Literature review:

*Literature search strategy:*

* Research 3D reconstruction techniques, in particular - MVS and SfM. SfS is also of interest.
* Research traditional IBR / video interpolation techniques – as this are combined with deep-learning
* Research deep-learning architectures, in particular – convolutional networks architectures
* Research data and data formats
* Research specific to video interpolation using deep-learning – study in detail articles that have done similar research

Chapter 2 State of the Art

# 2.1 3D Reconstruction

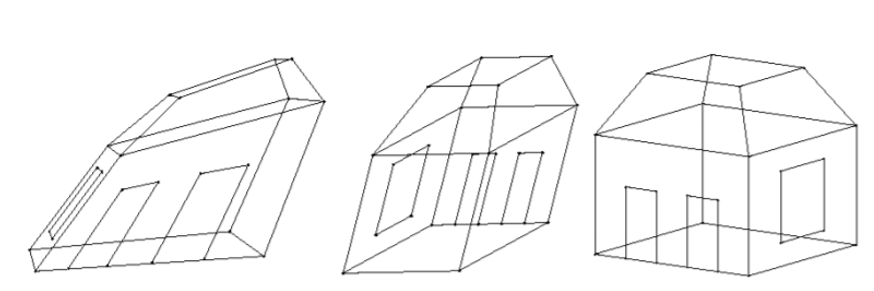
3D Reconstruction of object shapes from still images and video stream is an ongoing research topic that puzzled researches for decades. As the final aim of this paper is to improve the quality of the reconstruction and the final evaluation is also based on the reconstructed models, it is relevant to describe here the state-of-the-art of 3D Reconstruction methodologies. This will also justify the choices for the particular 3D reconstruction software used for the evaluation.

# 2.1.1 Orthographic projection

As early as 1992 (C.Tomasi and T.Kanade, 1992) obtained good results from a stream of images using orthographic, rather than perspective, projection and introduced *Factorization method*. The orthographic projection simplified processing, removing the depth dimension[[1]](#footnote-1). Examples of projective, affine and Euclidean projections are given in Figure 1. (Tomasi and Kanade, 1992) worked with affine and orthographic projections only. They decomposed the measurement matrix W (F frames, P tracked points forms 2\*F\*P matrix in 2D) into 2 matrices – R and S – representing the camera rotation and the object shape respectively plus the projection of the camera translation t along the image plane.

They were able to process input with *noisy measurements* by introducing 3x3 matrix Q (, ) and metric constraints to solve for Q.

They are also able to cope with *occlusions* by recovering position of feature points from 3 other positions of the feature (??Ulmann).



**Figure 1. Examples of Perspective, Affine and Orthographic projections** [Source: (Belongie, 2009, chapter 9)]

# 2.1.2 Perspective projection – epipolar geometry

When working in *perspective projection* several terms are useful:

*Calibrated cameras:*

First, a special case of calibrated cameras is described.

The *essential matrix E* for correspondence between 2 images was introduced by (Longuet-Higgins, 1981):

So, the same point in 2 images correspond as:

where is the position of the point in the second image, is the position of the same point in the first image and is the essential matrix – forming *epipolar constraint*. The essential matrix maps a point in the first image into a line in the second image - , which is called *epipole* (Szeliski, 2010).

Both translation and rotation of the 2nd camera – and traditionally, any subsequent cameras in a sequence of images – are taken with reference to the camera position of the first image in the sequence, i.e. the camera of the first image is the origin of the world co-ordinates and its orientation equals identity matrix.

If more than one feature point is available between the 2 images, the Essential Matrix can be determined from a series of equations: ⊗ , where ⊗ denotes point-wise multiplication, and i- is the index of the feature. The series of equations can be resolved with SVD (singular-value decomposition) algorithm. It has been shown by several researchers ((Hartley); (Torr and Murray, 1997); (Hartley and Zisserman, 2004)) that *7 point correspondences* (i.e. features) is sufficient to find the elements of the essential matrix (Szeliski, 2010).

In addition, (Hartley, 1995) suggests that the point co-ordinates should be translated and scaled to the centre of the object, so that the sum of x- and y- co-ordinates is 0 and the squared sum of both co-ordinates equals twice the number of points (, )

*Uncalibrated cameras*

The above equations describe *an ideal case*, where cameras are *perfectly* *calibrated*. The assumption of un-calibrated cameras adds an additional complexity of the calibration matrix K. The essential matrix becomes the fundamental matrix F:

Where K – is the camera calibration matrix. Or, , where e – is the *focus of expansion* and matrix is one of many possible *homographies* ((Hartley and Zisserman, 2004), (Faugeras, 1992)).

*Calibration Matrix*

While it is possible under certain constraints to convert projective reconstruction into a metric one, i.e. recover calibration matrices associated with each image – *self-calibration* (Hartley and Zisserman, 2004), most 3D reconstructions assume *pre-calibrated cameras* or images taken with a single camera with fixed intrinsic parameters.

where F is focal length, and – size of the camera sensor per pixel, and – translation of the camera centre with regard to the image. (Source: lectures CS7GV4 by Aljosa Smolic)

*Bundle adjustment*

Two images with 7 point correspondences is sufficient to estimate the Fundamental matrix, so each new image or each new point-correspondence overdetermines the system. A cost function can be introduced that aims to minimize the re-projection error. The system of equations can be solved with a non-linear method. There are two options for bundle adjustment: this can be done incrementally – as each new image is added - or at the end of the process with all images.

*Triangulation*

The last topic to discuss in the basics chapter is triangulation. This is a method to estimate depth to the object after the Fundamental matrix is known. …

The fundamental matrix and epipolar correspondence lie at the heart of SfM algorithm, as any estimation will start from finding the matrix correspondence between the 2 images.

2.1.3 Structure from Motion

The following two sections describe SfM (Structure from Motion) and MVS (Multi-view stereo) algorithms. These correspond to obtaining sparse 3D point cloud reconstruction and camera positions (SfM) and dense 3D surface reconstruction (MVS). The project will use COLMAP for 3D reconstruction, so particular implementation specific to COLMAP is specified.

The typical workflow of a SfM algorithm is as follows (Schonberger et al., 2016a)

// Marc PollefeysPollefeys M. Self-calibration and metric 3d reconstruction from uncalibrated image sequences[J]. Thesis K.u.leuven Departement Esat Afdeling Psi.phd.thesis, 1999.

//from (Smith et al., 2016):

1. *Feature extraction and feature descriptors*. Correspondence between images is found based on distinctive points, so the first step of SfM is to identify feature points and their descriptors for each image in the stream.

One of the early methods for finding the interesting points is - Autocorrelation function (ACF) (Szeliski, 2010), which finds if the point is unique in its surroundings. The following authors further expanded on ACF - (Harris and Stephens, 1988), (Lucas and Kanade, 1981), (Shi and Tomasi, 1994) etc.

Suitable features are then described in terms of their neighbourhood – this is to ensure rotation, scaling, perspective distortions, lighting changes etc invariance.

The proposed algorithms create a description of the point’s neighbourhood:

* SIFT (Lowe, 1999, 2001, 2004)
* SURF (Bay et al., 2008)
* BRIEF (Calonder et al., 2010)
* ASIFT (Morel and Yu, 2009)
* LDAHash (Strecha et al., 2012)

COLMAP uses RootSIFT for its feature extraction, which is an improved version of SIFT algorithm.

For this work, which intends to produce additional images based on deep-learning, it is possible to have blurriness in generated images and ghost artefacts, which may make feature detection more difficult or place features incorrectly.

1. *Feature Matching*

The above features are matched. The simplest approach is to test every image pair and for every feature in the first image to find the most similar feature in the second image (*similarity metric*).

This approach has computational complexity O(N2IMAGESN2FEATURES) and is prohibitive for large image collections (Schonberger et al., 2016a). There is research to improve the efficiency of the matching, for example, k-dimensional trees and ANN (Approximate Nearest Neighbour) (Smith et al., 2016).

1. *Identifying geometrically consistent matches*

Some feature matches may be excluded when they are checked for possible geometric transformations (homography, fundamental and essential matrices). If a valid transformation maps a sufficient number of features between the images, they are considered geometrically verified. RANSAC algorithm is usually used for the outlier detection (Schonberger et al., 2016a).

1. *Initialisation before reconstruction and image registration*

SfM chooses the appropriate initial pair of cameras that would represent the origin of the world co-ordinates. Typically, these will have many common features and a wide baseline (Smith et al., 2016).

Also, the order in which the images will be added is important. New images can be registered to the current model by solving the Perspective-n-Point (PnP) problem. The PnP problem involves estimating the pose of the camera for the new image and, for uncalibrated cameras, camera’s intrinsic parameters. Every new image provides additional 2D-3D correspondences. (Schonberger et al., 2016a).

1. *Triangulation*

Triangulation method is used to compute 3D space point from feature point correspondence (. Again, several different methods are proposed. (Hartley and Zisserman, 2004) describe triangulation suitable for different types of transformations (affine, projective etc). They discuss the differences between linear triangulation method (DLT, inhomogeneous) , error minimisation, Sampson approximation and solving a 6-degree polynomial. COLMAP developers ((Schonberger et al., 2016a) propose their own version of the triangulation method.

1. *Bundle Adjustment*

Bundle adjustment minimizes the reprojection error as more 2D-3D correspondences are added to the system. It performs a joint non-linear refinement of parameters Pc (Camera position and intrinsic parameters) and point positions – X. COLMAP uses the following formulae, where E – reprojection error, xj – co-ordinates of the point in image j, – loss function to down-weight the outliers and symbolises the function that converts scene points into image space ((Schonberger et al., 2016a).

The output of the SfM stage is a *sparse, unscaled 3D point cloud in arbitrary units along with camera models and poses*. This can be resolved into metric reconstruction if camera calibrations are known, or if metric parameters of some of the points are known (for example, ground-control points in case of georeferencing - (Smith et al., 2016) .

2.1.1 Multi-view stereo

Multi-view stereo (MVS) provides a complete 3D reconstruction or *dense modelling* of the object from a known sparse 3D cloud and known camera positions and intrinsic matrices. (Smith et al., 2016) summarises the review of MVS methods by As detailed by (Schonberger et al., 2016a) with reference to (Seitz et al., 2006), there is a wide variety of MVS algorithms, which can be classified into:

**TODO – add references**

1. Voxel-based methods which are 3D grids that are occupied to define the scene (for example, Seitz and Dyer, 1999)
2. Surface evolution-based methods that use iteratively evolved polygonal meshes (for example, Furukawa and Ponce, 2009)
3. Depth-map merging methods where individual depth maps showing the distance between the camera viewpoint to the 3D scene objects are combined into a single model (for example, Li et al. , 2010a)
4. Patch-based methods where collections of small patches or surfels represent the scene (for example, Lhuillier and Quan, 2005).

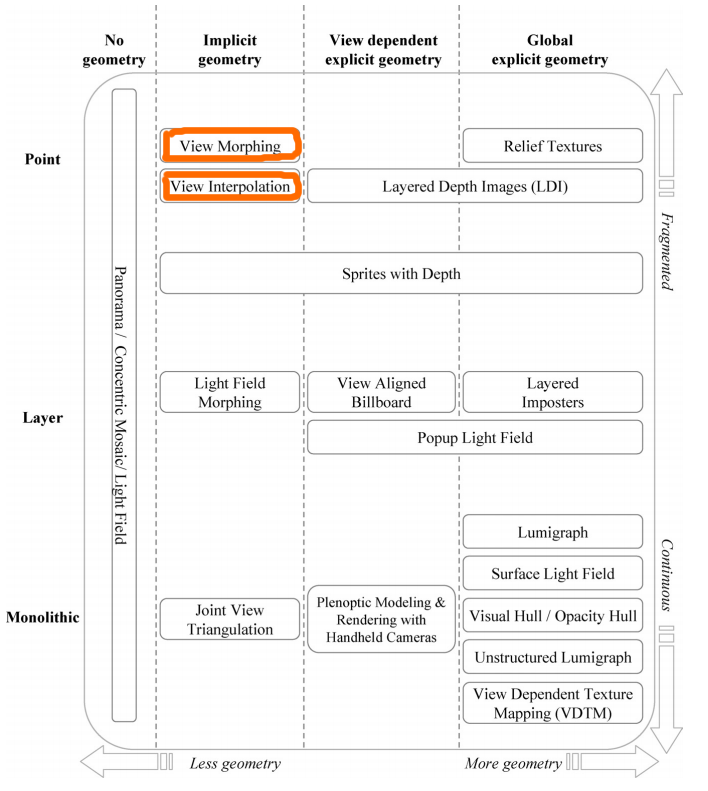
The latter method is also called PMVS and is the one utilised by COLMAP (Schonberger et al., 2016b). Schonberger suggests a new PMVS algorithm based on (Zheng et al., 2014).

The last step in Multi-view stereo is to generate a polygonal 3D mesh from dense point cloud. This can be achieved with *Poisson Surface Reconstruction* (PSR) (Kazhdan et al., 2006). Texture is then added with *Texture mapping*, which is another field of research in itself.

It can be seen that 3D reconstruction is a multi-step process, where different methodologies can be selected at every step. The particular choice of algorithms for each stage will strongly affect the accuracy of reconstruction, we have therefore specified the particular algorithms applicable to COLMAP, which will be used for 3D testing.

# 2.2 View interpolation

(Kang et al., 2007) proposes the following classification of Image-based rendering (IBR) techniques. View interpolation and view morphing are on the left of the continuum as relying on rendering with implicit geometry and acting on pixel-per-pixel basis.



# **Figure . IBR techniques classification(Kang et al., 2007)**

Motion interpolation and [view synthesis](http://apps.webofknowledge.com.elib.tcd.ie/OneClickSearch.do?product=WOS&search_mode=OneClickSearch&excludeEventConfig=ExcludeIfFromFullRecPage&colName=WOS&SID=E6lvDoNxLTSOwVpPNBU&field=TS&value=view+synthesis&uncondQuotes=true) are popular terms more recently that are also of interest.

*Single image interpolation techniques*

We will be interested here primary in the view interpolation between 2 views, but interpolation techniques are also applicable to *a single image*. These belong to a field of digital image processing and are required when images are re-sized, rotated or transformed. It is worth mentioning these interpolation techniques - as they can also be applied to the result of 2 frame interpolation. The interpolation techniques for a single image include: 2D nearest-neighbour interpolation, bilinear interpolation (2x2 neighbourhood), bicubic interpolation (4x4 neighbourhood), spline and sinc interpolation, natural neighbour interpolation using Voronoi cells, kriging based on Gaussian distribution (Wikipedia). When applied these can produce artefacts in the interpolated images: aliasing, blurring, edge halo (<https://www.cambridgeincolour.com/tutorials/image-interpolation.htm>). These can be rectified with anti-aliasing, interpolation that is “edge-aware” or “weighted edge-aware” (<http://techtidings.blogspot.com/2012/01/demosaicing-exposed-normal-edge-aware.html>).

*Interpolating between 2 views*

Coming back to the interpolation between 2 views – these belong to two broad categories:

* Spatial interpolation of image sequences, when camera position changes and the objects are static.
* Temporal interpolation of image sequences (video interpolation) – when objects in the images/ frames can move. In practice, video interpolation may combine both the moving camera and non-static objects hence incorporating special interpolation.

The simplest technique for *interpolation between 2 images* is *linear interpolation* where the intermediate pixel can be calculated at any intermediate point with:

Where and are pixel values in image 1 and 2. This method produces blurry result.

*Feature based image morphing*

An improved version of this algorithm applicable to *spatial interpolation* was proposed early on by (Chen and Williams, 1993). They worked with computer graphics images in order to improve the speed of generating CG views. Their technique first determined *pixel-by-pixel correspondences* between images and stored morph maps for further calculation. When required positions and colors of the points were linearly interpolated. As the authors worked with synthetic images - range data and the camera transformations were readily available. They were able to synthesize arbitrary intermediate points of view with bi-directional mapping. The new views only had view-independent shading.

(Seitz and Dyer, 1996) worked with natural images and assumed known camera projections. They expanded on view morphing techniques aiming to keep the shape of 3D objects. The authors first resolve the case of parallel views and prove that for parallel views with orthographic projection linear interpolation between feature points produces the correct result. For non-parallel views,   
*image reprojection* was used – this allows to move the image to a different plane using homography matrix. Their algorithm is composed of 3 steps:

1) Pre-warping – applies reverse homography camera matrices to the 2 images, to bring the images to a single plane and all 3 cameras to a single line.

2) Morph – linearly interpolate position and colors between 2 images.

3) Post-warping - apply homography of the target image camera to obtain the final view.

The work by (Seitz and Dyer, 1996) had a big influenced on subsequent research.

Before proceeding to the more recent interpolation techniques, two methods applicable to spatial view interpolation need to be described here: forward mapping and backward (inverse) mapping.

*Forward Mapping*

Forward mapping maps each pixel on the reference view(s) to the target view using some form of geometry, e.g., depth map (explicit geometry) or correspondences between views (implicit geometry). If – 2D point in the target image, – 2D point in the reference image, – point in 3D space, and – camera positions for reference and target images, and – camera projections, and – scaling factors.

The resultant pixel in the target image can be evaluated from the above equation. There is a problem with this approach – not all pixels in the target image may be populated and therefore will need to be interpolated, or at the same time as many pixels may land on the same pixel in the target image. Even after the interpolation there may still be holes due to magnification and disocclusion. (Kang et al., 2007)

Therefore, it is more traditional approach to use is the reverse of forward mapping:

*Inverse mapping*

In inverse mapping the pixel mapping in the target is found by tracing the ray from the target view back to the reference view:

Or, expressed in terms of homography :

Where defines the 2D planar perspective transformation from target screen to reference camera, is the epipole, is a scale factor and therefore defines epipolar line.

Inverse mapping ensures that there are no gaps in the target image, however, if is occluded in the reference view - the search yields no result. (Kang et al., 2007)

More recent references on view synthesis are able to work with un-calibrated cameras. (Fragneto et al., 2012) expands the work of (Seitz and Dyer, 1996), work with uncalibrated cameras and proposes a new algorithm based on interpolating homographies rather than pixel positions and colours. (Gurdan et al., 2014) combines …TODO

Navigating between camera views TODO

filing the holes after.. TODO

--

*Temporal (video) interpolation*

Frame interpolation for video is a special case of image-based rendering (IBR) where middle frames are interpolated from temporally neighbouring frames (Niklaus et al., 2017). For temporal interpolation the most common approach is to use optical flow. TODO

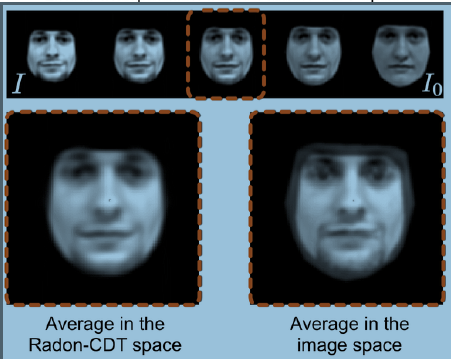
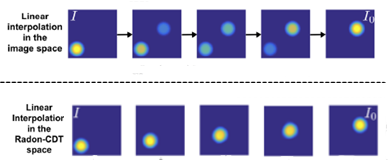
video interpolation with dense optical flow (1993)

Dense image interpolation.

based on motion estimation

---

Methods of *non-linear interpolation* that do not include deep-learning have advanced recently. For example, (Kolouri et al., 2016) proposes linear interpolation in Radon Cumulative Distribution Transform space, where the interpolated image is still linearly separable into the 2 original images. The pixel location information is encoded in transport flows (*optimal transport metric*), so each pixel and neighbourhood are considered from ‘Lagrangian’ point of view. The transform captures translation and scaling, as well as more complicated transformations – see example of capturing movement and face interpolation below.



**Fig. Comparing linear interpolation results with non-linear interpolation using Radon-CDT space (linear interpolation images at top in the left example and bottom-right in the right example (Kolouri et al., 2016)**

?? PCA for image interpolation

An alternative approach to PCA (Principal Component Analysis) for image interpolation, is *Isomap*, or *dimensionality reduction* problem. This approach works well for repetitive motion, i.e. camera rotating around a rigid object, person waving hand etc (TODO – add reference ROBASZKIEWICZ Interpolating images between video frames). The method finds feature point correspondences between the interpolated images and interpolating a curve between the data points in the feature space, before fitting the intermediate images to the curve. The method works well for preserving the original shape of the object.

As a conclusion for this section, we can summarize that that *important properties of the interpolated frame* for 3D reconstructionare*:*

* Keeping features and edges
* Correct location of features and edges
* Sharp
* No ghost artifacts
* No holes

# 2.3 Deep learning

2.3.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks have revolutionised image processing in the last decade. They were the first successful application of deep-learning architectures.

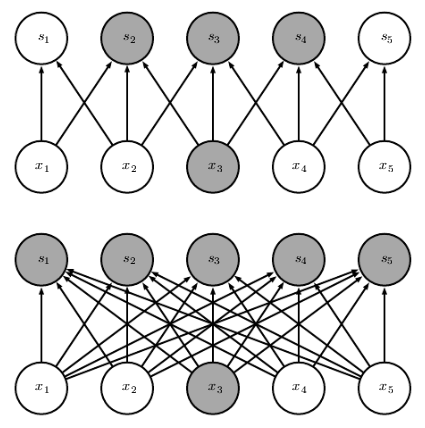
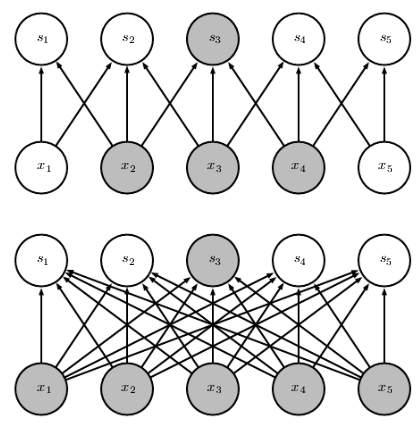
Their success in image processing is attributed to their *sparse connectivity*, which make processing images more *computationally efficient*. Also, *parameter sharing* – unlike traditional neural net where each weight is only applied once, the convolution kernel is applied to every pixel in the image, which makes it possible to extract the feature independent of the location in the image. CNNs are also *equivariant to translation* meaning that if input changes, the output changes in the same way, so convolution can create a 2D map of where certain features appear in the input (based on (Goodfellow et al., 2016).

*Convolution* is an operation on two functions of real-valued argument. Usually in image processing the first function is a 2D image or a 3D tensor (including time parameter) - in case of a video. The second function is the convolution *kernel*, or sometimes it’s called *feature map*. The latter is usually much smaller in size than the image.

In the discrete domain the formulae for convolution can be written as:

Or sometime for implementation this is re-written as equivalent - as operation is commutative. Also, alternatively, the calculation can be done for *cross-correlation* where the kernel is not flipped – this is used in implementation by many neural network libraries.

The layers in convolutional network are sparsely connected. This is illustrated in Figure.. – input x3 only affects some of the output pixels, where in a traditional neural network – all outputs are affected by all inputs. Same for the receptive field – output pixel s3 is only affected by 3 inputs rather than all.

**Figure … Comparison of connectivity in a traditional NN (bottom) and CNN (top). Image on the left shows the effect of a single input pixel x3, image on the right show the receptive field of a single output pixel s3 (Goodfellow et al., 2016)**

The connectivity can be even sparser if *stride* bigger than 1 is used, i.e. kernel is not applied to every pixel, but to every 2nd pixel, 3rd etc. This is equivalent to *downsampling* in full convolution function and is used for computational efficiency and low-rate sampling.

Usually, the convolution operation is combined with *pooling*. This can be a maximum value in the neighbourhood operation (*max pooling*) – usually neighbourhoods are size 2x2 or 4x4. Or average of a rectangular neighbourhood (*average pooling*), L2-norm of the rectangular neighbourhood or a weighted average based on a distance from the central pixel. Pooling operation removes small changes, makes features invariant to small translations, or can introduce an arbitrary invariance to other transformations depending on the parameters (Goodfellow et al., 2016).

So, a typical *layer of convolutional neural network* comprises of the following:

* Convolution
* Activation function (ReLU for example)
* Pooling

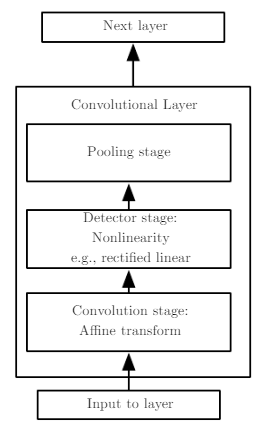
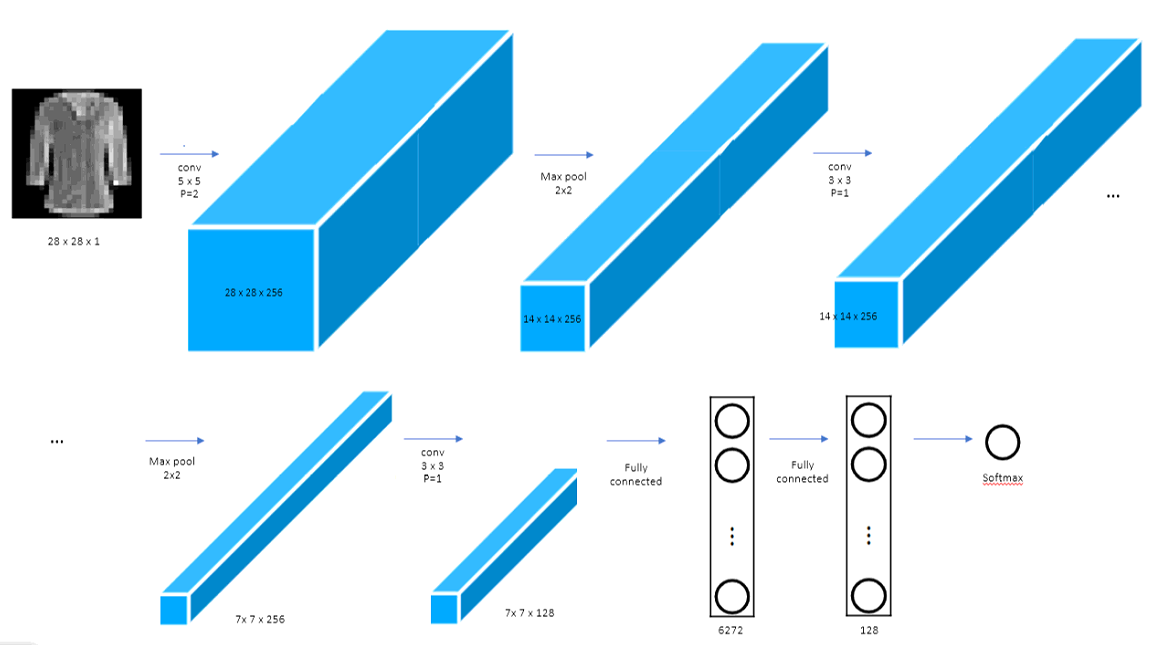


Figure .. Typical layer of convolutional neural network

The convolutional neural network benefits from having *multiple convolutional layers* – which is the “deep” part in deep-learning. For example, 16-19 layers in VGGNet, 22 layers in GoogleLeNet/ Inception, and up to 152 layers in ResNet – all of these neural networks were developed in 2014 – 2015 (V.Krylov lecture CS7GV1).



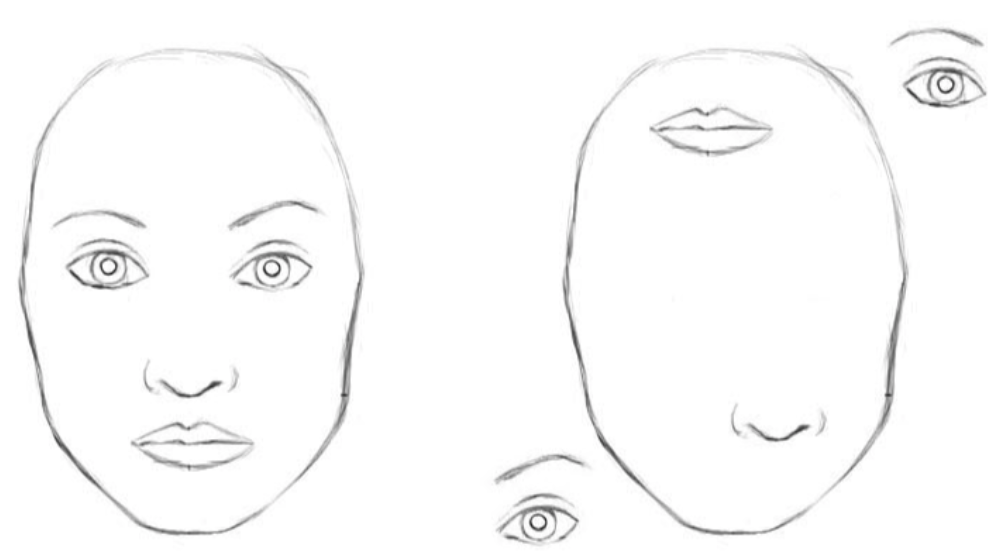
Figure…Example CNN architecture with 3 convolutional layers. Size of the square shows the size of input images, length of the cuboid – number of hidden layers. (source V.Olyunina, Assignment for submission for CS7GV1).

CNNs are used in computer vision extensively mainly for image classification and object recognition, but they can also be used for segmentation (no pooling is employed for segmentation). Their application to video interpolation is discussed in the next section.

2.3.2 Capsule Networks

There are certain drawbacks of CNNs – spatial relationships between components are not very important to a CNN. Figure .. illustrates this concept – both images in the figure are recognised as “face” by CNN. This is because in CNN features are combined into higher level features based on a weighted sum, so the positional parameter is lost.

(<https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b>)



**Figure .. Both the image on the right and the image on the left are classified as a “face” by CNN**

(Hinton et al., 2018) developed CapsuleNets to counteract this problem. CapusleNets use iterative process called “routing-by-agreement”, that updates the probability with which a part is assigned to a whole based on the proximity of the vote coming from that part to the votes coming from other parts that are assigned to that whole. This allows knowledge of familiar shapes to derive segmentation.

A capsule network consists of several layers of capsules. Each capsule has a 4x4 pose matrix, , and an activation probability, . CapsuleNets converts the whole set of activation probabilities and poses of the capsules in one layer into the activation probabilities and poses of capsules in the next layer using Expectation-Maximisation procedure (EM). (Hinton et al., 2018). The author trained CapsuleNet and CNN for comparison on the same small set of 3D objects (5 classes, 5 objects for each class) imaged from different azimuth, elevations etc. Each object had 18 different azimuth, 9 elevations and 6 different lighting conditions. The author shows that although there is no advantage over CNNs on familiar viewpoints – CapsuleNets performed considerably better on novel viewpoints (13% error vs 20% on different azimuth, 12% vs 18% error on different elevation).

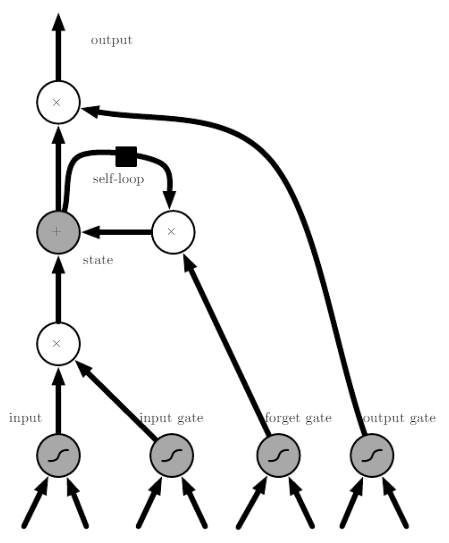
This research is very new and so far, was applied to classification and object segmentation tasks: ((Chen et al., 2018) , (LaLonde and Bagci, 2018) . It seems however that it should have good potential for video interpolation as this presents the subject from different novel points of view.

2.3.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are designed for processing sequential data, this can be a sequence of human limb positions in a video or a sequence of words in a sentence. If we had positions (states) in a sequence, the output from processing the position is passed for processing to position t+1. RNNs and in particular *gated RNNs* – Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) – have been shown to be very effective at natural language processing, handwriting recognition, speech recognition, image captioning and parsing (Goodfellow et al., 2016).

There are many different architectures of RNNs, the simplest once having one layer, but they have been shown to be more effective with multiple layers, i.e. deep RNNs (Pascanu, 2013). Some allow input not just from the previous state, but from all the previous states, some allow self-input.

Below is the architecture of LSTM recurrent network “cell”. LSTM NNs allow not only the input from the previous states in the sequence, but also self-input which is *gated* by the previous states. “Cells” are connected recurrently to each other.



**Figure … Architecture of LSTM recurrent network “cell” (Goodfellow et al., 2016)**.

While natural language processing is the most known use of RNNs, there are some relevant applications for video processing and video generation which are discussed in the next section.

2.3.4 Autoencoder

Autoencoders (AE) is one of the oldest types of neural networks that exists for decades. This neural network can generate new images from input images and is employed in generative neural networks discussed below.

The network consists of two parts:

* encoder function , where x is the input (image) and
* decoder function that produces a reconstruction based on the input.

- is the *code* used to represent the input, for example, the features of the input image (Goodfellow et al., 2016).

At its simplest meaning that the original image is copied, which is not very useful. The network therefore needs to be constraint by *regularizers*. *Undercomplete and sparse autoencoders* can learn features and be used for *dimensionality reduction*. *Denoising autoencoders* (DAE) were used to de-noise images. *Contractive autoencoders* (CAE) produce tangent vectors similar to PCA, so they learn a more powerful nonlinear generalization of PCA (Goodfellow et al., 2016).

While autoencoders themselves are used for dimensionality reduction and information retrieval tasks, they are the theoretical foundation for the more advanced generative networks.

2.3.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) have received recent attention. Pioneered in 2014 by (Goodfellow et al., 2014),

…

# 2.4 Deep-learning in view interpolation

Niklaus…

Depth estimation with deep-learning was researched by (Garg *et al.* 2016) and (Agrawal *et al.* 2015).

CHEN, R., JALAL, M. A., MIHAYLOVA, L. & MOORE, R. K. 2018. Learning Capsules for Vehicle Logo Recognition. ISIF.

CHEN, S. E. & WILLIAMS, L. 1993. View Interpolation for Image Synthesis. Proceedings of the 20th annual conference on Computer graphics and interactive techniques: SIGGRAPH

FAUGERAS, O. D. 1992. What can be seen in three dimensions with an uncalibrated stereo

rig? : Second European Conference on Computer Vision (ECCV’92).

FRAGNETO, P., FUSIELLO, A. & SOC, I. C. 2012. Uncalibrated View Synthesis with Homography Interpolation. *Second Joint 3dim/3dpvt Conference: 3d Imaging, Modeling, Processing, Visualization & Transmission (3dimpvt 2012)***,** 270-277.

GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016. *Deep Learning*, MIT Press.

GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. & BENGIO, Y. 2014. Generative Adversarial Networks. Available: <https://arxiv.org/abs/1406.2661>.

GURDAN, T., OSWALD, M. R., GURDAN, D. & CREMERS, D. 2014. Spatial and Temporal Interpolation of Multi-view Image Sequences. *Pattern Recognition, Gcpr 2014,* 8753**,** 305-316.

HARRIS, C. & STEPHENS, M. J. 1988. A combined corner and edge detector. *Alvey*

*Vision Conference.*

HARTLEY, R. & ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*.

HARTLEY, R. I. In defence of the 8-point algorithm. Proceedings of IEEE International Conference on Computer Vision. 1064-1070.

HARTLEY, R. I. In defence of the 8-point algorithm. Proceedings of IEEE International Conference on Computer Vision, 1995. 1064-1070.

HINTON, G., SABOUR, S. & FROSST, N. 2018. Matrix capsules with EM routing. *ICLR 2018 Conference.*

KANG, S. B., XIN TONG, Y. L. & SHUM, H.-Y. 2007. Image-Based Rendering.

KAZHDAN, M., BOLITHO, M. & HOPPE, H. 2006. Poisson Surface Reconstruction. Eurographics Symposium on Geometry Processing.

KOLOURI, S., PARK, S. R. & ROHDE, G. K. 2016. The Radon Cumulative Distribution Transform and Its Application to Image Classification. *Ieee Transactions on Image Processing,* 25**,** 920-934.

LALONDE, R. & BAGCI, U. 2018. Capsules for Object Segmentation. *MIDL*.

LONGUET-HIGGINS, H. C. 1981. A computer algorithm for reconstructing a scene from two

projections. *Nature.*

LUCAS, B. D. & KANADE, T. 1981. An iterative image registration technique with an

application in stereo vision. Seventh International Joint Conference on Artificial

Intelligence (IJCAI-81).

NIKLAUS, S., MAI, L. & LIU, F. 2017. Video Frame Interpolation via Adaptive Convolution. *CoRR,* abs/1703.07514.

PASCANU, R. A. G. C. A. C. K. A. B. Y. 2013. How to Construct Deep Recurrent Neural Networks.

SCHONBERGER, J. L., FRAHM, J. M. & IEEE. 2016. Structure-from-Motion Revisited. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 27-30 2016a Seattle, WA. NEW YORK: Ieee, 4104-4113.

SCHONBERGER, J. L., ZHENG, E. L., FRAHM, J. M. & POLLEFEYS, M. 2016b. Pixelwise View Selection for Unstructured Multi-View Stereo. *Computer Vision - Eccv 2016, Pt Iii,* 9907**,** 501-518.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D. & SZELISKI, R. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

SEITZ, S. M. & DYER, C. R. 1996. View Morphing. *SIGGRAPH,* Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques.

SHI, J. & TOMASI, C. 1994. Good features to track. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

SMITH, M. W., CARRIVICK, J. L. & QUINCEY, D. J. 2016. Structure from motion photogrammetry in physical geography. *Progress in Physical Geography,* 40**,** 247-275.

SZELISKI, R. 2010. *Computer Vision:*

*Algorithms and Applications*, Springer.

TOMASI, C. & KANADE, T. 1992. SHAPE AND MOTION FROM IMAGE STREAMS UNDER ORTHOGRAPHY - A FACTORIZATION METHOD. *International Journal of Computer Vision,* 9**,** 137-154.

TORR, P. H. S. & MURRAY, D. 1997. The development and comparison of robust methods for estimating the fundamental matrix. International Journal of Computer Vision.

ZHENG, E., DUNN, E., JOJIC, V. & FRAHM, J. M. 2014. Patchmatch based joint view selection and depthmap estimatio.

1. Orthographic projection is applicable when the distance from the object to the camera (Zavg) is more than 10 times the object’s width davg: Zavg ≥ 10 \* davg (Belongie, 2009, chapter 9) [↑](#footnote-ref-1)