International Journal of Multimedia Information Retrieval
https://doi.org/10.1007/s13735-019-00183-w

TRENDS AND SURVEYS

A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision

Theodoros Georgiou[1] · Yu Liu[1] · Wei Chen[1] · Michael Lew[1]

Received: 10 July 2019 / Revised: 8 October 2019 / Accepted: 30 October 2019
© The Author(s) 2019

Abstract
Higher dimensional data such as video and 3D are the leading edge of multimedia retrieval and computer vision research. In this survey, we give a comprehensive overview and key insights into the state of the art of higher dimensional features from deep learning and also traditional approaches. Current approaches are frequently using 3D information from the sensor or are using 3D in modeling and understanding the 3D world. With the growth of prevalent application areas such as 3D games, self-driving automobiles, health monitoring and sports activity training, a wide variety of new sensors have allowed researchers to develop feature description models beyond 2D. Although higher dimensional data enhance the performance of methods on numerous tasks, they can also introduce new challenges and problems. The higher dimensionality of the data often leads to more complicated structures which present additional problems in both extracting meaningful content and in adapting it for current machine learning algorithms. Due to the major importance of the evaluation process, we also present an overview of the current datasets and benchmarks. Moreover, based on more than 330 papers from this study, we present the major challenges and future directions.

Keywords High dimensional · Computer vision · Feature descriptors · Deep learning

1 Introduction

With the current growth of computing systems and technologies, three- and four dimensional data, such as 3D images and videos, are becoming a commodity in multimedia systems. Understanding and utilizing these data are the leading edge of modern computer vision. In this paper, we present a comprehensive study (including a categorization) of these high dimensional data types, as well as the methods developed to process them, accompanied with their strengths and weaknesses. Finally, we collect and give an overview of the main areas that utilize such representations.

One of the first steps toward developing, testing and applying methods on high dimensional data is the acquisition of complicated datasets, for instance datasets consisting of 3D models [35,314], three dimensional medical images and videos (MRI, Ultrasound, etc.) [43,111], large 2D and 3D video datasets for action recognition [175,242] and more. Different datasets are used for different data mining tasks. For example, object retrieval, movie retrieval and action classification tasks are performed on video data such as movies and YouTube clips. Clustering and classification tasks are performed on medical images for computer-aided diagnostics and surgery. Object classification and detection, as well as scene semantic segmentation, are usually applied on RGB-D images and videos retrieved by sensors such as the Microsoft Kinect [332].

We perform two types of categorization. The first is dataset and application driven, and the second is method driven. Although these datasets find applications in different fields, there are some similarities between the methods used. For example, deep learning techniques are used for 2.5D and 3D object classification (either retrieved from depth maps or designed models), action classification, video retrieval, as well as medical applications, for instance landmark detection and tracing in ultrasound video. Histograms of different metrics (e.g., gradients, optical flow or surface normals) are used as features that describe the content of the data.

One of the recent breakthroughs has been the development of new deep learning architectures which could

✉ Theodoros Georgiou
t.k.georgiou@liacs.leidenuniv.nl

[1] Leiden Institute of Advanced Computer Science, Universiteit Leiden, Niels Bohrweg 1, Leiden, The Netherlands

Published online: 22 November 2019

overcome (to some extent) the well-known vanishing gradient problem in training. In the case of neural networks, they changed the landscape from typically using a few layers to using hundreds of layers. These methods typically learn the features based on large datasets directly from the raw data and have the least supervision. The other main approach from the literature is the continuation of advances in traditional or "handcrafted"- and "shallow learning"-based features. 2D features in computer vision have had a major impact in computer vision and human–computer interaction across many applications [3,12,168,183,224,240,279,334], and many of the higher dimensional methods were inspired or adapted from the 2D versions. These approaches usually require significantly more supervision but also can be effective when large training datasets are not accessible.

High dimensional computer vision, with the definition given in this paper (i.e., higher than 2D), is a very broad field that contains many different research areas, data types and methods. There have been surveys on specific areas within high dimensional computer vision. For example, when it comes to the static world, some of surveys focus on specific research areas such as 3D object detection [84,229], semantic segmentation [74,85,324], object retrieval [58,273] or human action recognition [101,128,203]. Others focus on methodologies such as interest point detectors and descriptors [27,149,283], spatiotemporal salient point detectors and descriptors [157] or deep learning [117]. Finally, some surveys focus on datasets and benchmarks of a specific research area, such as human action recognition [96]. We differ from these since we focus on the generalization of methodologies with the increase in dimensionality, regardless of the research area or the type of data. The most relevant work to ours was done by Ioannidou et al. [117] where they focus on computer vision on static 3D data. There are two main differences with our work: (1) they focus only on deep learning methods and (2) they focus only on 3D representation of the static world which means that they neglect the temporal dimension, which is a significant part of this survey.

The rest of the paper is organized as follows. Section 2 gives an analysis of existing deep learning methods and categorizes their extensions to higher dimensional data. Section 3 gives an overview and a categorization of existing handcrafted features for several different data types. In Sect. 4, we describe existing large- scale datasets and benchmarks that contain high dimensional data. Section 5 gives an overview of the most researched areas that make use of higher dimensional data. In Sect. 6, we identify the difficulties and challenges that researchers face as well as the limitations of current state-of-the-art methods. Finally, in Sect. 7 we draw our conclusions.

2 Deep learning

Deep learning techniques refer to a cluster of machine learning methods that construct a multilayered representation of the input data. The transformation of the data in each layer is typically trained through algorithms similar to backpropagation. There are several deep learning methods. In this section, we will give a summary of the methods that have been used with high dimensional data. The main examples are the convolutional neural networks (CNNs), the recurrent neural networks (RNNs), auto-encoders (AE) and restricted Boltzmann machines (RBMs). For a detailed overview of deep learning in computer vision, the reader is referred to [86] and for a general deep learning overview to [79].

Deep learning approaches can be split into two main categories, supervised and unsupervised methods. Supervised methods define an error function which depends on the task the method needs to solve and change the model parameters according to that error function. These kind of methods provide an end-to-end learning scheme, meaning that the model is learning to perform the task from the raw data. Unsupervised methods usually define an error function to be minimized which depends on the reconstruction ability of the model. Together with the reconstruction error, depending on the method, an auxiliary error function might be defined which forces some characteristics to the learned representation. For example, sparse auto-encoders try to force the learned representation to be sparse, which helps the overall learning procedure and provides a more discriminative representation. The most commonly used deep learning method is CNNs. In the rest of this section, we give a small introduction to the basic deep learning methods and provide an in depth analysis on their generalization from the image domain to the higher dimensional problems.

2.1 Basic deep learning methods

2.1.1 Convolutional neural networks (CNN)

Convolutional neural networks consist of multiple layers of convolutions, pooling layers and activation functions. Usually, each layer will have a number of different convolutional kernels, a nonlinear activation function and, maybe, a pooling mechanism to lower the dimensionality of the output data. An example of such a layer is shown in Fig. 1. These networks were initially applied on handwritten digit recognition [151] but got the attention they have today after the introduction of LeNet [152] and more so after Krizhevsky et al.'s [140] work in 2012, where they won the ImageNet 2012 image classification competition with a deep-CNN. This recent success of the CNNs highly depends on the increased processing power of modern GPUs as well as the availability of large-scale and

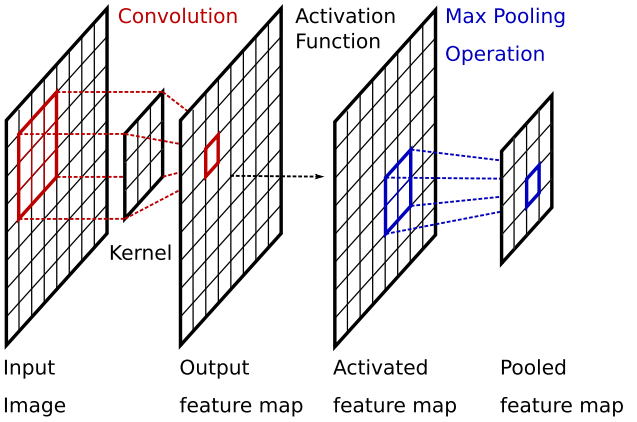

Fig. 1 Basic CNN block. A single layer is shown which applies a kernel on an input filter followed by an activation function and a max pooling operation

diverse datasets which made training models with millions of trainable parameters possible.

One of the main drawbacks of deep convolutional neural networks is that they tend to overfit the data. Moreover, they suffer from vanishing and exploding gradients. Resolving these issues has motivated a lot of research in various directions. More specifically, different elements of CNNs are studied and proposed, e.g., activation functions or normalization layers, training strategies and the generic network architecture, for example the inception networks [270]. Most of this research is based on image recognition as the established benchmark due to the availability of large-scale annotated datasets such as the ImageNet [225] and the Microsoft COCO [163]. Nonetheless, many of these methods have been generalized and adapted to be applicable to 2.5D and 3D data, such as videos, and RGB-D images.

Activation functions One of the main components of the successful AlexNet [140] on the ImageNet 2012 challenge is the rectified linear unit [120,188] activation function. The output of the function is max $(0, y)$, where y is the output of a node in the network. The main advantages of this layer are the sparsity it provides to the output as well as minimization of the vanishing gradients problem, compared to the more traditional hyperbolic tan and the sigmoid functions [78].

In the past years, many researchers have proposed new activation functions in order to improve the quality of neural networks. Some examples are the leaky ReLU (LReLU) [172], which instead of having always zero as output of negative inputs, it has a small response proportional to the input, i.e., $\alpha * y$. The parametric rectified linear unit (PReLU) [98], which learns the parameter α of LReLU. The exponential linear unit (ELU) [42] and its trainable counterpart parametric ELU (PELU) [286], and many more [2,80,124,134]. For a more detailed overview of activation functions, the reader is referred to [286].

Normalization The experimental results suggest that when networks have normalized inputs, with zero mean and standard deviation of one, they tend to converge much faster [140]. In order to take advantage of this finding, it is a common practice to rescale and normalize the input images [114, 140,254]. Besides the input normalization, many researchers try to also normalize the input of individual layers, in order to alleviate the covariate shift affect [248]. The traditional method of activation normalization is the local response normalization [120,140]. The most established work though is the later batch normalization technique [118]. In this work the output of each layer is rescaled and centered according to the batch-statistics of activations. The success of this method gave rise to more research in this direction like [8,115,233,287,297,313]. For a detailed overview and comparison of these methods, the reader is referred to [214,297,313].

Network structure In an attempt to increase their performance, a large group of works have also explored different architectures of the internal structure of CNNs. After the work of Krizhevsky et al. [140], researchers tried to understand how different parameters effected the quality of the networks. Here we will give a small overview of the main milestone works since then.

One of the first important works was the one of Simonyan and Zisserman [254] who proposed the VGG nets. In their work, they showed that with small convolutional kernels (3×3), deeper networks were able to be trained. They introduced an 11, 13, 16 and 19 weighted layered networks. One main constraint on the possible depth of neural networks is the vanishing gradients problem. In an attempt to alleviate this issue, HighWay networks [262] and residual networks (ResNet) [99] make use of "skip" or "shortcut" connections in order to pass information from one layer to one or several layers ahead (Fig. 2). Huang et al. [114] generalized this idea even further, with their DenseNet, by giving as input to the lth layer all previous l–1 layers. The building blocks of ResNet, Res Block, and DensNet, Dense Block, are shown in Fig. 2.

Besides skip connections, which helped deeper networks to be trained, different methods to increase the quality of networks have also been studied. Lin et al. [162] proposed the network in network (NiN) architecture. In their work, they substituted the linear convolutional nodes with small multilayer perceptron (MLP), giving to the network the ability to learn nonlinear mappings in a layer. Lee et al. [153] proposed the deeply supervised nets (DSN) which use secondary supervision signals directly to hidden layers of the network. Liu et al. [165] explore a different approach, where the final

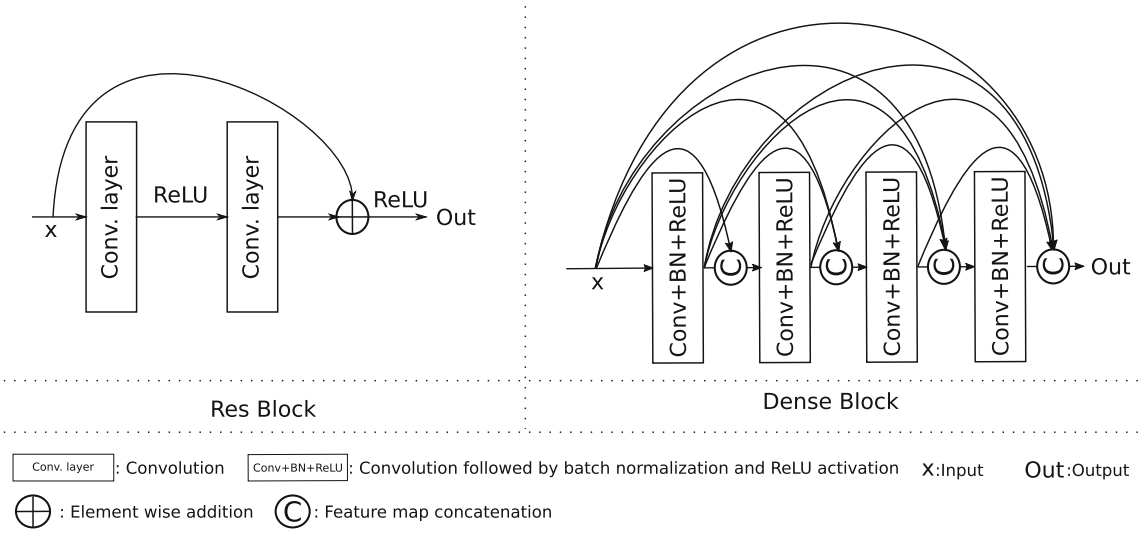

Fig. 2 On the left is the ResBlock, the building block of ResNet [99]. After two convolution operations, the input is added to the output in order to produce the residual learning function $H(x) = F(x) + x$. On the right is the building block of Dense Net [114]. The layer l gets as an input the output of all layers $[l-4, l-1]$

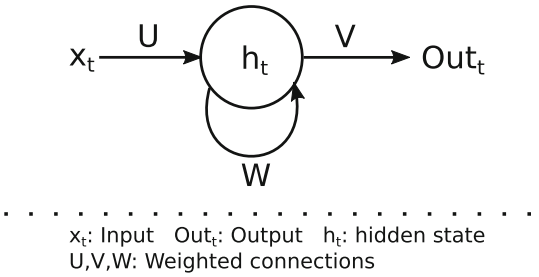

Fig. 3 Basic module of an RNN processing time step t

decision, either classification or any other task, is made not only by the information in the last layer but also from deeper layers. They do so with their convolutional fusion network (CFN), in which locally connected (LC) layers are used to fuse lower-level information from deeper layers with the high-level information of the top layer and make a more informative decision.

2.1.2 Recurrent neural networks (RNN)

Recurrent neural networks are a special class of artificial neural networks. A basic RNN module is composed by a feed forward node computing a "hidden state", a recurrent connection, which connects the hidden unit to the next time step input, and an output unit, as seen in Fig. 3. This recurrent connection gives the network the ability to make predictions not only according to the current input but also historic inputs that comprise a sequence of data.

Although this architecture was successful, in problems with a large number of time steps it could no longer maintain high performance. That happens due to the vanishing gradient problem in back- propagation through time (BPTT), a main stream training procedure of RNN. In order to counter this limitation a new architecture, the long short-term memory node (LSTM) was proposed by Hochreiter and Schmidhuber [109]. It contains several gates that control the flow of information and allow the network to store long-term information, if needed. Such an architecture has been used for many tasks that deal with sequential data, such as language modeling [330] and translation [171], action classification in videos [54], speech synthesis [62] and more.

Inspired from the success of the LSTM method, researchers proposed many variations. Some are generic and can be applied to any problem that simple LSTM is applied while others are application specific.

To the best of our knowledge, the first generic extension of LSTM was proposed in the work of Gers et al. [77]. They noticed that none of the gates have direct connections to the memory cell they are supposed to control. In order to alleviate that limitation, they proposed "peephole" connections from the memory cell to the input of each gate. Cho et al. [40] proposed an extension, the Gated Recurrent Unit (GRU) that simplified the architecture and reduced the number of trainable parameters by combining the forget and input gates. Laurent et al. [150] and Cooijmans et al. [44] proposed batch normalized LSTM. Although [150] batch normalized only the input of the node, Cooijmans et al. [44] did so also in the hidden unit. Zhao et al. [333] proposed a combination of several of the above extensions. Specifically, they proposed a bidirectional [238] GRU unit, combined with batch normal-

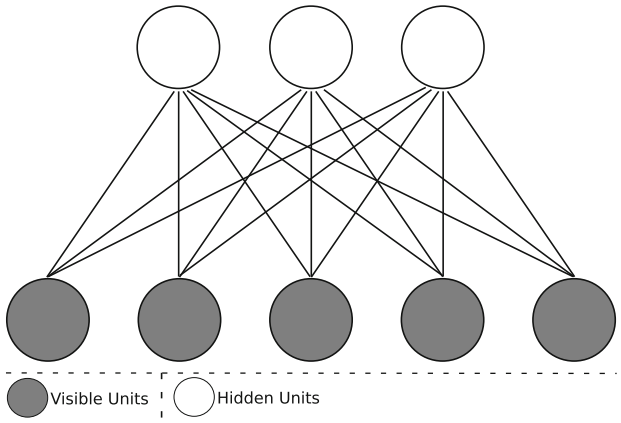

Fig. 4 RBM architecture. Notice that the connections are undirected

ization. For a more thorough review regarding LSTM and its variants, the reader is referred to [82].

As mentioned above, some extensions of the LSTM are application specific. For example, Shahroudy et al. [242] proposed the Part-Aware LSTM (PA-LSTM), an architecture tailored for skeleton-based data. Instead of having one memory cell for the whole skeleton, as is a common approach, they introduced one memory cell per joint of the skeleton, each with its own input, forget and output gates. Liu et al. [164] proposed the spatiotemporal LSTM unit with trust gates (ST-LSTM) for 3D human action recognition. This unit extends the recurrent learning with memory to the spatial domain as well.

2.1.3 Restricted Boltzmann machine (RBM)

The restricted Boltzmann machine (RBM) was first introduced by Hinton [108]. It is a two-layer, undirected, bipartite and undirected model (Fig. 4). It comprises of a set of visible units, which are either binary or real valued, and a set of binary hidden nodes. A configuration with visible vector $\mathbf{v}$ and hidden vector $\mathbf{h}$ is assigned with energy given by:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i\in\text{visible}} \alpha_i v_i \quad -\sum_{j\in\text{hidden}} b_j h_j \quad -\sum_{ij} v_i h_i w_{ij}, \tag{1}$$

where α_i, b_j, w_{ij} are the network parameters. Given this energy the network assigns to every pair $\mathbf{v}$, $\mathbf{h}$ a probability:

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{2}$$

where Z is the partition function and is given by summing overall possible pairs of visible and hidden vectors. Since there are no direct connections between the hidden or visible units, we can easily obtain an unbiased pair ($\mathbf{v}$, $\mathbf{h}$). Given the visible vector $\mathbf{v}$, the hidden unit h_j is assigned to one with probability:

$$P(h_j = 1|\mathbf{v}) = \sigma\left(b_j + \sum_i v_i w_{ij}\right), \tag{3}$$

where $\sigma(\cdot)$ is the logistic sigmoid function. Similarly, given a hidden vector $\mathbf{h}$ the probability of a visible unit v_i to be assigned to one is given by:

$$P(v_i = 1|\mathbf{h}) = \sigma\left(\alpha_i + \sum_j h_j w_{ij}\right), \tag{4}$$

Starting from the training data, the network parameters are tuned in order to maximize the likelihood of the visible and hidden vectors pair $\{\mathbf{v}, \mathbf{h}\}$.

RBMs are only two-layer deep models and thus are restricted in the complexity of the data they can represent. In order to alleviate this issue, a number of deeper models built on RBMs are designed. The most known models derived from RBMs are the deep belief networks (DBN) [106], deep Boltzmann machines (DBM) [232] and the deep energy models (DEM) [191]. They are all multilayer probabilistic models that perform nonlinear transformation to the data.

DBNs are trained in a greedy layer-wise manner, where each layer is trained as an RBM. The final model keeps only the top-down connections of the layers except the top two that remain undirected. Unlike DBNs, DBMs have undirected weights in all layers. Initially the weights are also trained in a greedy fashion, like a DBN. Since it is very computationally expensive to estimate and maximize the likelihood directly, Salakhutdinov and Larochelle [232] proposed an approximative algorithm which maximizes the lower bound of the log-likelihood [230,231]. Finally, DEM, the most recent deep model based on RBMs is a fully connected feedforward network with an RBM on top [191]. The non-stochastic nature of the hidden layers renders it possible to have an efficient training of the whole model simultaneously. For a more comprehensive review of these models, the reader is referred to [86].

2.1.4 Auto-encoders (AE)

Auto-encoders are a collection of neural network methods based on unsupervised learning. They were first introduced by Bourlard and Kamp [23] in 1988, as auto-association networks. The main idea is to reduce the dimensionality of the data with a fully connected layer and then try to recover the input from the reduced representation. In the case where the network is able to reconstruct the input, the intermediate low-dimensional representation should contain most of the information of the original data (Fig. 5). Since a single-layer network is able to perform only linear transformations, it is not sufficient for performing high dimensionality reduction in complicated data. Thus, Hinton and Salakhutdinov [107]

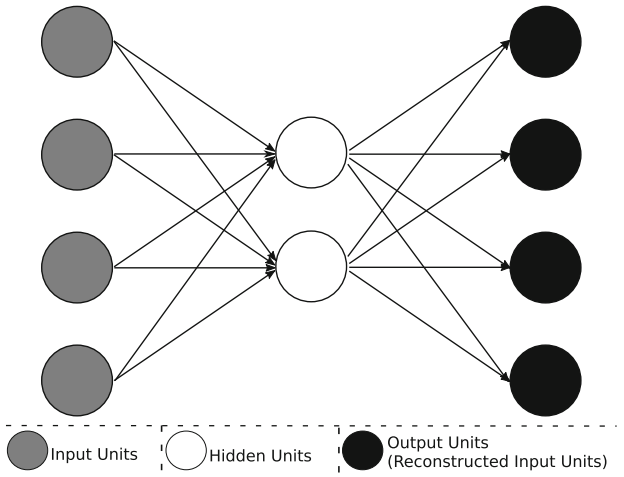

Fig. 5 Auto-association network. Notice that the output units are reconstructed input units

proposed a multiple layer version, called Auto-encoder (AE). It utilizes several layers to transform or "encode" the data. In some cases, if there is large error in the first layers, these models only learn the average of the training data. In order to alleviate this issue, [107] proposed to pre-train the network so the initial parameters are already close to a good solution. Since then, many variants of AEs have been proposed.

One of the first variations in AEs is the sparse auto-encoder. The basic idea behind it is to transform the data on an over-complete representation of higher dimensionality than the original. The benefits of such a transformation is that (1) there is a high probability that in the new representation the data will be linearly separable and (2) it can provide a simple interpretation of the input data in terms of a small number of "parts" by extracting the structure hidden in the data [204].

Vincent et al. [291,292] suggested that a good transformation should provide similar representation for two similar data points. In an effort to force the model to be more robust in small variations in the data, they proposed the Denoising AE (DAE), which tried to reconstruct the original data given slightly modified data as input. Rifai et al. [218] proposed a different method to achieve robustness to small input variations, the Contractive AE. They do so by penalizing the sensitivity of encoded representation with respect to the input data point.

Masci et al. [176] inspired by the success of CNNs, proposed a combination of AE with CNNs the Convolutional AE (CAE) and applied on image datasets, MNITST and CIFAR10. The architecture comprises of several stacked convolutional layers. The model is used as a pre-train mechanism for a CNN which is then trained in a supervised manner for object classification.

2.2 Deep learning for high dimensional data

In this section, we describe the main deep learning approaches applied on high dimensional data and provide a categorization of them. Specifically, we cluster the methods according to the type of generalization performed.

Most of the deep learning methods applied on higher than two dimensional data are generalized from lower dimensional counterparts, e.g., CNNs, CAEs, etc. The methods can be divided into two categories, namely increase in physical dimensions and increase in modalities. There are also several models that are developed for high dimensional data and were not generalized from lower dimensions, such as the PointNet [206]. It is important to note that all of the deep learning methods developed for 2D (images) and the generalization to 3D as well are either CNNs or a variation in them, like CAE.

2.2.1 Increase in physical dimensions

In this section, we describe the methods that were based on generalizing an existing approach to higher dimensions. Although this seems straightforward, due to the curse of dimensionality, as well as the large demand of memory and computational power of deep learning approaches, the extension from two to three dimensional data is not trivial. When considering the static world, i.e., time is not involved in some way, two main concepts exist. The straight forward extension to three dimensional kernels and the projection of data to fewer dimensions coupled with the use of an assembly of lower dimensional models, usually pre-trained on a large dataset, like the ImageNet 2012 [225].

The first approach to extend the 2D convolutional deep learning techniques to the 3D case is the work of Chang et al. [35] on ShapeNets. They implemented a convolutional DBN with three dimensional kernels with which they learned a 3D shape representation from CAD models. The three dimensional convolutional kernels (and pooling) have also been combined with other models, such as the feed forward CNNs [241], CAEs [26] and GANs [312]. Moreover, they have been utilized in many fields such as 3D medical Images [53], computational fluid dynamics (CFD) Simulations [76], 3D objects [179] and Videos [121]. The main drawback of these approaches is the high computational and memory demand of the resulted models, which limit both their size and the input resolution they can support. Although this is the case they are able to exploit relationships in all three dimensions, unlike the 2D methods.

The second cluster is the reduction in the data dimensionality to two, in order to be able to construct complicated models as well as take advantage of pre-trained ones. The reduction from three to two dimensions depends on the type of data in question. For example, when CAD models or 3D

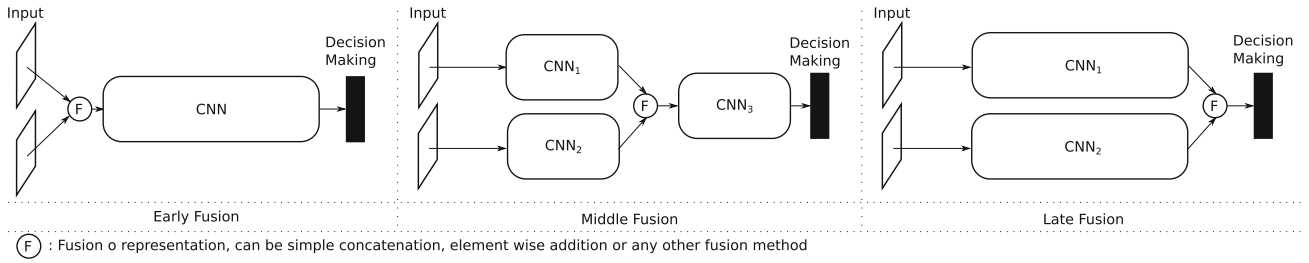

Fig. 6 Three naive approaches of fusing information from different modalities. Left shows the early fusion, which fuses before any processing. Middle shows mid-fusion and on the right is the late fusion approach

objects are concerned, the projection to two dimensions is done from an outside perspective, i.e., "taking photos" of the object from different angles [266]. Shi et al. [245] proposed an alternative representation of the 3D models. Specifically, they proposed a projection of the 3D shape on a cylinder around the object. The height of the cylinder is equal to the height of the object, making their representation invariant to scaling. Three dimensional medical images contain information in all three dimensional space, and the outside perspective misses all information relevant to most applications. In that case, the data are not projected but rather processed in a slice-by-slice manner [53]. In the case of videos, three strategies for lowering the dimensionality have been proposed. In the first one, each frame can be considered separately [54,285]. The second considers frames as extra channels [65,129,253,305]. This is usually done when passing to the networks the optical flow for several frames. Another approach is to try and compress the information of several frames into one. The work of Bilen et al. [16] is in that direction. They propose the Dynamic Image. More specifically, they adapt the method of Fernando et al. [66] that combines features from multiple frames to the pixel level. The result is an image which contains movement information, similar to a blurred one.

Due to the lower dimensionality of the transformed input data, it is possible to construct very complicated and large models. Moreover, a common approach is to use and fine-tune pre-trained models on very large and diverse datasets such as the ImageNet 2012 [225]. Although this is the case, as mentioned in the previous section, these methods lose the ability to explore the correlations in the data in all available dimensions.

2.2.2 Increase in modalities

The second type of generalization refers to the increase in the available modalities of the data. To be more precise, although the physical dimensions of the data remain the same, for example from 2D image to 2D image or 2D+time to 2D+time, the information given per point increases. Some examples are the RGB-D data, optical flow added to the videos and more. Depending on nature of the extra information, the resulted representation might result in a partial space-dimensionality increase. For example, the RGB-D data do not increase the dimensions to three. Nonetheless, the extra information is the distance to the sensor, which provides some information about the extra third physical dimension.

When dealing with this type of dimensionality increase, researches proposed various strategies to incorporate the extra information.

The most simple and naive approach is to consider the extra information as an extra channel and process with the same data dimensionality as before. This is very common when dealing with RGB-D data [46,294].

In the second category belong approaches that process the different types of information separately and fuse the extracted features by concatenating the feature maps [92, 167]. The extreme case that the fusion happens before any processing layers is the aforementioned first category. Some methods fuse the representations in a mid-stage [33,76,129] and some in a late stage [65,253,305], as shown in Fig. 6.

In the third category belong methods that do not apply a naive fusion of the different representations, such as concatenation. Many works propose more sophisticated strategies for fusing the different modalities. For example, Wang et al. [303] try to specifically learn modality-specific and common features during training. As a result, the total complexity of the model reduces. Moreover, one modality might be missing some of the common features due to noise, such as occlusion, clutter or illumination. In such a case, the quality of the representation will not drop since the other modality will provide the necessary information. Another example is the work of Hazirbas et al. [97], where they make the assumption that one of the modalities is the main source of information and the rest are complementary. They assign one CNN to each modality, and then, at several levels of the CNN's hierarchy they insert information from the complementary branches to the main one. Deng et al. [50] followed a different approach. Instead of having two streams, they introduced a third stream, the interaction stream, which is comprised by their newly found GFU unit. By using this interaction stream, the feature maps of all streams are updated at the

interaction points. Park et al. [202] propose the multimodal feature fusion module in order to combine information from different modality-specific branches. Valada et al. [288] proposed a fusion module (SSMA) that emphasizes areas and modality-specific feature maps according to the feature map contents, thus leveraging common and modality-specific features.

Finally, some researchers defined data specific solutions. For example, the work of Georgiou et al. [76] evaluates three different modality-processing strategies specific for CFD simulation output, which consist of four different modalities over six channels of information. Gupta et al. [92] propose a data transformation for the depth channel in RGB-D data, called HHA. Mainly, they introduced two more channels. Although the values of those channels are computed from the depth map itself, they are transformations that are not easily learnable, by convolutional kernels, namely height from ground and surface angle to gravity vector.

The benefits of using this transformation are twofold. First, the network gets more relative information to its input, and second, with the depth information transformed to a three-channel representation it is possible to use pre-trained networks on ImageNet for this modality as well. Eitel et al. [57] proposed three more encodings that transfer the depth data to a three-channel representation and compared them to each other and HHA. Their intuition was that since in object classification, all objects have similar elevation, not all channels of HHA are interesting. The projections they proposed are (1) copy the depth values to all channels, (2) transform to the surface normal vector field and (3) apply jet colormap of depth values to rgb, ranging from red (near), through green to blue (far). They argue that since the networks are pre-trained on RGB data, transforming depth to rgb might result in a more stable fine-tuning of the networks. The last method showed the best results on object classification. Nonetheless, they do not perform a comparison in the case where the elevation makes a difference, and thus, there is no objective comparison between their method and HHA. For a visual comparison of the four different schemes, the reader is referred to [57].

3 Traditional methods

Traditional methods vary a lot depending on the application and the type of data they are applied on. For example, when dealing with semantic segmentation the most common, non-deep, approach is to apply a graph model like a conditional random field (CRF) [51,133,250,265]. On the other hand, a large group of works utilize template matching approaches [87,103,105,219] in order to tackle object detection. Although there is a large diversity on the applied methods, there are some common practices between most of them. The data are not processed in their raw format, but they are transferred in a feature space in which they are represented and then processed by any machine learning pipeline.

Building from the very successful work of feature representation of images in many applications of computer vision, a lot of methods are developed that generalize them to be applicable to higher dimensional data as well. The main idea is to describe the content of an image using a number of points or neighborhoods instead of the whole image. The type of description can vary, from raw values to histograms of gradients and point-wise comparisons. In order to get a good content description and not background description, researchers develop specialized detectors which detect points according to several characteristics. This very well-known pipeline is extended and applied to higher dimensional data.

The most common types of higher dimensional data that people are dealing with are objects represented by surfaces and/or color, volumetric representation of the world, videos or sequences of images, or in the extreme scenario four dimensional data, a three dimensional representation evolving in time. A large group of works try to generalize the interest point detectors and descriptors of images to the data available. Because of the different nature of different data types, the definition and development of features change accordingly. The main categories of such features are surface features, volumetric features and spatiotemporal features.

3.1 Object surface features

Many people have tried to derive heuristics and encodings of 3D shapes and objects that help to process them in an efficient way. The first approaches date back to 1984 with the work of B. Horn, Extended Gaussian Images [112]. Since then, numerous approaches and features have been developed. The main common objective is to have a low dimensional yet discriminative description of three dimensional objects and shapes. There are many ways one can separate these methods according to their characteristics. A common distinction is global and local features. Global features describe the whole object, while local describe a small neighborhood around a point on the object. The final description of the object is comprised by a collection of such local descriptions.

3.1.1 Global features

Global features usually try to aggregate low-level structural and geometric statistics of the complete objects like point pair distances, surface normals and curvature. Their advantage is the very low dimensional representation they offer in comparison with local descriptors that make object retrieval much faster. Unfortunately, they require the whole object to be available and fully separated from the environment [88]. Thus, they are very limited in real-world scenarios where objects are partially occluded and usually blended in

their environment. Some examples of global methods are the Extended Gaussian Images [112], shape distributions [201], the light field descriptor (LFD) [36], the spatial structure circular descriptor (SSCD) [71] and the elevation descriptor (ED) [246]. For a more comprehensive review of global features, the reader is referred to [71,88,246].

3.1.2 Local features

Local features describe some properties of the local neighborhood of an object's surface points. In order to describe a complete object, a set of these local descriptors have to be used. Depending on the needs of an application, a different scheme of accumulating these local features is used. For example, for object recognition the local features of an object in the repository are added to a feature library. These features are searched for candidate correspondences with the features of a scene, which vote for specific objects and poses [84]. Bronstein et al. [28] incorporated the well-established "Bag of Features" model of computer vision to 3D shape retrieval, in which the local features are translated to "visual words", or in this case "shape words", in order to obtain a global compact description of the full object. When tackling the scene semantic segmentation task, these features are considered as the data primitives in order to construct geometric unary potentials that are considered in an CRF pipeline [250,251].

As mentioned above, local descriptors encode information of a neighborhood around a point. In order to exclude points that do not carry enough information, feature detectors are introduced. These detectors usually find points whose neighborhoods exhibit large variance of some property, e.g., fast and multiple changes of the surface normals. Given a detector, a set of "highly informative" points is detected. Then, one can extract local descriptors only for those points and describe an object or scene only using these points neighborhoods. Since most real-world applications deal with varying scales of objects, as well as a variety of occlusions and deformations, feature detectors and descriptors must be invariant to scaling, rigid and non-rigid deformations, as well as illumination changes. Moreover, they need to be repeatable and unique. A very comprehensive study on surface detectors and descriptors has been published in [84]. In this paper, we will give a brief overview of the available detectors and descriptors.

Detectors Interest point, salient or *keypoint* detectors are a classic first step to object description, since they define which points of the surface are the most important for describing the object. A generic and popular division of detectors depends on whether they are scale invariant or not [84,283]. Although scale invariance is an important feature, not all detectors have that ability. Some of them take the scale or neighborhood size, in which they will detect keypoints, as an input. Consequently, detectors are classified as fixed-scale or adaptive-scale keypoint detectors.

Most fixed-scale keypoint detectors have two common steps [283]. They first compute a quality measurement across all points. Then, the points are checked for saliency by checking whether they are local maxima of the quality measurement. As an example, we describe the detector defined by Mokhtarian et al. [184]. A point is declared as interest point if its curvature is larger than the curvature of every 1-ring neighbor, where the k-ring neighbors are defined as the neighbors that have k edges distance. On the other hand, adaptive-scale detectors, inspired by the works of image detectors, first construct a scale-space and then search for local maxima of a defined function along the scale-space [283]. For example, Zaharescu et al. [329] build a scale-space by applying Gaussian filters directly on the 3D mesh and detect points as the extrema of the DoG space. For an extensive review of keypoint detectors, the reader is referred to [84,283].

Descriptors Local surface descriptors can be subdivided according to different factors. For example, they can be subdivided according to the invariance properties, i.e., invariant to rigid or non-rigid transformations, invariant to scaling, etc. The most common division for surface features is according to their encoding, i.e., histograms, point signatures and transformations [84,281], which we will follow in this work as well.

Histograms are a broadly used type of feature description, not only in describing 3D surface features but also in image and video analysis. Histograms accumulate different measurements of the neighborhood of a point and use that as a feature. Histograms have been very popular due to their simplicity combined with high descriptive capabilities. Three dimensional surface histogram descriptors can be subdivided into spatial distribution histograms (SDH), geometric attribute histograms (GAH) and oriented gradient histograms (OGH) [84].

SDH accumulate in histograms the spatial relationship, e.g., pair point distances, of points in a neighborhood. One of the first examples of SDH descriptors is the spin images (SI) [125,126]. The spin image is a two- dimensional histogram. First, all the neighboring points are transferred to a cylindrical coordinate system starting from the interest point. The points are expressed with the radial distance α and the elevation distance β. The 2D histogram accumulates the number of points in squares of the $\alpha - \beta$ plane. Other examples include the extensions of the SI, scale invariant SI (SISI) [49] and Tri-SI [88,89], the generalization of shape context (SC) [15], 3DSC [69] and the Rotational Projection Statistics (RoPS) [87]. More recent examples are the Toldi [320], the RSM [210], the BroPH [336] and the MVD [83].

Table 1 Collection of surface descriptors with the most influence on the field, according to our study

Method	Year	Comments
SI [125,126]	1998	Most sited surface descriptor
PFH [228]	2008	Captures multiple characteristics
FPFH [227]	2009	Improved computational efficiency of PFH
2.5D SIFT [166]	2009	SIFT for depth images
HKS [268]	2009	Invariant to non-rigid transformations
mesh-HOG [329]	2009	Extension of HOG [48] descriptor for triangular meshes
3D-SURF [136]	2010	Extension of SURF [12] descriptor for triangular meshes
SI-HKS [29]	2010	Scale invariant extension of HKS
SHOT [281]	2010	Signatures of histograms, balance between descriptiveness and robustness
CSHOT [282]	2011	Extension of SHOT descriptor to incorporate texture information
WKS [7]	2011	Invariant to non-rigid transformations, scale invariant, outperforms HKS
TriSI [89]	2013	Rotation, scale invariant and robust extension of SI descriptor
RoPS [87]	2013	Unique and repeatable LRF, robust to noise and mesh resolution
3DLBP [178]	2015	Generalization of LBP to three dimensions
3DBRIEF [178]	2015	Generalization of BRIEF to three dimensions
3DORB [178]	2015	Generalization of ORB to three dimensions
LFSH [319]	2016	Combines depth map, point distribution and deviation angle between normals
Toldi [320]	2017	Robust to noise, resolution, clutter and occlusion LRF. Multi-view depth map descriptor
RSM [210]	2018	Uses multi-view silhouette instead of depth map. Outperforms RoPS
BroPH [336]	2018	Binary descriptor, combines depth map and spatial distribution
MVD [83]	2019	Extremely low dimensional. Performs similar to SoA descriptors in object recognition

The table shows the most important contribution of the work to the field. For a more comprehensive study of surface descriptors the reader is referred to [84]

GAH accumulate geometric properties of the neighborhood of a point, e.g., angle between surface normals. Soma examples are the Local Surface Patch (LSP) [37], THRIFT [68], the point feature histogram (PFH) [228], its fast counterpart fast point feature histogram (FPFH) [227] and the Signature of Histograms of Orientation (SHOT) [281].

OGH accumulate gradients of various metrics of the surface. This kind of descriptors is closely related and inspired from image descriptors like SURF [12] and SIFT [168,169]. Some examples are the 2.5D SIFT [166], the meshSIFT [173], the meshHOG [329], 3DLBP [178], 3DBRIEF [178] and 3DORB [178].

Yang et al. [319] proposed a descriptor (LFSH) which combines SDH and GAH. Specifically, they use histograms of a depth map, point distribution and deviation angle between normals.

Signatures describe the local neighborhood of a point by encoding one or more geometric measures computed individually at each point of a subset of the neighborhood [84,281]. Some examples of signature descriptors are the exponential map [195] and the binary robust appearance and normal descriptor (BRAND) [189], a binary descriptor that encodes geometrical and intensity information from a local patch. This is achieved by fusing intensity variations with surface normal displacement.

Transforms These descriptors perform a transformation of the surface to a different domain and describe the neighborhood according to the characteristics of the surface on that domain. For example, Rustamov [226] performed a Laplace–Beltrami transform, while Knopp et al. [136] performed a Hough transform on a voxelized representation of the surface. Other examples of transform descriptors are the heat kernel signature (HKS) [268], its scale invariant variation (SI-HKS) [29], as well as the more recent wave kernel signature (WKS) [7].

A collection of the most important, according to this study, surface features is shown in Table 1. The features are shown together with what, in our opinion, is their most important contribution to the field.

Rotation invariance A common goal for most descriptors is to achieve rotational invariance. In order to achieve that they try to find a repeatable and unique Reference Angle (RA) or local Reference Frame (LRF) to which the local patch or neighborhood is rotated before they describe it [126]. The first approaches used the surface normal as a reference vector in order to achieve rotation invariance. Although the surface

normal is easy and fast to compute, it is very sensitive to noise. Other methods use the singular value decomposition (SVD) or eigenvalue decomposition (EVD) [25,195,335]. Unfortunately, these methods do not produce a unique LRF and in order to tackle that, multiple descriptors are extracted per point. A good overview and comparison of these methods is given in [281]. Moreover, they propose their own method which is more robust to noise and tackles the limitations mentioned above. To do that, it computes the EVD of a weighted N-nearest neighbor covariance matrix, in combination with the sign swapping of [25].

3.2 Volume features

In some applications, the data of interest are not represented by surfaces, but by volumes. Some examples include voxelized representation of the objects, as well as 3D images, mainly medical images, like 3D ultrasound, CT scans and MRI scans [39,192]. In some cases, videos are considered as three dimensional data where the time dimension is considered equivalent to the two spatial ones [239]. In order to describe the content of these kind of data, scientists generalized one of the known interest point detector and descriptor of 2D images to 3D, namely Lowe's SIFT detector and descriptor [168,169].

Scovanner et al. [239] were one of the first that tried to generalize the SIFT descriptor to the three dimensional case. Although they did extend the SIFT descriptor, they did not generalize the detector as well. The method picks random points in the volume as salient points and then describes them in a similar fashion to the SIFT. Orientation invariance is achieved by computing the dominant solid angle of the gradient and rotating the neighborhood around the point so that the solid angle is equal to zero. Finally, the neighborhood is split into eight subregions and a gradient orientation histogram is computed per region. The final descriptor is the concatenation of these histograms, which results in a 2048-D vector. They tested their descriptor on action recognition and showed that their method performs better than the regular 2D-SIFT.

At the same time, Cheung and Hamarneh [39] developed independently their own generalization. In contrast to Scovanner et al.'s work [239], they generalized both the descriptor and the detector. Moreover, instead of generalizing to the 3D case, they generalized to the nD case making their method applicable to many more datasets and applications. They use $n-1$ directions, with β bins for each, resulting in β^{n-1} bins in total. The gradients are computed using hyperspherical coordinates. They tested their method on 3D MRI of the brain and 4D CT scans of a beating heart.

Allaire et al. [5] focused on the 3D case. They observed that the aforementioned methods failed to account for the tilt that a neighborhood can have, resulting in the need for an extra angle in order to have full orientation invariance. For detecting points, they extended Lowe's method by computing the Difference of Gaussians (DoG) similar to Lowe manner. The local minima/maxima of the DoG in the scale-space are picked as interest points. After detection in the scale-space, feature points are filtered and localized. The remaining points are described as follows. First, they find the dominant solid angle and for each angle with magnitude above 80% of the maximum, they calculate the tilt. As with the solid angle, every angle that has a magnitude more than 80% of the maximum is considered as a different interest point. They evaluated their method on 3D registration and segmentation of clinical datasets such as CT, MR and CBCT images.

Ni et al. [192] used a similar method to the one developed by Allaire et al. [5] and adapted it for optimal description of ultrasound content, which is very noisy. They used the same filtering techniques at the detection stage with different thresholds, necessary due to the increased noise of ultrasound images. Besides the extension of Lowe's detector, they also applied the Rohr3D detector developed by [221]. It first defines the cornerness as the determinant of the matrix C, given by Eq. 5.

$$C = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{5}$$

where I_{ij} are the second-order intensity gradients of a voxel. The local maxima of the cornerness are then detected as interest points. For description, they do not use all three angles defined by [5] but only the two constituting the solid angle, like in [239]. They evaluate their method on 3D ultrasound registration and compare it to the original 3D SIFT of Scovanner et al. [239].

An overview of the aforementioned methods, together with the milestone of each work, is given in Table 2.

3.3 Spatiotemporal features

As with images and three dimensional representation of objects, traditional approaches that deal with videos follow the same regime. First, a number of points are defined as interest points. These points are either detected through some saliency measurement, which means that their neighborhood is considered as very informative, or they are densely sampled, e.g., [131]. These points are then used to describe the whole sequence of frames (either 2D or 3D). There are many methods that try to detect and describe this kind of interest points.

First, traditional approaches deal with time-dependent data, like video, either used a collection of 2D features, i.e., image features, to describe the clip or consider time as an

Table 2 Extensions of the SIFT descriptor to 3D volumetric data

Method	Data type	Dimensionality	Comments
Scovanner et al. [239]	Video	3	First 3D SIFT
Cheung and Hamarneh [39]	3D MRI and 4D CT	n	Detector and nD
Allaire et al. [5]	3D CT, MRI, CBCT	3	Detector and account for tilt and 3D DoG
Ni et al. [192]	3D Ultrasound	3	Ultrasound specific noise filter and smoothing

extra dimensional equivalent to the spatial ones and thus represent the clip as a 3D volume. As such, simple extensions of the image features to the 3D case are used to describe the volume [239]. Although this method produced good results at the time, the different nature of the time dimension as well as the large variance in sampling frequencies by different sensors, i.e., frame rate, motivated scientists to develop methods that describe spatiotemporal volumes while regarding time separately. These features are called spatiotemporal features. The new interest points are known as Space–Time Interest Points (STIPs).

3.3.1 STIP detectors

The first STIP detector was proposed by Laptev [144]. It is an extension of the Harris corner [95], called Harris3D. The Harris3D operator considers different scales in the space and time dimensions. To achieve that, it convolves the video sequence f with a Gaussian kernel g given by Eq. 6.

$$L(\cdot;\sigma_l^2,\tau_l^2) = g(\cdot;\sigma_l^2,\tau_l^2) * f(\cdot) \tag{6}$$

where the spatiotemporal Gaussian kernel is given by:

$$\begin{aligned} g(\cdot;\sigma_l^2,\tau_l^2) &= \frac{1}{\sqrt{(2\pi)^3\sigma_l^4\tau_l^2}} \\ \times \exp&\left(\frac{-(x^2+y^2)}{2\sigma_l^2} - \frac{t^2}{2\tau_l^2}\right) \end{aligned} \tag{7}$$

where σ_l^2, τ_l^2 are the spatial and temporal variances, respectively, and x, y are the spatial coordinates while t is the temporal one. Given a space and a temporal scale, a corner or interest point is found by finding the local maxima of the corner function given by Eq. 8.

$$H = \det(\mu) - k\,\mathrm{trace}^3(\mu) \tag{8}$$

where μ is the 3 by 3 second-moment matrix weighted by a Gaussian function, given by Eq. 9. In a later work, Laptev and Lindeberg [146] extended the detector in order to be velocity adaptable, which provides invariance to camera motion. In order to achieve that they considered the transformation caused by camera motion as a Galilean transformation, which is computed iteratively. This approach was later used by [145] for motion recognition. Schuldt et al. [237] combined the feature size adaptation of [144] and the velocity adaptation [146] in a single framework.

$$\mu = g(\cdot;\sigma_i^2,\tau_i^2) * \begin{bmatrix} L_x^2 & L_xL_y & L_xL_z \\ L_xL_y & L_y^2 & L_yL_z \\ L_xL_z & L_yL_z & L_z^2 \end{bmatrix} \tag{9}$$

Another very popular spatiotemporal detector is the one developed by Dollár et al. [52], known as cuboids. The motivation behind their detector lies in the observations that (1) corners are very sparse in images and even sparser in videos and (2) there are movements, like opening and closing of a jaw that do not include corners, and thus, if only corners are chosen to represent a video clip, many actions will not be recognizable. STIP are detected at the local maxima of the response function given in Eq. 10.

$$R = (I * g * h_{\mathrm{ev}})^2 + (I * g * h_{\mathrm{od}})^2 \tag{10}$$

where $g(x, y; \sigma)$ is a 2D Gaussian smoothing function applied only on the spatial dimensions and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters, given by Eq. 11, applied temporally. The scale of the feature in the spatial dimensions is defined by the Gaussian (σ) while in the temporal dimension by the quadrature pair ($\tau, \omega = \frac{4}{\tau}$).

$$\begin{aligned} h_{\mathrm{ev}}(t;\tau,\omega) &= -\cos(2\pi t\omega)e^{-\frac{t^2}{\tau^2}} \\ h_{\mathrm{od}}(t;\tau,\omega) &= -\sin(2\pi t\omega)e^{-\frac{t^2}{\tau^2}} \end{aligned} \tag{11}$$

Bregonzio et al. [24] observed that the aforementioned detector has some drawbacks. The Gabor filters applied in the temporal dimension are very sensitive to noise and produce many false detections in textured scenes. Moreover, it fails to recognize slow movements. In order to deal with these drawbacks, they propose their own STIP detector which works in two steps. The first step is simple differencing between consecutive frames in order to produce regions of interest in which there is motion. The second step is to apply, spatially, a 2D Gabor filter.

Oikonomopoulos et al. [197] followed a different approach. They extended to the spatiotemporal case the approach

Table 3 Existing spatiotemporal detectors

Method	comments	Year
Harris3D [144]	First STIP detector	2003
Harris3D + velocity adaptation [146,237]	Limit camera motion detections	2004
Cuboids [52]	More dense point detection	2005
Bregonzio et al. [24]	Limit false detections and detect slow movements	2009
Oikonomopoulos et al. [197]	Information based saliency	2005
Wong et al. [311]	Use of local and global information	2007
V-FAST [325]	Efficient computation	2010
Chakraborty et al. [34]	Limit background detections	2012
Li et al. [158]	Unified motion and appearance	2018

In the left column shows the name of the descriptor together with the paper that proposes it, in the middle column the contribution of the method to the field, and in the right column the year the method was published

of Kadir and Brady [127]. They first defined a measure of saliency based on the amount of information change in a neighborhood, which they expressed by the entropy of the signal in the neighborhood. The extension to the spatiotemporal case is done by considering a cylindrical neighborhood instead if a two dimensional circle.

Wong and Cipolla [311] argued that all the above methods detect interest points using only local information, which produces a lot of false positives in the presence of noise. In order to counter this drawback, they proposed an alternative approach which uses global information in order to detect interest points in a video sequence. In order to do so, they applied nonnegative decomposition of the sequence, which is represented by a two- dimensional matrix, in which each column is a frame of the video. The result of the decomposition is a number of subspaces ϕ and transitions χ. By applying Difference of Gaussians (DoG) on the subspaces and the transitions, they detect spatiotemporal interest points. They compared their method with the aforementioned approaches on gesture recognition using the same description for all detectors and showed that their method outperforms the rest.

Inspired by the work of Laptev [144], Willems et al. [310] proposed an new detector which instead of utilizing the second moment matrix μ (given by Eq. 9) they utilized the Hessian matrix H given by Eq. 12. The points are detected at the local maxima of the saliency measurement S given by Eq. 13. Unlike the 2D case [13], maxima of S do not ensure positive eigenvalues of H which means that saddle points will also be detected.

$$H = \begin{bmatrix} L_{xx} & L_{xy} & L_{xz} \\ L_{xy} & L_{yy} & L_{yz} \\ L_{xz} & L_{yz} & L_{zz} \end{bmatrix} \tag{12}$$

$$S = |\det(H)| \tag{13}$$

Yu et al. [325] developed a generalization of the FAST [223] detector to the spatiotemporal case, which they call V-FAST. For each candidate point, they considered three 2D planes, the XY, XT and YT planes. They applied the FAST detector in each plane. If the point is detected as interest point in the spatial domain (XY plane) and at least one of the time comprising planes (XT or YT), then the point is considered as a STIP.

Cao et al. [32] observed that from all STIPs detected by Laptev's [144] detector, only the 18% belong to a specific action while the rest belong to the background. Inspired by this phenomenon, Chakraborty et al. [34] proposed an new pipeline for STIP detection. They initially detect spatial interest points (SIPs) using the Harris detector [95] and then apply background suppression and other temporal and spatial constraints in order to keep only features relative to the motion in the sequence.

Finally, Li et al. [158] proposed a new detector, the UMAM-detector. The video is transferred to a Clifford algebra-based representation. There a vector is extracted for each pixel which contains both motion and appearance information. In this new space, they apply a Harris corner detector to detect STIPs. According to their experiments, the UMAM-detector outperforms all the aforementioned detectors and some deep learning methods, in classification performance.

All the above detectors are summarized in Table 3, together with their contribution to the field.

3.3.2 STIP descriptors

In order for the STIPs to be in an optimal representation for machine learning pipelines, special descriptors are defined that try to capture important information for the neighborhood of the STIP. Most proposed descriptors can be categorized depending on the type of measurements they contain or the way they quantize that information. More specifically, the most typical measurements taken to describe a STIP are the N-jets [137], Gaussian gradient field (similar to HoG and SIFT [48,168]) or optical flow field [17].

These measurements are usually quantized or vectorized by histogramming or Principal Component Analysis (PCA) [145,147].

The N-Jets represent a collection of point derivatives (up to Nth order) at a specific scale of the scale-space representation L, given by Eq. 14.

$$J(g(\cdot;\sigma_0,\tau_0) * f) = \{\sigma L_x, \sigma L_y, \tau L_t, \sigma^2 L_{xx}, \ldots, \sigma\tau^{N-1} L_{yt..tt}, \tau^N L_{tt..tt}\} \quad (14)$$

The Gaussian first-order gradient field is also computed on the scale-space representation L, in order to make the descriptors invariant to scaling and noise. The optical flow field represents the movement in a clip at each pixel by a velocity vector field. There are a lot of methods that try to efficiently and accurately extract that vector field. For a good overview of the optical flow estimation field, the reader is referred to [267].

As mentioned above, there are many ways to accumulate information over the spatiotemporal neighborhood. The most common ones are histogramming and applying PCA. Histogramming is either applied globally, i.e., one histogram over the STIP neighborhood, or on several small neighborhoods around the STIP. In the later case, the separate histograms are concatenated in order to constitute a single descriptor. PCA is usually applied on a number of IP of a train set in order to obtain D most significant dimensions defined by the eigenvectors.

Laptev et al. [145,147] tested a number of different descriptors both in terms of measurements accumulated and in the type of accumulation. Their study showed that, on average, local histograms on adaptive scales perform better than the rest of the approaches. Moreover, methods based on the first-order gradient field outperform both optical flow and the N-Jets.

In a parallel work, Dollár et al. [52] performed a similar comparison. They tested normalized pixel values, first-order intensity gradients and optical flow values. They tried all the above measurements by flattening the cuboid and within global or local histograms. Finally, on all descriptors, they applied PCA to reduce the dimensionality. According to their experiments, histogramming did not benefit performance and thus concluded to the flattened values with PCA. As with Laptev et al.'s experiments, the gradient-based descriptors showed higher overall performance than the rest.

Niebles et al. [193] extended the aforementioned descriptor. They first smooth the image at a specific scale and then extract the intensity gradients. The apply this function for several scales and then apply PCA to get the final descriptor. Their method indeed outperforms Dollár et al.'s [52] method, but it is still outperformed by Laptev et al.'s [145] histogram of gradients, with velocity adaptation.

Laptev et al. [148] proposed a combined histogram of gradients with a histogram of optical flow. Their descriptor together with the nonlinear SVMs managed to outperform all previous methods on the KTH dataset [237]. Willems et al. [310] extended the known SURF descriptor [12] to the spatiotemporal case. Their implementation differentiates between the spatial and temporal dimensions by setting a different number of bins, as well as different scales (σ and τ). They evaluated their method on the mouse behavior dataset as well as the KTH, and they achieve comparable to the state-of-the-art results.

Klaser et al. [135] designed a new 3D HoG descriptor. They introduced a generalization of the orientation binning of the known SIFT descriptor by introducing a normal polyhedron, dodecahedron or icosahedron and considering each face of the polyhedron as a bin. The angle of the gradient vector to the surface normals of the faces is computed and if its smaller than a threshold, the projection of the gradient vector to the surface normal contributes to the respective face's bin. Moreover, they generalized the integral image method of [293] to the integral video method. The integral video is a representation of the video volume that helps the fast computation of average gradients. Given a video volume $v(x, y, t)$ and its three first- order partial derivatives $v_{\partial x}, v_{\partial y}, v_{\partial t}$, the integral video of direction j is given by:

$$iv_j(x,y,t) = \sum_{x'<x, y'<y, t'<t} v_{\partial j}(x', y', t') \quad (15)$$

A block of video **b** is first divided into SxSxS sub-blocks. For each sub-block, the average gradient and its contribution to the histogram bins are calculated. The final descriptor is a concatenation of several such histograms computed on MxMxN blocks around the STIP. Willems et al. [309], inspired by the quantization of Klaser et al. [135], extended the method of [310] to quantize the gradient orientations in the same way.

Yeffet and Wolf [323], inspired by the Local Binary Pattern descriptor [198], proposed the Local Trinary Pattern (LTP) a spatiotemporal motion descriptor. The main idea of the descriptor is to compare patches between frames instead of pixels within an image. Eight patches neighboring the pixel in question in the previous and next frames are defined, as well as a "central" patch which includes the pixel in question, as shown in Fig. 7. A trit is calculated for each spatial location (i, j) according to the following rule:

$$\begin{array}{l} -1 \; if \; \text{SSD1} < \text{SSD2} \\ 0 \; \; if \; \text{SSD1} = \text{SSD2} \\ +1 \; if \; \text{SSD1} > \text{SSD2} \end{array} \quad (16)$$

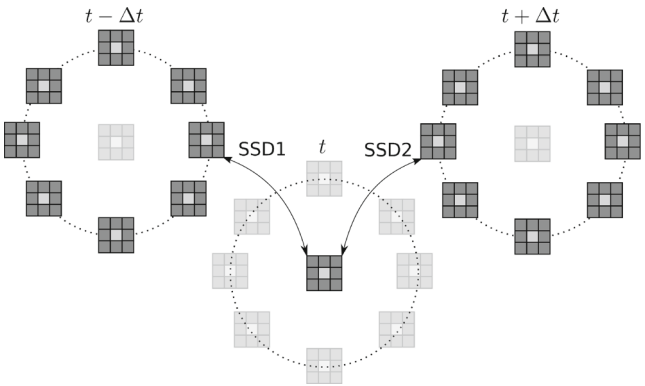

Fig. 7 Illustration of the encoding process of LTP. For each of eight different locations at time $t - \delta t$ and the same locations at $t + \delta t$, SSD distances of 3×3 patches to a central patch at time t are computed [323]

where SSD is the sum of square differences between the patches (Fig. 7). A global descriptor is calculated by combining the trinary patters for all available pixels in histograms. First, spatial histograms are created by splitting each frame in (m x n) patches. The resulted histograms are then merged temporally to create one global spatiotemporal descriptor.

3.3.3 3D space

Due to the inexpensive available sensors, scientists extended the STIPs to the 3.5 and four dimensional cases as well. To the best of our knowledge, the first to define detectors and descriptors for higher than 2+ time dimensional data are Xia and Aggarwal [315]. Their detector is similar to Dollár et al. [52]'s Cuboids. The motivation behind their method is that due to the nature of depth images, detectors developed for color-based STIP detection tend to find many points in the background and thus introducing a lot of noise in the description of a clip. In order to avoid that they introduced a correction function that smooths out depth map specific type of noise. After the detection of the Depth-STIPs (DSTIPs), the information of the spatiotemporal neighborhood is described by a occupancy histogram.

In later work, Oreifej and Liu [200] generalized the Histogram of surface Normals (HON) [272] to four dimensional surfaces (HON4D) and applied it on 3D action recognition. Finally, Rahmani et al. [212] proposed the histogram of oriented principal component (HOPC). Their descriptor calculates the principal components of the scatter matrix of spatiotemporal points around an interest point and create a histogram of principal components for all points in a neighborhood. In a later work, they also proposed a detector in order to filter out points that are irrelevant [211]. Their method first computes the ratio of sequential eigenvalues. If the surface is symmetric, then at least one of these ratios is going to be one. Thus, they define a threshold, and if a ratio is below that the point is excluded. Otherwise, the neighborhood of that point is considered informative enough to be of interest.

3.3.4 Trajectories

Driven by the poor generalization performance of the aforementioned approaches, researchers proposed a new strategy for handling the time dimension [177,182,269]. Instead of describing the change in the temporal dimension in a local manner as with the spatial ones, researchers tried to describe motion using trajectories of spatial interest points and their spatial description.

More specifically, Matikainen et al. [177] track features in a video using the standard KLT method [170]. For every tracked feature, they keep a vector of frame-by-frame position derivatives. The resulting vector is the trajectory feature. These features are then clustered, and the Bag of Words (BoW) model is implemented. The final action classification happens using an SVM. In parallel work, Messing et al. [182] proposed a very similar feature which they call velocity history. The difference with the aforementioned method is that they quantize the velocities in eight directions and five magnitudes. Moreover, the classification is done by a generative mixture model instead of the BoW approach. Sun et al. [269] proposed a different approach, but in the same direction. Instead of the KLT method, they find trajectories by applying frame-by-frame SIFT feature matching. According to their results, this is a more robust approach for feature tracking. Then, the visual characteristics of each trajectory is described by the average SIFT descriptor tracked. In order to describe the temporal dynamics of the trajectory, a Hidden Markov Chain (HMC) is employed that is trained on the spatial development of features. Finally, the inter-trajectory context is encoded with their proximity descriptor.

Wang et al. [298,299], inspired by the success of the aforementioned methods as well as the dense sampling of features in images [196], proposed a combination, the dense trajectories. The trajectories are sampled on multiple scales on a spatial grid via dense optical flow. Finally, the area around the trajectories is described by the HOG-HOF spatiotemporal descriptor. Their method achieved the state-of-the-art results at the time, on many benchmarks. In later work, Wang and Schmid [300] proposed an improvement on the dense trajectories. They tracked camera movement and used it to reject trajectories caused by it. Moreover, they applied the estimated camera movement as a correction to the optical flow, in order to extract camera motion invariant trajectories.

Table 4 Large-scale datasets and benchmarks for object understanding

Dataset	Data type	# Images	# Objects	# Object Cat.	6DoF pose
PSB [247]	Polygonal surface geometry	–	1814/6670	161/1271	–
ModelNet [314]	CAD	–	151,128	660	–
ShapeNet [35]	CAD	–	3M/220K	– /3135	–
shapeNetCore [35]	CAD	–	51,300	55	–
shapeNetSem [35]	CAD	–	12K	270	–
YCB [31]	RGB-D	600	75	–	No
Rutgers APC [216]	RGB-D	10K	24	24	Yes
SUD [41]	RGB-D	23M	> 10K	44	No

4 Datasets and benchmarks

One of the main motives behind the research on higher than two dimensional data is the large availability of datasets comprised by such representations. Depending on the application and the type of data different datasets and benchmarks are proposed, both small scale and large scale. In this section, we will give an overview of the well- known and current benchmarks and large datasets for the domain of computer vision in higher dimensions and we categorize them according to their intended application. To be more precise, numerous small-scale datasets and benchmarks exist that are meant for very specific applications. Nonetheless, for each type of data, i.e., 3D scene, action in video, objects, etc., there are some large- scale datasets that help evaluate the data representation methods that can be applied on many different tasks. These are the datasets that are presented here and are categorized according to the type of data they deal with, namely object understanding, scene understanding and video understanding. More specific concepts can be added, like video retrieval, but due to the small number of datasets, they are grouped together in a category called "other datasets".

4.1 Object understanding

There is a large collection of datasets with various 3D models of objects used for object understanding tasks, like detection and classification, shape understanding and more. These datasets either contain 3D images or scans of real objects, e.g., [235,247] or they might contain designed objects like CAD models [314]. Moreover, different datasets are used for different tasks. For example, the LINEMOD dataset [104] is used for object detection, classification and pose estimation, while the Princeton shape benchmark (PSB) [247] focuses on different classification themes. Besides these state-of-the-art datasets, there are also smaller but well-known datasets. Some of these are Lai et al.'s [143] dataset, the big bird [255] and the SHREC [154]. For a good overview of all these benchmarks and datasets, the reader is referred to [67]. Table 4 gives a comparison of the state-of-the-art datasets.

The largest datasets available, to date, are datasets that contain designed models and objects instead of real scans, largely due to the longstanding graphics communities. Some of the well-known datasets are the Princeton shape benchmark [247], which consists of 161 object classes and a total of 1814 models. The ModelNet [314], a dataset which consists of 151,128 3D CAD models in 660 categories. ShapeNet [35] is also a recent database, which tries to make even more detailed annotations than just object labels. The raw dataset consists of roughly 3 million models, from which 220,000 have been classified into 3135 categories. Besides the raw dataset, the authors also made two subsets. The first, called shapeNetCore, consists of 51,300 models in 55 common categories, with extra alignment annotations and the second, shapeNetSem, consists of 12,000 models from 270 categories. In addition to manually verified category labels and consistent alignments, they are also annotated with real-world dimensions, estimates of their material composition at the category level and estimates of their total volume and weight [35,236].

As mentioned above, there are also datasets with scanned real-life objects instead of designed models. One example is the YCB object and model set [31]. It consists of everyday object scans from 75 object categories. For each object, the dataset includes 600 RGB-D images coupled with 600 high-resolution RGB images, segmentation masks, as well as calibration information and texture-mapped 3D mesh models. The Rutgers APC RGB-D dataset [216] consists of more than 10 thousand RGB-D images. In total, it contains 25 objects along with their 6DoF pose. Choi et al. [41] created a dataset of scanned 3D objects with an RGB-D camera. The dataset provides a variety of different objects, from bottles of shampoo to sculptures and even an Howitzer. They grouped these objects in 44 categories. Besides the raw RGB-D videos, they also provide 3D reconstruction for some of the objects. Some example 3D reconstructions can be seen in Fig. 8. For more information about the reconstruction tech-

Fig. 8 Example scans of real objects from Choi et al.'s [41] dataset. Original figure from [41]

nique and the number of objects reconstructed, we refer the reader to the original paper [41]. All the above datasets are summarized in Table 4.

4.2 Scene understanding

Scene understanding is a domain that refers to machine learning pipelines that are able to perform several tasks given a scene, such as object detection and localization, scene semantic segmentation, scene classification and more. In general, it includes all methods that increase the understanding of a scene through visual means. Due to the significant qualitative difference in terms of applied sensors and the structure of indoor and outdoor scenes, they are considered as separate problems.

One of the first "bigger" datasets is Berkley's B3DO dataset introduced by Janoch et al. [119]. It is comprised by 849 from 75 scenes captured by an RGB-D camera. Overall, it includes more than 50 object classes. One of the most known datasets and most used benchmarks for indoor scene understanding is the NYUv2, created by Silberman et al. [251] in 2012. It is comprised by a set of indoor videos taken with RGB-D camera, resulting in 795 labeled images with 894 object classes. Xiao et al. [316] tried to provide a richer dataset, in the sense that the segmentation is not pixel-wise, but there is a better 3D representation of the objects. The result is the SUN 3D dataset [316] which also provides point cloud segmentation produced by Structure from Motion (SfM). Song et al. [258] realized that existing datasets were limited in (1) the number of scenes and sequences they include and (2) they have sequences from a single RGB-D camera type. They created a more large-scale and generic dataset, the SUN-RGBD dataset. They achieved that by taking images from existing datasets and also introducing their own. The result was a dataset with 10,335 RGB-D images of a total of 47 scene categories and 800 object classes. Hua et al. [113] created sceneNN, a dataset that contains 100 scenes with per-pixel annotation of objects. The scenes are 3D reconstructed on triangular meshes.

Most of the scene understanding datasets suffer from small variation in well-annotated scenes and limited number of objects. Handa et al. [94] created a method for dataset creation in order to tackle these problems. They claimed that their system is able to create virtually infinite number of scenes with various objects in them and perfect per-pixel annotation. They accomplish that by using computer graphics to artificially create scenes. They also acquired a large number of 3D CAD models, from some of the datasets mentioned in Sect. 4.1, and randomly placed them in the scenes. The resulted dataset can be used in order to properly pre-train a CNN which can be then fine-tuned on a real-world dataset. McCormac et al. [180] continued this work with the goal to create a dataset, called SceneNet RGB-D, with annotation not only for semantic segmentation, object detection and instance segmentation but also scene trajectories and optical flow. For comparison, example real scenes from the NYUv2 are shown in Fig. 10 and some artificial scenes from the SceneNet RGB-D in Fig. 9. Similar to their work, Song et al. [259] created a synthetic 3D scene dataset called SUN-CG, which contains 45,622 synthetic scene layouts created using Planner5D [259]. Dai et al. [47] introduced a much bigger dataset with real- world scenes than all the aforementioned. It consists of 1513 scenes with overall 2.5M RGB-D frames and more than 36K object instances. All scenes have been reconstructed and labeled manually.

For a good comparison, the datasets, together with their features and details, are shown in Table 5. As with the object datasets of the previous section, we can see that the artificial datasets are orders of magnitude larger than the datasets that contain images and videos of real scenes.

The aforementioned datasets focus only on indoor scenes and objects. When considering outdoor scenes, the availability of datasets decreases significantly. One of the reasons is the low quality of the RGB-D sensors in open space. Most of the existing datasets are limited to 2D RGB images, for example Richter et al.'s [217] dataset and the SYNTHIA dataset [222]. Nonetheless, the KITTI dataset [75], although built for pedestrian, car and cyclist detection on images, it also includes Velodyne 64E range scan data with 2D and 3D bounding boxes for 7500+ frames. Moreover, the Sydney Urban Objects dataset [209] contains labeled Velodyne LiDAR scans of 631 urban objects in 26 categories.

4.3 Video understanding

The most active areas in video understanding are action recognition and video retrieval. Most of video understanding-related researches focus on action recognition and more

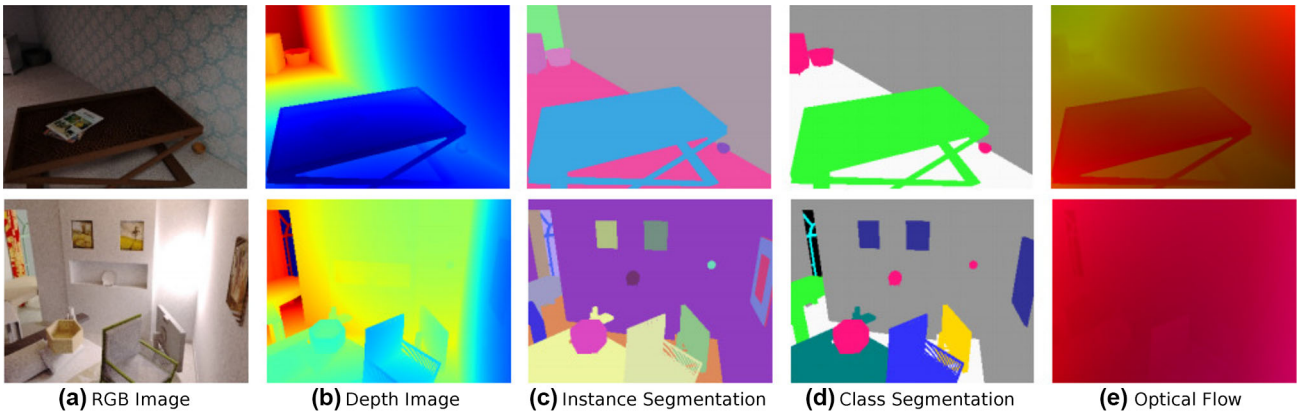

Fig. 9 Example images from the SceneNet RGB-D dataset [180]. **a** RGB image, **b** depth image, **c** ground truth instance segmentation, **d** ground truth class segmentation, **e** optical flow

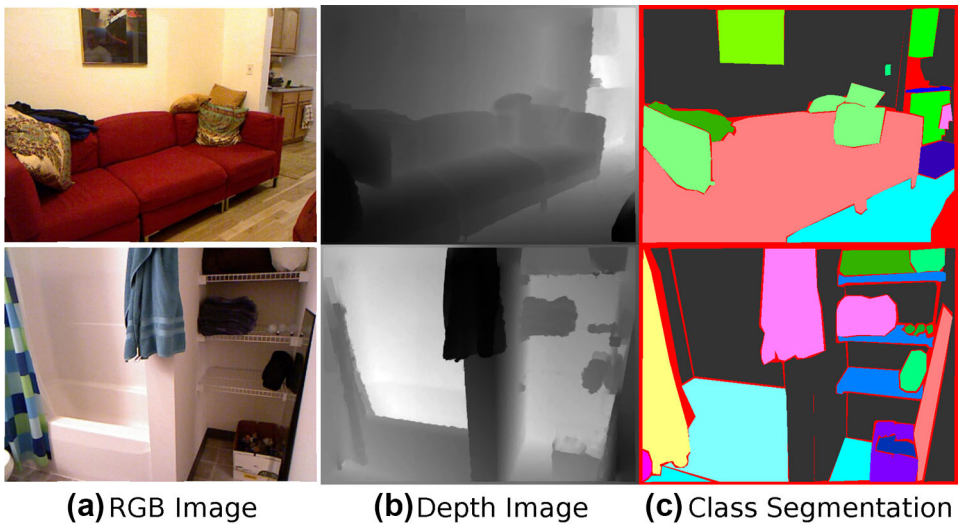

Fig. 10 Example images from the NYUv2 dataset [251]. **a** RGB image, **b** depth image, **c** ground truth segmentation

Table 5 Big-scale datasets and benchmarks for indoor scene understanding

Dataset (reference)	RGB-D video	Per-pixel annotation	traj. GT	RGB texture	# scenes	# layouts	# object classes	3D Models avail.
B3DO [119]	No	Key frames	No	Real	75	–	> 50	No
NYUv2 [251]	Yes	Key frames	No	Real	464	464	894	No
SUN 3D [316]	Yes	3D point cloud + Video	No	Real	254	415	–	Yes
SUN RGB-D [258]	No	Key frames	No	Real	–	–	∼ 800	No
sceneNN [113]	Yes	Video	Yes	Real	100	100	≥ 63	Yes
SceneNet [94]	No	Key frames	No	non-pr	57	1000	–	Yes
SceneNet RGB-D [180]	Yes	Video	Yes	pr	57	16,895	255	Yes
SUN-CG [259]	Yes	Video	Yes	non-pr	45,622	45,622	84	Yes
ScanNet [47]	Yes	3D + Video	?	Real	1513	?	≥ 20	Yes

The first column shows the name of the dataset, the second column shows whether the dataset provides RGB-D video of the scenes, the third one the level of the annotation, the forth one whether trajectory ground truth is included, and the fifth whether the data are real, or synthetic. "pr" means photorealistic, while "non-pr" means non-photorealistic. Sixth, seventh and eighth columns show the number of scenes, layouts and object classes, respectively, and the ninth, last, columns show whether the dataset provides 3D models of the objects present in the dataset

specifically human action recognition. Action recognition is the main research area for which new representation approaches and video understanding methods are developed and tested on. There is a large collection of datasets and benchmarks whose content relates a lot on the evolution of the "action recognition" research. Good overviews of these benchmarks and their historic value are given by Hassner [96] and Idrees et al. [116]. In this section, we will give an overview of the state-of-the-art datasets and benchmarks.

One of the well-known and used benchmarks today is the Human Motion Data Base (HMDB51) [141]. It consists of 6766 video clips, each representing one out of 51 "everyday" actions collected from various sources on the Internet. The annotation is done in a redundant way (each label is verified by at least two humans) in order to ensure its quality. Moreover, every video has some extra meta-data such as camera viewpoint and motion. Although, for todays standards, this consists a small- to medium-scale dataset, it is still widely used due to its very accurate ground truth. A similarly popular dataset is the UCF101 [261] dataset. It consists of 13,320 clips which belong to one of the 101 action classes of the dataset. These classes are single-person actions as well as person-to-person interactions. Caba Heilbron et al. [30] proposed the ActivityNet, a dataset of human activities. It contains about 20 thousand videos from 203 different human activities. Most videos are between 5 and 10 min long with a maximum of 20 min. In these videos, the classes are manually annotated and specified in time. This results in about 30 thousand human-annotated clips of a specific human action. Recently, Kay et al. [130] proposed the Kinetics dataset, the largest human action dataset to date. It consists of 306,245 trimmed clips from YouTube that include human–object and human–human interactions. The clips are classified to one of the 400 possible classes and were annotated using Amazon's Mechanical Turk (AMT) [130].

One of the largest datasets at the time of this paper is the Sports 1M dataset [129]. It consists of 1 million YouTube videos assigned to one of 487 classes. These classes are sport actions such as road bicycle training, track cycling and monster truck. These videos have been automatically annotated according to the video tags. Moreover, these are five-minute videos so the class might be a small proportion of the whole video. Due to the above reasons, the labeling of the data is very weak and thus hard to properly evaluate different algorithms. Jiang et al. [123] released the Fudan-Columbia Video Dataset (FCVID), a dataset that contains over 90 thousand videos from 239 categories. Most of these categories are actions like "making cake" while there are some object and scene categories as well. The videos are collected from YouTube and are manually labeled. Abu-El-Haija et al. [1] released the largest to date video dataset, the YouTube-8M. It consists of about 8 million videos with 4 thousand labels in total. Each label is supposed to shortly explain the content of the video. For example, a video of biking on dirt roads and cliffs would have a central topic/theme of Mountain Biking, not Dirt, Road, Person, Sky [1]. Possible labels are also filtered out according to some characteristics. For example, a label must be visually recognizable and should not require specialized knowledge.

Barekatain et al. [11] introduced an aerial view video dataset for human action recognition; it consists of 43 videos with varying camera position and motion. The videos are staged and include multiple actors that perform several actions out of the 12 defined classes. Goyal et al. [81] introduced the "something–something" dataset. It is an action recognition dataset where the labels are of the form "something" action "something", for example "Dropping [something] into [something]". The dataset is manually annotated and consists of about 108K short videos ($\tilde{4}$sec) with 174 action classes and more than 23K object names. Monfort et al. [185] introduced the "Moments in Time" dataset. A big dataset of one Million 3-second clips with 339 classes of verbs are picked from the VerbNet.

A summary of all the above datasets can be found in Table 6. For a more comprehensive review on human action recognition datasets, the reader is referred to [256].

4.4 Other datasets

Besides the scene understanding, object and action classification datasets mentioned in the previous sections, there are also datasets for a big variety of applications. For example, the Cornell dataset [122] is a dataset built with the goal of training robotic grasp detection on various objects. It contains 1035 RGB-D images with 280 graspable objects annotated with several positive and negative graspable rectangles. For the goal of shape deformation, Yumer et al. [327] created a dataset, containing objects from various categories and their deformations scales that was later also used for other research purposes, for example [328]. Garcia and Vogiatzis [73] proposed the MovieDB, a dataset for different image-to-video retrieval tasks [72]. The TACoS dataset [213], with action labels on videos as well as natural language descriptions with temporal locations, and the Charades-STA [70] have been used for text-to-clip video retrieval. The DiDeMo dataset [6] has been introduced for temporal localization given natural language, but has also been used for the purpose of text-to-clip video retrieval [317]. Recently, the Hollywood 3D dataset was proposed [93] which contains 650 stereo clips with 14 action classes, together with stereo calibration and depth reconstruction.

Table 6 Big-scale datasets and benchmarks for video understanding

Dataset	#Videos	#Clips	#Classes	Multi-label	Trimmed	Manually annotated
HMDB51 [141]	3312	6766	51	No	Yes	Yes
UCF101 [261]	2500	13,320	101	No	Yes	Yes
Sports 1M [129]	1M	–	487	No	No	No
ActivityNet [30]	19,994	28,108	203	No	Both	Yes
FCVID [123]	91,223	91,223	239	No	No	Yes
YFCC100M [280]	0.8M	–	–	–	No	–
YouTube-8M [1]	∼ 8M	–	4800	Yes	No	No
Kinetics [130]	306,245	306,245	400	No	Yes	Yes
Okutama—action [11]	43	43	12	Yes	Yes	Yes
Something–something [81]	108,499	108,499	174	No	Yes	Yes
Moments in time [185]	1M	1M	339	No	Yes	Yes

5 Research areas

5.1 Object classification and recognition

A very well researched topic that includes three dimensional representation of the world is 3D object classification and recognition. Given an object with a 3D representation, a system has to classify the category or the instance of the object. Although conceptually, a straight forward task, it constitutes a very complex problem because it requires efficient and complicated representation methods that are able to capture the high-level content from the raw representation. Moreover, it is a fundamental step in understanding the three dimensional world. As a result, it is considered a very good benchmark for 3D world representation methods. During our research, we identified two large clusters of object classification and recognition methods, depending on the data they process. These are methods that try to classify full 3D objects, usually available as CAD models, and methods that classify RGB-D images of objects.

5.1.1 RGB-D object recognition

The first methods applied for this task are inspired by the imaging community. Researchers were trying to develop handcrafted descriptors that were then used to discriminate between different objects. One of the first examples of such methods is the work of Lai et al. [142], which extracts spin images from the depth map and SIFT features from the RGB values. They create two different vocabularies using the efficient match kernel (EMK) method. The resulted representation is fed into a linear SVM (linSVM), a Gaussian kernel SVM (kSVM) and a random forest (RF) and compare their performance on their RGB-D object dataset [142,143]. Other works apply the well-known kernel descriptors (KDE) [20] on several characteristics of an RGB-D image, while other use the hierarchical kernel descriptor (HKDE) [18], which applies the kernel descriptor also on the kernel representation instead of only on the pixel level, creating a hierarchy of kernel descriptors.

With the recent success of deep convolutional neural networks (Deep CNN) in image analysis tasks, researchers try to extend these methods to the three dimensional representations as well. One of the first approaches toward training features from data from more than two dimensional representations was done by Bo et al. [21] who learned features in an unsupervised manner from RGB-D data and Socher et al. [257] who trained a convolutional-recursive neural network. Alexandre [4] proposed a transfer learning method where different networks are used for each channel (three color channels and depth map). Instead of training each network from scratch, they take as initialization method the weights of the best performing network trained so far. Since their experiments aim to test the increase in performance using the transfer learning method, they do not compare to other methods. Unfortunately, they also use a subset of the original dataset which makes the comparison to other methods impractical. Eitel et al. [57] propose a fusion architecture, in which two networks are trained, one on the RGB data, pretrained on ImageNet [225] and an other on the depth map. The two networks are combined with a late fusion to produce the final result.

We summarize the performance of all the above methods, on the RGB-D object recognition benchmark [142,143] in Table 7. The benchmark used for this comparison provides two different tasks. One is the category- level classification, where a classifier is supposed to label the type of object. The second is instance-level classification, where the classifier is supposed to identify the specific object from different views and in different environments.

Table 7 Performance of object recognition methods on the RGB-D object recognition dataset [142]

Method	Category	Instance
linSVM [142]	81.9 ± 2.8	73.9
kSVM [142]	83.8 ± 3.5	74.8
RF [142]	79.6 ± 4	73.1
KDE [20]	86.2 ± 2.1	84.5
HKDE [18]	84.1 ± 2.2	82.4
Upgraded HMP [21]	87.5 ± 2.9	92.8
CNN-RNN [257]	86.8 ± 3.3	–
Fus-CNN [57]	91.3 ± 1.4	–

The performance is measured by classification accuracy. Left column describes the method, the middle column presents the results on the category-level classification benchmark, and the right the instance- level classification performance

Table 8 Performance of object classification methods on the ModelNet 10 (MN10) and 40 (MN40) benchmarks [314]

Method	Type	MN10	MN40
shapeNet [314]	3D	83.54	77.32
MV-CNN [266]	2D proj.	–	90.1
VoxNet [179]	3D	92.0	83.0
DeepPano [245]	2D proj.	88.66	82.54
MVCNN-MultiRes [207]	2D proj.	–	91.4
MO-AniProbing [207]	3D	–	89.9
ORION [241]	3D	93.9	89.4
FusionNet [100]	Both	93.11	90.8
VRN [26]	3D	93.61	91.33
VRN-ensemble [26]	3D	**97.14**	**95.54**
Wang et al. [295]	2D proj.	–	93.8

The performance is measured by classification accuracy. The left column describes the method, the middle column presents the results on the ModelNet10 classification benchmark, and the right the performance on the ModelNet40 classification benchmark

5.1.2 3D object classification

As mentioned in Sect. 2.2.1, early deep learning approaches on learning from a three dimensional representation define two design concepts. The first approach is to train CNNs straight from a three dimensional representation of voxel grids [314], while the second one applies 2D projections. In the context of 3D object classification, the projection is done via a multi-view approach [266]. Most of the proposed methods for 3D object classification belong to one of these two categories.

Both strategies have received a lot of attention. The 3D kernel approach was first applied in this research area by Wu et al. [314].They utilize a 3D convolutional DBN, which is trained on their newly proposed ModelNet. The idea of 3D convolutional kernels is further explored with the works of Maturana and Scherer [179], who introduced a 3D CNN as well as a new representation approach. Later, Qi et al. [207] tried to improve the 3D CNN approach in three stages:1) new network structure, 2) data augmentation and 3) feature pooling. Sedaghat et al. [241] added an auxiliary task, namely pose estimation. Hegde and Zadeh [100] fused multi-view and 3D CNNs, while Brock et al. [26] defined blocks of layers based on the inception [270] and ResNet [99] architectures, namely Voxception, Voxception-downslample and Voxception-ResNet.

The projection to lower dimensions has also received a lot of attention. As mentioned above, Su et al. [266] proposed a multi-view approach, where pictures of the object are taken from 20 different views and processed by a pre-trained, on ImageNet, network. Shi et al. [245] proposed the projection of the shape on a cylinder, described in Sect. 2.2.1, and Qi et al. [207] improved the multi-view approach by introducing a multi-resolution extension of data augmentation. Wang et al. [295] argued that the view pooling approach of the multi-view strategies fails to take into account important information from different views since only one survives the pooling. In order to alleviate this issue, they introduced a recurrent clustering and pooling layer based on graph theory. With their approach, they achieved SoA performance on the ModelNet 40 dataset.

The performance of the above methods is summarized in Table 8. Although for the most part, multi-view approaches were outperforming the voxel-based approaches, the work of Brock et al. [26] with the Voxception-ResNet approach managed to outperform all multi-view approaches. Nonetheless, their strategy needs to train multiple big networks from scratch, while the work of Wang et al. [295] only needs to fine-tune the networks lowering the training time by multiple orders of magnitude while still having competitive performance.

5.2 Semantic segmentation

An important research area using such three dimensional datasets is semantic segmentation. Semantic segmentation or scene labeling is the procedure of labeling every pixel, or voxel, in an image, as shown in Figs. 9 and 10. Most methods tackle this problem by utilizing only RGB images. Since depth sensors became widely accessible, people started to use this extra information in order to make better predictions. The methods that utilize these features are heavily influenced by their RGB-only counterpart. In this work, we will only focus on the methods that utilize the depth information since we are interested in applications and methods that deal with higher than two dimensional data. Most traditional methods tackle this problem by utilizing handcrafted features, introduced in

Table 9 Performance evaluation of different methods on the NYU datasets (v1 and v2)

Method	Year	Shallow/deep	NYUv1	NYUv2					
				4 Classes		40 Classes			
			pixacc	pixacc	clacc	fwavacc	avacc	pixacc	clacc
SIFT+MRF [250]	2011	Shallow	56.6 ± 2.9	–	–	–	–	–	–
Silberman et al. [251]	2012	Shallow	–	58.6	–	–	–	–	–
KDES [215]	2012	Shallow	*76.1 ± 0.9	–	–	–	–	–	–
Gupta et al. [91]	2013	Shallow	–	–	–	45.1	26.1	57.9	*28.4
Hermans et al. [102]	2014	Shallow	59.5	69.0	–	–	–	–	–
RF + SP + CRF [186]	2014	Shallow	–	*72.3	*71.9	–	–	–	–
Khan et al. [133]	2014	Shallow	–	69.2	65.6	–	–	–	–
Gupta et al. [90]	2015	Shallow	–	–	–	45.9	26.8	58.3	–
Deng et al. [51]	2015	Shallow	–	–	–	*48.5	*31.5	*63.8	–
Stückler et al. [265]	2015	Shallow	–	70.9	67.0	–	–	–	–
Couprie et al. [46]	2013	Deep	–	64.5	63.5	–	–	–	–
R-CNN [92]	2014	Deep	–	–	–	47.0	28.6	60.3	35.1
FCN [167]	2015	Deep	–	–	–	49.5	34.0	65.4	46.1
Eigen and Fergus [56]	2015	Deep	–	**83.2**	–	**51.4**	34.1	65.6	45.1
Wang et al. [303]	2016	Deep	**78.8**	–	**74.7**	–	–	–	47.3
RDF-152 [202]	2017	Deep	–	–	–	–	**50.1**	**76.0**	**62.8**
3DGNN [208]	2017	Deep	–	–	–	–	43.1	–	59.5

The first column refers to the methods and the papers that present them. The second column is the year that the methods were published. The third column shows whether the method is follows a traditional approach, shallow learning or a deep learning approach. The fourth column shows the per-pixel average accuracy on the NYUv1 dataset using all 13 classes. The rest of the columns show the performance results on the NYUv2 dataset. The fifth column and sixth column refer to the four-class segmentation task, while the rest on the 40-class segmentation task [251]. pixacc refers to the average per-pixel accuracy, clacc refers to the average per class accuracy, fwavacc is the frequency weighted average accuracy, and avacc refers to the meanIU, or the mean Intersection over Union [97]. We highlight the per category (shallow or deep) best performance with a * and the overall best with bold

Sect. 3, in a conditional random field (CRF) or Markov random field (MRF) model. The usual pipeline is to oversegment the image in super pixels. Extract features from the superpixels and then use them to construct unary and pairwise potentials for the CRF or MRF model. With the success of deep learning in image classification, researchers try to adapt these methods for three dimensional semantic segmentation as well.

The first to tackle this problem in the higher than two dimensional representations is Silberman and Fergus [250]. In their work, they use a CRF-based approach and define unary potentials encoding spatial location and pairwise potentials encoding relative depth. The unary potentials are learned from a neural network using local descriptors. They evaluate their approach on their NYUv1 dataset, which they construct for the purpose of their project. Moreover, they test different descriptors, both image and depth descriptors, and compare their performance. They extended their work [251], by introducing a new extended version of NYU, NYUv2, which is still one of the most used datasets for benchmarking scene segmentation algorithms. Couprie [45] explored other CRF-like approaches in order to improve the computational complexity of the algorithm. Ren et al. [215] improved the segmentation performance by using kernel descriptors [19,20] and by combining superpixel MRF with segmentation trees for contextual modeling. Koppula et al. [138] oversegmented a 3D pointclound [59], while Gupta et al. [90,91] introduced gravity direction prediction. Hermans et al. [102] proposed an RDF classification which is refined using a Dense CRF. Deng et al. [51] proposed a method that jointly considers local and global spatial configurations in order to alleviate the local nature of handcrafted descriptors. Stückler et al. [264,265] proposed a method for real time semantic segmentation on RGB-D videos, which combined RGB-D SLAM and RFs, while Müller and Behnke [186] used the output of this method as a feature for unary node potentials on a CRF model. Khan et al. [133] introduced a new region growing algorithm to extract fundamental geometric planes and extract appearance and geometric unary potentials from these planes, utilized by a CRF model.

As mentioned above, a lot of methods that utilize deep learning have been also developed. Within this category, we can identify two clusters of methods. The first represents a transition from the aforementioned traditional methods to

Table 10 Performance evaluation of different methods on the SUN-RGBD dataset [258]

Method	Year	SUN-RGBD		
		clacc	avacc	pixacc
*FCN [167]	2015	41.13	30.46	68.35
LSTM-CF [159]	2016	48.1	–	–
FuseNet-SF5 [97]	2016	48.3	37.29	76.27
RDF-152 [202]	2017	**60.1**	**47.7**	**81.5**
SSMA [288]	2018	–	38.4	–

The first column refers to the method, and the second shows the year the method was published. The rest of the columns show the performance results on the SUN-RGBD 37 class benchmark. pixacc refers to the average per-pixel accuracy, clacc refers to the average per class accuracy, and avacc refers to the meanIU, or the mean Intersection over Union [97]. We highlight the best performance with bold. It should be noted that all methods shown on this table are deep learning methods. *FCN refers to the work of [167], but the performance on the SUN-RGBD is reported by [288]

Table 11 Performance evaluation of different methods on the ScanNet dataset [47]

Method	Year	avacc
SSMA [288]	2018	57.7
RFB-Net [50]	2019	**59.2**

The first column refers to the method, and the second shows the year the method was published. The third column shows the performance results on the ScanNet class semantic segmentation benchmark on the test set as reported by the benchmark website. avacc refers to the mIoU. We highlight the best performance with bold. It should be noted that all methods shown on this table are deep learning methods

the pure deep learning ones. In these, the networks are used in order to extract features that are then used to classify segments or superpixels either using graph models like CRF and MRF or some other classifiers. Some examples are the works of Couprie et al. [46] who adopted a multi-scale approach by adapting the previous work in semantic segmentation [63,64], Höft et al. [110] and Wang et al. [294] who proposed a multimodal unsupervised method that would automatically learn rich high- and low-level features from an auto-encoder.

The second cluster is initiated by the work of Long et al. [167], who introduced the fully convolutional networks (FCN) in order to produce per-pixel, dense, classifications. These networks are end-to-end trainable and do not rely on other methods. Eigen and Fergus [56] trained a multi-scale convolutional neural network to predict the depth map, surface normals and provide semantic segmentation. Wang et al. [303] designed two convolutional and deconvolutional networks, one trained on depth values and one at RGB values. These networks explicitly try to learn common features between different modalities (see Sect. 2.2.2). Li et al. [159,160] proposed an LSTM-CNN approach called LSTM-CF and Hazirbas et al. [97] extended the work of Noh et al. and Badrinarayanan et al. [10,194] to also utilize depth information. Finally, Park et al. [202] adapted the very successful work of Lin et al. [161], RefineNet, to use RGB-D data. They do that by introducing the multimodal feature fusion (MMF) block which fuses feature maps from an RGB-specific and a depth-specific network. These fused representations are used as input to the refine blocks of RefineNet [161]. Valada et al. [288] used the SSMA (Sect. 2.2.2) module to fuse geometric and color features, while Deng et al. [50] used the interaction stream that they introduced, described in Sect. 2.2.2 as encoders. The outputs of the streams are fused together and sent to a decoder to predict the class labels.

Qi et al. [208] introduced a method which combines the two methodologies. They do that by utilizing graph neural networks (GNN) instead of a CRF or MRF. They experiment with unary potentials extracted from a pre-trained VGG as well as a ResNet. Moreover, as an update function for the GNN they try both MLP and an LSTM.

The performance of the aforementioned methods on the NYU benchmarks [250,251] can be seen in Table 9. For all benchmarks, the highest performance is reported by deep learning methods and more specifically the second cluster of the deep learning methods. Nonetheless, the best performing traditional approaches still outperform the first cluster of the deep learning approaches. Table 10 shows the performance evaluation of the methods on the SUN-RGBD dataset. From both tables, it can be seen that the RDF-Net of Park et al. [202] outperforms all other methods by a large margin, on every benchmark tested. Table 11 shows the performance evaluation of the methods on the scanNet dataset. On this benchmark, the RFB-Net [50] outperfroms the SSMA [288]. Unfortnately, there is no overlap on the tested benchamrks between the RFB-Net and RDF-152, making it infeasible to compare the two methods.

5.3 Human action classification

To the best of our knowledge, human action classification is the most researched area concerning image sequences, or videos. Given a short video clip that contains humans performing an action, an automated system has to be able and classify the given action. Depending on the dataset, these actions might be single-human actions, like standing up or opening door, single-human actions in a sport environment, or person-to-person actions, like hugging or kissing. Like with many fields that deal with visual data, early approaches include template matching while a bulk of traditional approaches define interest points in order to describe small clips and using these interest point and special descriptors try to classify the actions. More recent approaches try to apply deep learning methods to this field as well.

5.3.1 Traditional methods

As stated above, the very early approaches are based on templates [22,243,244]. Unfortunately, these methods cannot define single templates for each activity which renders them insufficient [220]. Thus, researchers turned their attention to other models, like the Hidden Markov Model (HMM), Hidden Semi-Markov Model (HSMM), conditional random field (CRF) and support vector machines (SVMs). Another group of methods extract a representation that is derived using the STIP detectors and descriptors introduced in Sect. 3.3. Finally, a group of works exploit trajectories of points in order to describe and classify actions [177,182,269,298–300], as described in Sect. 3.3.4.

Yamato et al. [318] were the first to apply HMM on the action classification problem. Oliver et al. [199] follow a different approach. They first extract the human positions and their trajectories and utilize a coupled HMM (CHMM) in order describe pairwise human interactions. Wang and Mori [307] utilized the hidden CRF (HCRF) in order to classify actions, while Song et al. [260] proposed a hierarchical recursive sequence representation coupled with a CRF model for sequence learning. Fernando et al. [66] tried to model the evolution of the actions in an video. In order to do that he used the "learning to rank" framework on the Fisher Vector representation of each frame.

As mentioned above, many methods followed the classical approach for image classification, utilizing interest points. Schuldt et al. [237] proposed a local SVM approach combined with the BoF representation in order to classify single-human actions in videos. Later, Laptev et al. [148] test both HoG and HoF to describe the STIPs. They use them to generate a BoF representation of the clips. From the combinations, they tested the best performing one was the HoF features.

Sun et al. [269] were one of the first to explore trajectories. They extract SIFT trajectories from the clips and measure the average SIFT descriptor along those trajectories. Wang and Schmid [300] used dense trajectories with corrected camera motion, encodes them using Fisher Vectors and finally classify them using a linear SVM. Kovashka and Grauman [139] proposed a hierarchical feature approach. They created different vocabularies for a BoF representation for multiple scales. From all the aforementioned methods, the only approach that still stands out today and can be compared to the state-of-the-art deep learning methods which is the trajectory-based improved dense trajectories (IDT) of Wang and Schmid [300], and thus, it is the only for which we report results.

5.3.2 Deep learning

Many deep learning approaches have been proposed for tackling the HAR task. The main bulk of works can be divided into three schemes, namely full 3D CNNs, two-stream networks and CNN-LSTM approaches. Regardless of the class of the method, besides a small number of works, the input to the networks is a small part of the video, usually referred to as clip. The length of these clips can vary from five to sixteen frames. A more detailed overview of the methods is given bellow.

To the best of our knowledge, the first to apply deep learning on HAR were Taylor et al. [274]. In their work, they proposed a special RBM, the convolutional gated RBM (convGRBM), which is a generalization of the gated RBM (GRBM) [181]. Their method alleviates a limitation of GRBM, the fact that it cannot scale up to large inputs. Their method shares weights in all locations of an image and thus can scale to large inputs. As an old approach, this work does not fit with our classification scheme.

Ji et al. [121] proposed the first 3D CNN for action recognition. Their network has five 3D convolutional layers, one 2D convolutional layer and the output, classification layer. Since their network takes as an input only seven frames, they use a feature vector from a long span of frames as auxiliary input through a hidden layer. In a later work, Tran et al. [284] delved into optimizing the architecture of 3D convNets for spatiotemporal learning. Their experiments indicated that uniform kernels (3x3x3) give the best overall performance. Karpathy et al. [129] did a detailed research on what architecture can exploit the time dimension better. They tested four different strategies, namely single frame network, early, late and slow fusion networks. Interestingly enough, the single frame network has similar performance to the rest, which means that these first approaches toward spatiotemporal understanding using deep CNNs are not able to exploit the temporal dimension as well.

Baccouche et al. [9] also proposed a 3D convolutional neural network. They deal with the long-term actions by building an RNN-LSTM network which takes as input the output of the 3D CNN network. Donahue et al. [54] proposed a very similar architecture; they stacked an LSTM on top of a CNN network and called the complete architecture long-term recurrent convolutional neural network (LRCN). The two main differences with the model of [9] are that they train their network end-to-end and that the CNN is pre-trained on ImageNet.

Simonyan and Zisserman [253] proposed a new strategy, the two-stream networks. In this architecture, one network processes the RGB values of a single frame, while an other processes ten stacked frames of optical flow fields. The spatial network is first pre-trained on ImageNet and thus increasing the performance of the approach. The final decision on the class of a clip is done by averaging the classification results of the separate networks. Wang et al. [305] identified as drawbacks of deep learning approaches on HAR, the lack of large data and the limitation of the complexity

Table 12 Performance evaluation of different methods on the UCF-101 [261] and HMDB-51 [141] datasets

Method	Year	+IDT	RGB	Flow	UCF-101	HMDB-51
IDT [300]	2013	–	–	–	86.4	61.7
Two-Stream [253]	2014	No	Yes	Yes	88.0	59.4
Karpathy et al. [129], Sport 1M pre-train	2014	No	Yes	No	65.2	–
TDD [304]	2015	No	Yes	Yes	90.3	63.2
C3D ensemble [284], Sport 1M pre-train	2015	No	Yes	No	85.2	–
Very deep two-stream [305]	2015	No	Yes	Yes	91.4	–
Two-stream fusion [65]	2016	No	Yes	Yes	92.5	65.4
LTC [289], Kinetics pre-train	2017	No	Yes	Yes	91.7	64.8
Two-stream I3D [33], Kinetics pre-train	2017	No	Yes	Yes	**97.9**	**80.2**
(2 + 1)D [285], Kinetics+Sports 1M pre-train	2018	No	Yes	Yes	97.3	78.7
TDD + IDT [304]	2015	Yes	Yes	Yes	91.5	65.9
C3D ensemble + IDT [284], Sport 1M pre-train	2015	Yes	Yes	No	90.1	–
Dynamic Image Networks + IDT [16]	2016	Yes	Yes	No	89.1	65.2
Two-stream fusion + IDT [65]	2016	Yes	Yes	Yes	93.5	69.2
LTC + IDT [289], Kinetics pre-train	2017	Yes	Yes	Yes	92.7	67.2

The first column refers to the method, and the second shows the year the method was published. The third column specifies whether IDT is used in combination with the networks. The forth and fifth column shows whether the method is utilizing RGB and optical flow inputs, respectively. The sixth and seventh columns show classification accuracies of the methods on the UCF-101 and HMDB-51 datasets, respectively

and depth of the networks applied. In order to alleviate these issues, they proposed some "good practices" for training very deep two-stream networks. The first important step is that the temporal network is also pre-trained on images and thus able to be much deeper. Second, they utilized state-of-the-art very deep networks, (VGG19 [254] and GoogleNet [271]) for both streams. Furthermore, they proposed more data augmentation techniques for the videos and applied smaller learning rates. Feichtenhofer et al. [65] identified two drawbacks with the two-stream strategy as applied until then. (1) It was not able to learn correlations between spatial and temporal features since the fusion happened after the classification, and (2) the temporal scale was limited since the temporal network only considered ten frames. Also inspired by the work of [190], they proposed a temporal fusion two-stream network. They applied feature map fusion before the last convolutional layer. They fused the two streams and activations from several frames with a 3D convolutional layer followed by a 3D pooling layer. Carreira and Zisserman [33] proposed to inflate existing architectures from images to three dimensions. They do that not only in terms of architecture but also inflate the trained parameters. Given this starting point, they trained two networks, one on RGB values and one on optical flow. Finally, they averaged the outputs in order to provide a unified prediction.

Ng et al. [190] followed a different approach, where they make predictions while processing the whole video sequence rather than short clips. They tested several architectures including two-stream networks, LSTM and other temporal feature pooling mechanisms. Applying max pooling over the temporal dimension in the last convolutional layer (i.e., convPooling) and the LSTM are the two best performing strategies for temporal handling. Their convPooling network takes as input 120 frames while the LSTM 30 and both give the similar results. In similar work, Varol et al. [289] proposed a long-temporal convolutional network (LTC). Their network is processing 60 frames per video clip. They defined a number of 3D convolutional networks, each processing different resolutions and modality, i.e., RGB and optical flow. The classification scores of all networks are averaged out in order to produce the final prediction.

Wang et al. [304] proposed the trajectory-pooled CNNs (TDDs). Inspired by the work of [300] and the lack of CNNs in exploiting long-term temporal relationships, they proposed the trajectory-pooled deep convolutional descriptors (TDDs), where they compute descriptors by computing trajectories of CNN features maps using the method of [300] and encoding them using Fisher Vectors.

Tran et al. [285] proposed to decompose the spatial to the temporal convolution, thus creating the (2+1)D convolution which is a 2D spatial convolution followed by a 1D convolution exploiting the temporal dimension. Their top performing network is a (2+1)D, two-stream network which has a much lower complexity than the top performing 3D networks, while keeping the performance competitive.

We summarize the results of some of the above methods in Table 12. There are several conclusions we can derive from these results. Simple 3D networks seem to be outperformed by CNN-LSTM as well as two-stream networks, but the combination of them outperforms the "single solution"

networks. Moreover, pre-training on large datasets with not very accurate annotation, such as Sports 1M [129], benefit the quality of the networks. Last but not least, as with many applications, the best performing traditional approach, IDT [300], is outperformed by most recent deep learning approaches. Nonetheless, the combination of IDT and networks produces better results, by a constantly large margin, driving us to the conclusion that the high-level handcrafted features seem to capture information that is not learned by the networks, rendering them complementary.

5.4 Other areas

There are numerous more research areas and applications that deal with high dimensional data. Some examples are:

Outdoor object detection Outdoor object detection is a very well-studied research topic with many real-life applications, like autonomous vehicles and security. Some more specific examples of object detections are pedestrian detection, vehicle detection, like cars motorcycles and bicycles. Traditional methods first segmented the input point cloud and then classified the segments with various methods [14,275,276,296]. For example, Behley et al. [14] used the BoW model to describe each segment and used it to classify it. State-of-the-art methods take advantage of deep neural networks. Some examples are [61,205]. Qi et al. [205] use the pointnet++ as a base, while [61] utilizes 3D convolutional kernels and [155] utilizes a 2D FCN with the depth data as an extra modality. To the best of our knowledge, [205] achieves the state-of-the-art performance on the KITTI benchmark [75].

Structure from Motion (SfM) and simultaneous localization and mapping (SLAM) are very challenging tasks. SLAM is the process where the algorithm is trying to identify the position of the camera or sensor in the environment while constructing a map of the environment. SLAM is a very challenging while very interesting and important in the field of robotics as well as augmented reality. Traditionally, people were trying to match new environment parts to the constructed map by matching features (usually handcrafted) and RANSAC-like algorithms. Some representative work can be found in [59,60,132,187,263,308]. SfM is the process of building a 3D representation of a scene/environment of a camera by using multiple views and more specifically views from the same camera as it moves in the space. It usually is part of SLAM since it tries to built a 3D representation of the local environment of the camera. A comprehensive survey on SLAM and SfM was recently published by Saputra et al. [234].

Action recognition in 3D videos is a relatively new research field. As with video action recognition, the target of the task is the classification of human actions in different kinds of categories. The methods applied in this field can be divided into two categories depending on the type of data they process. More precisely, they process skeleton data or depth data [211]. Also methods that process color data have been proposed but since these are much closer to the 2D Video action recognition, described in Sect. 5.3, than the rest of these methods we do not consider it as part of this section. Skeleton-based approaches first extract the joints positions, usually using the OpenNI tracking framework [249], and then either use them [322], or information from the area around them [301,302], to describe the motion. Depth-based approaches use either silhouettes [156,290] or 4D histogram descriptors [200,211,321] in a BoW framework to describe each action and then try to classify them. In recent years, plenty of DL approaches have been proposed as well. They usually utilize an RNN-LSTM on joints and skeletons [55,164,242] or process directly the depth data in time [306]. For a good overview of deep learning approaches, the user is referred to [333].

6 Discussion

Although this field has come a long way, there are still a lot of challenges that the researchers face. Since most of these methods are generalized from successful methods developed for two dimensional images, all limitations and problems that arise when dealing with two dimensional images existing here as well. For example, when it comes to deep learning, the models are typically not understood and treated as black boxes [86]. Although researchers know how these models update their parameters and learn from the data, retrieving the information that they have learned is still an open research area. More specifically, although there has been done research on feature visualization [252,326,331], it is still unknown how to discover or understand what the networks learn and how they behave. Another inherit limitation is the typical lack of rotation invariance of the models, although some methods try to work around it. For example, Cheng et al. [38] train a specific layer to be orientation invariant. They do that by adding a penalty term to the loss function to force the layer to become rotation invariant. Although the result of the specific layer is rotation invariant, the rest of the network is not. In cases where information from multiple layers is needed, such as semantic segmentation, this solution does not suffice. An other example is the work of Marcos et al. [174]. They rotate the kernels and convolve with the rotated kernels and thus obtain responses from all possible orientations. The rotation invariance of this strategy is also limited since the information of the orientation is getting lost during the orientation pooling operation.

Besides the inherited difficulties from the two dimensional case, other problems arise when trying to extrapolate to more dimensions, either when the increase is an increase in physical dimensions or if it is an increase in available modalities. A common limitation to all state-of-the-art methods that deal with higher than two dimensional data is the high demand of resources. This limits the possible size of the deep learning methods. Moreover, as shown from the two dimensional case, these methods highly depend on the complexity and size of the resulted models [86,99,114,270], which combined with the increased complexity of the data as well as the increase in demand renders very difficult to efficiently apply them.

According to the results of the previous sections, the state-of-the- art performance on volumetric data is achieved using deep learning models. As described above, these methods have many drawbacks, both inherited from the drawbacks of deep learning in general as well as drawbacks regarding computational complexity. Moreover, it is still unclear which strategy for dealing with the higher dimensionality of the data is better. To be more precise, it is still unclear whether reducing the dimensionality to two is better than using three dimensional kernels. In the later case, it is still unclear which representation of the data works best. All these are questions left unanswered while the computational complexity of the models together with the lack of very large-scale, high dimensional, diverse and well-annotated datasets make the unbiased comparison between approaches very hard.

Difficulties arise when processing spatiotemporal data as well. Although the current results show that methods that utilize optical flow outperform methods that do not, it is still unclear how to optimally include this information. Moreover, the difference of space and time is still a challenging concept. It is still not clear how to process them in order to acquire as much information as possible from both spatial contexts as well as their temporal interactions. Furthermore, most approaches process only short-term interactions and only a few process more that 16 frames long clips, thus encoding long-term interactions [289]. Processing many frames though becomes very computationally expensive, and thus, the question of how to optimally perform temporal and spatial pooling arises. Although there has been significant development in the field the long-term impact and directions for continued advances are still unclear. Some of the limiting factors are the fundamental theory for understanding the strengths and limitations of the networks, approaches for learning with small training sets and/or the availability of accurately annotated, diverse and large- scale real-life datasets.

6.1 Major challenges

In summary, the major challenges as described by the research community are:

- Deep learning in high dimensional data is very computationally and memory expensive, limiting the capabilities of the applied approaches.
- Deep learning approaches lack invariance in many transformations, such as scale and rotation, which are usually tackled by very computationally expensive approaches.
- There exist many competing strategies for handling high dimensional data, and it is still not clear which approaches are suited better for which type of data and more importantly why.
- For many applications, there are not enough labeled data to properly train and test methods. Nonetheless, the past few years, in some research areas, this issue has been slowly tackled by introducing large-scale datasets such as the ScanNet [47] and the Moments in Time [185].

6.2 Future work

According to our study, there is significant room for improvement in all research areas covered by this survey. Nonetheless, we can identify some common issues to most of them. In most cases, deep learning approaches are too computationally expensive for many real- world applications, while the traditional counterparts have much lower performance. It is important to get as high-performing approaches while minimizing computational complexity and memory demands. Moreover, being able to leverage information from different modalities without performing unnecessary computations for common features while not missing modality-specific information is very important to the whole field. Although there are similarities in the type of dimensionality increase in different research areas, the solutions applied are usually unique to the research area. It would be interesting to acquire knowledge from multiple and create unified solutions.

7 Conclusions

This paper presents a comprehensive review of methodologies, data types, datasets, benchmarks and applications of computer vision on high dimensional data (higher than 2D). Based on the recent research literature, we identify four main data sources, namely image videos, RGB-D images and videos and 3D object models, such as CAD models. Moreover, we identify common practices between methods that are applied on all data types despite their qualitative difference. For example, deep learning approaches and handcrafted features, such as histograms, are developed and applied on all data types and research areas mentioned in this paper. Most of the methods are inspired by the previous work in computer vision on 2D data.

Regarding deep learning methods, we discuss the interrelationships and give a categorization of generalization of

methods to higher dimensions, namely generalization in case of increase in physical dimensions and generalization in case of increase in modalities, or information per physical position. Finally, we review and discuss the state-of-the-art methods on the most researched areas using these data, such as 3D object recognition, classification and detection, 3D scene semantic segmentation, human action recognition and more.

According to our study, we can draw some conclusions regarding the top performing approaches. Deep learning approaches seem to outperform handcrafted feature-based approaches when it comes to recognition performance in all tested settings (i.e., object classification, recognition and detection, semantic segmentation and human action classification). Nonetheless, handcrafted feature based have much lower time complexity. In some cases, they can produce similar performance to the state-of-the-art deep learning method, as shown in object detection by Tejani et al. [278]. As shown in human action recognition, with the IDT approach [299], the handcrafted features can provide complementary information to the deep learning features increasing the overall performance of a system by a large margin. When the number of physical dimensions is increasing, although early experiments showed that projecting information to lower dimensions and taking advantage of large available systems outperformed the raw processing of the high dimensional data; nowadays, we see an opposite trend. For example, the work Brock et al. [26] on object detection as well as Carreira and Zisserman [33] on HAR outperform 2D projection methods. Finally, late fusion seems to be the best performing naive strategy across the board for combining different modalities, while fusion in multiple levels and fusion on multiple stages of the process seem to outperform all other methods, e.g., Wang et al. [303] and Park et al. [202].

Understanding the world around us is a difficult task [165]. Although there is a lot of progress in this area, there are still a lot of room for improvement. For most data types, there is no clear solution or approach that properly handles the extra dimensions. For example, even in the well-studied area of video understanding, there is not a definitive way to handle the difference between space and time. Similarly, in the three dimensional static world even the optimal raw format of the data, e.g., point cloud, 3D mesh or voxelized, is unknown.

Acknowledgements This work is part of the research program DAMIOSO with project number 628.006.002, which is partly financed by the Netherlands Organization for Scientific Research (NWO) and partly by Honda Research Institute-Europe (GmbH).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abu-El-Haija S, Kothari N, Lee J, Natsev P, Toderici G, Varadarajan B, Vijayanarasimhan S (2016) Youtube-8m: a large-scale video classification benchmark. arXiv preprint arXiv:1609.08675
2. Agostinelli F, Hoffman M, Sadowski P, Baldi P (2014) Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830
3. Alahi A, Ortiz R, Vandergheynst P (2012) Freak: fast retina keypoint. In: Proceedings of the CVPR. IEEE, pp 510–517
4. Alexandre LA (2016) 3D object recognition using convolutional neural networks with transfer learning between input channels. In: Intelligent autonomous systems, vol 13. Springer, pp 889–898
5. Allaire S, Kim JJ, Breen SL, Jaffray DA, Pekar V (2008) Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. In: Proceedings of the CVPRW. IEEE, pp 1–8
6. Anne Hendricks L, Wang O, Shechtman E, Sivic J, Darrell T, Russell B (2017) Localizing moments in video with natural language. In: ICCV. IEEE, pp 5803–5812
7. Aubry M, Schlickewei U, Cremers D (2011) The wave kernel signature: a quantum mechanical approach to shape analysis. In: ICCVW. IEEE, pp 1626–1633
8. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450
9. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding. Springer, pp 29–39
10. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder–decoder architecture for image segmentation. Trans Pattern Anal Mach Intell 39:2481–2495
11. Barekatain M, Martí M, Shih HF, Murray S, Nakayama K, Matsuo Y, Prendinger H (2017) Okutama-action: an aerial view video dataset for concurrent human action detection. In: Proceedings of the CVPRW. IEEE, pp 28–35
12. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: Proceedings of the ECCV. Springer, pp 404–417
13. Beaudet PR (1978) Rotationally invariant image operators. In: Proceedings 4th international joint conference pattern recognition, Tokyo, Japan, 1978
14. Behley J, Steinhage V, Cremers AB (2013) Laser-based segment classification using a mixture of bag-of-words. In: IROS. IEEE, pp 4195–4200
15. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. Trans Pattern Anal Mach Intell 24:509–522
16. Bilen H, Fernando B, Gavves E, Vedaldi A, Gould S (2016) Dynamic image networks for action recognition. In: Proceedings of the CVPR. IEEE, pp 3034–3042
17. Black MJ, Jepson AD (1998) Eigentracking: robust matching and tracking of articulated objects using a view-based representation. Int J Comput Vis 26:63–84
18. Bo L, Lai K, Ren X, Fox D (2011) Object recognition with hierarchical kernel descriptors. In: Proceedings of the CVPR. IEEE, pp 1729–1736
19. Bo L, Ren X, Fox D (2010) Kernel descriptors for visual recognition. In: Advances in neural information processing systems, vol 23. Curran Associates, Inc., pp 244–252
20. Bo L, Ren X, Fox D (2011) Depth kernel descriptors for object recognition. In: IROS. IEEE, pp 821–826
21. Bo L, Ren X, Fox D (2013) Unsupervised feature learning for RGB-D based object recognition. In: Desai J, Dudek G, Khatib O, Kumar V (eds) Experimental robotics. Springer, Heidelberg, pp 387–402

22. Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. Trans Pattern Anal Mach Intell 23:257–267
23. Bourlard H, Kamp Y (1988) Auto-association by multilayer perceptrons and singular value decomposition. Biol Cybern 59:291–294
24. Bregonzio M, Gong S, Xiang T (2009) Recognising action as clouds of space-time interest points. In: Proceedings of the CVPR. IEEE, pp 1948–1955
25. Bro R, Acar E, Kolda TG (2008) Resolving the sign ambiguity in the singular value decomposition. J Chemometr 22:135–140
26. Brock A, Lim T, Ritchie J, Weston N (2016) Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236
27. Bronstein A, Bronstein M, Ovsjanikov M (2010) 3D features, surface descriptors, and object descriptors. Imaging Anal Appl 3D:1–27
28. Bronstein AM, Bronstein MM, Guibas LJ, Ovsjanikov M (2011) Shape google: geometric words and expressions for invariant shape retrieval. Trans Graph 30:1
29. Bronstein MM, Kokkinos I (2010) Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Proceedings of the CVPR. IEEE, pp 1704–1711
30. Caba Heilbron F, Escorcia V, Ghanem B, Carlos Niebles J (2015) Activitynet: a large-scale video benchmark for human activity understanding. In: Proceedings of the CVPR. IEEE, pp 961–970
31. Calli B, Singh A, Walsman A, Srinivasa S, Abbeel P, Dollar AM (2015) The ycb object and model set: towards common benchmarks for manipulation research. In: ICAR. IEEE, pp 510–517
32. Cao L, Liu Z, Huang TS (2010) Cross-dataset action detection. In: Proceedings of the CVPR. IEEE, pp 1998–2005
33. Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the CVPR. IEEE, pp 4724–4733
34. Chakraborty B, Holte MB, Moeslund TB, Gonzàlez J (2012) Selective spatio-temporal interest points. Comput Vis Image Underst 116:396–410
35. Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H et al (2015) Shapenet: an information-rich 3D model repository. arXiv preprint arXiv:1512.03012
36. Chen DY, Tian XP, Shen YT, Ouhyoung M (2003) On visual similarity based 3D model retrieval. In: Computer graphics forum. Wiley Online Library, pp 223–232
37. Chen H, Bhanu B (2007) 3D free-form object recognition in range images using local surface patches. Pattern Recogn Lett 28:1252–1262
38. Cheng G, Zhou P, Han J (2016) RIFD-CNN: rotation-invariant and fisher discriminative convolutional neural networks for object detection. In: Proceedings of the CVPR. IEEE, pp 2884–2893
39. Cheung W, Hamarneh G (2007) N-SIFT: N-dimensional scale invariant feature transform for matching medical images. In: 2007 4th IEEE international symposium on biomedical imaging: from nano to macro. IEEE, pp 720–723
40. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
41. Choi S, Zhou QY, Miller S, Koltun V (2016) A large dataset of object scans. arXiv:1602.02481
42. Clevert DA, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289
43. Cocosco CA, Kollokian V, Kwan RKS, Pike GB, Evans AC (1997) Brainweb: online interface to a 3D MRI simulated brain database. In: NeuroImage. Citeseer
44. Cooijmans T, Ballas N, Laurent C, Gülçehre Ç, Courville A (2016) Recurrent batch normalization. arXiv preprint arXiv:1603.09025
45. Couprie C (2012) Multi-label energy minimization for object class segmentation. In: EUSIPCO. IEEE, pp 2233–2237
46. Couprie C, Farabet C, Najman L, LeCun Y (2013) Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572
47. Dai A, Chang AX, Savva M, Halber M, Funkhouser T, Nießner M (2017) Scannet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the CVPR. IEEE, pp 5828–5839
48. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the CVPR. IEEE, pp 886–893
49. Darom T, Keller Y (2012) Scale-invariant features for 3-D mesh models. IEEE Trans Image Process 21:2758–2769
50. Deng L, Yang M, Li T, He Y, Wang C (2019) RFBNet: deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. arXiv preprint arXiv:1907.00135
51. Deng Z, Todorovic S, Jan Latecki L (2015) Semantic segmentation of RGBD images with mutex constraints. In: ICCV. IEEE, pp 1733–1741
52. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: Workshop on visual surveillance and performance evaluation of tracking and surveillance. IEEE, pp 65–72
53. Dolz J, Desrosiers C, Ayed IB (2017) 3D fully convolutional networks for subcortical segmentation in MRI: a large-scale study. NeuroImage 170:456–470
54. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the CVPR. IEEE, pp 2625–2634
55. Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the CVPR. IEEE, pp 1110–1118
56. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV. IEEE, pp 2650–2658
57. Eitel A, Springenberg JT, Spinello L, Riedmiller M, Burgard W (2015) Multimodal deep learning for robust RGB-D object recognition. In: IROS. IEEE, pp 681–687
58. ElNaghy H, Hamad S, Khalifa ME (2013) Taxonomy for 3D content-based object retrieval methods. Int J Res Rev Appl Sci 14:412–446
59. Endres F, Hess J, Engelhard N, Sturm J, Cremers D, Burgard W (2012) An evaluation of the RGB-D slam system. In: ICRA. IEEE, pp 1691–1696
60. Endres F, Hess J, Sturm J, Cremers D, Burgard W (2014) 3-d mapping with an RGB-D camera. Trans Robot 30:177–187
61. Engelcke M, Rao D, Wang DZ, Tong CH, Posner I (2017) Vote3deep: fast object detection in 3D point clouds using efficient convolutional neural networks. In: ICRA. IEEE, pp 1355–1361
62. Fan Y, Qian Y, Xie FL, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Fifteenth annual conference of the international speech communication association
63. Farabet C, Couprie C, Najman L, LeCun Y (2012) Scene parsing with multiscale feature learning, purity trees, and optimal covers. In: Proceedings of the ICML. Omnipress, pp 1857–1864
64. Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. Trans Pattern Anal Mach Intell 35:1915–1929
65. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the CVPR. IEEE, pp 1933–1941

66. Fernando B, Gavves S, Mogrovejo O, Antonio J, Ghodrati A, Tuytelaars T (2015) Modeling video evolution for action recognition. In: Proceedings of the CVPR. IEEE, pp 5378–5387
67. Firman M (2016) RGBD datasets: past, present and future. In: Proceedings of the CVPRW. IEEE, pp 19–31
68. Flint A, Dick A, Van Den Hengel A (2007) Thrift: local 3D structure recognition. In: DICTA. IEEE, pp 182–188
69. Frome A, Huber D, Kolluri R, Bülow T, Malik J (2004) Recognizing objects in range data using regional point descriptors. In: Proceedings of the ECCV. Springer, pp 224–237
70. Gao J, Sun C, Yang Z, Nevatia R (2017) Tall: temporal activity localization via language query. In: ICCV. IEEE, pp 5267–5275
71. Gao Y, Dai Q, Zhang NY (2010) 3D model comparison using spatial structure circular descriptor. Pattern Recognit 43:1142–1151
72. Garcia N (2018) Temporal aggregation of visual features for large-scale image-to-video retrieval. In: Proceedings of the 2018 ACM on international conference on multimedia retrieval. ACM, pp 489–492
73. Garcia N, Vogiatzis G (2017) Dress like a star: Retrieving fashion products from videos. In: ICCVW. IEEE, pp 2293–2299
74. Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J (2017) A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857
75. Geiger A (2012) Are we ready for autonomous driving? The kitti vision benchmark suite. In: Proceedings of the CVPR. IEEE, pp 3354–3361
76. Georgiou T, Schmitt S, Olhofer M, Liu Y, Bäck T, Lew, M (2018) Learning fluid flows. In: IJCNN. IEEE, pp 1–8
77. Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with LSTM recurrent networks. J Mach Learn Res 3:115–143
78. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: AISTATS, pp 315–323. PMLR
79. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT Press, Cambridge
80. Goodfellow IJ, Warde-Farley D, Mirza M, Courville A, Bengio Y (2013) Maxout networks. In: Proceedings of the ICML. Omnipress, pp III–1319–III–1327
81. Goyal R, Kahou SE, Michalski V, Materzynska J, Westphal S, Kim H, Haenel V, Fruend I, Yianilos P, Mueller-Freitag M, et al. (2017) The "something something" video database for learning and evaluating visual common sense. In: ICCV. IEEE, p 3
82. Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2017) LSTM: a search space odyssey. Trans Neural Netw Learn Syst 28:2222–2232
83. Guo W, Hu W, Liu C, Lu T (2019) 3D object recognition from cluttered and occluded scenes with a compact local feature. Mach Vis Appl 30:763–783
84. Guo Y, Bennamoun M, Sohel F, Lu M, Wan J (2014) 3D object recognition in cluttered scenes with local surface features: a survey. Trans Pattern Anal Mach Intell pp 2270–2287
85. Guo Y, Liu Y, Georgiou T, Lew MS (2018) A review of semantic segmentation using deep neural networks. Int J Multi Inf Retrieval 7:87–93
86. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
87. Guo Y, Sohel F, Bennamoun M, Lu M, Wan J (2013) Rotational projection statistics for 3D local surface description and object recognition. Int J Comput Vis 105:63–86
88. Guo Y, Sohel F, Bennamoun M, Wan J, Lu M (2015) A novel local surface feature for 3D object recognition under clutter and occlusion. Inf Sci 293:196–213
89. Guo Y, Sohel FA, Bennamoun M, Lu M, Wan J (2013) TriSI: a distinctive local surface descriptor for 3D modeling and object recognition. In: GRAPP/IVAPP, pp 86–93
90. Gupta S, Arbeláez P, Girshick R, Malik J (2015) Indoor scene understanding with RGB-D images: bottom-up segmentation, object detection and semantic segmentation. Int J Comput Vis 112:133–149
91. Gupta S, Arbelaez P, Malik J (2013) Perceptual organization and recognition of indoor scenes from RGB-D images. In: Proceedings of the CVPR. IEEE, pp 564–571
92. Gupta S, Girshick R, Arbeláez P, Malik J (2014) Learning rich features from RGB-D images for object detection and segmentation. In: Proceedings of the ECCV. Springer, pp 345–360
93. Hadfield S, Lebeda K, Bowden R (2017) Hollywood 3D: what are the best 3D features for action recognition? Int J Comput Vis 121:95–110
94. Handa A, Patraucean V, Badrinarayanan V, Stent S, Cipolla R (2016) Understanding real world indoor scenes with synthetic data. In: Proceedings of the CVPR. IEEE, pp 4077–4085
95. Harris C, Stephens M (1988) A combined corner and edge detector. In: Alvey vision conference. Citeseer, pp 10–5244
96. Hassner T (2013) A critical review of action recognition benchmarks. In: Proceedings of the CVPRW. IEEE, pp 245–250
97. Hazirbas C, Ma L, Domokos C, Cremers D (2016) Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In: ACCV. Springer, pp 213–228
98. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. IEEE, pp 1026–1034
99. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the CVPR. IEEE, pp 770–778
100. Hegde V, Zadeh R (2016) Fusionnet: 3d object classification using multiple data representations. arXiv preprint arXiv:1607.05695
101. Herath S, Harandi M, Porikli F (2017) Going deeper into action recognition: a survey. Image Vis Comput 60:4–21
102. Hermans A, Floros G, Leibe B (2014) Dense 3D semantic mapping of indoor scenes from RGB-D images. In: ICRA. IEEE, pp 2631–2638
103. Hinterstoisser S, Holzer S, Cagniart C, Ilic S, Konolige K, Navab N, Lepetit V (2011) Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: ICCV. IEEE, pp 858–865
104. Hinterstoisser S, Lepetit V, Ilic S, Holzer S, Bradski G, Konolige K, Navab N (2012) Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: ACCV. Springer, pp 548–562
105. Hinterstoisser S, Lepetit V, Rajkumar N, Konolige K (2016) Going further with point pair features. In: Proceedings of the ECCV. Springer, pp 834–848
106. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
107. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
108. Hinton GE, Sejnowski TJ (1986) Learning and releaming in Boltzmann machines. In: Parallel distributed processing: explorations in the microstructure of cognition, vol 1, p 2
109. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780
110. Höft N, Schulz H, Behnke S (2014) Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. In: Joint German/Austrian conference on artificial intelligence. Springer, pp 80–85
111. Holmes DR, Workman EL, Robb RA (2005) The NLM-Mayo image collection: common access to uncommon data. In: MICCAI workshop

112. Horn BKP (1984) Extended Gaussian images. In: Proceedings, pp 1671–1686
113. Hua BS, Pham QH, Nguyen DT, Tran MK, Yu LF, Yeung SK (2016) Scenenn: a scene meshes dataset with annotations. In: 3DV
114. Huang G, Liu Z, Weinberger KQ, van der Maaten L (2017) Densely connected convolutional networks. In: Proceedings of the CVPR. IEEE, pp 2261–2269
115. Huang L, Yang D, Lang B, Deng J (2018) Decorrelated batch normalization. In: Proceedings of the CVPR. IEEE, pp 791–800
116. Idrees H, Zamir AR, Jiang YG, Gorban A, Laptev I, Sukthankar R, Shah M (2017) The thumos challenge on action recognition for videos "in the wild". Comput Vis Image Underst 155:1–23
117. Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I (2017) Deep learning advances in computer vision with 3D data: a survey. ACM Comput Surv 50:20
118. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the ICML, pp 448–456. Omnipress
119. Janoch A, Karayev S, Jia Y, Barron JT, Fritz M, Saenko K, Darrell T (2013) A category-level 3D object dataset: putting the kinect to work. In: Fossati A, Gall J, Grabner H, Ren X, Konolige K (eds) Consumer depth cameras for computer vision. Springer, Berlin, pp 141–165
120. Jarrett K, Kavukcuoglu K, LeCun Y, et al. (2009) What is the best multi-stage architecture for object recognition? In: ICCV. IEEE, pp 2146–2153
121. Ji S, Xu W, Yang M, Yu K (2013) 3D convolutional neural networks for human action recognition. Trans Pattern Anal Mach Intell 35:221–231
122. Jiang Y, Moseson S, Saxena A (2011) Efficient grasping from RGBD images: learning using a new rectangle representation. In: ICRA. IEEE, pp 3304–3311
123. Jiang YG, Wu Z, Wang J, Xue X, Chang SF (2018) Exploiting feature and class relationships in video categorization with regularized deep neural networks. Trans Pattern Anal Mach Intell 40:352–364
124. Jin X, Xu C, Feng J, Wei Y, Xiong J, Yan S (2016) Deep learning with s-shaped rectified linear activation units. In: AAAI conference on artificial intelligence, pp 1737–1743
125. Johnson AE, Hebert M (1998) Surface matching for object recognition in complex three-dimensional scenes. Image Vis Comput 16:635–651
126. Johnson AE, Hebert M (1999) Using spin images for efficient object recognition in cluttered 3D scenes. Trans Pattern Anal Mach Intell 21:433–449
127. Kadir T, Brady M (2003) Scale saliency: a novel approach to salient feature and scale selection. In: VIE, pp 25–28. IET
128. Kang SM, Wildes RP (2016) Review of action recognition and detection methods. arXiv preprint arXiv:1610.06906
129. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the CVPR. IEEE, pp 1725–1732
130. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P et al. (2017) The kinetics human action video dataset. arXiv preprint arXiv:1705.06950
131. Ke Y, Sukthankar R, Hebert M (2005) Efficient visual event detection using volumetric features. In: ICCV. IEEE, pp 166–173
132. Kerl C, Sturm J, Cremers D (2013) Dense visual slam for RGB-D cameras. In: IROS. IEEE, pp 2100–2106
133. Khan SH, Bennamoun M, Sohel F, Togneri R (2014) Geometry driven semantic labeling of indoor scenes. In: Proceedings of the ECCV. Springer, pp 679–694
134. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. In: Advances in neural information processing systems, vol 30. Curran Associates, Inc., pp 971–980
135. Klaser A, Marszałek M, Schmid C (2008) A spatio-temporal descriptor based on 3D-gradients. In: BMVC, pp 275–1. BMVA Press
136. Knopp J, Prasad M, Willems G, Timofte R, Van Gool L (2010) Hough transform and 3D surf for robust three dimensional classification. In: Proceedings of the ECCV. Springer, pp 589–602
137. Koenderink JJ, van Doorn AJ (1987) Representation of local geometry in the visual system. Biol Cybern 55:367–375
138. Koppula HS, Anand A, Joachims T, Saxena A (2011) Semantic labeling of 3D point clouds for indoor scenes. In: Advances in neural information processing systems, vol 24. Curran Associates, Inc., pp 244–252
139. Kovashka A, Grauman K (2010) Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Proceedings of the CVPR. IEEE, pp 2046–2053
140. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, vol 25. Curran Associates, Inc., pp 1097–1105
141. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: ICCV. IEEE, pp 2556–2563
142. Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset. In: ICRA. IEEE, pp 1817–1824
143. Lai K, Bo L, Ren X, Fox D (2013) RGB-D object recognition: features, algorithms, and a large scale benchmark. In: Consumer depth cameras for computer vision. Springer, pp 167–192
144. Laptev I (2005) On space-time interest points. Int J Comput Vis 64:107–123
145. Laptev I, Caputo B, Schüldt C, Lindeberg T (2007) Local velocity-adapted motion events for spatio-temporal recognition. Comput Vis Image Underst 108:207–229
146. Laptev I, Lindeberg T (2004) Velocity adaptation of space-time interest points. In: ICPR. IEEE, pp 52–56
147. Laptev I, Lindeberg T (2006) Local descriptors for spatio-temporal recognition. In: MacLean WJ (ed) Spatial coherence for visual motion analysis. Springer, Berlin, pp 91–103
148. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Proceedings of the CVPR. IEEE, pp 1–8
149. Lara López G, Pena Pérez Negrón A, De Antonio Jiménez A, Ramírez Rodríguez J, Imbert Paredes R (2017) Comparative analysis of shape descriptors for 3D objects. Multimed Tools Appl 76:6993–7040
150. Laurent C, Pereyra G, Brakel P, Zhang Y, Bengio Y (2016) Batch normalized recurrent neural networks. In: ICASSP. IEEE, pp 2657–2661
151. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems, pp 396–404
152. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings 86(11):2278–2324
153. Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: AISTATS. PMLR, pp 562–570
154. Li B, Lu Y, Li C, Godil A, Schreck T, Aono M, Burtscher M, Fu H, Furuya T, Johan H, et al. (2014) Shrec'14 track: extended large scale sketch-based 3D shape retrieval. In: Eurographics workshop on 3DOR, pp 121–130
155. Li B, Zhang T, Xia T (2016) Vehicle detection from 3D lidar using fully convolutional network. arXiv preprint arXiv:1608.07916

156. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3D points. In: Proceedings of the CVPRW. IEEE, pp 9–14
157. Li Y, Xia R, Huang Q, Xie W, Li X (2017) Survey of spatio-temporal interest point detection algorithms in video. IEEE Access 5:10323–10331
158. Li Y, Xia R, Xie W (2018) A unified model of appearance and motion of video and its application in stip detection. Signal Image Video Process 12:403–410
159. Li Z, Gan Y, Liang X, Yu Y, Cheng H, Lin L (2016) Lstm-cf: Unifying context modeling and fusion with LSTMs for RGB-D scene labeling. In: Proceedings of the ECCV. Springer, pp 541–557
160. Li Z, Gan Y, Liang X, Yu Y, Cheng H, Lin L (2016) RGB-D scene labeling with long short-term memorized fusion model. arXiv preprint arXiv:1604.05000
161. Lin G, Milan A, Shen C, Reid I (2017) Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the CVPR. IEEE
162. Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400
163. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Proceedings of the ECCV. Springer, pp 740–755
164. Liu J, Shahroudy A, Xu D, Wang G (2016) Spatio-temporal LSTM with trust gates for 3D human action recognition. In: Proceedings of the ECCV. Springer, pp 816–833
165. Liu Y, Guo Y, Georgiou T, Lew MS (2018) Fusion that matters: convolutional fusion networks for visual recognition. Multimed Tools Appl 77:1–28
166. Lo TWR, Siebert JP (2009) Local feature extraction and matching on range images: 2.5 d SIFT. Comput Vis Image Underst 113:1235–1250
167. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the CVPR. IEEE, pp 3431–3440
168. Lowe DG (1999) Object recognition from local scale-invariant features. In: ICCV. IEEE, pp 1150–1157
169. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
170. Lucas BD, Kanade T et al (1981) An iterative image registration technique with an application to stereo vision. In: IJCAI. Vancouver, BC, Canada
171. Luong MT, Sutskever I, Le QV, Vinyals O, Zaremba W (2014) Addressing the rare word problem in neural machine translation. arXiv preprint arXiv:1410.8206
172. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML. Omnipress, p 3
173. Maes C, Fabry T, Keustermans J, Smeets D, Suetens P, Vandermeulen D (2010) Feature detection on 3D face surfaces for pose normalisation and recognition. In: BTAS. IEEE, pp 1–6
174. Marcos D, Volpi M, Tuia D (2016) Learning rotation invariant convolutional filters for texture classification. In: ICPR. IEEE, pp 2012–2017
175. Marszalek M, Laptev I, Schmid C (2009) Actions in context. In: Proceedings of the CVPR. IEEE, pp 2929–2936
176. Masci J, Meier U, Cireşan D, Schmidhuber J (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: ICANN. Springer, pp 52–59
177. Matikainen P, Hebert M, Sukthankar R (2009) Trajectons: action recognition through the motion analysis of tracked features. In: ICCVW. IEEE, pp 514–521
178. Matsuda T, Furuya T, Ohbuchi R (2015) Lightweight binary voxel shape features for 3D data matching and retrieval. In: International conference on multimedia big data. IEEE, pp 100–107
179. Maturana D, Scherer S (2015) Voxnet: A 3D convolutional neural network for real-time object recognition. In: IROS. IEEE, pp 922–928
180. McCormac J, Handa A, Leutenegger S, Davison AJ (2016) Scenenet RGB-D: 5m photorealistic images of synthetic indoor trajectories with ground truth. arXiv preprint arXiv:1612.05079
181. Memisevic R, Hinton G (2007) Unsupervised learning of image transformations. In: Proceedings of the CVPR. IEEE, pp 1–8
182. Messing R, Pal C, Kautz H (2009) Activity recognition using the velocity histories of tracked keypoints. In: ICCV. IEEE, pp 104–111
183. Mikolajczyk K, Schmid C (2005) A performance evaluation of local descriptors. Trans Pattern Anal Mach Intell 27:1615–1630
184. Mokhtarian F, Khalili N, Yuen P (2001) Multi-scale free-form 3D object recognition using 3D models. Image Vis Comput 19:271–281
185. Monfort M, Andonian A, Zhou B, Ramakrishnan K, Bargal SA, Yan Y, Brown L, Fan Q, Gutfreund D, Vondrick C et al. (2019) Moments in time dataset: one million videos for event understanding. Trans Pattern Anal Mach Intell 1–1
186. Müller AC, Behnke S (2014) Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images. In: ICRA. IEEE, pp 6232–6237
187. Mur-Artal R, Tardós JD (2017) Orb-slam2: an open-source slam system for monocular, stereo, and RGB-D cameras. Trans Robot 33:1255–1262
188. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the ICML. Omnipress, pp 807–814
189. Nascimento ER, Oliveira GL, Vieira AW, Campos MF (2013) On the development of a robust, fast and lightweight keypoint descriptor. Neurocomputing 120:141–155
190. Ng JYH, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: deep networks for video classification. In: Proceedings of the CVPR. IEEE, pp 4694–4702
191. Ngiam J, Chen Z, Koh PW, Ng AY (2011) Learning deep energy models. In: Proceedings of the ICML. Omnipress, pp 1105–1112
192. Ni D, Chui YP, Qu Y, Yang X, Qin J, Wong TT, Ho SS, Heng PA (2009) Reconstruction of volumetric ultrasound panorama based on improved 3D SIFT. Comput Med Imaging Graph 33:559–566
193. Niebles JC, Wang H, Fei-Fei L (2008) Unsupervised learning of human action categories using spatial-temporal words. Int J Comput Vis 79:299–318
194. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: ICCV. IEEE, pp 1520–1528
195. Novatnack J, Nishino K (2008) Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In: Proceedings of the ECCV. Springer, pp 440–453
196. Nowak E, Jurie F, Triggs B (2006) Sampling strategies for bag-of-features image classification. In: Proceedings of the ECCV. Springer, pp 490–503
197. Oikonomopoulos A, Patras I, Pantic M (2005) Spatiotemporal salient points for visual recognition of human actions. Trans Syst Man Cybern B (Cybern) 36:710–719
198. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Trans Pattern Anal Mach Intell 24:971–987
199. Oliver NM, Rosario B, Pentland AP (2000) A bayesian computer vision system for modeling human interactions. Trans Pattern Anal Mach Intell 22:831–843
200. Oreifej O, Liu Z (2013) Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences. In: Proceedings of the CVPR. IEEE, pp 716–723

201. Osada R, Funkhouser T, Chazelle B, Dobkin D (2002) Shape distributions. Trans Graph 21:807–832
202. Park SJ, Hong KS, Lee S (2017) Rdfnet: RGB-D multi-level residual feature fusion for indoor semantic segmentation. In: ICCV. IEEE, pp 4990–4999
203. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28:976–990
204. Poultney C, Chopra S, Cun YL et al. (2007) Efficient learning of sparse representations with an energy-based model. In: Advances in neural information processing systems, pp 1137–1144
205. Qi CR, Liu W, Wu C, Su H, Guibas LJ (2017) Frustum pointnets for 3D object detection from RGB-D data. arXiv preprint arXiv:1711.08488
206. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the CVPR. IEEE
207. Qi CR, Su H, Nießner M, Dai A, Yan M, Guibas LJ (2016) Volumetric and multi-view CNNs for object classification on 3D data. In: Proceedings of the CVPR. IEEE, pp 5648–5656
208. Qi X, Liao R, Jia J, Fidler S, Urtasun R (2017) 3D graph neural networks for RGBD semantic segmentation. In: ICCV. IEEE, pp 5199–5208
209. Quadros A, Underwood JP, Douillard B (2013) Sydney urban objects dataset. http://www.acfr.usyd.edu.au/papers/SydneyUrbanObjectsDataset.shtml
210. Quan S, Ma J, Ma T, Hu F, Fang B (2018) Representing local shape geometry from multi-view silhouette perspective: a distinctive and robust binary 3D feature. Signal Process Image Commun 65:67–80
211. Rahmani H, Mahmood A, Huynh D, Mian A (2016) Histogram of oriented principal components for cross-view action recognition. Trans Pattern Anal Mach Intell 38:2430–2443
212. Rahmani H, Mahmood A, Huynh DQ, Mian A (2014) Hopc: histogram of oriented principal components of 3D pointclouds for action recognition. In: Proceedings of the ECCV. Springer, pp 742–757
213. Regneri M, Rohrbach M, Wetzel D, Thater S, Schiele B, Pinkal M (2013) Grounding action descriptions in videos. Trans ACL 1:25–36
214. Ren M, Liao R, Urtasun R, Sinz FH, Zemel RS (2016) Normalizing the normalizers: comparing and extending network normalization schemes. arXiv preprint arXiv:1611.04520
215. Ren X, Bo L, Fox D (2012) Rgb-(d) scene labeling: features and algorithms. In: Proceedings of the CVPR. IEEE, pp 2759–2766
216. Rennie C, Shome R, Bekris KE, De Souza AF (2016) A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. Robot Autom Lett 1:1179–1185
217. Richter SR, Vineet V, Roth S, Koltun V (2016) Playing for data: Ground truth from computer games. In: Proceedings of the ECCV. Springer, pp 102–118
218. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y (2011) Contractive auto-encoders: explicit invariance during feature extraction. In: Proceedings of the ICML. Omnipress, pp 833–840
219. Rios-Cabrera R, Tuytelaars T (2013) Discriminatively trained templates for 3D object detection: a real time scalable approach. In: ICCV. IEEE, pp 2048–2055
220. Rodriguez MD, Ahmed J, Shah M (2008) Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: Proceedings of the CVPR. IEEE, pp 1–8
221. Rohr K (1997) On 3D differential operators for detecting point landmarks. Image Vis Comput 15:219–233
222. Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the CVPR. IEEE, pp 3234–3243
223. Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. In: Proceedings of the ECCV. Springer, pp 430–443
224. Rublee E, Rabaud V, Konolige K, Bradski GR (2011) Orb: An efficient alternative to SIFT or SURF. In: ICCV, vol 11. Citeseer, p 2
225. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115:211–252
226. Rustamov RM (2007) Laplace-beltrami eigenfunctions for deformation invariant shape representation. In: Proceedings of the ESGP. Eurographics Association, pp 225–233
227. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: ICRA. IEEE, pp 3212–3217
228. Rusu RB, Blodow N, Marton ZC, Beetz M (2008) Aligning point cloud views using persistent feature histograms. In: IROS. IEEE, pp 3384–3391
229. Saeed Mian A, Bennamoun M, Owens R (2004) Automated 3D model-based free-form object recognition. Sens Rev 24:206–215
230. Salakhutdinov R (2008) Learning and evaluating boltzmann machines. Technical Report, Technical Report UTML TR 2008-002, Department of Computer Science, University of Toronto
231. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: AISTATS. PMLR, pp 448–455
232. Salakhutdinov R, Larochelle H (2010) Efficient learning of deep boltzmann machines. In: AISTATS. PMLR, pp 693–700
233. Salimans T, Kingma DP (2016) Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: Advances in neural information processing systems, vol 29. Curran Associates, Inc., pp 901–909
234. Saputra MRU, Markham A, Trigoni N (2018) Visual slam and structure from motion in dynamic environments: a survey. CSUR p. 37
235. Savarese S, Fei-Fei L (2007) 3D generic object categorization, localization and pose estimation. In: ICCV. IEEE, pp 1–8
236. Savva M, Chang AX, Hanrahan P (2015) Semantically-enriched 3D models for common-sense knowledge. In: Proceedings of the CVPRW. IEEE, pp 24–31
237. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: ICPR. IEEE, pp 32–36
238. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. Trans Signal Process 45:2673–2681
239. Scovanner P, Ali S, Shah M (2007) A 3-dimensional SIFT descriptor and its application to action recognition. In: Proceedings of the ICM, pp 357–360. ACM
240. Sebe N, Lew MS, Huang TS (2004) The state-of-the-art in human–computer interaction. In: International workshop on computer vision in human–computer interaction. Springer, pp 1–6
241. Sedaghat N, Zolfaghari M, Amiri E, Brox T (2016) Orientation-boosted voxel nets for 3D object recognition. arXiv preprint arXiv:1604.03351
242. Shahroudy A, Liu J, Ng TT, Wang G (2016) Ntu rgb+ d: a large scale dataset for 3D human activity analysis. In: Proceedings of the CVPR. IEEE, pp 1010–1019
243. Shechtman E, Irani M (2005) Space-time behavior based correlation. In: Proceedings of the CVPR. IEEE, pp 405–412
244. Shechtman E, Irani M (2007) Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them? Trans Pattern Anal Mach Intell 29:2045–2056
245. Shi B, Bai S, Zhou Z, Bai X (2015) Deeppano: Deep panoramic representation for 3-d shape recognition. Signal Process Lett 22:2339–2343

246. Shih JL, Lee CH, Wang JT (2007) A new 3D model retrieval approach based on the elevation descriptor. Pattern Recognit 40:283–295
247. Shilane P, Min P, Kazhdan M, Funkhouser T (2004) The princeton shape benchmark. In: Shape modeling applications, 2004. Proceedings. IEEE, pp 167–178
248. Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. J Stat Plan Inference 90(2):227–244
249. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: Proceedings of the CVPR. IEEE, pp 1297–1304
250. Silberman N, Fergus R (2011) Indoor scene segmentation using a structured light sensor. In: ICCVW. IEEE, pp 601–608
251. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from RGBD images. In: Proceedings of the ECCV. Springer, pp 746–760
252. Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034
253. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, vol 27. Curran Associates, Inc., pp 568–576
254. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
255. Singh A, Sha J, Narayan KS, Achim T, Abbeel P (2014) Bigbird: a large-scale 3D database of object instances. In: ICRA. IEEE, pp 509–516
256. Singh T, Vishwakarma DK (2019) Video benchmarks of human action datasets: a review. Artif Intell Rev 52:1107–1154
257. Socher R, Huval B, Bath BP, Manning CD, Ng AY (2012) Convolutional-recursive deep learning for 3d object classification. In: Advances in neural information processing systems. Curran Associates, Inc., p 8
258. Song S, Lichtenberg SP, Xiao J (2015) Sun RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the CVPR. IEEE, pp 567–576
259. Song S, Yu F, Zeng A, Chang AX, Savva M, Funkhouser T (2017) Semantic scene completion from a single depth image. In: Proceedings of the CVPR. IEEE, pp 1746–1754
260. Song Y, Morency LP, Davis R (2013) Action recognition by hierarchical sequence summarization. In: Proceedings of the CVPR. IEEE, pp 3562–3569
261. Soomro K, Zamir AR, Shah M (2012) Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402
262. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. arXiv preprint arXiv:1505.00387
263. Strasdat H, Davison AJ, Montiel JM, Konolige K (2011) Double window optimisation for constant time visual slam. In: ICCV. IEEE, pp 2352–2359
264. Stückler J, Biresev N, Behnke S (2012) Semantic mapping using object-class segmentation of RGB-D images. In: IROS. IEEE, pp 3005–3010
265. Stückler J, Waldvogel B, Schulz H, Behnke S (2015) Dense real-time mapping of object-class semantics from RGB-D video. J Real-Time Image Process 10:599–609
266. Su H, Maji S, Kalogerakis E, Learned-Miller E (2015) Multi-view convolutional neural networks for 3d shape recognition. In: ICCV. IEEE, pp 945–953
267. Sun D, Roth S, Black MJ (2014) A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int J Comput Vis 106:115–137
268. Sun J, Ovsjanikov M, Guibas L (2009) A concise and provably informative multi-scale signature based on heat diffusion. In: Computer graphics forum. Wiley Online Library, pp 1383–1392
269. Sun J, Wu X, Yan S, Cheong LF, Chua TS, Li J (2009) Hierarchical spatio-temporal context modeling for action recognition. In: Proceedings of the CVPR. IEEE, pp 2004–2011
270. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI conference on artificial intelligence
271. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A, et al. (2015) Going deeper with convolutions. In: Proceedings of the CVPR. IEEE, pp 1–9
272. Tang S, Wang X, Lv X, Han TX, Keller J, He Z, Skubic M, Lao S (2012) Histogram of oriented normal vectors for object recognition with a depth sensor. In: ACCV. Springer, pp 525–538
273. Tangelder JW, Veltkamp RC (2004) A survey of content based 3D shape retrieval methods. In: Shape modeling applications, 2004. IEEE, pp 145–156
274. Taylor GW, Fergus R, LeCun Y, Bregler C (2010) Convolutional learning of spatio-temporal features. In: Proceedings of the ECCV. Springer, pp 140–153
275. Teichman A, Levinson J, Thrun S (2011) Towards 3D object recognition via classification of arbitrary object tracks. In: ICRA. IEEE, pp 4034–4041
276. Teichman A, Thrun S (2012) Tracking-based semi-supervised learning. Int J Robot Res 31:804–818
277. Tejani A, Kouskouridas R, Doumanoglou A, Tang D, Kim TK (2017) Latent-class hough forests for 6 DoF object pose estimation. Trans Pattern Anal Mach Intell 40:119–132
278. Tejani A, Kouskouridas R, Doumanoglou A, Tang D, Kim TK (2018) Latent-class hough forests for 6 dof object pose estimation. Trans Pattern Anal Mach Intell 40:119–132
279. Thomee B, Huiskes MJ, Bakker E, Lew MS (2008) Large scale image copy detection evaluation. In: ICMIR. ACM, pp 59–66
280. Thomee B, Shamma DA, Friedland G, Elizalde B, Ni K, Poland D, Borth D, Li LJ (2015) The new data and new challenges in multimedia research. arXiv preprint arXiv:1503.01817
281. Tombari F, Salti S, Di Stefano L (2010) Unique signatures of histograms for local surface description. In: Proceedings of the ECCV. Springer, pp 356–369
282. Tombari F, Salti S, Di Stefano L (2011) A combined texture-shape descriptor for enhanced 3D feature matching. In: ICIP. IEEE, pp 809–812
283. Tombari F, Salti S, Di Stefano L (2013) Performance evaluation of 3D keypoint detectors. Int J Comput Vis 102:198–220
284. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: ICCV. IEEE, pp 4489–4497
285. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the CVPR. IEEE, pp 6450–6459
286. Trottier L, Gigu P, Chaib-draa B, et al. (2017) Parametric exponential linear unit for deep convolutional neural networks. In: ICMLA. IEEE, pp 207–214
287. Ulyanov D, Vedaldi A, Lempitsky V (2016) Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022
288. Valada A, Mohan R, Burgard W (2019) Self-supervised model adaptation for multimodal semantic segmentation. Int J Comput Vis
289. Varol G, Laptev I, Schmid C (2017) Long-term temporal convolutions for action recognition. Trans Pattern Anal Mach Intell 40:1510–1517
290. Vieira AW, Nascimento ER, Oliveira GL, Liu Z, Campos MF (2012) Stop: space-time occupancy patterns for 3D action recog-

nition from depth map sequences. In: Iberoamerican congress on pattern recognition. Springer, pp 252–259
291. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the ICML, pp 1096–1103. ACM
292. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
293. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the CVPR. IEEE, p 3
294. Wang A, Lu J, Wang G, Cai J, Cham TJ (2014) Multi-modal unsupervised feature learning for RGB-D scene labeling. In: Proceedings of the ECCV. Springer, pp 453–467
295. Wang C, Pelillo M, Siddiqi K (2019) Dominant set clustering and pooling for multi-view 3D object recognition. arXiv preprint arXiv:1906.01592
296. Wang DZ, Posner I, Newman P (2012) What could move? finding cars, pedestrians and bicyclists in 3D laser data. In: ICRA. IEEE, pp 4038–4044
297. Wang G, Luo P, Wang X, Lin L, et al. (2018) Kalman normalization: Normalizing internal representations across network layers. In: Advances in neural information processing systems, vol 31. Curran Associates, Inc., pp 21–31
298. Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: Proceedings of the CVPR. IEEE, pp 3169–3176
299. Wang H, Kläser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 103:60–79
300. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: ICCV. IEEE, pp 3551–3558
301. Wang J, Liu Z, Wu Y, Yuan J (2012) Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the CVPR. IEEE, pp 1290–1297
302. Wang J, Liu Z, Wu Y (2014) Learning actionlet ensemble for 3D human action recognition. Trans Pattern Anal Mach Intell 36:914–927
303. Wang J, Wang Z, Tao D, See S, Wang G (2016) Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In: Proceedings of the ECCV. Springer, pp 664–679
304. Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the CVPR. IEEE, pp 4305–4314
305. Wang L, Xiong Y, Wang Z, Qiao Y (2015) Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:1507.02159
306. Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona PO (2016) Action recognition from depth maps using deep convolutional neural networks. Trans Hum Mach Syst 46:498–509
307. Wang Y, Mori G (2011) Hidden part models for human action recognition: probabilistic versus max margin. Trans Pattern Anal Mach Intell 33:1310–1323
308. Whelan T, Salas-Moreno RF, Glocker B, Davison AJ, Leutenegger S (2016) Elasticfusion: real-time dense SLAM and light source estimation. Int J Robot Res 35:1697–1716
309. Willems G, Becker JH, Tuytelaars T, Van Gool LJ (2009) Exemplar-based action recognition in video. In: BMVC. BMVA Press, p 3
310. Willems G, Tuytelaars T, Van Gool L (2008) An efficient dense and scale-invariant spatio-temporal interest point detector. In: Proceedings of the ECCV. Springer, pp 650–663
311. Wong SF, Cipolla R (2007) Extracting spatiotemporal interest points using global information. In: ICCV. IEEE, pp 1–8
312. Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J (2016) Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in neural information processing systems, vol 29. Curran Associates, Inc., pp 82–90
313. Wu Y, He K (2018) Group normalization. In: Proceedings of the ECCV. Springer, pp 3–19
314. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D shapenets: A deep representation for volumetric shapes. In: Proceedings of the CVPR. IEEE, pp 1912–1920
315. Xia L, Aggarwal J (2013) Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: Proceedings of the CVPR. IEEE, pp 2834–2841
316. Xiao J, Owens A, Torralba A (2013) Sun3d: A database of big spaces reconstructed using sfm and object labels. In: ICCV. IEEE, pp 1625–1632
317. Xu H, He K, Sigal L, Sclaroff S, Saenko K (2018) Text-to-clip video retrieval with early fusion and re-captioning. arXiv preprint arXiv:1804.05113
318. Yamato J, Ohya J, Ishii K (1992) Recognizing human action in time-sequential images using hidden markov model. In: Proceedings of the CVPR. IEEE, pp 379–385
319. Yang J, Cao Z, Zhang Q (2016) A fast and robust local descriptor for 3D point cloud registration. Information Sciences 346:163–179
320. Yang J, Zhang Q, Xiao Y, Cao Z (2017) Toldi: an effective and robust approach for 3D local shape description. Pattern Recognit 65:175–187
321. Yang X, Tian Y (2014) Super normal vector for activity recognition using depth sequences. In: Proceedings of the CVPR. IEEE, pp 804–811
322. Yang X, Tian YL (2012) Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: Proceedings of the CVPR. IEEE, pp 14–19
323. Yeffet L, Wolf L (2009) Local trinary patterns for human action recognition. In: ICCV. IEEE, pp 492–497
324. Yu H, Yang Z, Tan L, Wang Y, Sun W, Sun M, Tang Y (2018) Methods and datasets on semantic segmentation: a review. Neurocomputing 304:82–103
325. Yu TH, Kim TK, Cipolla R (2010) Real-time action recognition by spatiotemporal semantic and structural forests. In: BMVC. BMVA Press, p 6
326. Yu W, Yang K, Bai Y, Yao H, Rui Y (2014) Visualizing and comparing convolutional neural networks. arXiv preprint arXiv:1412.6631
327. Yumer ME, Chaudhuri S, Hodgins JK, Kara LB (2015) Semantic shape editing using deformation handles. ACM Trans Graph 34:86
328. Yumer ME, Mitra NJ (2016) Learning semantic deformation flows with 3D convolutional networks. In: Proceedings of the ECCV. Springer, pp 294–311
329. Zaharescu A, Boyer E, Varanasi K, Horaud R (2009) Surface feature detection and description with applications to mesh matching. In: Proceedings of the CVPR. IEEE, pp 373–380
330. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv preprint arXiv:1409.2329
331. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Proceedings of the ECCV. Springer, pp 818–833
332. Zhang Z (2012) Microsoft kinect sensor and its effect. IEEE Multimed 19:4–10
333. Zhao R, Ali H, Van der Smagt P (2017) Two-stream RNN/CNN for action recognition in 3D videos. In: IROS. IEEE, pp 4260–4267
334. Zheng L, Yang Y, Tian Q (2017) SIFT meets CNN: a decade survey of instance retrieval. Trans Pattern Anal Mach Intell 40(5):1224–1244

335. Zhong Y (2009) Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: ICCVW. IEEE, pp 689–696
336. Zou Y, Wang X, Zhang T, Liang B, Song J, Liu H (2018) BRoPH: an efficient and compact binary descriptor for 3D point clouds. Pattern Recognit 76:522–536

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.