a single-input encoder-decoder structure. Multi-scale spatial resolution is achieved by placing a Feature Pooling Module (FPM) at the end of the encoder. Within the FPM, parallel dilated convolutional layers[10] with different dilation rates allow the network to incorporate spatial information from multiple scales without re-scaling the image prior to training.

FgSegNet v2 (61), the third version of FgSegNet, maintained the structure of FgSegNet S but modified the FPM module and the decoder. The updated decoder had a significant impact (+ 1-2%) on accuracy when the number N of training examples was low (25-50), but its impact was negligible (< 0.01%) for $N = 200$, which was the configuration that gave the highest accuracy. The exact impact of the modified FPM module cannot be evinced directly from the original paper of FgSegNet v2, since the authors did not include in the paper a detailed ablation study, highlighting the contribution of each module of their model to overall performance.

In typical images for background subtraction, the ratio of background pixels to foreground pixels is in the order of 100:1, 1000:1, or even 10000:1 (60)(61). In supervised learning, having an imbalanced number of training examples for different class categories causes problematic bias during classification (64)(46);

[9] For more details on this, please refer to (33).

[10] Dilated convolutions are described in detail in Section 3.2.1.

such bias makes it harder for the network to generalise to new data (50). FgSegNet deals with the issue of the imbalanced data classes by penalising the loss more if a foreground pixel is classified as a background pixel than the contrary. In other words, the rare class (foreground) is given a larger weight than the dominating class (background) when computing the loss function. The weights for the two classes are derived on a frame by frame basis by considering the foreground/background pixel ratio for that specific frame.

Datasets

In 2014, a review paper by Bouwmans (35) on methods for background subtraction reported 9 "traditional" datasets (like the Wallflower (40) dataset) and 8 "recent" datasets. Since then, the ChangeDetection 2014 (CDnet2014) (27) and Background Models Challenge 2012 (BMC) (65) datasets, both of which appeared in Bouwmans' "recent" category, have established themselves as the benchmark background subtraction datasets on which new algorithms are tested.

CDnet2014 contains 53 video sequences (for a total of over 11,000 frames) divided into 11 categories which reflect specific challenging scenarios: Baseline[11], Dynamic Background, Night, Shadows, etc. The videos were obtained using different cameras which had different resolution, frame rate, and compression parameters. Therefore, the resolution of the frames in CDnet2014 ranges from 320x240 to 720x576 pixels. However, because most of the images are of size 320x240, some authors (42)(47) resize all images to this resolution before training their networks. The ground truths made available for testing algorithms on

[11] The baseline category contains a mixture of mild challenges that belong to the other categories, and therefore is the easiest category for algorithms to analyse.

CDnet2014 (see Fig. 4 for an example) were segmented manually by human operators under the following guidelines:

- A pixel should be labelled as foreground only if it is not part of the background; • Foreground objects are people, animals, vehicles, or man-made objects;
- A moving object that suddenly stops (i.e. an abandoned bag) should be detected as foreground for a short period before being considered as background (where the definition of the word “short” is intrinsically subjective);
- Reflections and spotlight halos are not considered as foreground;
- Hard shadows should be manually labelled in order to enable the comparison of algorithms based on their robustness to shadows.

Another popular dataset for background subtraction, the BMC dataset (65) is comprised of 20 synthetic videos rendered with the SiVIC simulator (66) and of 9 real

Figure 4. On the left: example of an image in CDnet2014. On the right: labelled ground-truth for the image on the left.

videos acquired by static surveillance cameras. The 20 synthetic videos concern two urban scenes (a roundabout and a street) under different conditions, such as bad

weather, artificially added noise, or dynamic background. Though the BMC dataset is still used in the literature today (28), CDnet2014 is by far the most commonly used dataset. In particular, the BScGAN algorithm is the only algorithm of the ones discussed in Section 2.2 that reported results on the BMC dataset. Therefore, the BMC dataset is not included in Section 2.5, where we compare methods, since a comparison of the algorithms discussed in Section 2.2 would not be possible on this dataset.

FgSegNet S and FgSegNet v2 were also tested on the SBI2015 dataset (67) and on the UCSD dataset (68). The SBI2015 dataset, which was originally designed as a benchmark for background initialisation algorithms, contains 14 videos with ground truth labels for each frame; of the modern background subtraction algorithms listed in Section 2.2, only FgSegNet _S, FgSegNet v2, and Cascade MSCNN were tested on the SBI2015 dataset. The UCSD dataset contains 18 videos (with ground truth labels) which showcase highly dynamic backgrounds, and therefore it constitutes an excellent tool for gauging the effectiveness of a background subtraction algorithm in a complex environment such as those encountered during the recording of human movement activities. The only modern algorithms that were tested on the UCSD dataset were FgSegNet S and FgSegNet v2. Therefore, similarly to BMC, this dataset is not considered in Section 2.5.

The shift towards a single, large, general-purpose dataset since Bouwmans' review (35) is a positive change for the field of background subtraction. If algorithms are tested on different datasets, their direct comparison becomes ambiguous, or in some cases impossible. Furthermore, by testing an algorithm on a dataset that is too narrowly focused on a specific challenge (for example, dynamic

backgrounds), the algorithm will tend to overfit the dataset, thus losing generalisability. However, some authors still choose to test their algorithms on datasets with very narrow focuses. For example, 3D-Net (42) was tested on CDnet2014 and on the ESI dataset (69), which focuses on a challenge (rapidly changing scene illumination) that is absent from CDnet2014. This practice is not to be discouraged. Indeed, authors should be incentivised to test their algorithm on general-purpose datasets like CDnet2014 to demonstrate that their model generalises well and to allow a direct comparison with other methods, and then to test their algorithm on a dataset that reflects the specific challenge that their algorithm wants to address.

Metrics

The CDnet2014 framework includes 7 metrics to measure the accuracy of background subtraction algorithms (27). The metrics included in CDnet2014 are *Specificity, False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classification (PWC), Precision, Recall, and F-measure*.

Intuitively, *Recall* is the ability of a model to find all the relevant cases (called *True Positives*, or *TP*) within a dataset, while *Precision* is a measure of how many of the cases labelled as relevant actually were relevant. An algorithm is considered accurate if it achieves high *recall* without sacrificing *precision*. The *F-measure* is a weighted harmonic mean of *precision* and *recall*, and as such it allows to express the accuracy of a model with a single parameter, thus enabling an immediate comparison between algorithms. For this reason, the *F-measure* is the most widely reported parameter of accuracy for background segmentation algorithms. However,

since it does not incorporate true negatives in its computation, it is sensitive to imbalanced data, and background subtraction ground truths are inherently imbalanced, as mentioned in Section 2.2.4. Such an imbalance between the two classes is particularly problematic for deep learning methods, which suffer from bias when trained on heavily imbalanced data. For this reason, Lim and Keles (50), developers of FgSegNet, have used, along with the metrics of CDnet2014, the *Matthews Correlation Coefficient (MCC)*. The *MCC* metric is defined as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}} \quad (1)$$

where *FP*, *FN*, *TN* and *TP* denote the number of False Positives, False Negatives, True Negatives, and True Positives, respectively. The use of the *MCC* metric for imbalanced data was proposed by Boughorbel et al. (70). Unlike the *F-measure*, which assumes values in the interval [0,1], the *MCC* takes values in the interval [-1, 1], with 1 = complete agreement, −1 = complete disagreement, and 0 = uncorrelation between the prediction and the ground truth. Although the use of the *MCC* in background subtraction has been limited to only a few studies (50)(60), it is likely that with the growing popularity of deep learning methods it will also gain popularity.

Vacavant et al. (65), authors of the BMC dataset, advocate the use of the *Peak Signal to Noise Ratio (PSNR)* metric, defined as:

$$PSNR = \frac{1}{2}\sum_{i=1}^{n} 10 \log_{10} \frac{m}{\sum_{j=1}^{m} ||S_i(j) - G_i(j)||^2} \quad (2)$$

where $S_i(j)$ and $G_i(j)$ are the j^{th} pixels of image i (of size m) in the sequences S and G, respectively (where "sequence S" and "sequence G" are two of the videos in the test set of the BMC dataset). Vacavant et al. also propose the use of what they call "application quality metrics": the *Structural Similarity (SSIM)* (71) and *D-score* (72) metrics. The *D-score* evaluates the localisation of the errors (i.e. where on the image the false positives are located), whereas the *SSIM* is a perception-based metric that measures the perceived change in structural information in an image (71). To date, no other researchers have used these metrics for background subtraction.

Wang et al. (47) suggest two metrics (*FPED = False Positive Error Distance*, and *FNED = False Negative Error Distance*) to quantify how far from the nearest foreground object the wrongly classified pixels are. Both of these metrics are conceptually similar to the D-score used by Vacant et al. and, similarly to the *D-score*, both of them are unused in the background subtraction literature.

Finally, authors often report the computational speed of their method (42)(47)(50)(60)(61), expressed in terms of training time or *frames per second (FPS)*. This metric is fundamental in the field of intelligent video surveillance, where real-time computation is often necessary. Therefore, in this field it is sometimes necessary to sacrifice accuracy for speed of execution.

Evaluation of Methods

Table 1. Results of Background Subtraction Algorithms Tested on CDNET2014

Algorithm	Average F-Measure	Average Recall	Average Precision	Average PWC	Average MCC	FPS (320x240)
Cascade MSCNN	0.9209	0.9506	0.8997	0.4052	0.9274	12.5 (GPU)
3D-Net	0.9507	0.9609	0.9499	0.2650	-	-
FgSegNet	0.9770	0.9836	0.9758	0.0559	0.9863	17.99 (GPU)
FgSegNet_S	0.9804	0.9896	0.9751	0.0461	-	21 (GPU)
FgSegNet v2	0.9339	0.9476	0.9232	0.3281	-	-

Table 1 compares the accuracy of the algorithms presented in Section 2.2, in terms of the metrics discussed in Section 2.4, on the CDnet2014 dataset. Since they are not part of the CDnet2014 challenge, the *FPED*, *FNED*, and *D-score* metrics were omitted from Table 1; the *MCC* metric, due to its relevance for deep-learning-based methods, was included, although most authors do not report this metric. The results in Table 1 were collected by reviewing the actual papers and by browsing the online leaderboard for CDnet2014[12]. The CDnet2014 leaderboard was particularly useful for determining the processing speed of the algorithms, which not all authors reported in their papers.

FgSegNet S and FgSegNet _v2 were the only algorithms to be tested on the UCSD dataset. Therefore, the inclusion of the UCSD dataset here would defy the purpose

[12] http://changedetection.net/

of this section, which is to compare the best-performing algorithms on the most popular datasets available. The SBI2015 and BMC datasets were omitted for the same reason.

Table 1 shows how FgSegNet v2 is the most accurate method under all metrics considered, except for the *MCC* which was not reported by the authors and is not one of the metrics available on the CDnet2014 website. However, the processing speed of FgSegNet v2 is far slower than that of BScGAN, which in fact is orders of magnitude faster than any other method reported in Table 1.

Semantic Segmentation

Overview

Garcia-Garcia et al. (33) defined semantic segmentation as the task of "assigning each pixel in an image to an object class". A more precise definition is given by Thoma (73), who defines semantic segmentation as "the task of clustering together parts of images which belong to the same object class". Zhu et al. (74) give yet another definition, arguing that semantic segmentation is the task of "dividing a natural image into some non-overlapped meaningful regions". Using a clear vocabulary is essential in order to avoid confusion of terms. For instance, the meaning of the terms "object" and "meaningful" in Zhu et al.'s definition is subjective: people who are told to segment the "meaningful" parts of the "objects" in an image will most likely segment different things (75). This means that, if the definition of what is to be segmented is not clear, semantic segmentation is an ill-posed problem. To clarify the terminology used in the rest of this section, we

propose the following categorisation of semantic segmentation algorithms (following (73)), based on four criteria:

- Operation state (interactive vs passive). Examples of interactive algorithms are the segmentation tools present in Adobe Photoshop and MATLAB [98], which require the user to click on the background to mark it or to provide a coarse initial segmentation which the algorithm will then refine (73). Passive algorithms, on the other hand, do not allow the users to interact with the image or manipulate it in any way;
- Allowed classes (multiple vs binary). As stated earlier, a clear definition of the object classes that need to be segmented is fundamental. Most algorithms fix a priori the number of classes, which can be multiple (cat, house, car, person, etc) or binary (e.g. foreground vs background; cat vs non-cat). In this sense, by taking the definition of semantic segmentation of Garcia-Garcia et al. (33) reported above, background subtraction could be interpreted as a sub-category of semantic segmentation in which only the foreground and background classes exist. However, as pointed out by Zhu et al. (74) and Thoma (73), semantic segmentation clusters together groups of pixels, thus nullifying the issue of isolated false positives/negatives found in background subtraction (see Section 2.2.1) and distinguishing the two fields;
- Type of input data (greyscale vs coloured, 2D vs 3D data, including vs excluding depth data);
- Degree of supervision (unsupervised vs weakly-supervised vs fully-supervised). Unsupervised methods do not have access to a label or ground

truth during training, whereas fully-supervised algorithms have at their disposal a ground-truth for each image to be segmented (74). Weakly-supervised methods use partial or coarse annotations, and as such often belong to the "interactive" operation state category.

Currently, semantic segmentation research is focused on the multi-class category, because it has more widespread applications (33)(76). Nevertheless, multi-class algorithms can easily be re-trained for a binary classification problem such as the one encountered during the first step of a visual hull pipeline (i.e. segmenting the foreground, class1, from the background, class0). For this reason, this section of the review will not be restricted to binary classification algorithms. However, because 3D data and depth data cannot be integrated into a shape-from-silhouette pipeline, algorithms that deal with such data will not be analysed here.

The applications of semantic segmentation range from the detection of road signs (77), to the segmentation of brain scans for the detection of tumours (78)(79), to the detection of objects in satellite images (80)

State of the Art

Traditional approaches to semantic segmentation relied heavily on domain knowledge to build an algorithm with domain-specific features (73), colour being the feature used most commonly (73)(74). Although no colour space has been proven to be superior to all others in all contexts (81), RGB is often chosen due to its simplicity and support by programming languages; occasionally, the HIS colour space is chosen due to its property of being invariant to illumination (82)(83)(84)(85). An example of domain-specific features are the "poselets"

introduced by Bourdev and Malik (86) for human pose estimation. Poselets are manually added extra keypoints such as “left shoulder” or “right shoulder” which aid in the task of detecting the poses of people in a scene (87)(88).

As was the case for background subtraction, the field of semantic segmentation was revolutionised by the advent of Deep Learning, and in particular of Fully Convolutional Neural Networks (FCNs, which are a particular kind of CNN): a glance at the most popular semantic segmentation datasets (89)(90) will reveal that almost the entirety of the new research in semantic segmentation adopts networks based on the FCN architecture. For this reason, and because traditional semantic segmentation algorithms are not used in the visual hull literature like traditional background subtraction algorithms were, this review will omit the analysis of traditional methods for semantic segmentation; for a review of this category of algorithms, please refer to (73)(74).

CNNs applied to semantic segmentation face two challenges: 1) because their structure was originally designed for the task of image classification, they lose feature resolution at each layer (in other words, the deeper the layer the more complex the features it encodes, but also the lower its awareness of spatial resolution; this is because the dimension of the feature maps is halved at each depth; 2) objects may exist at multiple scales, thus requiring the network to have both large and small fields of view. Both of these challenges have been addressed extensively by the algorithms discussed in this section.

Including in the following analysis every algorithm that uses CNNs for semantic segmentation would not help us reach the objective of this paper. As was the case for Section 2.2, then, this section focuses on the top-performing methods in this

category of algorithms. The algorithms discussed in this section were chosen by looking at those with an *Average Precision* (described in Section 3.4) on the PASCAL VOC 2012 dataset (89) (described in Section 3.3) of at least 90% in the "person" category. Although the threshold of 90% was chosen arbitrarily, it allowed us to single out the top ten algorithms for the semantic segmentation of humans, thus providing ample choice to users interested in applying one such tool to a visual hull pipeline.

DeepLab

Different versions of DeepLab, an algorithm developed by researchers at Google, have been at the top of semantic segmentation leaderboards for years (33)(76)(89). One of the main features of DeepLab is its use of atrous convolutions to solve the issue of the loss of feature resolution in the deep layers of CNNs. Atrous convolutions (from the French "a trous", with holes), also known as dilated convolutions, expand the resolution of the filter in the convolutional layer according to a parameter called *dilation rate*; in practice, this process fills with zeros the empty elements of a dilated filter (see Fig. 5). Atrous convolutions allow to control the resolution at which features are computed within the network, and to scale their resolution multiple times without having to learn new parameters like in an encoder-decoder structure. For a two-dimensional signal, for each location i on the output feature map y and a convolution filter w, an atrous convolution is applied over the input feature map x as follows:

$$y[i] = \sum_{k} x[i + r \cdot k] \cdot w[k] \tag{3}$$

where the atrous rate *r* determines the stride with which the input signal is sampled. During training, DeepLab uses patches cropped from the original image. Because of the atrous convolutions present in DeepLab, a large crop size is required to avoid that filter weights with large atrous rates be applied to the zero-padded region (i.e. the empty space within the dilated filter).

DeepLabv3 was introduced in 2017 (91) and has undergone several iterations since. Its backbone is ResNet-101 (46), a network for image recognition pre-trained on the ImageNet dataset (92). In DeepLabv3, the last blocks of ResNet are re-purposed using Atrous Spatial Pyramid Pooling (ASPP). First proposed by (93), the ASPP is a module of four cascaded atrous convolutional layers with different atrous rates that allows DeepLab to capture multi-scale information at the level of the features learned by the network (91), instead of at the level of the input features (like in Cascade MSCNN). In two separate experiments, DeepLabv3 was pre-trained on the MS-COCO dataset [110] and on the JFT-300M dataset[13] [114]; both pre-training regimens noticeably improved the performance of DeepLabv3(94), as we discuss in Section 3.5.

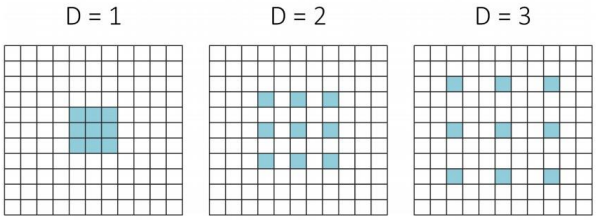

Figure 5. Atrous convolutions allow the network to obtain varying spatial resolution by changing the dilation rate (D). The number of parameters to learn (in this figure, 9, one for each coloured square) does not change, since the empty elements within the filter are filled with zeros).

[13] The version of DeepLab3 pre-trained on the JFT-300M dataset takes the name of "DeepLab3-JFT", while the version of DeepLab3 pre-trained on the MS-COCO dataset simply takes the name of "DeepLabv3".

DeepLabv3+ (95) further improved upon DeepLab3 by implementing an encoderdecoder structure that uses DeepLab3 as the encoder module and a simple series of upsampling layers and convolutional layers as the decoder module. Furthermore, it reduced the computational complexity of the algorithm by adopting depthwise separable convolutions: the standard convolution operation is factorised into a depthwise convolution, which performs a spatial convolution independently for each input channel, and a point-wise convolution, which combines the output from the depthwise convolution step. Another reason why DeepLabv3+ performs better than DeepLab3 is because it adopts the more powerful Xception (96)(97), instead of ResNet-101, as its backbone network. Furthermore, it was pre-trained on ImageNet and JFT-300M in two separate instances.

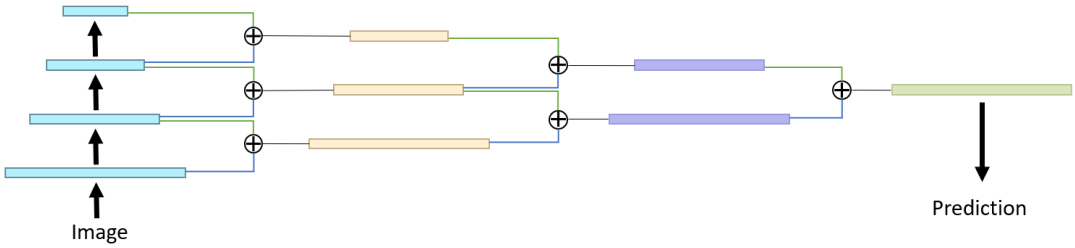

Figure 6. Architecture of the MSCI network. On the left, a traditional convolutional neural network structure encodes features into layers of progressively smaller reception fields. Each pair of adjacent layers is connected "horizontally" using bi-directional connections, which progressively incorporate information from different receptive fields into the final prediction (to the right). (Image from (98))

MSCI

In order to capture the multi-scale information present in the data, the MSCI algorithm (98) combines the outputs of pairs of adjacent layers, as shown in Fig. 6. Whereas in most multi-scale architectures the information flows in a unidirectional fashion, in MSCI the connections are bi-directional, with long short-term memory (LSTM) chains connecting feature maps of different resolutions. These intertwined

connections are present both horizontally and vertically in the network structure (illustrated in Fig. 6), granting exceptional integration of the different levels of information present at each layer. Using the structured edge detection toolbox (99), MSCI divides the input image and each subsequent feature map into sets of non-overlapping regions called super-pixels. The neurons that correspond to neighbouring super-pixels are densely connected with bi-directional LSTM connections, which allow great local consistency to the segmentation: the super-pixels form a sort of "patchwork" of small patches of high resolution, which are tightly interwoven together to make seams disappear, thus granting high fidelity to the contours of the segmented objects. MSCI uses ResNet-152 [64] as its backbone network and the model is pre-trained on the MS-COCO dataset (90) following (100)(101).

ExFuse

Many networks that use an encoder-decoder architecture gradually fuse the information from the bottom layers, which is low-level[14] but high-resolution, with the information from the top layers, which is high-level but low-resolution. It is the case, for example, of U-Net (59), which was adopted by several authors as the backbone network for their semantic segmentation algorithm (102)(103)(100)(104). Zhang et al., authors of ExFuse (105), argue that fusing "pure" low-level and high-level features is inefficient and leads to inaccurate results, and propose to introduce more semantic information in low-level features and more spatial information in

[14] Information is progressively encoded as the layers get deeper. Therefore, the first layers will contain information that is scarcely encoded, and which is consequently defined as low-level. An example of a low-level feature is an edge map of the original image, which requires little encoding to obtain.

high-level features, thus increasing the content overlap between distant layers that will be fused together. They introduce more semantic information into low-level features using three strategies:

- they rearrange the layers of the backbone network (ResNeXt 101 (106)) to be more evenly distributed instead of being clumped up in the deep blocks;
- they use auxiliary supervision at the early stages of the encoder, a practice inspired by Deeply Supervised Learning (107)(108);
- they do not fuse layers in a binary fashion as in U-Net (59); instead, before being fused with its high-level counterpart, each low-level layer is combined with the ones directly above it using a novel module called "semantic embedding branch", the purpose of which is to embed more semantic information into low-level features before they are fused with high-level features.

They introduce more spatial information into high-level features using two strategies: 1) they use auxiliary supervision on the first deconvolutional module of the decoder; furthermore, the original deconvolution of the module is replaced with Sub-Pixel Upsample (109), which enlarges the feature map just by reshaping the spatial dimensions; 2) they introduce the "Densely Adjacent Prediction" mechanism, which enables feature points of the decoder to estimate the semantic information of adjacent points; the final segmentation for each point is obtained by averaging all the associated scores.

DPC

Inspired by the Neural Architecture Search (NAS) model (110)(111), the Dense Prediction Cell (DPC) method developed by Chen et al. (112) does not rely on human expertise to manually construct a neural network for semantic segmentation. Instead, they construct a space of possible network architectures and use an optimisation tool (113) to select the most optimal architecture within the space. They populate the space using most state-of-the-art semantic segmentation algorithms, such as (91)(93)(97)(101). As the optimisation tool selects random architectures from within the search space to evaluate them, the selected architecture has to be trained. However, training large networks for semantic segmentation is a time-consuming task, and iterating through a search space of several architectures would be prohibitively expensive in terms of computational time. Therefore, the authors developed a proxy task on which to train the candidate architectures. The objective of a proxy task is to provide the candidate architecture a task that is quick to evaluate and that gives an output that is easily relatable to the large-scale task (112). To achieve this goal, the authors employed, as a proxy task, a smaller network backbone and cached the feature maps produced by the network backbone on the training set, and then directly build DPC on top of it. The optimisation tool is then run on the architecture search space using the proxy task; after optimisation has ended, the selected architecture is tested on the large-scale task.

Datasets

The dataset most referenced in recent semantic segmentation research is undoubtedly the PASCAL VOC 2012 dataset (89). The PASCAL Visual Object

Classes (VOC) project ran challenges evaluating performance on object class recognition algorithms from 2005 to 2012, each year developing a new or modified dataset; starting from 2007, a semantic segmentation challenge was added. The 2012 dataset consists of 28,952 images split into 50% for training-validation (with public ground-truths) and 50% for testing (with private ground-truths); of these 28,952 images, only 9,993 are labelled for segmentation (see Fig. 7 for an example of an image in PASCAL VOC 2012). For each image in the dataset, the bare-minimum label consists of bounding boxes that surround objects that belong to one of the following twenty-one categories: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, tv/monitor, and background. All human labellers, the number of which is not shared by PASCAL VOC, were provided with the same guidelines for the segmentation of the ground-truths:

- only segment objects whose bounding boxes have been labelled;
- labelled pixels MUST be the object; pixels outside a 5-pixel border area MUST be background. Border pixels can be either;
- pixels which are mixed e.g. due to transparency, motion blur or the presence of a border should be considered to belong to the object whose colour contributes most to the mix;
- aim to capture thin structures where possible, within the accuracy constraints. Structures of roughly one-pixel thickness can be ignored e.g. wires, rigging, whiskers;
- if a number of small objects are occluding an object e.g. cutlery/silverware on a dining table, they can be considered part of that object. The exception is if

they are sticking out of the object (e.g. candles) where they should be truncated at the object boundary.

PASCAL VOC also provides a public leaderboard that reports the accuracy of the methods submitted to the website. The size of the images in PASCAL VOC 2012 varies but is generally within 500x500 pixels.

Figure 7. On the left: example of an image in PASCAL VOC 2012. On the right: labelled ground-truth for the image on the left.

The Microsoft Common Objects in Context (MS-COCO) dataset (90) is also widely referenced in the semantic segmentation community. For the object segmentation task, MS-COCO includes over 200,000 images (all of which are fully annotated) split into 80 categories. However, MS-COCO focuses on instance segmentation[15] rather than on semantic segmentation. In instance segmentation, all objects in the image that belong to different instances of the same object class must be labelled separately (see Fig. 8 for an example). Therefore, the methods reported in Section 3.2, which are semantic segmentation algorithms, were never tested on MS-COCO. Nevertheless, most of the algorithms in Section 3.2 use MS-COCO to pre-train their network, under the assumption that a network pre-trained on a large dataset like MS-COCO will perform better than a randomly pre-trained network

[15] This review focuses on methods for silhouette extraction that can be applied to markerless motion capture. Because such a silhouette extraction algorithm would only have to deal with a single object in the image (i.e. the human subject being recorded), instance segmentation algorithms, which focus on the segmentation of multiple objects of the same class, were not considered in this review.

(33). The size of the images in MS-COCO varies but is generally within 640x640 pixels.

Another popular dataset in the semantic segmentation literature is the Cityscapes dataset (114), which focuses on urban scenes[16]. Semantic segmentation algorithms are often applied to autonomous driving or urban scene recognition tasks, making

Figure 8. Example of an image in MS-COCO with superimposed ground-truth. Each instance of an object class is labelled separately. For example, each person in this image is segmented using a different colour.

Cityscapes a particularly important dataset on which to test algorithms. The dataset consists of 5,000 fully-labelled images and 20,000 coarsely-labelled images (see Fig. 9 for an example of an image in Cityscapes) extracted from videos shot in 50 different cities during daytime. The labels are divided into 30 classes, including "person" and "rider" to distinguish pedestrians from people on vehicles. The images in Cityscapes are considerably larger than those in other popular datasets, with a resolution of 2040x1016 pixels.

Finally, some semantic segmentation algorithms (like MSCI) are tested on scene labelling datasets like NYUDv2 (115), PASCAL-Context (116), and SUN-RGBD (117); MS-COCO also has a scene labelling challenge. Unlike the object-centric

[16] Although this dataset is more pertinent to algorithms designed for self-driving cars or similar applications, we include it in this review paper because it features human beings as one of its object classes, and because it is such a widely recognised benchmark dataset that many readers will be familiar with it.

PASCAL VOC, these datasets focus on the segmentation of scenery and "stuff" like grass, sky, and wall. Because this application does not match the one on which this review focuses, these datasets will not be discussed further here.

Metrics

The concepts of *recall* and *precision* were introduced in Section 2.4. In background subtraction, these metrics are often combined into a single metric, the *F-measure*; in semantic segmentation, *recall* and *precision* are used to calculate a metric called *average precision* (*AP*). Let us assume that a segmentation model has a confidence threshold *T* on its predictions: the model gives a certain label to a pixel if its confidence in the label exceeds *T*. To this model score threshold correspond a value of *recall* and a value of *precision*. The *Average Precision* summarises the shape of the *precision-recall* curve obtained by varying the threshold of the model so that eleven equally-spaced *recall* levels are obtained (r = [0,0.1,0.2,...,1]). In other words, the *AP* is the mean *precision* at a set of eleven equally spaced *recall* levels:

$$AP = \frac{1}{11}\sum_{r\epsilon\{0,0.1,\ldots,1\}} p_{interpolated}(r) \qquad (4)$$

The *precision* at each *recall* level r is interpolated by taking the maximum *precision*

Figure 9. Top: example of a finely labelled image in Cityscapes. Bottom: examples of a coarsely labelled image in Cityscapes.

measured for a method for which the corresponding *recall* exceeds *r*:

$$p_{interpolated}(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r}) \qquad (5)$$

where $p(r)$ is the measured *precision* at *recall* r. This metric penalises methods which e e detect only a fraction of examples with high *precision*, since it forces the algorithm to have *precision* at all levels of *recall* [109]. But how is the threshold T set, and what does it represent? In the case of PASCAL VOC 2012, T corresponds to an *Intersection over Union* (*IoU*) value greater than 0.5 (89). The *IoU* metric measures how well the ground-truth object overlaps the object predicted by the model:

$$IoU = \frac{Area\ of\ intersection}{Area\ of\ union} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (6)$$

In the PASCAL VOC 2012 dataset, the metric reported is *AP* at $IoU > 0.5$; in other words, a prediction is considered positive if $IoU > 0.5$, and this threshold allows to calculate eleven values of *recall* with corresponding values of *precision*,

which in turn allow to calculate the *AP*. In the MS-COCO dataset, the metric used is the *meanAP* (*mAP*): the *AP* is averaged over 10 values of *IoU*, providing a much stronger metric that rewards methods that are better at precise localisation (118).

The Cityscapes dataset does not use *AP* as its main metric and focuses on *IoU* instead. However, since the *IoU* metric is known to be biased toward object instances that cover a large image area (114), Cityscapes also introduces a metric called *instance-level IoU* (*iIoU*). *iIoU* is computed exactly as in equation 18, but *TP* and *FN* are computed by weighting the contribution of each pixel by the ratio of the class' average instance size over the size of the respective ground truth instance. Because this metric only pertains to cases where multiple instances of the same object class are present in an image (a condition that will never occur in human motion capture scenarios), the *iIoU* metric is not be considered further in this review.

PASCAL VOC 2012 and MS-COCO do not report values of running time for the algorithms described in Section 3.2. On the website for the Cityscape dataset[17], the running time of 43 of the top 125 methods is reported; none of the 43 methods for which the running time is reported are in the top 20 list.

Evaluation of Methods

Table 2 compares the accuracy of the algorithms presented in Section 3.2, in terms of the metrics discussed in Section 3.4, on the PASCAL VOC 2012 dataset. Results on the MS-COCO datasets are not reported in this section because MS-COCO deals with instance segmentation, which is not pertinent to the focus of this review paper.

[17] https://www.cityscapes-dataset.com/

Similarly, because the Cityscapes dataset focuses on urban traffic scenes, results of methods tested on this dataset are not be reported in this section.

The results reported in Table 2 correspond to the *AP* values of each method at $IoU > 0.5$. The rightmost column reports the *AP* values for the "person" category, while the centre column reports the *AP* values averaged across all twenty categories of PASCAL VOC 2012. Table 2 shows how the top ten algorithms for semantic segmentation evaluated on PASCAL VOC 2012 are very close to each other in terms of *AP* in the "person" category. Nevertheless, the more recent versions of DeepLab, in particular DeepLabv3+ JFT (95) and DeepLabv3+ AASPP (still unpublished), are the most accurate semantic segmentation methods available to date.

Table 2. Results of Semantic Segmentation Algorithms Tested on PASCAL VOC 2012

	AP (%)	
Algorithms	**mean**	**person**
DeepLabv3	85.7	92.1
DeepLabv3-JFT	86.9	92.3
DeepLabv3+	87.8	92.8
DeepLabv3+ JFT	89	93.8
DeepLabv3+ AASPP (unpublished)	88.5	93
MSCI	88	92.8
ExFuse	87.9	92.3
DPC	87.9	92.5
SRC-B-MachineLearningLab (unpublished)	88.5	92.9
DFN (unpublished)	86.2	91.7

Multi-View Segmentation

Overview

To reconstruct a visual hull, multiple images of the same object recorded by cameras in different positions are required, and from each image the silhouette of the object has to be extracted. So far, this review has dealt with algorithms that solve silhouette extraction in a monocular manner (i.e. one camera view at a time). Therefore, in a visual hull pipeline, the algorithms of background subtraction or semantic segmentation described in Sections 2 and 3 would need to be applied to each camera view independently. This monocular approach does not take advantage of the redundancy present in a set of co-temporal images of the same object. Multi-view segmentation algorithms attempt to exploit such redundancy, usually in the pursuit of one of two goals: to improve the overall accuracy of the system (21), or to reduce its processing time (119). If colour (119)(120) or geometric (21)(121)(122) consistency are enforced across camera views (i.e. we expect to see similar colours in all camera views at a certain point in space), it becomes possible to correct errors in the segmentation in one camera view using the information present in another. As argued by Nobuhara et al. (21), however, merely relying on colour consistency between camera views may lead to errors. In the monocular case, background subtraction algorithms operate on the assumption that the colour of the background is different from the colour of the foreground. In the multi-view case this assumption is not self-evident, because if the object has a colour that is completely different from the background in one viewpoint but is similar to that of another view, extraction of consistent silhouettes can be difficult. Therefore, some authors

(21)(121)(122) prefer to model their algorithms using geometric consistency constraints instead.

State of the Art

Research in this field is not as active as in the fields of background subtraction and semantic segmentation. To the best of our knowledge, the latest article proposing a new method of multi-view segmentation was published in 2016 (123). For comparison, in 2018 alone there have been three versions of FgSegNet (see Section 2.2.4) (50)(60)(61). Although research in this area could be said to be stagnant, methods published in the early 2010s already achieved exceptional accuracy, and the fact that most of these algorithms were developed specifically for integration within a visual hull pipeline makes them relevant to this review paper. As was the case for Sections 2.1, 2.2, and 3.2, only the top-performing algorithms of this category are reported in the following section. However, unlike in the fields of background subtraction and semantic segmentation, multi-view segmentation does not have a benchmark dataset that ranks methods based on their accuracy. Therefore, the algorithms reviewed in this section were selected by thoroughly reviewing the literature and establishing which algorithms performed best in terms of accuracy and/or runtime. Also, algorithms which could not be seamlessly implemented in a visual hull pipeline for markerless motion capture are not included in this section. For example, methods that rely heavily on interactive inputs from users (124)(125)(126) would complicate the overall system excessively, and are therefore not considered in this section.

Because the authors of the papers reviewed in this section did not name their algorithms like the authors of background subtraction and semantic segmentation algorithms did (e.g. FgSegNet, DeepLab, etc), in the following sections we will refer to the multi-view segmentation algorithms being discussed by using the last name of the first author of the paper, followed by the year in which the paper was published.

Nobuhara 2009

The method developed by Nobuhara et al. (21) tries to solve multi-view segmentation and visual hull reconstruction simultaneously using an iterative process. Initially, rough silhouettes are extracted and used to create a rough visual hull, which in turn is projected back onto each camera view. Geometric constraints are then employed to correct the silhouettes, which are used to create a slightly more refined visual hull. This iterative process continues until a convergence criterion (described later in this section) is met. The constraints used in Nobuhara 2009 are the following:

- Intersection constraint (IC): the projection of the visual hull on every camera view should be equal to the silhouette on that camera view;
- Projection constraints (PC): each 2D camera view can be segmented into regions so that each region fully belongs to either the foreground or the background. This constraint operates on the assumption that two adjacent points on the surface of a 3D object with similar properties, different from those of the background, have similar projections on the 2D camera view (127);

- Background subtraction constraint (BC): the sum of differences of pixel intensity between the regions identified as background and the regions identified as foreground should be greater than a certain threshold; this constraint is equivalent to applying basic background subtraction (as described in Section 2.1) on top of the multi-view segmentation algorithm.

The entire region of each camera view is used as the initial silhouette for that camera view; these initial "silhouettes" produce a gross over-estimation of the visual hull volume. The PC constraint is then used to establish the initial foreground and background regions, yielding a new approximation of the silhouette of the object, which is polished using the BC. At this point, a new visual hull is constructed and then projected onto each camera view, where its projection is subjected to the carving effect of the PC, IC, and BC constraints, in that order. This iterative process is repeated until the carving effect of the PC and BC violates the IC; in this sense, the IC serves as a stop criterion for the algorithm.

Djelouh 2012

Methods, like Nobuhara 2009, that rely on the joint extraction of 2D silhouettes and reconstruction of the 3D object (121)(122)(128) are computationally expensive (129). Instead of reconstructing a dense 3D representation of the object from which to define geometric constraints, Djelouah 2012 (129) proposes to use sparse samples that only convey colour information. Given a sample point in 3D space, in each camera view its projection is identified by a certain pixel intensity value (i.e. a colour); the *n* colours present at the *n* pixel projections of the sample define a colour *n*-tuple, which is the basic unit on which Djelouah 2012 operates. The *n*-tuple does

not convey 3D positional information like visibility or neighbourhood (see "Projection Constraint" in Section 4.2.1), which means that the complexity of the problem is reduced. Furthermore, using sparse samples (10,000 in total) instead of a dense 3D representation (for example, (130) used 8,000,000 voxels for their dense 3D representation) reduces the running time of the algorithm.

In Djelouah 2012, background and foreground colour models are built for each camera view independently following (131)(127). The *n*-tuple is then checked against the colour distributions in each camera view: for a sample to be labelled as "foreground", it needs to lie in the foreground region of all *n* camera views; if even a single camera view labels the projection of the sample as background, the sample is labelled as background and the other camera views do not need to be checked, thus additionally saving computational power. The sparse 3D sampling, which consists of 10,000 samples, is then combined with the per-view colour models, yielding a dense segmentation in all camera views.

Kowdle 2012

Like Nobuhara 2009, the algorithm developed by Kowdle et al. (132) performs 2D segmentation and 3D reconstruction jointly. Firstly, a piecewise planar, layer-based depth map is estimated for each camera view by combining stereo matching (obtained via semi-global matching (133)) with appearance cues based on colour models. The depth maps consist of a set of 3D planes, and each pixel in a camera view is assigned to one of these planes. The depth maps are then refined using a Gaussian Mixture Model learnt for each surface. Secondly, the algorithm establishes which planar surfaces belong to the object by using appearance and depth cues, as

well as by verifying that they be visible in multiple camera views. After each surface has been labelled as either "object" or "background", their projection in each 2D camera view yields an initial segmentation. Thirdly, these initial segmentations are fused across multiple views using a probabilistic model, thus generating the final segmentation; this third step is similar to enforcing silhouette consistency in multiple views (see "PC" in Section 4.2.1). However, since in this case the 3D representation is provided by the depth maps and not by a dense 3D geometric reconstruction of the object as in Nobuhara 2012, Kowdle 2012 achieves higher computational efficiency, which is further boosted by the fact that the 3D representation is computed only once, and not repeated through several iterations as in Nobuhara 2012.

Djelouah 2013

Like Djelouah 2012, Djelouah 2013 (134) uses sparse 3D samples to obtain an initial sparse segmentation which is later refined. Unlike in Djelouah 2012, though, colour consistency across views is not the only cue that is used. Djelouah 2013 can be divided into an initialisation phase, an iteration phase, and a final segmentation phase. During initialisation, the image is divided into superpixels using the SLIC algorithm (135), and similar superpixels are linked together using appearance descriptors; for each superpixel, a colour model is initialised using a custom-made k-means model. Links between superpixels from successive frames are established using a time consistency descriptor based on optic flow. The spatial and time links between superpixels are then used to minimise a Markov Random Field (MRF) energy function, which in turn is used to update the colour models for each camera

view; the MRF optimisation is iterated until an arbitrary convergence threshold is reached, after which time a pixel-level graphcut segmentation (136) is performed using the colour models derived from the superpixel segmentation. The innovation of Djelouah 2013 is not its use of superpixel-segmentation as an intermediate step (a strategy already reported by Campbell et al. (136)), but rather its use of sparse 3D samples (100,000, unlike the 10,000 in Djelouah 2012) to link superpixels, and the presence of time links between consecutive frames. This latter innovation allows Djelouah 2013 to exploit the redundancy between adjacent temporal frames to further increase segmentation accuracy, a strategy reminiscent of the background subtraction algorithm 3D-Net described in Section 2.2.2.

Datasets

Unlike in background subtraction and semantic segmentation, in multi-view segmentation there is no single benchmark dataset on which all algorithms are tested. Instead, most authors develop their own image sets to evaluate their method or use image sets developed by other authors.

Djelouah 2012 (129) was tested on eight datasets (Arts Matriaux, Bear (132), Bike (132), Bust (137), Couch (132), Car (132), Pig (130), Rabbit (130)). The Couch and Bear datasets were also used by Kim et al. (138), who also developed the Tree, Lion2, and Lion3 datasets, which were not used by any other authors. Nobuhara 2009 (21) was first tested on virtually-constructed cubes, and then tested on images that the authors acquired themselves and which are not public. Gallego et al. (128) also developed their own private image set. Kowdle 2012 [156] was tested on the Couch, Teddy, Bike, Chair1, Chair2, and Car datasets, all of which were

developed by the authors of Kowdle 2012. Djelouah 2013 (134) reports results on the Couch, Bear, and Car datasets used in Djelouah 2012, and additionally use the Chai1 dataset from (132). Kolev et al. (130) report results on, amongst others, a "Bust" dataset: closer inspection will reveal that this dataset is not the same as the one with the same name used by Djelouah 2012. Similarly, Djelouah 2013 reports results on a "Buste" dataset, which is actually the "Bust" dataset from Djelouah 2012. Although linguistically trivial, these differences and similarities between dataset names may lead to confusion if care is not taken in analysing the results reported by different papers. Furthermore, the lack of a cohesive benchmark dataset on which to test algorithms means that different methods cannot be easily compared to one another. Additionally, the datasets listed above usually comprise between 8 and 45 images of a single object. This limited amount of data means that methods tested on such datasets could be subject to overtraining and may not generalise well to new examples. Therefore, care needs to be taken when evaluating their results.

Metrics

The accuracy metrics used by authors to evaluate their methods are not consistent across papers. For example, Gallego et al. (128) only report qualitative results in the form of images, while Nobuhara et al. (21) report results in terms of *F-measure* (see Section 2.4). Djelouah 2012 (129) uses three metrics defined by Lee et al. (139) (*Mean Error*, *Hit Rate*, and *False Alarms*), and also defines two novel metrics, *Accuracy* and *Missed Rate*. These metrics are defined as follows:

$$Mean\ Error = \frac{N(W_F^B)+N(W_B^F)}{number\ of\ pixels} \qquad (7)$$

$$Hit\ Rate = \frac{N(W_F^F)}{N(W_F^F)+N(W_B^F)} \quad (8)$$

$$False\ Rates = \frac{N(W_F^B)}{N(W_F^B)+N(W_B^B)} \quad (9)$$

$$Accuracy = 1 - Mean\ Error \quad (10)$$

$$Missed\ rate = 1 - Hit\ Rate \quad (11)$$

where $N(W_b^a)$ refers to the number N of pixels W labeled as a in the ground truth and b by the algorithm. It can be proven that *Mean Error*, *Hit Rate*, and *False Alarms* are identical to, respectively, the *PWC*, *Recall*, and *FPR* metrics mentioned in Section 2.4 for background subtraction, while *Accuracy* and *Missed Rate*, as defined in equations 22 and 23, correspond to the *IoU* (see equation 18) and *FNR* (see Section 2.4) metrics, respectively. Kowdle 2012 (132) and Djelouah 2013 (134) report results in terms of *Intersection over Union* (*IoU*). The *IoU* metric reported for multiview segmentation algorithms should not be confused with the results for semantic segmentation algorithms reported in Table 2, which are expressed in terms of *Average Precision* at fixed values of *IoU*. In semantic segmentation, the *IoU* value is fixed to allow the network to determine its confidence in making a prediction for the label of a pixel; *Average Precision* is then used to evaluate the method. Therefore, multi-view segmentation algorithms reported in Section 4.5 should not be directly compared to semantic segmentation algorithms reported in Section 3.5. In other words, semantic segmentation algorithms and multi-view segmentation algorithms cannot be put in the same table for easy comparison. This does not mean that general conclusions cannot be drawn regarding the validity of

one algorithm from the first domain over an algorithm from the other; indeed, in the discussion we present an analysis of which algorithms perform best under different conditions.

Evaluation of Methods

Nobuhara 2009 achieved an *F-measure* of 0.975 on the data used by the authors (which is private). Since this was the only metric reported for the method and since no other methods were tested on this data, this method is not included in any tables in this section. The processing time of Nobuhara 2009 was not reported by the authors.

The results of Djelouah 2012 on the Arts Martiaux, Bear, Bike, Bust, Couch, Car, Pig, and Rabbit datasets using the *Mean Error*, *Hit Rate*, and *False Alarms* metrics are reported in Table 3. Furthermore, the results of Djelouah 2012 on the Couch, Bear, Car, and Chair1 datasets using the *Accuracy* metric are reported in Table 4. The results of Kowdle 2012 and Djelouah 2013 for the above datasets are also reported in Table 4.

Table 3. Results of Djelouah 2012 on the datasets considered

	Metrics		
Dataset	**Mean Error (%)**	**Hit Rate (%)**	**False Alarm (%)**
Bust	0.2±0.1	99.4±0.01	0.7±0.3
Arts Martiaux	0.5±0.2	97.5±0.3	2.7±1.4
Couch	1.2±0.8	97.0±2.8	0.1±0.1
Bear	2.7±1.5	94.5±0.8	7.0±9.0
Car	2.8±0.8	98.8±0.8	16.7±8.8
Bike	2.4±1.1	96.7±2.1	25.0±13.3

Table 4. Results of Djelouah 2013, Djelouah 2012, and Kowdle 2009 on the datasets considered

	Method		
Dataset	**Djelouah 2013**	**Djelouah 2012**	**Kowdle 2009**
Couch	99.0±0.2	98.8±0.8	99.6±0.1
Bear	98.0±1.0	98.8±0.4	98.8±0.4
Car	97.0±0.8	-	98.0±0.7
Chair1	98.6±0.3	88.0±2.0	99.2±0.4

Table 4 shows how Kowdle 2012 outperforms both Djelouah 2012 and Djelouah 2013, albeit by small margins; indeed, if the standard deviation reported in Table 4 is taken into account, the difference between the three methods is almost trivial, with the exception of Djelouah 2012 on the Chair1 dataset, for which it achieves 10% lower accuracy than the other two methods. Djelouah 2012 relies entirely on colour information to perform its segmentation, and its reliance on colour makes it susceptible to errors when the foreground and background present similar colours, which is the case for the Chair1 dataset, thus explaining the lower accuracy of Djelouah 2012 on this dataset.

What separates Kowdle 2012 from Djelouah 2012 and 2013 is the processing time: for a 640x480 pixels image, Kowdle 2012 takes 2 minutes (132), Djelouah 2012 takes 2 seconds (129), and Djelouah 2013 takes 10 seconds (134).

Discussion and Future Work

Most images used in biomechanics have a resolution of 1280x1024 pixels or above (2), whereas the resolution of the images in background subtraction and semantic segmentation datasets varies from 320x240 to 640x480 pixels. This fact has two implications. Firstly, the methods described in Sections 2.2 and 3.2 would need to

be re-trained on larger images to better understand how their computational times scale with image size. Although computational time is not as strict a constraint for biomechanics as it is for applications such as intelligent surveillance (see Section 2.1), in a case where two algorithms have almost equal accuracy the faster one will be the obvious choice. For example, a biomechanist would be more likely to prefer a method such as Djelouah 2013 (134) over Kowdle 2012 (132), which is only marginally more accurate but almost 12 times slower.

The second way in which larger images would affect the training of silhouette extraction methods has to do with the patch-based approach described in Sections 2.2 and 3.2. When dealing with small images such as the ones in CDnet2014, the use of patches of 30x30 pixels is justified by the fact that, since the images themselves are no larger than 640x480 pixels, the objects in the images do not exceed the size of 30x30 pixels by much. When the scale of the object and that of the patch are comparable, it is sufficient to use a multi-scale approach as in Cascade MSCNN (47) and FgSegNet (50) to introduce enough global information to solve local inconsistencies. When dealing with large images, however, the objects to be segmented could be considerably larger than 30x30 pixels, and such a small patch could lack the global information necessary to classify the pixel correctly. Two ways to address this issue that would not evert the architecture of existing networks would be to either use larger patches (which, however, could cause a loss of fine-grained detail in the segmentation), or the use of a "deeper" multi-scale approach. For example, Cascade MSCNN resizes the original image twice, thus feeding to the network an image of size 1, one of size 0.75, and one of size 0.5; in other words, it uses a multi-scale "depth" of 3. It is left to future works to test whether a deeper

multi-scale approach (e.g. depth = 5, using images of size 1, 0.8, 0.6, 0.4, and 0.2 of the original image size) would be able to retain more global information than shallower multi-scale approaches (e.g. depth = 3) when dealing with large images. A third option, adopted for example by the background subtraction algorithm BScGAN (see Section 2.2.3), would be to dispense with the patch-based approach altogether. Implementing this strategy into networks that rely on patches for training, though, could require significant modification of the network's architecture.

A glaring issue with the state of the literature is the fact that algorithms that belong to different categories are tested on different datasets, using different metrics. This makes it difficult to compare methods that belong to different categories. Nevertheless, some conclusions can be drawn from Tables 1 to 4. For instance, FgSegNet v2 (61) and DeepLabv3+ JFT (95) are the most accurate algorithms in their respective categories, and both could be applied to a visual hull pipeline for markerless motion capture. Accuracy metrics, however, should not be the sole determiner of what algorithm to choose for a visual hull pipeline: the choice should be guided by the specific needs of the task at hand. For instance, BScGAN [36] is one of the most accurate background subtraction algorithms available to date (see Section 2.5), but it requires, for each training example, a frame unobstructed by the foreground object so that it can build its background model. In open-world human motion capture, having a frame clear of the foreground object for each frame to be analysed is not always possible, and therefore BScGAN may not be an ideal choice for this specific task. Similarly, as discussed in Section 4.5, Kowdle 2012 (132)

does not outperform other multi-view algorithms by a large enough margin to warrant its much longer computational time, and therefore it would probably not be a good fit within a markerless motion capture pipeline. Also, it should be noted that although the accuracy of multi-view segmentation algorithms seems exceptionally high, with Kowdle 2012 achieving $IoU = 99.6 \pm 0.1$ (see Section 4.5), these methods were tested on very few images and it is possible that they overfit the very specific cases presented to them during training; there is no information relative to the accuracy of multi-view segmentation algorithms on datasets different from the ones on which they were trained.

Supervised methods are inherently dependent on the type of data they are trained on. New algorithms should be tested on general-purpose datasets like CDnet2014 and PASCAL VOC 2012 to show that they can be used for many different tasks. However, were algorithms like FgSegNet and DeepLab to be trained on images that exclusively show humans performing movements, their accuracy on the task of segmenting humans performing movements would increase. Therefore, the results presented in Tables 1 and 2 should not be taken at face value, and further studies on human-specific datasets are required to understand the full potential of supervised methods for silhouette extraction in specialised tasks. Conversely, multi-view segmentation methods do not explicitly learn from data, so the amount of data available is not necessarily correlated with the accuracy of the model. Therefore, it is reasonable to assume that supervised methods of background subtraction and semantic segmentation will out-scale multi-view segmentation algorithms as more task-specific data is acquired. Nevertheless, multi-view segmentation methods are

promising and should be researched further, particularly because of their direct link with shape-from-silhouette methods. This link also means, however, that multi-view segmentation algorithms cannot be directly compared to background subtraction and semantic segmentation algorithms: multi-view segmentation algorithms require multiple image of the same object to function, whereas background subtraction and semantic segmentation algorithms are always trained and tested on single images. A solution to this issue would be to establish a large multi-view dataset that allowed the direct comparison of algorithms that belong to all three categories. Such a dataset would need to contain multiple objects seen from multiple views (16+, as suggested by Mu¨ndermann et al. (22)) per object. Such a dataset could bridge the gap between background subtraction, semantic segmentation, and multi-view segmentation algorithms that does not allow their direct comparison. The problem would lie in the selection of the images to put into the dataset: too general-purpose, and supervised algorithms would not display their full potential on specific tasks (like human segmentation for markerless motion capture); too narrow, and it would limit the universal applicability of the algorithms and the interest of the researchers. A possible solution would be to establish a general-purpose dataset like CDnet2014 and PASCAL VOC 2012 on which to benchmark the algorithms, and similar but task-specific datasets to demonstrate the task-specific accuracy of the algorithms.

Along with the datasets, the metrics used to evaluate methods should also be standardised. Background subtraction algorithms often report results in terms of *F-measure* because it is an easy metric to compute and it is intuitive to understand what it represents. However, as pointed out by Lim and Keles (50), the *F-measure* is

susceptible to errors in the presence of imbalanced classes. Therefore, the *AP* metric used in PASCAL VOC 2012 (see Section 3.3) would likely constitute a better choice for a standard metric to use across categories of silhouette extraction algorithms. Additionally, the *MCC* metric used by (50) should be reported more often by authors, given its inherent resilience against imbalanced data.

Conclusion

The applicability of the visual hull algorithm to the task of markerless motion capture of human motion is hindered by the reliance of the visual hull on perfect 2D silhouettes of the object from each camera view. Traditionally, the capture volume is manipulated in such a way as to make the distinction of the foreground from the background trivial, thus removing the requirement for advanced silhouette extraction methods. However, the background cannot be easily manipulated in most open-world motion capture settings, and therefore highly accurate silhouette extraction methods are necessary in order to apply the visual hull to this task. This paper reviewed the literature on silhouette extraction methods in search of the algorithms most relevant to the application of the visual hull to biomechanics. Therefore, only the most accurate algorithms in their respective category were reported and discussed. What emerged from this review was that FgSegNet_v2 (a background subtraction algorithm), DeepLabv3+ JFT (a semantic segmentation algorithm), and Djelouah 2013 (a multi-view segmentation algorithm) are the most accurate and promising methods for the extraction of silhouettes from 2D images. Furthermore, Section 5 provided some preliminary guidelines for future works in this field. In particular, options for the establishment of a new dataset that enables

the direct comparison of methods from different categories were discussed, as well as recommendations as to which metrics of accuracy to use in the future.

References

[1] R. Bartlett, J. Wheat, and M. Robins, "Is movement variability important for sports biomechanists?" *Sports biomechanics*, vol. 6, no. 2, pp. 224–243, 2007.

[2] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[3] M. Windolf, N. G¨otzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the vicon-460 system," *Journal of biomechanics*, vol. 41, no. 12, pp. 2776–2780, 2008.

[4] L. Mu¨ndermann, S. Corazza, and T. P. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," *Journal of neuroengineering and rehabilitation*, vol. 3, no. 1, p. 6, 2006.

[5] A. Cappozzo, A. Cappello, U. D. Croce, and F. Pensalfini, "Surface-marker cluster design criteria for 3-d bone movement reconstruction," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 12, pp. 1165–1174, 1997.

[6] A. Cappozzo, F. Catani, A. Leardini, M. Benedetti, and U. Della Croce, "Position and orientation in space of bones during movement: experimental artefacts," *Clinical biomechanics*, vol. 11, no. 2, pp. 90–100, 1996.

[7] J. P. Holden, J. A. Orsini, K. L. Siegel, T. M. Kepple, L. H. Gerber, and S. J. Stanhope, "Surface movement errors in shank kinematics and knee kinetics during gait," *Gait & Posture*, vol. 5, no. 3, pp. 217–227, 1997.

[8] M. Sati, J. A. de Guise, S. Larouche, and G. Drouin, "Quantitative assessment of skinbone movement at the knee," *The Knee*, vol. 3, no. 3, pp. 121–138, 1996.

[9] C. Reinschmidt, A. Van Den Bogert, B. Nigg, A. Lundberg, and N. Murphy, "Effect of skin movement on the analysis of skeletal knee joint motion during running," *Journal of biomechanics*, vol. 30, no. 7, pp. 729–732, 1997.

[10] A. Fern'ndez-Baena, A. Sus´ın, and X. Lligadas, "Biomechanical validation of upperbody and lower-body joint movements of kinect motion capture data for rehabilitation treatments," in *Intelligent networking and collaborative systems (INCoS), 2012 4th international conference on*. IEEE, 2012, pp. 656–661.

[11] T. B. Moeslund, A. Hilton, and V. Kru¨ger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[12] E. Ceseracciu, Z. Sawacha, S. Fantozzi, M. Cortesi, G. Gatta, S. Corazza, and C. Cobelli, "Markerless analysis of front crawl swimming," *Journal of biomechanics*, vol. 44, no. 12, pp. 2236–2242, 2011.

[13] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 2, pp. 150–162, 1994.

[14] A. L. Sheets, G. D. Abrams, S. Corazza, M. R. Safran, and T. P. Andriacchi, "Kinematics differences between the flat, kick, and slice serves measured using a markerless motion capture method," *Annals of biomedical engineering*, vol. 39, no. 12, p. 3011, 2011.

[15] G. D. Abrams, A. H. Harris, T. P. Andriacchi, and M. R. Safran, "Biomechanical analysis of three tennis serve types using a markerless system," *Br J Sports Med*, vol. 48, no. 4, pp. 339–342, 2014.

[16] S. Corazza, L. Muendermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. P. Andriacchi, "A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach," *Annals of biomedical engineering*, vol. 34, no. 6, pp. 1019–1029, 2006.

[17] S. Corazza, L. Mu¨ndermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *International journal of computer vision*, vol. 87, no. 1-2, p. 156, 2010.

[18] M. Mikhnevich and D. Laurendeau, "Shape from silhouette in space, time and light domains," in *Computer Vision Theory and Applications (VISAPP), 2014 international conference on*, vol. 3. IEEE, 2014, pp. 368–377.

[19] K. Grauman, G. Shakhnarovich, and T. Darrell, "A bayesian approach to image-based visual hull reconstruction," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–I.

[20] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1746– 1753.

[21] S. Nobuhara, Y. Tsuda, I. Ohama, and T. Matsuyama, "Multi-viewpoint silhouette extraction with 3d context-aware error detection, correction, and shadow suppression," *IPSJ Transactions on Computer Vision and Applications*, vol. 1, pp. 242–259, 2009.

[22] L. Mundermann, S. Corazza, A. M. Chaudhari, E. J. Alexander, and T. P. Andriacchi, "Most favorable camera configuration for a shape-from-silhouette markerless motion capture system for biomechanical analysis," in *Videometrics VIII*, vol. 5665. International Society for Optics and Photonics, 2005, p. 56650T.

[23] S. Nobuhara and T. Matsuyama, "Deformable mesh model for complex multi-object 3d motion estimation from multi-viewpoint video," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*. IEEE, 2006, pp. 264–271.

[24] D. Vlasic, I. Baran, W. Matusik, and J. Popovi´c, "Articulated mesh animation from multi-view silhouettes," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 97.

[25] Y. Furukawa and J. Ponce, "Carved visual hulls for image-based modeling," in *European Conference on Computer Vision*. Springer, 2006, pp. 564–577.

[26] S. Lazebnik, Y. Furukawa, and J. Ponce, "Projective visual hulls," *International Journal of Computer Vision*, vol. 74, no. 2, pp. 137–165, 2007.

[27] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, P. Ishwar *et al.*, "Changedetection. net: A new change detection benchmark dataset." in *CVPR Workshops*, no. 2012, 2012, pp. 1–8.

[28] M. Bakkay, H. Rashwan, H. Salmane, L. Khoudour, D. Puigtt, and Y. Ruichek, "Bscgan: Deep background subtraction with conditional generative adversarial networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4018–4022.

[29] M. Camplani, L. Maddalena, G. M. Alcover, A. Petrosino, and L. Salgado, "A benchmarking framework for background subtraction in rgbd videos," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 219–229.

[30] S.-C. S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 14, p. 726261, 2005.

[31] Y. Tian, A. Senior, and M. Lu, "Robust and efficient foreground analysis in complex surveillance videos," *Machine vision and applications*, vol. 23, no. 5, pp. 967–983, 2012.

[32] A. W. Senior, Y. Tian, and M. Lu, "Interactive motion analysis for video surveillance and long term scene monitoring," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 164–174.

[33] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. GarciaRodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[34] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

[35] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11, pp. 31–66, 2014.

[36] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent patents on computer science*, vol. 1, no. 1, pp. 32–54, 2008.

[37] L. Unzueta, M. Nieto, A. Cort´es, J. Barandiaran, O. Otaegui, and P. S´anchez, "Adaptive multicue background subtraction for robust vehicle counting and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 527–540, 2012.

[38] P. Blauensteiner and M. Kampel, *Visual surveillance of an airport's apron-An overview of the AVITRACK project*. na, 2004.

[39] A. Leykin and M. Tuceryan, "Detecting shopper groups in video sequences," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 2007, pp. 417–422.

[40] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 255–261.

[41] D. E. Butler, V. M. Bove, and S. Sridharan, "Real-time adaptive foreground/background segmentation," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 14, p. 841926, 2005.

[42] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–19, 2017.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[44] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 20, 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008.

[47] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, 2017.

[48] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[49] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[50] L. A. Lim and H. Y. Keles, "Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding," *arXiv preprint arXiv:1801.02225*, 2018.

[51] C. Wojek and B. Schiele, “A dynamic conditional random field model for joint labeling of object and scene classes,” in *European Conference on Computer Vision*. Springer, 2008, pp. 733–747.

[52] P. Kr¨ahenbu¨hl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in neural information processing systems*, 2011, pp. 109– 117.

[53] Z. Luo, P.-M. Jodoin, S.-Z. Li, and S.-Z. Su, “Traffic analysis without motion features,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3290–3294.

[54] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Twelfth annual conference of the international speech communication association*, 2011.

[55] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8609–8613.

[56] D. Cire¸san, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” *arXiv preprint arXiv:1202.2745*, 2012.

[57] A. Newswanger and C. Xu, “One-shot video object segmentation with iterative online fine-tuning,” in *CVPR Workshop*, vol. 1, 2017.

[58] M. Mirza and S. Osindero, “Conditional generative adversarial networks,” *Manuscript: https://arxiv. org/abs/1709.02023*, 2014.

[59] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[60] L. A. Lim and H. Y. Keles, “Foreground segmentation using convolutional neural networks for multiscale feature encoding,” *Pattern Recognition Letters*, vol. 112, pp. 256– 262, 2018.

[61] ——, “Learning multi-scale features for foreground segmentation,” *arXiv preprint arXiv:1808.01477*, 2018.

[62] H. Olkkonen and P. Pesola, “Gaussian pyramid wavelet transform for multiresolution analysis of images,” *Graphical Models and Image Processing*, vol. 58, no. 4, pp. 394–398, 1996.

[63] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[64] N. V. Chawla, N. Japkowicz, and A. Kotcz, “Special issue on learning from imbalanced data sets,” *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[65] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequi`evre, “A benchmark dataset for outdoor foreground/background extraction,” in *Asian Conference on Computer Vision*. Springer, 2012, pp. 291–300.

[66] J. C. Crane, M. P. Olson, and S. J. Nelson, “Sivic: open-source, standards-based software for dicom mr spectroscopy workflows,” *Journal of Biomedical Imaging*, vol. 2013, p. 12, 2013.

[67] P.-M. Jodoin, L. Maddalena, A. Petrosino, and Y. Wang, “Extensive benchmark and survey of modeling methods for scene background initialization,” *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5244–5256, 2017.

[68] V. Mahadevan and N. Vasconcelos, “Spatiotemporal saliency in dynamic scenes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 171–177, 2010.

[69] L. Vosters, C. Shan, and T. Gritti, “Real-time robust background subtraction under rapidly changing illumination conditions,” *Image and Vision Computing*, vol. 30, no. 12, pp. 1004–1015, 2012.

[70] S. Boughorbel, F. Jarray, and M. El-Anbari, “Optimal classifier for imbalanced data using matthews correlation coefficient metric,” *PloS one*, vol. 12, no. 6, p. e0177678, 2017.

[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment:

from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[72] C. Lallier, E. Reynaud, L. Robinault, and L. Tougne, "A testing framework for background subtraction algorithms comparison in intrusion detection context," in *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2011, pp. 314–319.

[73] M. Thoma, "A survey of semantic segmentation," *arXiv preprint arXiv:1602.06541*, 2016.

[74] H. Zhu, F. Meng, J. Cai, and S. Lu, "Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 12–27, 2016.

[75] X. Hou, A. Yuille, and C. Koch, "Boundary detection benchmarking: Beyond fmeasures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2123–2130.

[76] S. Chilamkurthy. (2017) A 2017 guide to semantic segmentation with deep learning. [Online]. Available: http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review

[77] S. Maldonado-Basc´on, S. Lafuente-Arroyo, P. Gil-Jimenez, H. G´omez-Moreno, and F. L´opez-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 264– 278, 2007.

[78] N. Behzadfar and H. Soltanian-Zadeh, "Automatic segmentation of brain tumors in magnetic resonance images," in *Biomedical and Health Informatics (BHI), 2012 IEEEEMBS International Conference on*. IEEE, 2012, pp. 329–332.

[79] N. Moon, E. Bullitt, K. Van Leemput, and G. Gerig, "Automatic brain and tumor segmentation," in *International Conference on Medical Image Computing and ComputerAssisted Intervention*. Springer, 2002, pp. 372–379.

[80] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS journal of photogrammetry and remote sensing*, vol. 65, no. 1, pp. 2–16, 2010.

[81] H.-D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[82] Y. Raja, S. J. McKenna, and S. Gong, "Segmentation and tracking using colour mixture models," in *Asian Conference on Computer Vision*. Springer, 1998, pp. 607–614.

[83] B. Kim, J. Son, and K. Sohn, "Illumination invariant road detection based on learning method," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1009–1014.

[84] B. Li and M. Q.-H. Meng, "Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments," *Computers in biology and medicine*, vol. 39, no. 2, pp. 141–147, 2009.

[85] C. Rotaru, T. Graf, and J. Zhang, "Color image segmentation in hsi space for automotive applications," *Journal of Real-Time Image Processing*, vol. 3, no. 4, pp. 311–322, 2008.

[86] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1365–1372.

[87] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *European conference on computer vision*. Springer, 2010, pp. 168–181.

[88] T. Brox, L. Bourdev, S. Maji, and J. Malik, "Object segmentation by alignment of poselet activations to image contours," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2225–2232.

[89] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[90] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[91] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[92] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[93] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[94] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 843–852.

[95] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611*, 2018.

[96] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, pp. 1610–02357, 2017.

[97] H. Qi, Z. Zhang, B. Xiao, H. Hu, B. Cheng, Y. Wei, and J. Dai, "Deformable convolutional networks–coco detection and segmentation challenge 2017 entry," in *ICCV COCO Challenge Workshop*, vol. 15, 2017.

[98] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[99] P. Doll´ar and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[100] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." in *Cvpr*, vol. 1, no. 2, 2017, p. 5.

[101] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[102] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 519–534.

[103] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—improve semantic segmentation by global convolutional network," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1743–1751.

[104] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[105] Z. Zhang, X. Zhang, C. Peng, D. Cheng, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," *arXiv preprint arXiv:1804.03821*, 2018.

[106] S. Xie, R. Girshick, P. Doll´ar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5987–5995.

[107] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[108] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[109] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

[110] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[111] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *arXiv preprint arXiv:1707.07012*, vol. 2, no. 6, 2017.

[112] L.-C. Chen, M. D. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens, "Searching for efficient multi-scale architectures for dense image prediction," *arXiv preprint arXiv:1809.04184*, 2018.

[113] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley, "Google vizier: A service for black-box optimization," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1487–1495.

[114] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[115] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[116] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[117] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[118] T. Arlen. (2018) Understanding the map evaluation metric for object detection. [Online]. Available: https://medium.com/@timothycarlen/ understanding-the-map-evaluation-metric-for-object-detection-a07fe6962cf3

[119] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. P´erez, "Sparse multi-view consistency for object segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1890–1903, 2015.

[120] C. Reinbacher, M. Ru¨ther, and H. Bischof, "Fast variational multi-view segmentation through backprojection of spatial constraints," *Image and Vision Computing*, vol. 30, no. 11, pp. 797–807, 2012.

[121] N. D. Campbell, G. Vogiatzis, C. Hern´andez, and R. Cipolla, "Automatic 3d object segmentation in multiple views using volumetric graph-cuts," *Image and Vision Computing*, vol. 28, no. 1, pp. 14–25, 2010.

[122] T. Feldmann, L. Dießelberg, and A. W¨orner, "Adaptive foreground/background segmentation using multiview silhouette fusion," in *Joint Pattern Recognition Symposium*. Springer, 2009, pp. 522–531.

[123] A. Djelouah, J.-S. Franco, E. Boyer, P. P´erez, and G. Drettakis, "Cotemporal multiview video segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 360–369.

[124] A. Kowdle, D. Batra, W.-C. Chen, and T. Chen, "imodel: interactive co-segmentation for object of interest 3d modeling," in *European Conference on Computer Vision*. Springer, 2010, pp. 211–224.

[125] M. Sormann, C. Zach, and K. Karner, "Graph cut based multiple view segmentation for 3d reconstruction," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*. IEEE, 2006, pp. 1085–1092.

[126] J. Xiao, J. Wang, P. Tan, and L. Quan, "Joint affinity propagation for multiple view segmentation," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–7.

[127] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[128] J. Gallego, J. Salvador, J. R. Casas, and M. Pardas, "Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 997–1000.

[129] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. P´erez, "N-tuple color segmentation for multi-view silhouette extraction," in *European Conference on Computer Vision*. Springer, 2012, pp. 818–831.

[130] K. Kolev, T. Brox, and D. Cremers, "Fast joint estimation of silhouettes and dense 3d geometry from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 493–505, 2012.

[131] V.-Q. Pham, K. Takahashi, and T. Naemura, "Foreground-background segmentation using iterated distribution matching," in *CVPR 2011*. IEEE, 2011, pp. 2113–2120.

[132] A. Kowdle, S. N. Sinha, and R. Szeliski, "Multiple view object cosegmentation using appearance and stereo cues," in *European Conference on Computer Vision*. Springer, 2012, pp. 789–803.

[133] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[134] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. P´erez, "Multi-view object segmentation in space and time," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2640–2647.

[135] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Su¨sstrunk *et al.*, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[136] N. D. Campbell, G. Vogiatzis, C. Hern´andez, and R. Cipolla, "Automatic object segmentation from calibrated images," in *Visual Media Production (CVMP), 2011 Conference for*. IEEE, 2011, pp. 126–137.

[137] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 3, pp. 418–433, 2005.

[138] S.-H. Kim, Y.-W. Tai, J. Park, and I. S. Kweon, "Multi-view object extraction with fractional boundaries," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3639– 3654, 2016.

[139] W. Lee, W. Woo, and E. Boyer, "Silhouette segmentation in multiple views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1429– 1441, 2011.