Technical Report

Pattern Recognition and Image Processing Group
Institute of Visual Computing and Human-Centered Technology
TU Wien
Favoritenstrasse 9-11/193-03
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801 - 18661
Fax: +43 (1) 58801 - 18697
E-mail:
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-143 April 5, 2018

Proceedings of the Seminar in Scientific Working Winter Term 2017/18

Christopher Dick and Elias Franta and Max Sbardellati and Anna Sebernegg
Editors: Walter G. Kropatsch and Nicole M. Artner

Abstract

This is a collection of articles from the seminar of scientific working of the winter termn 2017/18. Articles are ordered alphabetically by the last names of the authors/students.

Segmentation of Motion Capture Data for Human Dance Movement: the case of *tango argentino*

Christopher S. Dick
e00946375

Scientific Working
TU Wien, Austria
e0946375@student.tuwien.ac.at

Abstract. The application of motion capture and pattern recognition techniques are not yet established within the field of academic dance research. This paper gives an overview of features and methods that can be used for automated dance movement segmentation. They comprise aspects that are represented in the growingly diverse literature on motion capture data and time series analysis. In the special case of *tango argentino*, a model for application-specific segmentation is suggested. This includes the idea of knowledge generation through segmentation and considerations on feature selection for tango.

Keywords: Motion Capture, Time Series Segmentation, Dance Movement, Tango Argentino

1 Introduction

The sophisticated capturing of human movement has become available for a broad audience in the past years. While modern digital motion capture technology, such as inertial or optical systems, was introduced already during the last few decades, it has mostly been a privilege of the entertainment industry. With more affordable technology available, these systems have been incorporated in many diverse fields of the study of human movement, providing large amounts of data for analysis. Especially the science and the studies of dance are getting more involved with digital approaches of capturing and analysis for their object of research. They record sequences of dance, movement to music, short segments, or just single motional patterns for in-depth examination. Specifically long recordings of dance are typical as they show all facets that come with the movement. But often particular parts of these long sequences are needed, which asks for segmentation of these recordings. Doing this manually can be cumbersome and imprecise. Automatic segmentation can reduce the work load and simultaneously function as a feedback model that might reveal facts with its segmentation about the underlying dance structures.

While methods for automated time series segmentation in general (and especially for large databases, such as stock market data, climate data, and so on)

are developed already since the 1990s, the research on dance motion data analysis has only begun to grow in about the last 10 to 15 years. Before that, and still today, many segmentation approaches for movement data were motivated by the need of the entertainment industry for short motion sequences, e.g. for use in animation of movies or games. The methods naturally have other requirements than what is needed in scientific movement analysis, yet they are based on common ground and are easily adaptable.

The paper seeks to give a compact overview of time series segmentation approaches that are relevant for the segmentation of human dance movement data. This summary is collected with the aim of application in the special case of *tango argentino*, the dance that is the research subject in the ongoing transdisciplinary research project "Tango-Danceability of Music in a European Perspective"[1]. The further goal is to segment recorded sequences of *tango argentino* to extract certain movement patterns, in particular the basic step. Due to the format and scope of this paper, the focus will be on the overview of techniques and initial theoretical considerations towards the case of *tango argentino* step detection.

Overview The remainder of this paper is structured as follows. In section 2 a short overview of time series segmentation is given, representing the current state of the art. Following, section 3 lists features and representation for both music and dance data that are both well established, as well as highly relevant for the particular case presented later. Section 4 presents approaches of time series segmentation strongly represented in the literature and applicable for the specific case example. A distinction between features and segmentation methodology is explicitly mentioned in the given literature. Both aspects often are combined into methods based on a certain feature. The chosen combination depends strongly on the application. The given features and methods, therefore can be combined interchangeably and are described separately for a better overview. Finally, a possible approach for applying the presented aspects in the special case of *tango argentino* is described in section 5. This is based on a two-phase approach, divided into (1) a low-level segmentation with unbiased feature selection, which is used for knowledge generation to be applied in (2) a semantic approach based on the characteristic kinematic features of the first phase, to be applied for further segmentation.

2 Overview of Time Series Segmentation

Early approaches of time series segmentation were often based in the domain of larger data bases. They comprised mainly financial data, weather data and the like and many segmentations were basic pre-processing operations for forecasting and trend detection [2,21,22]. Another early focus was set on speech recognition, working with audio data [13,18,20]. While the specific application of these approaches differ considerably, their common ground is the time series

[1] For more information see: dancetangomusic.com

analysis and segmentation, which can often be generalized. These by now well established methods can be used as a basis for the analysis of movement time segmentation.

First approaches to segmenting movement data can be found in [4,35,34]. However, the approaches differ fundamentally as some are working mainly on low-level features, focusing on the signal as information source itself [3,33]. Others, especially more recent studies, are trying to incorporate additional factors and pursue a high-level view of what Bouchard [5] calls "semantic segmentation", that is, informed segmentation, incorporating knowledge about the movement in question [5,12,28,29]. In the domain of dance movement, an additional aspect that can be considered is the music played or sound produced in general, which provides even more features for automatic segmentation or detection of dance movements [23,30,34].

The choice of representation of the captured movement data and the corresponding extracted features also vary among different methods. Since capturing absolute data (usually 3D position data) is confined by its governing coordinate system (the one used by the motion capture system, [27]), many seek to use some form of relative representation of the human body and its motion. This often involves joint angles, or positions of data points relative to the body [24,31]. Another option for representation and use as features are simple kinematic transformations, such as velocity or acceleration and their angular counterparts [3,33]. In the context of feature selection additional research has to be mentioned which works on the relationship of sound and movement and develops the analysis of the two. Important studies in this field are [6,7], which link rhythmical elements of music to their embodiment by dancers, or [10,26,38], which propose new methods of analyzing the intricate relationship of dance and music.

3 Feature Selection

A crucial part of automated segmentation is the selection of suitable features (or the representation of the data, respectively). The following section therefore sums up some of the most commonly used features in movement and music analysis that can be considered common ground within the respective fields.

3.1 Music Features

Concerning possible features of music, the field of Music Information Retrieval (MIR) already established a lot of possible techniques, algorithms, and de-facto standards in music analysis[2]. As this extends beyond the scope of this paper, only a few selected aspects are summarized, which might be of interest for the given case of *tango argentino*. Additionally, it can be noted that some of the given low-level features resemble those presented in subsection 3.2. This is no coincidence, as many kinematic analyses were inspired by audio analysis and

[2] For an overview see [9].

general signal analysis, on which all these approaches are based on. Therefore they are listed first.

Zero-Crossing A basic feature of signal analysis that is heavily used in speech processing and MIR is the occurrence of zero-crossings in a signal. It is described by the change of the sign of the signal's value, either from positive to negative, $v_i > 0 \wedge v_{i+1} < 0$, or from negative to positive $v_i < 0 \wedge v_{i+1} > 0$. Counting the instances of crossings, or determining their mean then results in the *zero-crossing rate*. In speech recognition this is used to detect endpoints and describes unvoiced sounds, or percussive sounds, regarding music.

Energy In the context of audio signals, the energy usually is connected to the wave form representation of digital audio, i.e. the amplitude. A typical feature then is the root-mean-square (RMS) energy, computed by taking the root average of the square of the amplitude (as described for example in [19]):

$$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2} \tag{1}$$

Applying this operation framewise yields an energy curve for the given signal. Changes of high and low energy can then point to possible segmentation points, indicating the presence of noise/sound/speech.

Tempo A simple description of music in the time domain can be given by the tempo of the music. Seeing that tempo is a construct that arose mainly in the context of classical western music, it can be more generally defined as beats per minute (BPM). This describes the period of a recurring reference entity in the music, usually described as the beat. The exact definition of the beat depends on the specific kind of music (e.g. a steady bass drum, the implications of a meter, etc.).

3.2 Kinematic Features

There is not yet a comprehensive, systematized overview of movement features. Therefore, based on the given literature, some of the most applied aspects as well as some that are found to be applicable in the special case of dance, are considered. This includes only features relevant for motion data typically captured by a motion capture system (i.e. the representations described in *Position, Velocity, Acceleration* and *Angles*), excluding classical image processing features such as optical flow and the like.

Position, Velocity, Acceleration A basic way of representing movement data, and also the most used one by motion capture systems, is position data and its time derivatives. This might refer to the position of captured markers, virtual markers, or body parts. It describes the position of a selected entity in regard

to its position components in two (x, y), or three (x, y, z) dimensional space respectively. An important aspect that has to be considered in using position data as movement representation is their adherence to the coordinate system in which they are captured. Since those are usually arbitrarily chosen depending on the given capturing system, a transformation into a particular coordinate system (e.g. those of a joint, or one defined by the PCA components of the data [26,27]), or transformation of the data into a relative description, might be more suitable. An example for this can be measuring the distance between hands to define clapping movements. The local extrema of the Euclidean distance between hand positions give a clue towards the endpoint of the movements.

An often more suitable representation, as it more closely describes the movement itself, is a certain time derivative of the position. Inertial motion capture systems usually deliver data in this format, using inertial measurement units and gyroscopes to determine local acceleration and rotation. Since the derivatives naturally depend on each other, velocity and acceleration are often used in conjunction, looking for extrema in either one, via the other. A simple application is for instance the detection of pivot points of a certain movement pattern [17], i.e. the point in time where velocity and acceleration are close to zero, indicating no motion and therefore an endpoint of motion patterns. The third derivative known as jerk (jolt, surge, lurch), is generally used to describe the smoothness of a movement (e.g. based on the minimum jerk model [11]). Higher order derivatives are rarely encountered.

Angles Analogous to the linear displacement features relating to position as described above, their angular counterparts are commonly used. The basis here is the rotation of a body (marker, body part, etc.) in a superimposed reference system. This also includes the time derivatives of angular velocity, angular acceleration, and angular jerk (where again, higher order derivatives are rarely encountered). Typically these descriptions refer to the joint angles, that is, the minimum angle between two body segments connected by a joint (e.g. between upper and lower arm), or the rotation of a body segment according to its parent segment (e.g. the rotation of the shin in relation to the thigh, defined by the knee joint). Similar to above, end and pivot points can be defined as angular velocity and acceleration close to zero.

Zero-Crossing This description comes from the acoustical feature of the same name (as described in section 3.1). It detects points in a movement signal (usually velocity or acceleration) where its value changes the sign. This results in possible end or pivot points of movements as described above. As this approach is highly sensitive to noise, an advanced zero-crossing technique is presented in [33]. It incorporates an additional threshold ϵ around zero, taking into account noise that might be, for instance, generated by the capturing system or slight motion of the body[3]. Another adaptation of the basic method is proposed in [4], where

[3] The human body is not able to stand completely still. This is subject of recent studies exploring the phenomenon of *micromovements* [15,14].

a weighted sum of zero-crossing of all joints is calculated, giving more weight to body parts that define certain motion (e.g. hands and feet in unconstrained movement).

Energy In movement, energy usually refers to the physical entity of energy, in particular kinetic energy, which is often decomposed into translational and rotational energy. This can be used in the context of semantic analysis, as it can describe the abstract concept of effort put into a movement [27] (see section 4.2). The calculation of kinetic energy however is not trivial anymore and only approximates real values, as some assumptions have to be made regarding the mass of different body segments. In the field of biomechanics several models have been established for this, of which the Dempster's model is usually sufficient for a satisfying approximation in movement feature extraction tasks [8].

Curvature When using velocity and acceleration features to determine end and pivot points, the trajectory itself is not taken into account. To incorporate this aspect, Zhao [39] presents the computation of curvature c_i, which is defined as the rate of change of the velocity trajectory. It is computed in three-dimensional capturing spaces by taking the cross product of the velocity v_i and acceleration a_i at time i:

$$c_i = v_i \times a_i; v_i = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, a_i = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{2}$$

This feature is suited to describe changes in direction of a motion trajectory, even for non-zero velocity magnitudes.

4 Time Series Segmentation Techniques

This section presents an overview of methods and techniques that can be considered relevant for the automated segmentation of human movement data. Especially low-level features and approaches are explained, and a less detailed summary of high-level works is given as they are mostly designed for very particular applications. In order to have a common basis for all presented topics, the definition of a time series must be given, so misunderstandings stemming from domain-specific interpretations can be avoided. A time series is therefore generally defined as a sequence of time dependent values. To use this definition within the domain of human movement data, an adaptation of [4] and [20] is followed. A time series T can then be defined more closely as an ordered sequence of n pairs, consisting of vector v_i and a point in time t_i (as seen in Equation 4), whereby v_i is the value of an m-dimensional function.

$$T = \{(v_1, t_1), (v_2, t_2), \ldots, (v_{n-1}, t_{n-1}), (v_n, t_n), \} \tag{3}$$

Since nearly all digital capturing techniques work on the basis of fixed frame rates, in the case of human movement capturing it is more practical to consider

only regular time series. These are defined by isochronous sample rates, i.e. a constant time step Δt between consecutive pairs of the time series: $\forall i, 1 \leq i < n : \Delta t = t_{i+1} - t_i$. A segment then is a contiguous subset of a time series, which is a time series itself, and a segmentation splits a given time series into different segments. This does not result in a partition of the original time series and the segments may overlap in t. Depending on the context, the latter factors may be constrained to require no overlaps or a complete partition of the original series. The dimensionality and type of the input sequence depends on the chosen representation format. For most low-level features and segmentation techniques, they are not relevant but work on generalized numeric vectors and scalars.

Processing a time series can be separated into the different processes of feature extraction, segmentation, and the actual application (e.g. of analysis, modification, etc.). Feature extraction describes the transformation of original data to a suited representation (if not already so), application of filters, dimensionality reduction and so forth. The segmentation process itself can be classified according to how the data is processed. In [4], the distinction is between explicit, implicit, and hybrid segmentation, where explicit describes techniques where the segmentation phase happens in one-pass, implicit segmentation uses a multi-pass approach incorporating error feedback for optimization of temporary results, and the hybrid uses explicit segmentation with additional feedback modes. Additional classifications can be made according to the often used concept of online and offline algorithms, describing the processing of data in a sequential manner, or the entire data set respectively [16]. These classifications are of importance especially in regard to their performance and ability for real-time application.

4.1 Low-Level Approaches

The approaches described here work mainly on a data-centric level, preferably taking into account kinematic analysis. What they have in common, is the description of the general approach of analyzing and iterating over a given time series, yielding an algorithm for processing time series. The exact segmentation boundaries then depend on the chosen data representation. They can, however, be combined with higher-level approaches, e.g. to include semantics and additional information. The given examples originate in database analysis but today represent basic approaches for general time series analysis. The given descriptions follow [4,16]. Some additional information is needed for these approaches: first, some function that defines the error of a given segment (usually relating to some optimal result or model); second, a distance measure might be needed to describe the distance between two selected segments. As opposed to a feature-based method, where certain value thresholds of a distinct feature define the segmentation boundary, these algorithms require an error information, therefore being implicit.

Top-Down The top-down approach takes the original segment, i.e. the entire data set, and splits it recursively into smaller segments. It starts by considering

all possible partitionings and their pairwise distances. The goal is to find the partitioning where adjacent segments have the highest possible distance. This is repeated recursively on the found segments until the errors of each individual segment are below a given threshold. Alternatively, the choice of the partitions can be based solely on the error of a given segment and a specified threshold, ignoring thereby the distance between the two created segments. This yields a complete partitioning with minimum error for all segments.

Bottom-Up Related to the top-down algorithm, bottom-up begins with a complete partitioning of the time series. The level of granularity of these segments can be chosen depending on the application. It then joins adjacent segments given their distance, or the cost to merge them respectively. This is repeated until the distance/cost meets a certain threshold and no more join operations are possible. The approach can be slightly faster than top-down, depending on the distance measure/cost function.

Sliding Window This simple online algorithm is the basis for many signal analysis methods. It considers all possible segments by sliding windows of different sizes of the entire time series, choosing appropriate segments by selecting those with the smallest error. If a complete partitioning is desired, an easy approach is to initialize the starting point of the window w_1 as the first point of the time series t_1. The size of the window, defined by its end point w_{end}, then is incremented until the error is below a certain threshold. Then the window is slided by setting the new w_1 to the old w_{end} until the end of the time series is met. If only a subset of segments are desired, the size of the window must be predefined, as well as the step size by which the window is slided. Depending on the applications, several iterations with different window sizes and step sizes might be used.

4.2 High-Level Approaches

The approaches in this section describe segmentation techniques for the specific case of movement analysis and represent a small selection of the application-dependent variety. Most of these work on a semantic level, taking into account external information about the analyzed movement, e.g. knowledge about existing movement patterns, or motion features that characterize the given movement.

Hidden Markov Models Hidden Markov Models (HMM) are statistical models that include different states and the transitions between them (for more information see [32]). They can be well used for the recognition of different states in a temporal sequence. This makes them suitable for the use with movement sequences, such as gesture recognition, or dance applications [12]. Since HMM are a supervised learning approach, training data is needed, i.e. the states of the model have to be known in advance. This limits the application to certain kinds of tasks, where uninformed segmentation is not possible.

Motion Templates In [24], Müller and Röder present a model for capturing the spatio-temporal characteristics of an entire motion class, i.e. a defined movement sequence, in a matrix representation called *motion templates*. They can be learned from a given set of samples of a motion class or artificially constructed. The used features are called *relational motion features*, which are boolean representations of geometric relations of particular points of a body (cf. subsection 3.2). Given dimensions f (number of features) and K (number of time samples or frames), a motion template is a real-valued matrix $X \in [0,1]^{f \times K}$. Learning a motion template from given training data yields a template for the represented motion class. These can then be used for further segmentation and identification, with the template as a reference entity for determining an error or distance (such as in subsection 4.1).

LMA Classifier The LMA Classifier is a semantic segmentation approach developed by Bouchard [5]. It is based on the theory of Laban Movement Analysis (LMA), a theoretical framework for describing and interpreting human motion. The basic principle is the relation between the internal state of an individual and the effect it has on the motion. The classifier is based on the concept of effort (the quality of motion) in the four dimensions of space, time, weight, and flow. Each of these dimensions is represented by a neural network. Given a movement time series, the LMA classifiers can be used to determine segment boundaries which are similar to the original data used to train the classifiers. This approach gives the most freedom in regard to the quality of the analyzed motion, and works therefore well with general movement data, i.e. the type of movement is not known. However, it strongly depends on the training data of the LMA classifier. It can be possible to train the latter with samples of a particular movement repertoire (such as a certain dance) in order to apply it to specialized data. What has to be considered for the given approach, is the difficulty of objectifying a complex concept such as effort. The selection of low-level features strongly influences the resulting classifier. Employing LMA specialists is therefore a crucial factor in describing the movements used for the training data.

5 Considerations in Applications: *Tango Argentino*

Tango Argentino is a performing arts genre that involves music, dance, and lyrics. For this paper, the dance and music are of interest. It is an improvised couple dance, characterized by a close embrace and close connection of the dance partners from the waist upwards, and flexible and elaborate footwork made possible mainly by horizontal dissociation movements in both dancers waists. The embrace is occasionally opened for short intervals of more elaborate movements. The improvisation is based on a particular, limited movement repertoire and set rules of basic body position, stride and turning, for both leader and follower. Movements are executed for communication, travelling, turning, and also embellishing. This leads to three constituting elements of the dance, regarding movement sequences: standing, walking, and turning. The element in question in

the research project "Tango-Danceability of Music in a European Perspective" [36] is the tango step, as this is the core aspect of the movement repertoire. Following are the combinations of the basic elements, i.e. turning while walking or standing.

As most other dances, *tango argentino* has a close relationship with the music, even more so, since the music is the primary resource of inspiration for the improvisations of the dancers. The classical tango repertoire[4] used in this study, is based on even meters (typically four quarter notes $\frac{4}{4}$, i.e. four beats are grouped together as a bar), and an absolute tempo of about 120 beats per minute (BPM), with only slight variations in tempo. The intricate relationship of the movements of the dance and the sound of the music is one of the questions of the research project. What can be determined *a priori* is the temporal relation of the steps and the beat. The tempo and steady beat, with emphasis on the first and third beat of each bar, are connected to even and consistent steps that are carried out on the first and third beat. This leads to a steady, basic walking pace of approximately 60 BPM. Variations of these steps are then, for instance, double- or half-time steps, resulting in short intervals of a walking pace of 120 BPM or 30 BPM, respectively.

Data Collection In the project [36], motion capture data of professional tango dancers is collected. For this, an optical system with passive markers and eight cameras by OptiTrack[5] is used. This system captures markers attached to a body, yielding in marker position data (x, y, z), which can be fitted to reconstruct a model of the captured body in a 3D environment. The level of granularity is within millimetres in position and milliseconds in timing (the used frame rate is 120 Hz). For both dancers a given marker set is used, as seen in section 5. This is adapted to requirements of the dance, which includes removing the chest marker, as it will be occluded most of the time, due to the close embrace, as well as changing the leg markers from the front to the back, so as not to restrict their freedom of movement (e.g. when the legs of the couple are connected). Further processing then usually happens within a joint representation, a transformation of the original marker set that represents anatomical landmarks in a human body model. From this representation joint angles are easily computed. Music for dancing is played and recorded back separately and synchronized manually by means of a clapper (as used in film recordings) attached with markers as a reference point in both recordings. Couples are asked to perform several movement tasks to four different audio stimuli: three recordings of classical tango music, and one click track with 120 BPM for reference. The tasks are performed separately and as a couple. Performing as a couple includes free dancing with no

[4] This comprises music from what is known as the golden age, the *época de oro* of *tango argentino*. It took place in Argentina in the 1930s and 1940s. Musical compositions, performers, and dance styles of this period are generally considered a form of benchmark against which all subsequent innovations and developments are measured.

[5] `http://www.optitrack.com/`

restriction, dancing with restriction to walking and turning, and dancing with restriction to only walking. Single tasks include tapping the foot to the beat, and tango walking. This produces a wide variety of data suitable for the different research questions that can also be used as reference entities between them.

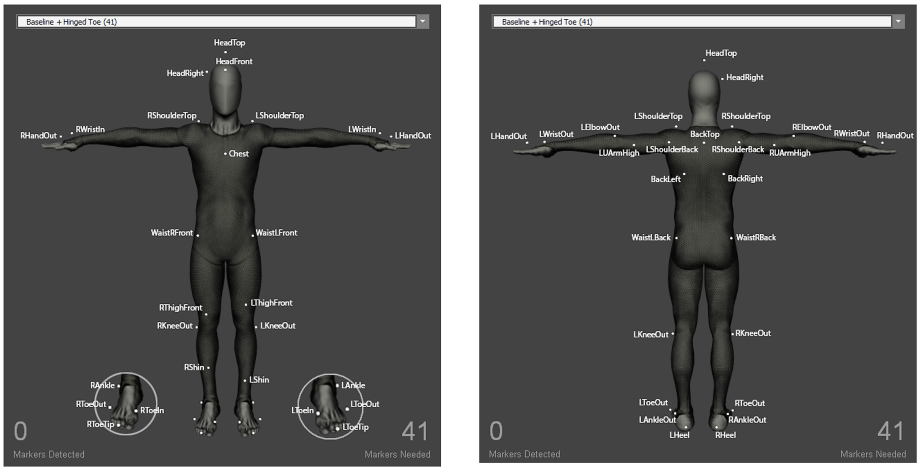


Fig. 1. The original full body marker set [25] used in the capturing of professional tango dancers. It is modified according to the requirements of the close embrace and footwork: the chest marker is removed, thigh and shin markers are moved to the back of the leg.

Application The idea for the application specific segmentation of *tango argentino* data consists of two phases, which determine the selected features and techniques used. As one aim of the project is to explain the characteristics of tango dance, and consequently the tango step, in multiple dimensions, supervised segmentation techniques are not suitable as a first approach, seeing that training data does not exist. In many studies manual segmentation is used to generate this training data, which is not considered here, as this might be biased by the person's experience and personal preferences. Instead, the idea is to use low-level techniques based on generally accepted, simple *a priory* knowledge that can be used to determine features with as little bias as possible. Additionally, the questions dealt with are working on the level of micro-timing, which cannot be achieved by manual segmentation[6]. The type of data for this phase should

[6] This does not imply that manual segmentation is not a valid approach for segmentation, but rather that it is not appropriate in this particular context. It might, however, well be considered for instance, as a measure in the evaluation of the segmentation, or a reference entity for interpreting the embodiment and conceptualization sound-movement relationships in dancers.

therefore be the captures of a couple dancing to music, restricted to tango walking, as this can be used as representations of a potential prototype tango step. Thus, the first phase consists of knowledge generation and extraction of possible training data for the second phase. This can then work on the basis of a supervised model, using semantic approaches, which are informed by the previous phase. The features are chosen according to the kinematic characteristics of the found segments. By generalizing the model to the abstract concept of a tango step, it can be used in further segmentation tasks of all other kinds of tango dancing data, as the basic concept of tango is always based on walking (also in combination with turning, double- and half time, and so on). The length of the segment is not relevant in this case, since it can be easily accounted for by methods such as Dynamic Time Warping [1,37] or choosing time-independent features.

For the first phase, the fact that regular tango steps are generally half the absolute tempo of the music can be used as a constraint for the segments' lengths. Based on the low-level approaches described in subsection 4.1, sliding windows with the length of a step and a time range around this can be applied. Choosing appropriate error measures to determine whether a candidate segment is accepted or rejected then might prove to be challenging, as features have to be chosen, that are not subject to individual interpretation. One possibility is to take general parameters that originate in gait analysis, as these determine what is understood as a step. This comprises a resting phase with little to almost no movement between steps. This includes, for instance, that feet cannot be lifted off the ground at start and end points and have to be in spatial proximity, or boundaries which are characterized by a minimum of overall motion. Alternatively, potential segment boundaries can be searched for directly by iterating over the time series, using the length of a segment in a subsequent iteration as a decision rule for matching candidate points, and finally accepting candidate segments based on the parameters described before. To detect potential segment boundaries, features such as zero crossings or energy minima can be suited. All segmentations can then be referenced to the music, accepting only those starting at a stressed beat, i.e. the first or third in a bar.

Applying the first phase to a diverse data set of different dancers with different music, results in training data used in the second phase. A simple approach for this is then the sliding window, as the features for error or distance measurement can be taken from the training data. These can consist of, for instance, any kinematic feature averaged over the different training samples, and a simple error measurement such as Euclidean Distance to the prototype step. Another possibility is step recognition that adapts the idea of face recognition based on eigenfaces. In the case of movement data, so called eigenmovements or eigenmoves [3] can be computed with Principal Components Analysis of selected kinematic features [6], especially position and its time derivatives and joint angles. To allow a greater variance in the movement data itself (potentially up to combined steps, such as turning midways), semantic approaches such as HMM, Motion Templates or LMA classifiers can be used. In the given case, it would be

useful to start with motion templates, as relational motion features can easily be computed from previously generated training data. When including additional motion sequences, such as pivoting while standing, or embellishing movements such as kicks, HMM can offer deeper insight, as the temporal dependence of the given motion states are taken into account.

6 Conclusion

Using pattern recognition for motion capture data is a popular approach for its segmentation nowadays. The applications of the theoretical findings, however, are mostly prominent with the entertainment industry, animation, media design, or robotics. Therefore this paper presented an overview for the application within the field of dance research, together with a specific example of its possible use. It can be noted that a lot is moving within the analysis of human dance movement data. Many features and segmentation techniques have already been established or adapted from related disciplines. Some of the most present ones in current literature are therefore listed, giving a compact overview. To illustrate their use and to suggest a possible application within the academic research of dance, the case of *tango argentino* was used. The suggested two-phase model of segmentation might improve the work with dance motion capture data, while serving as an analytical tool at the same time. The simple requirement of unbiased feature selection in the first phase can deliver insights into aspects of the dance by its segmentation. In the case of *tango argentino*, it could deliver a descriptive set of kinematic features, i.e. of what defines a tango step. These results can be subsequently used in the second step for automated segmentation of other tango data for further analysis. The next steps therefore include an application and evaluation of the suggested model.

References

1. Kevin Adistambha, Christian H. Ritz, and Ian S. Burnett. Motion classification using dynamic time warping. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*. IEEE, Oct 2008. doi:10.1109/mmsp.2008.4665151.
2. Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. *Foundations of data organization and algorithms*, pages 69–84, 1993.
3. Frédéric Bevilacqua, Jeff Ridenour, and David J. Cuccia. 3d motion capture data: Motion analysis and mapping to music. In *Proceeding of the Workshop/Symposium on Sensing and Input for Media-centric Systems*, 2002.
4. Durell Bouchard. Automated time series segmentation for human motion analysis. Technical report, Center for Human Modeling and Simulation University of Pennsylvania, Pennsylvania, 2006. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.485.664&rep=rep1&type=pdf.
5. Durell Bouchard and Norman I. Badler. Segmenting motion capture data using a qualitative analysis. In Unknown, editor, *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games - SA '15*, pages 23–30, New York, New York, USA, 2015. ACM Press. doi:10.1145/2822013.2822039.

6. Birgitta Burger, Marc R. Thompson, Geoff Luck, Suvi Saarikallio, and Petri Toiviainen. Influences of rhythm- and timbre-related musical features on characteristics of music-induced movement. *Frontiers in Psychology*, 4, 2013. URL: https://doi.org/10.3389/fpsyg.2013.00183, doi:10.3389/fpsyg.2013.00183.
7. Birgitta Burger, Marc R. Thompson, Geoff Luck, Suvi H. Saarikallio, and Petri Toiviainen. Hunting for the beat in the body: on period and phase locking in music-induced movement. *Frontiers in Human Neuroscience*, 8, nov 2014. doi:10.3389/fnhum.2014.00903.
8. Birgitta Burger and Petri Toiviainen. Mocap toolbox - a matlab toolbox for computational analysis of movement data. In *Proceedings of the 10th Sound and Music Computing Conference, Stockholm, Sweden*, Aug 2013. URL: https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mocaptoolbox/MocapToolboxProceeding.
9. J. Stephen Downie and David Bainbridge Ismir. Music information retrieval. *Annual Review of Information Science and Technology*, pages 295–340, 2003.
10. Edith Van Dyck, Dirk Moelants, Michiel Demey, Alexander Deweppe, Pieter Coussement, and Marc Leman. The impact of the bass drum on human dance movement. *Music Perception: An Interdisciplinary Journal*, 30(4):349–359, apr 2013. URL: https://doi.org/10.1525/mp.2013.30.4.349, doi:10.1525/mp.2013.30.4.349.
11. Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.
12. Jules Françoise, Agnès Roby-Brami, Natasha Riboud, and Frédéric Bevilacqua. Movement sequence analysis using hidden markov models. In Sarah Fdili Alaoui, Philippe Pasquier, Thecla Schiphorst, Jules Françoise, and Frédéric Bevilacqua, editors, *Proceedings of the 2nd International Workshop on Movement and Computing - MOCO '15*, pages 29–36, New York, New York, USA, 2015. ACM Press. doi:10.1145/2790994.2791006.
13. A. Ganapathiraju, L. Webster, J. Trimble, K. Bush, and P. Kornman. Comparison of energy-based endpoint detectors for speech signal processing. In *Southeastcon '96. Bringing Together Education, Science and Technology., Proceedings of the IEEE*, pages 500–503, Apr 1996. doi:10.1109/SECON.1996.510121.
14. Alexander Refsum Jensenius. Exploring music-related micromotion. In Clemens Wöllner, editor, *Body, Sound and Space in Music and Beyond: Multimodal Explorations*, pages 29–48. Taylor & Francis, Justus-Liebig-Universitt Gieen, 2017.
15. Alexander Refsum Jensenius and Kari Anne Vadstensvik Bjerkestrand. Exploring micromovements with motion capture and sonification. In Anthony L. Brooks, editor, *Arts and Technology: Second International Conference, ArtsIT 2011, Esbjerg, Denmark, December 10-11, 2011, Revised Selected Papers*, pages 100–107. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. URL: https://doi.org/10.1007/978-3-642-33329-3_12, doi:10.1007/978-3-642-33329-3_12.
16. Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. In Mark Last, Abraham Kandel, and Horst Bunke, editors, *Data mining in time series databases*, Series in machine perception and artificial intelligence, pages 1–22. World Scientific Publication, Singapore, 2005.
17. Gerhard Kubik. Transkription afrikanischer musik vom stummfilm: Methoden und probleme. In Arthur Simon, editor, *Musik in Afrika: Mit 20 Beitrgen zur Kenntnis traditioneller Afrikanischer Musikkulturen*, pages 202–216. Reiter Druck, Berlin, 1983.

18. Lori Lamel, Lawrence Rabiner, Aaron Rosenberg, and J Wilpon. An improved endpoint detector for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(4):777–785, 1981.
19. Olivier Lartillot. *MIRToolbox 1.6.1: User's Manual*. Aalborg University, Denmark, Dec 2014. URL: `https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox/MIRtoolbox1.6.1guide/view`.
20. M. Last, Y. Klein, and A. Kandel. Knowledge discovery in time series databases. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(1):160–169, Feb 2001. `doi:10.1109/3477.907576`.
21. Chung-Sheng Li, Philip S. Yu, and Vittorio Castelli. Malm: A framework for mining sequence database at multiple abstraction levels. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM '98, pages 267–272, New York, NY, USA, 1998. ACM. URL: `http://doi.acm.org/10.1145/288627.288666`, `doi:10.1145/288627.288666`.
22. Spyros Makridakis. *Forecasting : methods and applications*. John Wiley & Sons, New York, 1998.
23. Aymeric Masurelle, Slim Essid, and Gael Richard. Multimodal classification of dance movements using body joint trajectories and step sounds. In *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4. IEEE, 2013. `doi:{10.1109/WIAMIS.2013.6616151}`.
24. Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In Carol O'Sullivan, editor, *Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer Animation*, pages 137–146, Aire-la-Ville, Switzerland, 2006. Eurographics Association.
25. NaturalPoint. Baseline hinged toe (41) - naturalpoint product documentation ver 2.0. online. URL: `https://v20.wiki.optitrack.com/index.php?title=Baseline_%2B_Hinged_Toe_%2841%29`.
26. Luiz Naveda and Marc Leman. The spatiotemporal representation of dance and music gestures using topological gesture analysis (tga). *Music Perception: An Interdisciplinary Journal*, 28(1):93–111, 2010. URL: `http://mp.ucpress.edu/content/28/1/93`, `arXiv:http://mp.ucpress.edu/content/28/1/93.full.pdf`, `doi:10.1525/mp.2010.28.1.93`.
27. Luiz Naveda, Isabel C. Martínez, Javier Damesón, Alejandro Pereira Ghiena, Romina Herrera, and Manuel Alejandro Ordás. Musical meter, rhythm and the moving body: Designing methods for the analysis of unconstrained body movements. In *Music, Mind, and Embodiment*, pages 42–57. Springer International Publishing, 2016. `doi:10.1007/978-3-319-46282-0_3`.
28. N. Okada, Naoya Iwamoto, Tsukasa Fukusato, and Shigeo Morishima. Dance motion segmentation method based on choreographic primitives. In José Braz, Julien Pettré, and Paul Richard, editors, *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications*, pages 332–339. SciTePess - Science and and Technology Publications, 2015. `doi:{10.5220/0005304303320339}`.
29. Costas Panagiotakis, Antonis Argyros, and Damien Michel. Temporal segmentation and seamless stitching of motion patterns for synthesizing novel animations of periodic dances. In *2014 22nd International Conference on Pattern Recognition*, pages 1892–1897. IEEE, 2014. `doi:{10.1109/ICPR.2014.331}`.
30. Costas Panagiotakis, Andre Holzapfel, Damien Michel, and Antonis A. Argyros. Beat synchronous dance animation based on visual analysis of human motion and audio analysis of music tempo. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Os-

car Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Baoxin Li, Fatih Porikli, Victor Zordan, James Klosowski, Sabine Coquillart, Xun Luo, Min Chen, and David Gotz, editors, *Advances in Visual Computing*, volume 8034 of *Lecture Notes in Computer Science*, pages 118–127. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:{10.1007/978-3-642-41939-3{\textunderscore}12}.

31. Shu-Juan Peng. Motion segmentation using central distance features and low-pass filter. In *Proceedings of the 2010 International Conference on Computational Intelligence and Security*, CIS '10, pages 223–226, Washington, DC, USA, 2010. IEEE Computer Society. doi:10.1109/CIS.2010.54.
32. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. doi:10.1109/5.18626.
33. Sebastian Schulz and Annika Woerner. Automatic motion segmentation for human motion synthesis. In Francisco J. Perales and Robert B. Fisher, editors, *Articulated Motion and Deformable Objects*, volume 6169 of *Lecture Notes in Computer Science*, pages 182–191. Springer, Berlin, Heidelberg, 2010. doi:{10.1007/978-3-642-14061-7{\textunderscore}18}.
34. T. Shiratori, A. Nakazawa, and K. Ikeuchi. Detecting dance motion structure using motion capture and musical information. online. URL: https://www.cs.cmu.edu/~siratori/pub/VSMM2004shiratori.pdf.
35. T. Shiratori, A. Nakazawa, and K. Ikeuchi. Rhythmic motion analysis using motion capture and musical information. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI2003*, pages 89–94. IEEE, 2003. doi:{10.1109/MFI-2003.2003.1232638}.
36. Kendra Stepputat. dance.tango.music. online. URL: http://www.dancetangomusic.com/.
37. Adam Switonski, Henryk Josinski, Hafedh Zghidi, and Konrad Wojciechowski. Motion data classification on the basis of dynamic time warping with a cloud point distance measure. *AIP Conference Proceedings*, 1738(1):180019, 2016. URL: http://aip.scitation.org/doi/abs/10.1063/1.4951966, arXiv:http://aip.scitation.org/doi/pdf/10.1063/1.4951966, doi:10.1063/1.4951966.
38. Petri Toiviainen, Geoff Luck, and Marc R. Thompson. Embodied meter: Hierarchical eigenmodes in music-induced movement. *Music Perception*, 28(1):59–70, sep 2010. URL: https://doi.org/10.1525/mp.2010.28.1.59, doi:10.1525/mp.2010.28.1.59.
39. Liwei Zhao. *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2001. AAI3015399.

Super-human play in classic board games

Elias Frantar
e01527171

Scientific Working
TU Wien, Austria
elias.frantar@gmail.com

Abstract. Playing classic board games such as Chess, Go or Backgammon at a super-human level has always been an active area of research in artificial intelligence as games provide excellent controlled environments for testing complex concepts. While many games were already solved more than 20 years ago using specialized algorithms designed with the help of human experts, some have remained extremely challenging for computers until very recently. New machine learning techniques that combine classical search with convolutional neural networks trained purely by playing against itself have made it possible to address even the most difficult board games, like Go, which previously seemed impossible for conventional methods. Those techniques are very general and could also lead to advancements in other fields. In this paper we present an overview of the current state of the art in automated game playing including the latest developments such as DeepMind's AlphaGo and AlphaZero. We analyze successful solutions and discuss the key techniques necessary for achieving super-human performance in many board games.

Keywords: board games, machine learning, reinforcement learning, neural networks, MiniMax, Monte-Carlo tree search

1 Introduction

Humans have been playing board games like Chess or Go for hundreds if not thousands of years. Those games present captivating intellectual challenges as they test not only memorization and calculation skills but also require on-point long-term strategy and rapid adaption to previously unseen situations. Despite the rules of most classic games being quite simple, playing them at a highest level requires immense amounts of knowledge, practice and talent. For some games (especially Go) it is very difficult to pass on those skills to other players, even more so to formalize it, as they rely heavily on *intuition*, having the right feeling about which move to play next.

Trying to design computer programs that beat humans at something they are extremely good at is already an interesting challenge by itself. What makes classic games even more attractive for researches is that they represent simple, controlled environments for testing potential solutions to complex concepts.

Techniques that achieve super-human play are likely also applicable in other fields eventually leading to advances in solving actual real world problems.

IBM's Deep Blue [3] defeating the world champion in Chess back in 1997 is still considered a milestone in artificial intelligence. More recently, in 2016 DeepMind's AlphaGo [14] won against a top human player in Go, a game far more difficult for computers than Chess. Especially developments in machine learning and reinforcement learning have provided new possibilities for creating game playing agents that learn entirely from data (generated by playing matches against itself) without any human intervention resulting in novel strategies no human thought of before.

In this paper we describe and analyze a number of successful solutions for various games to figure out what it takes to design a program that achieves super-human play in a board game with the hope of those techniques generalizing to other areas of application.

1.1 Overview of the paper

Section 2 briefly describes several fundamental techniques necessary for understanding the state of the art game playing agents. In section 3 we describe a representative selection of state of the art solutions for several different games. These will then be analyzed in section 4 to deduce what is in general necessary for achieving super-human performance in board games.

2 Fundamentals

First, we introduce some fundamentals that are necessary to understand the current state of the art presented in section 3.

2.1 The Value Function

In this paper we deal with games where (starting with player one) two players alternate in taking actions leading from one game state s to another game state s'. When the game ends, i.e. when a state s^* is reached in which, according to the game rules, no more move is possible, both players receive a certain defined reward. Usually the winner gets $+1$ and the loser -1 with both parties receiving 0 in case of a draw.

To play effectively, it is necessary to know how good a certain game state is. This is usually captured by a concept called the *value function*. The value function is a function $v(s)$ that describes the reward player one is expected to receive when the game is in state s and both players continue playing optimally. In case of deterministic games this expected reward is precisely the best possible outcome assuming no player makes a mistake, i.e. if $v(s) = 1$ player one can force a win from this position regardless of what player two does. This is because if there was an action for player two that leads to a state s' with $v(s') < 1$, then $v(s)$ would not be equal to 1 in the first place.

Given the value function, it is easy to play perfectly. As player one simply selects the action that leads to a state s' with maximal $v(s')$ and as player two do the same but for minimal $v(s')$. When applied to the true value function, this greedy strategy is indeed optimal.

While this brief introduction should be sufficient for following the remainder of the paper, the interested reader can find a detailed discussion of the value function concept in [10].

2.2 Computing the Value Function

The actual value function can be defined recursively in a similar manner to the optimal strategy described in section 2.1. Let s' denote a state reachable by taking an action in s, then $v(s) = \max_{s'} v(s')$ if it is player one's turn and $v(s) = \min_{s'} v(s')$ if it is player two's turn. For terminal states the value function is defined by the game rules.

This recursion can be evaluated by a straight-forward depth first search, called the *Minimax* algorithm [11]. The emerging structure of a game state s connected to all the states reachable by taking an action in s is commonly referred to as the *game-tree*. The value of a node is computed by taking the max/min (as discussed in the previous paragraph) over the values of all child nodes. This new value is then passed upwards to be used for the evaluation of the parent node. Figure 1 shows an example of Minimax search in a game-tree.

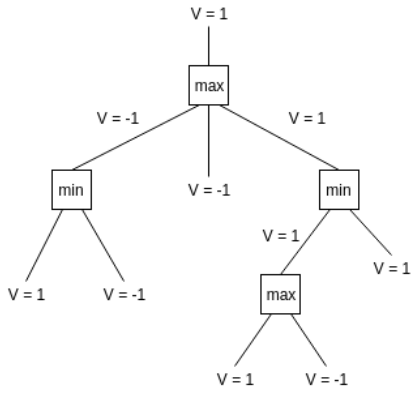


Fig. 1. An example of a Minimax search tree

Minimax always needs to examine the entire game-tree. For a game with branching factor b and an average length of t it has $O(b^t)$ nodes. It is worth noting that this number is typically much larger than the count of distinct games states due state and consequently subtree repetitions. Hence, the Minimax algorithm in its simplest form can only be applied to games with a small game-tree, like for

instance Tic-Tac-Toe with less than 9! nodes. There exist various optimizations to reduce the search tree while still retaining optimality. *Alpha-beta pruning* [11] immediately cuts off branches that cannot improve the optimum anymore. *Transposition tables* [11] keep track of already visited states to ensure that every value is only computed exactly once even if a state is reachable through different sequences of play. These and other techniques make it possible to address much more difficult problems. For instance, they enable perfect play in Connect 4, a game with to the order of 10^{12} distinct states, guaranteeing a win for the first player to make a move [1].

As an alternative to enumerating the whole search-space, one could sample tree trajectories and estimate the value of a position by averaging over the results from all sampled games visiting this state (see Figure 2). This approach is called Monte-Carlo tree search (or short MCTS). Usually trajectories are generated by choosing the child node with the highest $V(n) + U(n)$ where $V(n)$ is the current estimate of the value of node n and $U(n)$ is an upper bound. $U(n)$ gets smaller the more often the node is visited and at the same time larger the more simulations are run. This ensures that the algorithm generally selects nodes that are already known to have a high value, which means that they are probably good moves the opponent is likely to pick, but also explores some other moves with still uncertain value. It is in some sense similar to classical Tabu search [5] where particular good states are temporarily blocked to promote exploring different parts of the search space. A common choice for $U(n)$ is using the Upper Confidence Bound 1 (derived from the Hoeffdings Inequality [7]), usually referred to as the *UCT* algorithm, which guarantees that the search eventually converges towards the Minimax result [8]. The most interesting property of Monte-Carlo methods is that the estimate of the value function improves as more simulations are run, in contrast a Minimax algorithm that explores only half of the state-space is essentially useless. A survey of different MCTS variants can be found in [2].

Even though the way of computing the value function explained in this section is often referred to as searching, classical search algorithms like for example A* [11] are not directly applicable in this context. This is mainly because they are designed to find an action sequence that leads to a certain goal and not to determine the value of specific nodes. They are also not optimized to consider different actors with adversarial objectives.

2.3 Value Function Approximation

Estimating the value of a position in extremely difficult games like Chess or Go with $\approx 10^{46}$ [4] (not counting repetitions) and $\approx 10^{171}$ [21] distinct board positions respectively (and even more gigantic game-trees) just by using the algorithms described in section 2.2 is completely intractable, different techniques are necessary.

[1] John Tromp, http://tromp.github.io/c4/c4.html, visited: 2018-01-14

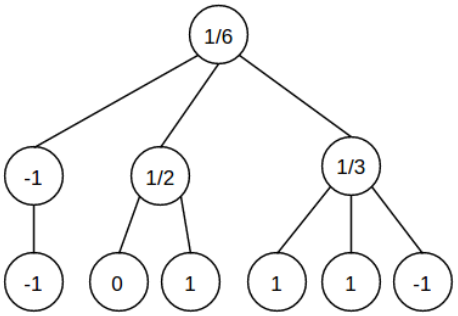


Fig. 2. An example of Monte-Carlo search

The value of a node is computed as the average of all its successor nodes.

A more promising approach is to estimate the value of a state from characteristics of the board position (i.e. which pieces are left on the board, where are they, etc.) either by using human expert knowledge to construct game-specific algorithms or by having the computer learn an evaluation function directly from data. However, since in many games just slightly changing the location of a single piece can turn a winning position into a losing one, just relying on an estimator is in many cases not sufficient for creating a strong player. Most strong game playing programs thus combine an efficient search algorithm with accurate value function approximation.

3 State of the Art

This section describes a representative selection of concrete solutions for playing different kind of board games at expert (to super-human) level. The discussed systems were chosen in way that they cover most of the common and successful techniques. In particular Stockfish (section 3.1) introduces search with value estimation, TD-Gammon (section 3.2) learning by self-play via TD-learning and Giraffe (section 3.3) a combination of those two approaches. AlphaGo (section 3.4) is an example where learning without supervision vastly outperforms all other previous methods and AlphaZero (section 3.5) is the strongest engine for several of the most difficult board games to date.

Other interesting systems not covered here include a self-learned Chess program *Knightcap* [1], a world-championship caliber Scrabble program *Maven* [13] and the perfectly playing Checkers program *Chinook* [12].

3.1 Stockfish

Stockfish [2] is one of the strongest chess-engines in the world playing much stronger than top human players. It uses the same general ideas as *Deep Blue* [3]

[2] T. Romstad, M. Costalba, and J. Kiiski, www.stockfish.com, visited: 2018-01-17

(the first ever computer program to beat a top human player in Chess) but several optimizations developed over the past 20 years lead to even higher playing strength leaving humans essentially chance-less.

The core of Stockfish is a highly efficient implementation of a depth-limited Minimax search while heavily utilizing alpha-beta pruning (described in section 2.2). As the latter is only effective when good moves are discovered early on since only then many branches can be cut off quickly, several heuristics are in place to make sure more promising options are explored early. Furthermore, for leaf-nodes with certain properties an additional special search is run to make sure critical situations, like for instance queen exchanges, are not misjudged. Other leaf-nodes are directly evaluated by a static scoring function. This metric uses a large number of hand-crafted features, many of which were designed by human grandmasters, incorporating a big part of what is known about chess to date. Some of those features are: synergy between different pieces, existence of certain promising board patterns, mobility of pieces or safety of the king (considering the structure of attackers/defenders). Stockfish can also use an opening book, that contains moves believed to be ideal in the early stages of the game (determined by humans and extensive precomputation), and an endgame solver that enables perfect play when only very few pieces remain on the board. While most of the evaluation function is hard-coded, some parameters like the weighting of different factors were learned by automatically trying out many different settings and evaluating them by self-play, some polynomial regression models are also part of the scoring function.

Stockfish's main strength comes from the evaluation function that gives high quality estimates while being cheap to execute due to its highly specialized implementation. This allows searching a very large number of positions (several millions) in little time.

3.2 TD-Gammon

The first program that beat the world champion in the game of Backgammon was *TD-Gammon* [20], a neural network trained exclusively by playing against itself and applying temporal difference learning [17].

TD-Gammon only uses a relatively shallow neural network with a single hidden layer to score positions without any kind of search. To select the next move to play, the neural network evaluates all positions reachable by legal moves (given the dice rolls) and then selects the action which leads to the position with the highest expected value (as the game is inherently stochastic this represents a real expectation and not just an approximation of a deterministic result). The network takes as input the raw board as well as some expert features developed for a predecessor program NeuroGammon [19].

The network was trained entirely by self-play using temporal difference learning (TD-learning). The main idea of TD-learning is to make the output of the network at time t similar to the output at time $t + 1$. An intuition behind this is that the value estimation should not rate a state as being good when one of the opponent's next moves can turn it into a bad position since if such move is

available, the initial position is definitively not a favorable one. When a game playing against itself has concluded, the error for every encountered state s_t is computed as the sum of the differences of the value predicted for s_t and the value predicted for each state in the future weighted by an exponentially decaying parameter λ. This assigns more importance to deviations from predictions in the immediate future. The network parameters are then updated in standard fashion in direction of the gradient as computed by back-propagation on the errors for every state.

Due to the stochasticity of Backgammon, the value function is generally relatively smooth as slight variations of the board state don't change the value too drastically (most of the time at least), thus the neural network by itself works well enough and additional searching does not seem to be necessary.

3.3 Giraffe

Stockfish (described in section 3.1) relies heavily on its carefully hand-tuned state evaluation. Several attempts have been made to replace this component by an algorithm learned exclusively from raw data, in particular a neural network. One of the most successful of those attempts is *Giraffe* [9].

Giraffe uses a relatively standard chess engine search (similar to Stockfish) but performing position evaluation via a small three-layer neural network. In addition to the actual board state encoded as a list of the coordinates of every piece (instead of a bitmap with a 1 indicating that a certain piece-type is present at that position), the network is also supplied with some low-level features, like the number of squares a piece can move in each direction or the lowest valued attacker and defender. While those things could theoretically also be figured out by the network itself, this would require a substantially larger architecture which would consequently cause both training and evaluation to be slower. Furthermore, to save even more computation time, the first layer of the network is not fully connected but the input features are split into three different groups that are each only connected to a third of the neurons in the first hidden layer respectively (the other two layers are fully connected though). Training data was taken from both expert human and computer games. Additional samples were generated by applying some random moves to the existing data so that the network also learns to properly judge bad positions that occur frequently during the search even though they are rather rare in real games. The neural network was trained by randomly sampling a position from the training data, playing twelve moves and then applying TD-learning as explained in section 3.2. To speed up training significantly, they also bootstrapped the neural network with a very simple evaluation function before starting the self-play learning.

Overall, Giraffe performed only slightly worse than Stockfish on a standardized dataset for testing chess-engines when enough computation time was available but reached just candidate master level in real games. One of the main reasons for this is that Giraffe can only consider significantly less positions (around one order of magnitude) in the same time due to the neural network being much slower than Stockfish's specialized evaluation function.

3.4 Alpha Go

Alpha Go was the very first Go program to beat a top human professional back in 2016. AlphaGo put together Monte-Carlo tree search with several different neural networks that were trained both from expert data and also via reinforcement learning by self-play [14].

AlphaGo uses a value-network that predicts the value of a board position as well as two policy networks that estimate move probabilities $\mathbf{p}$ (indicating how good it thinks every available move is). The value network and the slow policy network use a convolutional architecture [6] receiving as input the raw board positions and some simple tactical features like the number of liberties or the result of ladder-searches (play-sequences that span the entire board but can be easily evaluated with a simple algorithm). The fast policy network is just a linear softmax over small pattern based features (i.e. the presence/abscence of particular piece configurations on the board). All those networks are initially trained to predict moves/game outcomes based on a database of expert games. Additionally, the slow policy network is improved by reinforcement learning via self-play (just by sampling from the policy, without the Monte-Carlo search explained in the paragraph below) using the policy-gradient method [18] where the policy is adjusted in the direction of maximum expected reward, in this case with the highest probability of winning. The value network is also improved by additional training on the generated self-play data.

For actually playing, AlphaGo uses the neural networks as an integral part of a Monte-Carlo tree search. The next node is selected by a slight variation of the UCT algorithm explained in section 2.2 where the calculation of $U(n)$ takes into account the prior move probabilities estimated by the slow policy network, hence choosing states with higher priors more often. When a leaf node is encountered, both the slow policy network and the value network evaluate the position exactly once. Then one full game (also called rollout) starting from this position is played by sampling from the fast rollout policy. The value of the leaf-node is finally calculated as a linear combination of the estimated value from the value network and the result of this rollout. It is then propagated upwards the tree to update the predecessor values in standard Monte-Carlo fashion. AlphaGo ultimately plays the action that leads to the node with the highest visit count.

3.5 Alpha Zero

AlphaGo as described in section 3.4 was consecutively improved to learn entirely from self-play [16]. This eventually lead to a more general version, called *AlphaZero*, which achieves superhuman performance not only in Go but also in Chess and Shogi [15].

Similar to its predecessor, it combines Monte-Carlo tree search with a deep neural network. AlphaZero utilizes a single convolutional neural network with residual blocks [6] to estimate both the value of a state v and a vector of move probabilities $\mathbf{p}$ at the same. Every leaf-node encountered during the Monte-Carlo search is evaluated exactly once by this neural network. The estimated v is then

immediately backed up to refine the move probabilities π of its predecessors, no full rollouts are played. The neural network is trained entirely by playing games against itself always using the most current version of the network. Whenever a game is complete, the final move probabilities $\pi_{\mathbf{t}}$ (after the search) together with the real game outcome z are used to update the parameters of the neural network via gradient descent.

The input of the neural network is presented in form of an $N \times N$ plane for every piece-type and every player with a one indicating that a certain piece of a certain player is present at this location on the board. Additionally, they also include a small number of extra planes to encode special rules, for example the number of move repetitions or whether or not castling is still available in Chess. The output for Chess and Shogi is encoded in rather interesting fashion, the network returns a full plane of probabilities for every type of move where the position in the plane indicates from where to pick up the piece and the number of the plane which action to take, for instance moving a certain figure three fields to the east.

4 Analysis

We will now analyze the different solutions described in section 3 to deduce what is in general required to construct a program for playing a complex board game at super human level.

Structured searching is hard to avoid as the value function of most classic board games is highly non-smooth, i.e. a slight change of the board position can lead to a completely different outcome. Accurately handling this without any kind of look-ahead seems difficult (or even impossible). For games that involve randomness this is not necessarily the case, in Backgammon moving a single stone by one field does not make a big difference most of the time as all dice roll results have equal chance of happening. This is also the main reason why TD-Gammon could get away using just a neural-network with no (or in the improved version [20] a two-turn) look-ahead. All the other programs discussed in this paper rely heavily on searching, DeepMind reports that using just the slow policy network for playing Go achieves results comparable with the existing programs before AlphaGo but gets nowhere near the strength necessary to beat top human players [14].

While searching is necessary, it is definitively not sufficient due to the absolutely enormous amount of states in complex games. The key to making search useful is to ensure that most of the time is spent considering actually interesting positions, i.e. positions that are likely to occur in real games between competent players. Stockfish achieves this by considering the (according to some heuristics) most promising moves first so that alpha-beta pruning can quickly cut off bad branches as moves with high value are typically already discovered early on. AlphaGo goes a step further and uses a deep neural network to predict the value of every possible move in one pass, so that the Monte-Carlo search can focus mostly on the moves the network considers relevant. But even with those

optimizations it is only possible to search up to a certain depth (that might vary depending on how interesting a position is) after which the search has to terminate and the program needs to rely on a different way of estimating the value of this leaf-nodes.

There are many different approaches towards designing such a value function estimator. One could either design a highly specialized algorithm based on human understanding of the game (as it is done in Stockfish) or learn a function using a more general estimator, like a neural network. One of the main problems here is that the latter ones are usually a lot slower with the consequence that the search can consider much fewer positions resulting in overall lower playing strength. For instance this can be observed in the Giraffe project. Learned functions must thus yield more accurate estimates than their hand-engineered counterparts to make up for their lower execution speed. AlphaGos convolutional neural network outperforms previous evaluation methods by a fairly significant margin and can thus achieve much higher playing strength despite the estimator being rather slow to evaluate. A common trick to get away with a smaller and hence also faster neural network is to pass some useful additional features that are rather difficult to learn from data but very easy to compute with specialized algorithms (this is done both in Giraffe and AlphaGo).

In addition to training the value estimator using data from human expert games (done in AlphaGo), training by self-play is not only possible (see TD-Gammon, Giraffe and AlphaZero) but even seems to be essential for reaching super-human performance with learned value estimation. AlphaGo with the neural networks solely trained with expert data was not enough to beat human professionals, first continuous improvement by self-play eventually lead to beating one of the best human players. AlphaZero learns entirely from playing itself and becomes stronger than the best engines to date in both Shogi and Chess in which previous programs relied mostly on hand-engineered features. Another big advantage of this approach is that the program is in no way biased by what we currently think is the best way to play but it can discover new (and possibly better) strategies completely on its own. Both TD-gammon and AlphaGo had a significant impact on how humans play the games as they demonstrated that several moves which were generally believed to be suboptimal are actually stronger than their alternatives.

The most effective way of training a network by self-play seems to be to use the refined values estimates after the search as new targets. This causes the estimator to learn to approximate the result of the search leading to a better overall value estimate and therefore to an even better approximation when searching with the updated network, and so on. One can either use the results of the search directly to update the network for the position the search was run for (as it is done in AlphaZero) or use the results of some state in the future, i.e. via TD-learning (done in TD-Gammon and Giraffe). The latter promotes temporal consistency, which means that when the network predicts that state s is good, then it should not predict that a successor state s' is bad because then s should not have been classified as a good state in the first place.

Previously most successful game playing programs were designed similar to Stockfish, an efficient search with a hand-crafted evaluation metric with the main strength coming from being able to check a huge number of positions. This is very unlike how humans play. Humans are able to judge positions very accurately based on certain properties/features of the situation and only actively consider (calculate) very few (in comparison to computers) actual move sequences. This is because humans can identify promising moves and rule out clearly bad ones extremely quickly, something computers struggle with. However, as it can be seen at the examples of AlphaGo and AlphaZero, developments are going in the direction of playing more human-like. AlphaZero considers only a fraction of the positions (around 16000) Stockfish does ($> 10^6$) but therefore uses a deep convolutional neural network for significantly better judgment of the encountered states.

In general, AlphaZero seems to strike a very good middle ground between what machines excel at, i.e. searching many different position, and what humans excel at, recognizing patterns on the board to evaluate a position. The former allows it to find the perfect local play sequence and the latter allows it develop general strategy, something conventional Chess-engines like Stockfish are typically not very good at (but which is not really necessary to beat error-prone humans).

This very effective combination of two of the most powerful techniques in computer science, search and neural networks, does not only seem applicable to other classic board games but can probably also be extended to lead to advances in other domains, especially as neural networks already work well for processing natural data such as images or audio. Combining them with a search algorithm could be a way to make up for their weakness in performing complex structured strategic considerations.

5 Conclusion

In this paper we discussed several interesting approaches towards building agents for playing classical board games at expert level.

Combining an efficient search algorithm with an accurate value approximation function can lead to super-human performance in classic games that have been studied for centuries. Furthermore, this static evaluation of game states does not need to be hand-designed using existing knowledge about the games but can instead be learned exclusively by having the machine play itself. This can lead to the discovery of novel strategies that were never though of before and which eventually change the way humans play the game.

Since games like Go test incredibly complex concepts that are also very relevant in real-world applications, it is very likely that recent breakthroughs, like AlphaZero, can be adapted to other domains leading to significant advances of the current state of the art.

References

1. Jonathan Baxter, Andrew Tridgell, and Lex Weaver. Knightcap: A chess programm that learns by combining td (lambda) with game-tree search. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 28–36. Morgan Kaufmann Publishers Inc., 1998.
2. Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.
3. Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.
4. Shirish Chinchalkar. An upper bound for the number of reachable positions. *ICCA JOURNAL*, 19(3):181–183, 1996.
5. Fred Glover. Tabu searchpart i. *ORSA Journal on computing*, 1(3):190–206, 1989.
6. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
7. Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.
8. Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *ECML*, volume 6, pages 282–293. Springer, 2006.
9. Matthew Lai. Giraffe: Using deep reinforcement learning to play chess. Master's thesis, Imperial College London, 2015.
10. Kevin Leyton-Brown and Yoav Shoham. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2(1):1–88, 2008.
11. Stuart Russell and Peter Norvig. *A modern approach*. Prentice Hall, 1995.
12. Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. Checkers is solved. *science*, 317(5844):1518–1522, 2007.
13. Brian Sheppard. World-championship-caliber scrabble. *Artificial Intelligence*, 134(1-2):241–275, 2002.
14. David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
15. David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.
16. David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.
17. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
18. Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

19. Gerald Tesauro. Neurogammon wins computer olympiad. *Neural Computation*, 1(3):321–323, 1989.
20. Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.
21. John Tromp and Gunnar Farnebäck. Combinatorics of go. In *International Conference on Computers and Games*, pages 84–99. Springer, 2006.

Understanding of urban intersections in autonomous driving

Maximilian Sbardellati
e01526262

Scientific Working
TU Wien, Austria
e01526262@student.tuwien.ac.at

Abstract. In this paper different approaches for urban traffic scene understanding are discussed and compared. The focus will be on the detection and understanding of urban traffic scenes like intersections based on systems using light detection and ranging (LIDAR) and monocular or stereo vision. Since the input data produced by LIDAR and monocular or stereo vision are so different, most systems use either one or the other. Nevertheless a system that can work with both LIDAR and vision based input will also be discussed.

Keywords: autonomous driving, LIDAR, stereo vision, traffic pattern, scene understanding

1 Introduction

Autonomous driving is a topic that caught the interest of researchers all over the world. With sensors becoming better and more available enormous progress has been made in recent years [1].

But even though simple scenarios like driving on highways or country roads have well working solutions, autonomous driving on urban roads is still a big challenge. Urban roads have lots of characteristics that make them way more complex than just driving on the right lane with the correct speed on a highway.

The focus of this paper is to discuss and compare different approaches for understanding one of the key characteristics of urban traffic scenes, intersections. Intersections feature a wide set of difficulties for self driving vehicles. The first being the very detection of intersections and the classification of the intersection style (T-shaped, +-shaped or even more complex shapes) [2]. Another challenge is the semantic understanding of the current traffic scene that unfold while a vehicle is coming near an intersection. To make the correct driving decision the vehicle has to be able to understand not only which type of intersection it is about to cross, but also what all the other traffic participants are about to do [3, 4]. This task can get troublesome in urban environments since they often feature blind intersections without mandatory stopping as illustrated in Fig.1.

Most of todays vehicles are already equipped with localization systems consisting of a Global Positioning System (GPS) and a Geographic Information

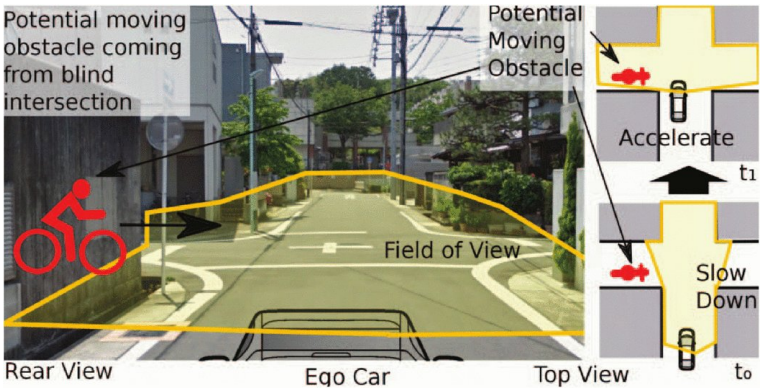


Fig. 1: A blind intersection without a line stop. The vehicle does not see if there are any obstacles approaching from left or right. The vehicle should slow down to be able to stop if necessary [5].

System (GIS). Even though one might think that GPS and GIS would be sufficient for the correct detection of intersections, the task remains difficult. The main disadvantage of GPS and GIS is, that they do not provide information about other traffic participants. Additionally GIS is often missing and GPS data is invalid or not reliable enough in many places. Also longitudinal and lateral errors that are created by interference or noise affect the intersection detection notably [2, 6].

Using vision based approaches was the common way to tackle the problem of getting input data for autonomous driving systems in the past [7]. Under optimal circumstances vision based systems are reliable. However, if the conditions are not optimal, for example in adverse weather or on streets without lane markings or missing traffic signs, vision based approaches often reach their limits. Gehring et al. [8] introduce a method to increase the robustness of stereo vision during adverse weather. Their approach uses Semi-Global Matching (SGM) [9] and combines it with a temporal prior, to predict future events, and a scene prior, which is calculated from a generated representative traffic scene. With this combination Gehring et al. [8] achieved a reduction of false positives (false obstacle detection) by a factor of three and also could increase the correct detection rate by more than 2%. Fig.2 shows the improvement achieved during rain and snowfall.

LIDAR has only become popular in recent years [1]. It works similar to radar, but instead of sending and receiving radio waves, LIDAR sends light pulses and detects the light that has been reflected by its surroundings. Conventional LIDAR systems consist of a single laser that fires into a rotating mirror. The resolution of the point cloud, that can be retrieved from such systems, is way to low to be used efficiently in autonomous driving. High-resolution LIDAR systems, that are used nowadays, were first used in the Defence Advanced Research

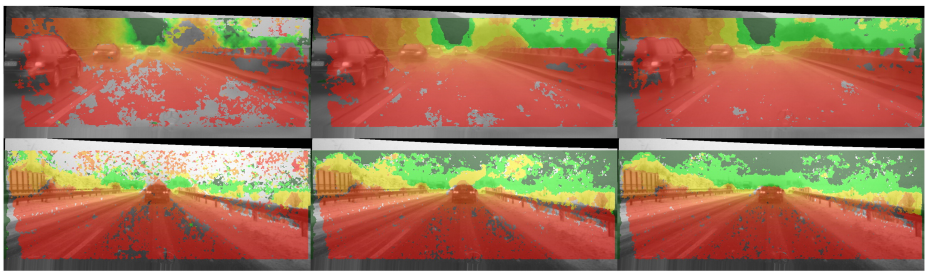


Fig. 2: Rain scene(top) and snow scene (bottom). Stereo reconstruction(red = near, green = far) for this scene using SGM(left, SGM with temporal prior(center) and width both temporal and scene prior(right) [8].)

Project Agency's (DARPA) "Grand Challenge" in 2007[1]. Most of these systems use a rotating head that features 64 lasers, resulting in remarkably high resolution point clouds. The downside of this much data is, that it becomes a troublesome task to compute and evaluate the data in real time [10].

1.1 Overview of the paper

The rest of the paper is organized as follows. In Section 2 approaches for understanding urban traffic scenes that will be discussed in detail after a short introduction into Hidden Markov Models (HMM) is given in Section 2.1. Section 3 will feature the experiment results of the introduced approaches and their comparison followed by a conclusion in Section 4.

2 Scene understanding approaches

In this section three approaches for classifying intersections and understanding their semantics will be discussed. Since Hidden Markov Models (HMM) are often used in these approaches for classification of the intersection style or the lane association of objects, a short introduction to HMM is given.

2.1 Hidden Markov Models

Hidden Markov Models have been introduced in the early 1970s, but only gained in popularity in the late 1980s. The main reason for their wide use is, that when applied correctly they can be used for solving many real world problems [11].

Before introducing the problems we can solve with HMMs $\lambda = (A, B, \pi)$ we need to define its elements using a simple coin toss as example [11, 12], that is illustrated in Fig.3. Imagine 2 coins, $c1$ and $c2$, being randomly tossed.

[1] DARPA Urban challange: http://archive.darpa.mil/grandchallenge

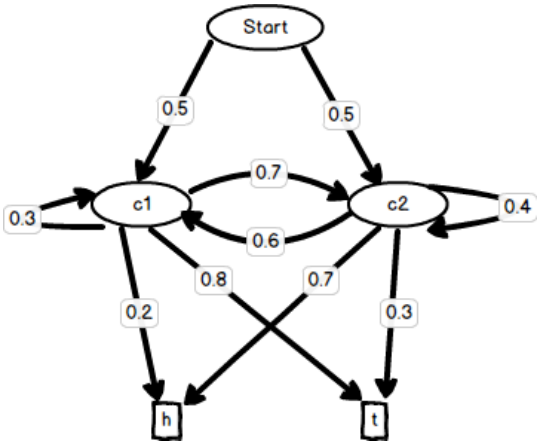


Fig. 3: Example for a HMM. $S = \{c1, c2\}, V = \{h, t\}, A =$ see Eq:(2), $B = \{b_{c1}(h) = 0.2, b_{c1}(t) = 0.8, b_{c2}(h) = 0.7, b_{c2}(t) = 0.3\}, \pi = \{0.5, 0.5\}$

- The **states** of the HMM in this example would be the tossing of $c1$ or $c2$. We denote the states as $S = \{c1, c2\}$. N is the number of states in the model. In our case $N = 2$. We also have a sequence of states $Q = (q_1, q_2, ...q_t)$ where $q_t \in S$ is the state at time t [11].
- The **observations** are the physical output of the modelled system. If you are tossing coins the output is the side the coin lands on, head(h) or tail(t). $V = \{h, t\}$ is the set of observations and $M = 2$ the size of V. If we toss coins t times we get an observation sequence $O = (o_1, o_2, ..., o_t)$ where $o_t \in V$ is the observation at time t [11].
- Using **Hidden** Markov Models, in contrast to Markov Models, sometimes we do not know in which state we currently are. We only know the probability of changing from one state to another, or staying in the current state. Therefore we get a **probability matrix** $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N \tag{1}$$

  describes the probability of state S_i being followed by state S_j in the state sequence [11].
  In our example

$$A = \begin{pmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{pmatrix}. \tag{2}$$

- Same as the states the observations can also be hidden (unknown). In both cases we need the **probability of an observation during a given state** $j, B = \{b_j(k)\}$, where

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \tag{3}$$

  [11].
  In our scenario $B = \{b_{c1}(h) = 0.2, b_{c1}(t) = 0.8, b_{c2}(h) = 0.7, b_{c2}(t) = 0.3\}$.

– Since we have to start assuming that one of the states was chosen as the first one, we also need the **initial state distribution** $\pi = \{\pi_i\}$, π_i describing the probability of state S_i being q_1 (the first state in the state sequence Q). $\pi = \{0.5, 0.5\}$ in our coin toss example [11].

Given all these parameters, we can solve problems of HMM, that are useful in real-world applications. The most important of these are [12]:

1. The learning of a HMM λ, given a training data set of observations O
2. Estimation of a HMM $\lambda = (A, B, \pi)$ that maximises $P(O|\lambda)$
3. Find the best state sequence Q given O and λ
4. Compute the probability that an observation sequence O was created by a given HMM λ, $P = (O|\lambda)$

2.2 LIDAR based intersection recognition

The approach by Zhu et al. [2], that is discussed in this section, aims at recognising intersections and classifying them as road segements, T-shaped intersections or +-shaped intersection. For this purpose they use a Velodyne HDL-64ES2 LIDAR sensor[2], a high-end sensor with 64 lasers that produce 2.2 million points per second with an accuracy of less than 2 centimeters and a range of 120 meters. They preprocess the point cloud, that is received from the sensor and then apply a beam model on the data to create feature vectors.

What distinguishes this approach from related ones is that it models the intersection detection as a classification problem. Therefore any method that is based on machine learning can be used to solve the issue. Furthermore the starting point of the beam model, used to create the feature vectors, is not fixed. It alters with the velocity of the vehicle. The faster the vehicle is going, the further the starting point is away. Also their algorithm is capable of real time intersection detection and classification [2].

Data preprocessing: The output of the LIDAR sensor is a 3D point cloud of the surrounding area. To make the following steps easier this 3D point cloud is projected into 2D space and other traffic participants, like car or pedestrians, are removed using following steps:

1. The data is looked at from above. For each frame of data a **grid map** is created. For all quadratic cells in the grid map the variance of elevation of the corresponding points is computed [2].
2. If the variance of elevation is greater than a given threshold the value of the grid cell in set 1, else it is set 0. The result of this operation is a 2D **binary image** that resembles the scene in bird-eye view [2].
3. All regions of ones are surrounded by bounding boxes [2].
4. Using the width and length of the bounding boxes, vehicles and pedestrians are detected [2].

[2] Velodyne LiDAR HDL-64E: http://velodynelidar.com/hdl-64e.html

5. All regions belonging to vehicles or pedestrians get set 0. The rest of the binary image, now without any unwanted obstacles on the road, is used for the further process [2].

Beam model and feature creation: Next a beam model is used on the **binary image**. First the distance D between vehicle and starting point of the model is calculated like

$$D = 5 + v * t \tag{4}$$

where v is the velocity of the vehicle and t is a pre-defined constant [2].

The starting point is in front of the vehicle with distance D. From there 360 beams, with a 1°angle between 2 adjacent beams, are sent. Each beam scans the binary image linearly and stops, if it hits an obstacle (a 1 in the binary image). If it doesn't hit anything it will stop after a pre-defined length L [2].

With the normalized length of all 360 beams a 360-D feature vector is created for classification. This can be done, because the normalized length histogram for each class we want to classify (road segment, T-shaped intersection, +-shaped intersection) is distinctive. A beam pointing towards a road segment is longer than a beam pointing directly to a curb, because it can travel further before hitting an obstacle. So in the normalized length histogram there is a peak for each possible direction the vehicle can go. As illustrated in Fig.4 +-shaped intersections have 4 and road segments have 2 local peaks [2].

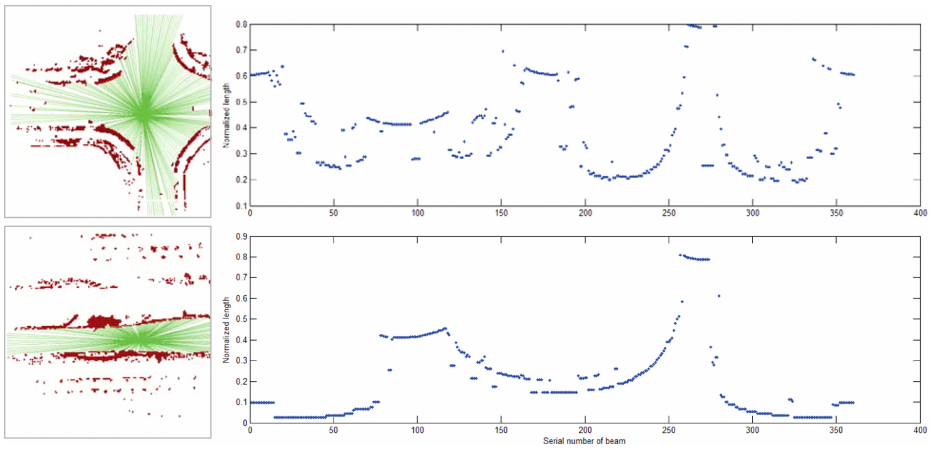


Fig. 4: Example of beam model histogram in intersection(top) and road segment(bottom) [2].

Classification: After all features are created, the problem of classification is then formulated as a supervised machine learning problem with labelled classes.

In the approach of Zhu et al. [2] it is considered a 2 stage two-class classification problem. First between intersection and road segment and then, if it was classified as intersection, between +-shaped and T-shaped intersections. They use a Support Vector Machine (SVM) [13] because of its good ability to learn from small training sets.

2.3 Small-scale intersection scan model

Yang et al. [6] discuss the problem of intersection recognition and classification on a similar basis as discussed in Section 2.2.

What separates them from the approach of Zhu et al. [2] is that their classification of intersections, besides road segments, T-shaped and +-shaped intersection, also differentiates Y-shaped intersections. Additionally they use not only real-time detection results, but also historic ones. To do so they use a particular Hidden Markov Model with the real-time and historic detections used as the observation states. Another unique selling point is the capability of this approach to deal with both LIDAR and visual, in this case monocular vision, data [6].

The Intersection Scan Model: The Intersection Scan Model (ISM), similar to the beam model in 2.2, is used to detect the traversable detections a vehicle could possible go on an approaching intersection.

The output of the ISM is an intersection model which is represented by

$$M_i = \{N_a, P_{TD}, P_c\}, \tag{5}$$

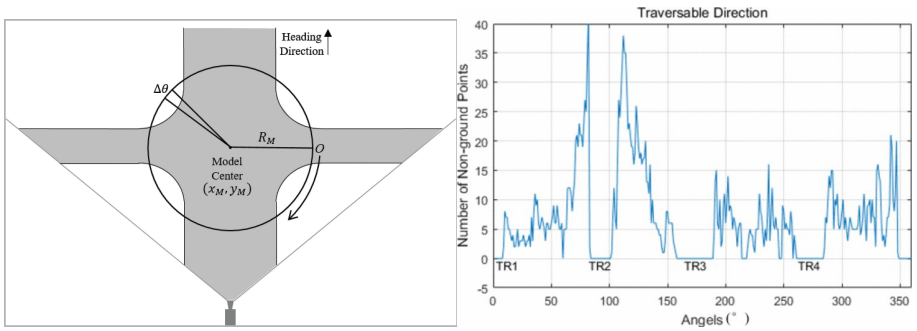
where N_a is the number of traversable directions, $P_{TD} = \{TR_{c1}, TR_{c2}, ..., TR_{cNa}\}$ contains the angle for each traversable direction and P_c is the intersection center [6]. M_i will later be used as input for the Hidden Markov Model.

As seen in Fig.5a the ISM only scans in the region in front of the vehicle. It has some parameters like the scan radius R_M, the scan resolution $M = 360/\Delta\theta$ and the contour C of the model that can be adjusted to achieve better detection results and are adaptive to the environment. For easier understanding $\Delta\theta = 1$ resulting in a resolution of 360 beams when scanning 360°around the ISM center [6].

Instead of stopping a beam when an obstacle is hit, as in the approach of Zhu et al. [2], each beam continues the scan till it reaches the length of R_M and returns the number of Non-ground Points (obstacles like road curbs) it hit until that. From these values a graph can be generated (see Fig.5b) that shows the traversable directions [6].

Even though continuous troughs in the graph indicate that there are no obstacle in this direction, these angle intervals can only be marked as traversable directions TR_x, if the interval is also long enough. An interval too short means, that the vehicle could not fit through this gap and that it possibly isn't even a road. To simply further calculations, only the center TR_{cx} of each interval TR_x is used resulting in the previously mentioned angles for each traversable intersection $P_{TD} = \{TR_{c1}, TR_{c2}, ..., TR_{cNa}\}$ [6].

Based on the number of angles N_a and P_{TD} a classification between different intersection types can be made. For example a T-shaped and a Y-shaped intersection both have 3 traversable directions, but can still be seperated by their angles P_{TD}, which differ between the 2 intersection types. To further improve the detection results not only results from a single frame are used, but also historic ones. For this a HMM is used [6].



(a) (b)

Fig. 5: (a) The ISM. (b) The graph for a +-shaped intersection as can be seen by the 4 continuous troughs of the curve [6].

Hidden Markov Model: To be able to use a HMM effectively for the intersection detection, the driving scene is considered a Markov process, which means that we can be sure that the current scene only is related to the last few scenes but not to scenes longer in the past. The thought behind using a HMM is that it is perfectly suited to detect hidden states from observable observations, as discussed in Section 2.1 [6].

In [6] the hidden states are road, T-shaped intersection, +-shaped intersection and other shaped intersection,

$$S = \{s_r, s_T, s_+, s_o\}. \tag{6}$$

The observations for each frame consist of the output of the ISM (see Eq.5) and the previous detections results [6],

$$V = \{M_i, v_{pre}\}. \tag{7}$$

The goal is to find the optimal state sequence $Q*$ given an observation sequence Q and a HMM λ, which is one of standard problems that can be solved with HMMs as stated in Section 2.1. On of the most used algorithms for this

problem is the Viterbi algorithm [14]. The Viterbi algorithm needs the probability matrix $A = \{a_{ij}\}$ and the probability of an observation during a given state $B = \{b_j(k)\}$ for it's calculations [6].

Since we do not know both the Baum-Welch algorithm [15] is used to train them to the HMM. The Baum-Welch algorithm in an expectation maximization algorithm and finds the maximum likelihood estimation of the parameters A and B given a set of observed feature vectors [6].

The ISM in LIDAR: The ISM can handle point cloud input from LIDAR systems. Yang et al. used a Velodyne HDL-64E LIDAR which is equipped on the top of a vehicle 2.25 meters above the ground. Since each frame sent by this LIDAR system contains more than 130.000 points, again the data needs to be preprocessed before further calculations can be made [6].

The pretreatment consists of building a grid map, the segmentation between ground and obstacles and the clustering of objects as shown in Fig.6 and similar to the preprocessing described in Section 2.2. The output of the pretreatment is a 2D grid map that labels points as ground or obstacle [6].

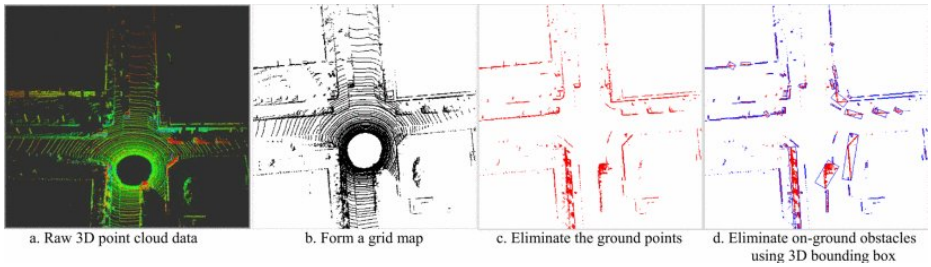


Fig. 6: Petreatment of point cloud data [6].

Afterwards the ISM is used to recognise the road environment in front of the vehicle. The center of the ISM is set about 9 meters in front of the vehicle and C is set to be a circle (see Fig.5a). C could be any shape as long as the values returned are normalized. The detection result of the ISM is then combined with some consecutive historic frame to improve the the accuracy of the intersection detection [6].

The ISM in monocular vision: Yang et al. also describe the use of the ISM on monocular vision data. In comparison to stereo vision monocular vision data lacks depth information which makes the task of scene understanding an even harder one.

For the ISM to work we need pixel wise segmentation of the input image. To achieve this a Fully Convolution Network - SegNet [16] is used. SegNet is able to model appearance (road, building), shape (Cars, pedestrians) and understand

the spatial-relationship (context) between classes like road and footpath. Additionally it is efficient in terms of memory and computational time, which makes it suitable for driving scene understanding [6].

Still the result by only using SegNet is not sufficient, because it returns us many fuzzy boundaries. To get rid of those a sophisticated segmentation by Dense Conditioned Random Field (Dense CRF) [17] is used. This method uses a fully connected CRF that is defined on the complete set of pixels in an image and is able to handle the immense graph, that this operations returns, extraordinarily fast [17]. As shown in Fig.7 after the Dense CRF there are no more fuzzy boundaries [6].

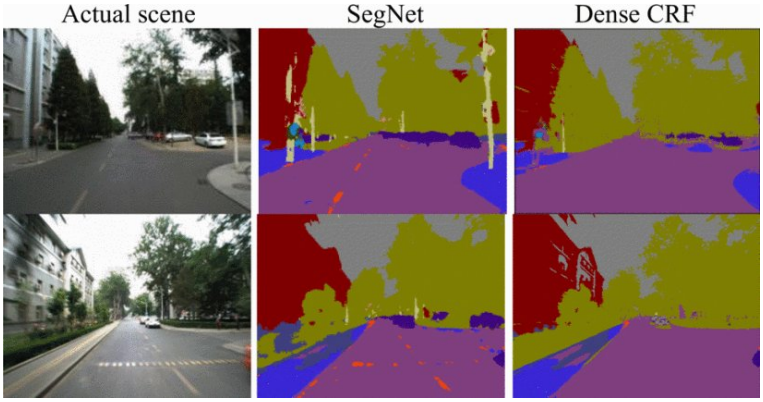


Fig. 7: Comparision between segmentation results of SegNet and Dense CRF [6].

When using monocular vision the model contour of the ISM is set as a rectangle to deal with the projection relationship of the camera (see Fig.8a). The model center is then set at the center of this rectangle. The result of the scan is shown in Fig.8b. Due to the non-linear angle changes it is harder to correctly interpret the result. The Moving Windows method is used for this purpose with the window size adapting according to angular distribution as is shown in Fig.8b where it is large for $TR1$ while small in $TR2$. Again the HMM is used for consideration of historic detection result in combination with the current one [6].

2.4 Understanding semantics by modelling traffic patterns

In Section 2.2 and 2.3 the main goal was to classify the intersection type. The approach by Zhang et al. [4] look at the intersection problem in a different way. Their main goal is not just to classify intersections into T-shaped and +-shaped, but also to understand the high-level semantics of the current traffic scene in the form of traffic patterns as seen in Fig.9. The experiments of [4] showed that only a small number of learned traffic patterns is enough to handle most of the scenarios at signalized intersections. Additionally their model is able to detect

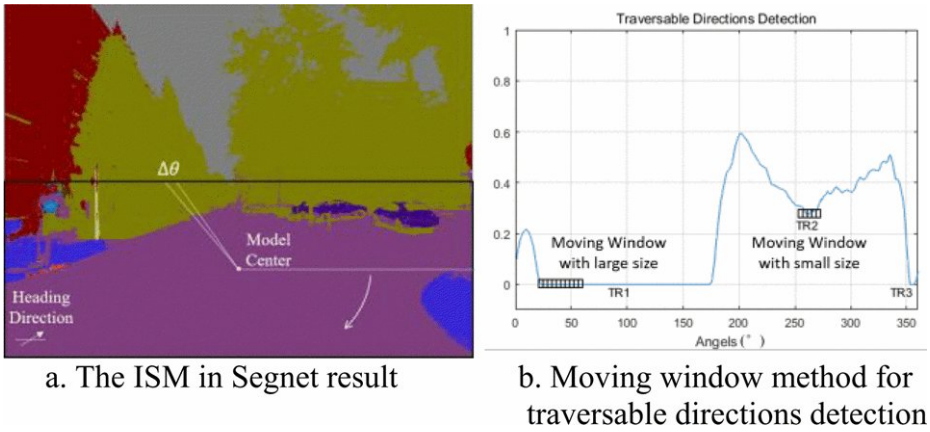


Fig. 8: Detection process in monocular vision [6].

other high-level knowledge like the current traffic light scene, without explicitly detecting it.

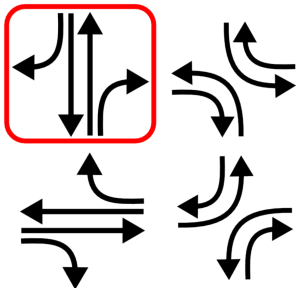
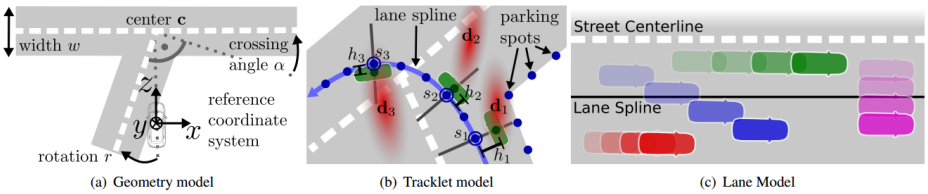


Fig. 9: Example of traffic patterns [4].

Modelling Traffic Patterns: As stated before the model of [4] first learns a subset of predominant pattern from training data. To interpret traffic scenes these patterns are recovered from short monocular video sequences and then vehicles are associated with the corresponding lanes.

The geometry model is defined as $R = \{c, r, w, \alpha\}$ where c is the intersection center, r the orientation of our own car in respect to the intersection, w defines the street width and α is the intersection crossing angle (see Fig.10a) [4].

For the understanding of traffic pattern and the vehicle-to-lane association, semantic segmentation, 3D tracklets and vanishing points are employed. The generative model is defined as:



(a) Geometry model (b) Tracklet model (c) Lane Model

Fig. 10: (a) road model parameters $R = \{c, r, w, \alpha\}$, (b) tracklet with 3 detections. Red: uncertain object detections d_i in 3D. Green: True location of the vehicle along the normal of the lane spline at distance h_i. Blue: lane spline with associated s_is (blue circles). (c) possible locations of a vehicle inside a lane [4].

$$p(\varepsilon, R) = p(R)p(T|R)p(V|R)p(S|R) \tag{8}$$

where the image evidence $\varepsilon = \{T, V, S\}$ consists of the vehicle tracklets $T = \{t_1, ..., t_n\}$, vanishing points $V = \{v_f, v_c\}$ and semantic labels S [4]. For the likelihood calculation of V and S a state of the art approach by Geiger et al. [3] is used. To determine the vehicle tracklets T a unique **traffic-aware tracklet model** has been developed by Zhang et al. [4].

Traffic-Aware Tracklet Model: To achieve the goal of estimating the lane association of vehicle, the tracklets of all vehicles in the scene must be determined. Drivable locations are symbolized by splines, that connect incoming and outgoing lanes of an intersection. The latent variable l represents the lane index a tracklet is associated with. Additionally the stop-or-go state, the longitudinal position and the lateral position of the tracklet are modelled [4].

The quality of the tracklet observations are essential for this approach. A 3D tracklet is defined as a collection of object detections $t = \{d_i, ..., d_M\}$ and each detection $d_i = \{f_i, b_i, o_i\}$ consists of the frame index $f_i \in \mathbb{N}$, the object bounding box $b_i \in \mathbb{R}^4$, and an orientation probability histogram $o_i \in \mathbb{R}^8$. For the detection of objects a well known detector by Felzenszwalb et al. [18] is used [4].

The computation of tracklets takes place in tow stages: The first stage builds short contiguous tracklets by associating detections using the hungarian method [19]. Additionally bounding boxes are predicted over time with the use of a Kalman filter [20]. In the second stage, again using the hungarian method, this time on tracklets, occlusion is overcome by joining tracklets that are up to 20 frames apart and the bounding boxes are projected into 3D using error propagation [4].

To model the detected object positions in relation to the given spline curves a few more variables need to be introduced. $s \in \{1, ..., S\}$ describes the anchor point of an object at the spline curve, which corresponds to its longitudinal position, and $h \in \mathbb{R}$ denotes its true location along the normal direction of the spline, or it lateral position, as illustrated in Fig.10b. $b \in \{stop, go\}$ stands for the stop-and-go status of a tracklet and g_i represents the pair $g_i = \{s_i, b_i\}$ to

simplify further notation. Note that h and g are hidden states, that will later be used in a HMM [4].

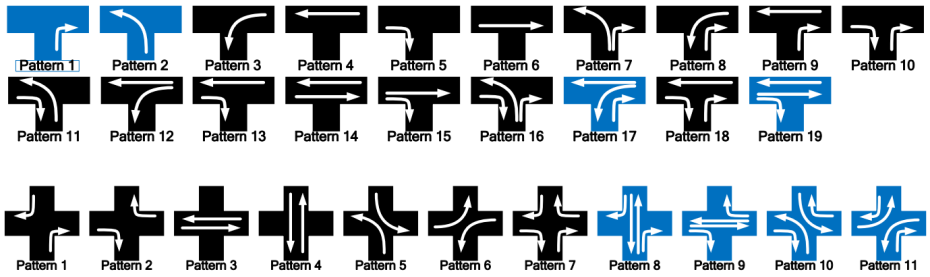


Fig. 11: All possible traffic flow patterns [4]

To be able to discuss traffic semantics, the latent variable a, which represents the possible traffic flow pattern (see Fig.11), is introduced. With that now the probability distribution over all tracklet observations can be described as

$$p(T, R) = \sum_{a} p(a) \prod_{n} \sum_{l_n} p(l_n) p(t_n | l_n, a, R) \tag{9}$$

where uniform priors are assumed for the patterns a and the lane assignments l since we don't know any distributions of these values. The likelihood of a single tracklet $p(t|l, a, R)$ is defined as

$$p(t|a, l, r) = p(d_1|a, l, R) \prod_{i} p(d_i|d^{i-1}, a, l, R) \tag{10}$$

where d_1 is the first object detection and $d^{i-1} = \{d_1, ..., d_{i-1}\}$ are all detections up to frame i [4]. For the further definition of $p(d_i|d^{i-1}, a, l, R)$ I refer to the derivation of Zhang et al. [4].

The output of the traffic-aware tracklet model is the collection of all current tracklets $T = \{t_1, ..., t_N\}$. With T given the road parameters, the traffic patterns, the lane associations and the hidden states can be inferred [4].

Inference Even though the parameters R, a, l and the hidden states h and g were already used, their real values are still unknown since we used uniform priors for most of them.

We start with the road parameters R. Considering that Eq.8 cannot be computed in a finite number of operations, R is calculated by using Metropolis-Hastings samplings [21]. A proposal R' given the current road parameters R is accepted with a probability

$$A = min\{1, \frac{p(\varepsilon, R')q(R|R')}{p(\varepsilon, R)q(R'|R)}\} \tag{11}$$

where $q(R'|R)$ is the proposed distribution. To calculate the transition kernel a combination of local moves, which change R slightly with symmetric Gaussian mixture proposals, and global moves that sample R directly from $p(R)$ [4].

The maximum-a-posteriori (MAP) [22] traffic pattern a are inferred by building the product over all tracklets t_n and marginalizing the lane associations l_n, assuming a uniform prior on traffic patterns $p(a)$

$$p(a|T,R) \propto \prod_{n=1}^{N} \sum_{l_n} p(t_n|a,l_n,R) \quad [4]. \tag{12}$$

Similar to that the lane association l of a tracklet, given the traffic pattern a and road parameters R, is obtained as the maximum of

$$p(l_n|a,t_n,R) \propto p(t_n|a,l_n,R) \quad [4]. \tag{13}$$

With the MAP estimate of the traffic pattern and the lane association given, the MAP assignment of the hidden states $\{g_1,...,g_M\}$ is calculated by marginalizing out the hidden states $\{h_1,...,h_M\}$, and running the Viterbi algorithm on the resulting HMM [4].

3 Experiment Results

A well-founded comparison between the experiment results of the three introduced approaches for intersection understanding is not possible due to two reasons. First the different goals of the approaches are not similar enough. Especially the approach of Zhang et al. that is introduced in Section 2.4 with it's primary goal to associate vehicles to lanes, stands out in this regard. Secondly the used test data sets are also not quite comparable. Nevertheless I will compare them after first introducing the different experiments.

LIDAR based intersection recognition The approach by Zhu et al. [2], explained in Section 2.2, used 2 different data sets for their experiments. Data set 1 focuses on frames with good road conditions with less interference from other vehicles or pedestrians and contains 264 frames of +-shaped intersections, 136 of T-shaped intersections and 400 frames of road segments with no intersections. Data set 2 has more challenging conditions with many other vehicles and pedestrians and contains 44 frames of +-shaped intersections, 56 frames of T-shaped intersections and 100 frames of straight road segments. As stated in Section 2.2 a Velodyne HDL-64ES2 LIDAR is used for the data acquisition.

The experiments are separated in 2 parts. First only the **classification between intersections and road segments** was tested. The manually annotated training set for this first test consists of 1300 frames. 150 of those are +-shaped intersection, 150 T-shaped intersections and the remaining 1000 frames are intersection free road segments. For evaluation all intersections are labeled as positive examples and the **true positive rate** (TPR), the **true negative rate** (TNR),

the **total accuracy** and the **area under curve** (AUR) of the ROC curve are used. The results are shown in Table 1a [2].

The second part of the experiment by [2] focuses on the **classification between +-shaped and T-shaped intersections**. The training set for this test consists of the same 150 +-shaped and 150 T-shaped intersection of test 1. Here T-shaped intersections are labeled as positive and +-shaped ones as negative. The results can be seen in Table 1b.

SVM PERFORMANCE ON INTERSECTION AND ROAD SEGMENT CLASSIFICATION

	TPR	TNR	Accuracy	AUC
Test Data 1	91.25%	96%	93.625%	0.987
Test Data 2	81%	84%	82.5%	0.938

(a)

SVM PERFORMANCE T-SHAPED AND +-SHAPED INTERSECTION CLASSIFICATION

	TPR	TNR	Accuracy
Test Data 1	93.382%	80.681%	85%
Test Data 2	85.714%	79.545%	83%

(b)

Table 1: (a) Classification between intersections and road segments. (b) Classification between +-shaped and T-shaped intersections [2].

Small scale intersection scan model What separates the intersection recognition approach by Yang et al. [6] from others is that it can be used with **LIDAR** and **monocular vision** data (see Section 2.3). Hence of course also the experiment data needs to be recorded by both a LIDAR system, in this case a Velodyne HDL-64E LIDAR, and a Point Grey camera for the monocular vision input. The test data was acquired by driving around in the same urban environment 4 times (see Fig.12), each trip with a different speed. The environment contains small-scale urban intersections and different scenes like trees, side-walks, dense buildings and bushland. While driving around, the vehicle identifies all intersections that it passes using a personal computer (Inter(R) Core(TM) i7, 2.7GHz, NVIDIA GTX970).

The results of the experiment are shown in Table 2. The qualitative result of the LIDAR and the monocular vision are similar, with the LIDAR having the advantage of about 100ms faster processing time. As stated by [6] the few misses that occurred, happened because of illumination changes or sensor failures.

Understanding semantics by modelling traffic patterns As introduced in Section 2.4, the approach of Zhang et al. [4] focuses on lane association of vehicles. Their experiments build on the comparison of their results with results of an older approach [23] by the same authors. The used dataset consists of 11 T-shaped and 49 +-shaped signalized intersection sequences. The sequences last frame is captured when the vehicle is entering the intersection and the

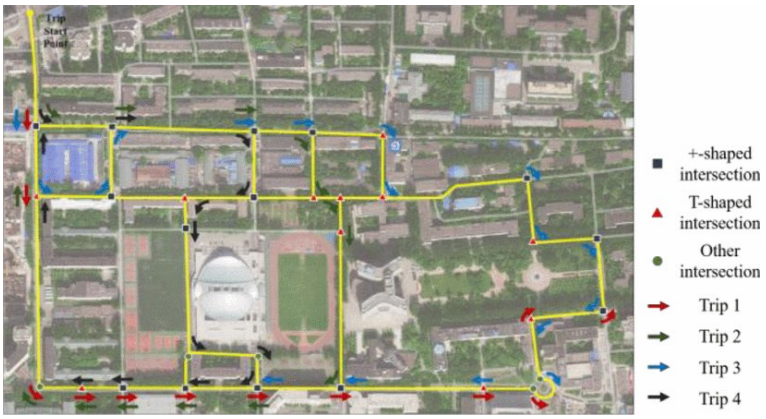


Fig. 12: Experiment path (highlighted in yellow). Experiment trip 1 to 4 are labeled in each intersection [6].

Trip	Velocity (km/h)	Intersection Detection			Time (ms)
		Type	**Numbers**	**Hit**	
1	5±20%	+-shaped	6	6	203
		T-shaped	4	3	189
		Other	3	3	211
2	10±10%	+-shaped	9	9	197
		T-shaped	5	4	189
		Other	1	1	206
3	15±10%	+-shaped	10	10	215
		T-shaped	6	6	198
		Other	2	2	200
4	20±5%	+-shaped	9	8	206
		T-shaped	3	2	193
		Other	3	2	192

(a)

Trip	Velocity (km/h)	Intersection Detection			Time (ms)
		Type	**Numbers**	**Hit**	
1	5±20%	+-shaped	6	5	302
		T-shaped	4	4	311
		Other	3	3	351
2	10±10%	+-shaped	9	8	316
		T-shaped	5	4	352
		Other	1	1	311
3	15±10%	+-shaped	10	9	308
		T-shaped	6	6	321
		Other	2	2	360
4	20±5%	+-shaped	9	8	352
		T-shaped	3	3	374
		Other	3	2	341

(b)

Table 2: (a) Detection results using LIDAR. (b) Detection results using monocular vision [6].

autonomous system would have to make a decision on what to do. The method of [23] was run using the improved tracklets by [4] to make the comparison fair.

The results of the test (see Table 3) showed, that the approach by [4] achieves better results at the estimation of the road parameters location (center of intersections), intersection arm orientation, and the overlap of the inferred road area. It also solves the problem of tracklet to lane association with astonishing better accuracy as the approach by [23].

	T-L error (all)		T-L error (>10m)		Location		Orientation		Overlap		Pattern error	
Method	3-arm	4-arm	3-arm	4-arm	3-arm	4-arm	3-arm	4-arm	3-arm	4-arm	3-arm	4-arm
[10]	46.7%	49.9%	17.9%	30.1%	**4.3 m**	5.4 m	3.3 deg	8.0 deg	58.7%	56.0%	–	–
Ours	**15.2%**	**30.1%**	**3.6%**	**14.0%**	5.7 m	**4.9 m**	**2.4 deg**	**4.3 deg**	**61.5%**	**61.3%**	**18.2%**	**19.4%**

Table 3: Comparision between the test results of [4] (Ours) and [23] ([10]).

Comparison: After introducing the experimental results of all three approaches lets make a comparison of their results. First I have to state, that the experiments of the "Intersection Scan Model" [6] and "Traffic Pattern" [4] approaches are not very meaningful, because their test sets are very small with 61 and 49 test cases. I narrowed the comparison to T-shaped and +-shaped intersections only, because only the "Intersection Scan Model" also differentiates other intersection types too.

In Table 4 the percentage of correctly classified intersections is shown. For data set 1 and 2 of the "LIDAR Beam Model" [2] these values were already given in their experiments. I calculated the average between the 2 data sets to get a better overall precision value. The experiment of the "Intersection Scan Model" showed the overall numbers of correctly classified intersections while driving with varying speed. I accumulated these numbers to calculate the precision percentage.

Precision Comparison			
Approach	Evaluated Data	T-shaped	+-shaped
LIDAR Beam Model	Data Set 1	94,382%	80,681%
	Data Set 2	85,714%	79,545%
	Average of Data Set 1&2	90,048%	80,113%
Intersection Scan Model	LIDAR	83,333%	97,058%
	Monocular Vision	94,444%	88,235%
Traffic Pattern	Monocular Vision	81,8%	80,6%

Table 4: Percentages of correctly classified intersections

For the "Traffic Pattern" approach it was difficult to decide what values to choose for this comparison, since their goal was not to classify between T-shaped

and +-shaped intersections, but to associate vehicles to lanes. The thought was between the "Orientation" results and the "Pattern error" (see Table 3). The "Orientation" value indicates how precise the angle of the intersections was calculated. The "Pattern error" shows how many patterns were incorrectly classified. Since "Orientation" does not say anything about the classification result and "Pattern error" is an even more in-depth classification than simply T-shaped or +-shaped, both values are not quite comparable with the results by the two other approaches. In the end I decided to take 100−"Pattern error" as the precision of the "Traffic Pattern" approach, which then indicates the percentage of correctly classified patterns.

The best result were achieved by the "Intersection Scan Model" using monocular vision data. It achieved a precision of **94,444**% on T-shaped intersections and **88,235**% on +-shaped intersections. A general remark about a precision difference between LIDAR and monocular vision cannot be made, because of the small test data sets.

Another very important value when talking about autonomous driving is the computation time an approach needs to come to a decision. This is important, because to use a method of intersection classification in the real world it has to be able to make decisions in time. In Table 5 the needed computation time by the "Intersection Scan Model" and the "Traffic Pattern" approach is shown. The "LIDAR Beam Model" approach did not state how fast its algorithm can make decisions, but states that it's capable of real-time decision making [2].

Computation Time Comparison		
Approach	Evaluated Data	Time(ms)
Intersection Scan Model	LIDAR	200
	Monocular Vision	320
Traffic Pattern	Monocular Vision	1000

Table 5: Computation time comparison

As can be seen in Table 5 the "Intersection Scan Model" is way faster than the "Traffic Pattern" approach. It only needs an average about **200ms** for the classification when using LIDAR data and about **320ms** when using monocular vision data. On the other hand the "Traffic Pattern" approach needs about 1 second to decide for a pattern and that only when the road parameters are already given. The calculation of the road parameters themselves already need about 1,5 minutes when using 15000 samples [4]. So there is no way that this approach can be used in the real world, because it is way to slow.

4 Conclusion

In this paper I introduced three different approaches for the problem of urban intersection understanding in regards to autonomous driving. It was shown that

the approaches use LIDAR or vision based systems and that the processing of those 2 different systems requires diverse methods. Nevertheless some methods can deal with both types. Additionally a short overview over the experiment results of the three approaches was given and a comparison between their results made. I also briefly discussed Hidden Markov Models, since they are used in two of the three approaches.

In future work I plan to develop novel solutions to the discussed problems.

References

1. A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.
2. Q. Zhu, L. Chen, Q. Li, M. Li, A. Nüchter, and J. Wang, "3d lidar point cloud based intersection recognition for autonomous driving," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 456–461, IEEE, 2012.
3. A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.
4. H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3056–3063, 2013.
5. Y. Morales, Y. Yoshihara, N. Akai, E. Takeuchi, and Y. Ninomiya, "Proactive driving modeling in blind intersections based on expert driver data," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 901–907, IEEE, 2017.
6. Y. Yang, L. Hao, H. Zhu, L. Ningyi, and S. Wenjie, "Small-scale intersection scan model for ugv in urban environment," in *American Control Conference (ACC), 2017*, pp. 4229–4235, IEEE, 2017.
7. U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold, and C. Wohler, "Autonomous driving goes downtown," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 6, pp. 40–48, 1998.
8. S. Gehrig, N. Schneider, R. Stalder, and U. Franke, "Stereo vision during adverse weatherusing priors to increase robustness in real-time stereo vision," *Image and Vision Computing*, 2017.
9. H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 807–814, IEEE, 2005.
10. B. Schwarz, "Lidar: Mapping the world in 3d," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, 2010.
11. L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
12. M. N. Murty and V. S. Devi, *Hidden Markov Models*, pp. 103–122. London: Springer London, 2011.
13. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
14. H. Zen, K. Tokuda, and T. Kitamura, "A viterbi algorithm for a trajectory model derived from hmm with explicit relationship between static and dynamic features," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 1, pp. I–837, IEEE, 2004.

15. L. C. Perez, "Hidden markov models and the baum-welch algorithm," *IEEE Information Theroy Society Newsletter*, vol. 53, no. 4, pp. 1–13, 2003.
16. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.
17. P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, pp. 109–117, 2011.
18. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
19. H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 2, no. 1-2, pp. 83–97, 1955.
20. P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*. No. Bd. 190 in Fundamentals of Kalman filtering: a practical approach, American Institute of Aeronautics and Astronautics, Incorporated, 2000.
21. S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The american statistician*, vol. 49, no. 4, pp. 327–335, 1995.
22. J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.
23. A. Geiger, C. Wojek, and R. Urtasun, "Joint 3d estimation of objects and scene layout," in *Advances in Neural Information Processing Systems*, pp. 1467–1475, 2011.

Object recognition within baggage - A comparison of 2D and 3D X-ray screening

Anna Sebernegg
e01526184

Scientific Working
TU Wien, Austria
e1526184@student.tuwien.ac.at

Abstract. X-ray screening, that generates either 2D or 3D images, has a major impact on security aspects such as baggage handling. These systems help to detect threat items like explosives or weapons within closed luggage. Despite many research and developments regarding X-ray technologies, automated object recognition is not precise enough to be able to completely rely on it. Baggage screening still depends on human operators. In this paper, different approaches for baggage screening are compared to show the respective advantages and disadvantages. It will also discuss the difference between industrial and medical X-ray screener and finally, problems that currently exist in terms of X-ray screener and object recognition are listed.

Keywords: X-ray, baggage screening, CT, object recognition, threat detection

1 Introduction

X-ray screening and inspection systems are used wherever objects and defects need to be detected non-destructively. Typically that's the case with industrial quality control, analysis of products like food inspection of cargos and for archaeological discoveries [14]. However X-ray screeners are also used as a security technology, the best known are probably baggage screener at security checkpoints in airports [5]. Since 9/11 and various other incidents, the security measures at airports and other public places have increased [24] [5]. Since then, more and more security research has been conducted to produce supporting technologies. Especially in the field of baggage screening breakthroughs were accomplished, which made it possible to scan every luggage before a flight takes place and without producing too much extra loss of time. Different countries and researchers took part in researching more precisely X-ray screener and therefore various approaches where achieved. Two predominant approaches lead to either 2D or 3D X-ray projection images. These images can be obtained by various approaches and have different advantages and disadvantages [24].

1.1 Overview of the paper

This paper is intended to provide an overview and understanding of object recognition within X-ray systems and therefore describes physic fundamentals (see Section 2) as well as the current state of the art of X-ray technologies used for baggage screening (see Section 3). Afterwards, some differences to medical X-ray scanners are explained (see Section 4) and advantages, as well as limitations of these technologies, are analyzed (see Section 5). Finally, the use cases are inspected and the usage of automatic algorithms is summarized (see Section 6).

2 X-ray Fundamentals

2.1 Electromagnetic waves

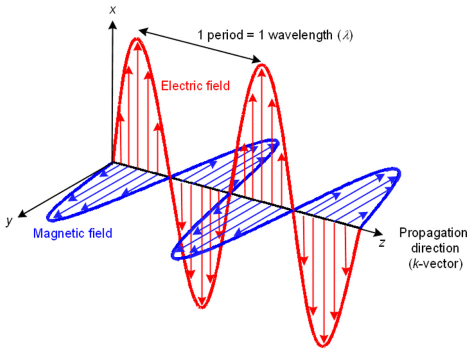


Fig. 1. Representation of electromagnetical waves [12]

Electromagnetic radiation is a form of energy that consists of transverse waves of electric and magnetic fields. These waves can travel through a space of vacuum at the speed of light (approximately 3.0108 m/s). The electric and magnetic fields components are orthogonal to each other as shown in Figure 1. Electromagnetic waves vary in wavelength, which is a measure of distance between two identical peaks (1 period), and frequency, which indicates how often the wave oscillates per second. Wavelength and frequency are inversely proportional. Waves with higher frequencies have shorter wavelengths and travel faster than waves with a longer wavelength and therefore lower frequencies. Furthermore the energy an electromagnetic radiation emits is inversely proportional to the wavelength as well [12] [4].

X-Rays have short wavelengths (between 10nm - 0.01nm) and high frequencies, hence they are most energetic beside of gamma-rays. Figure 2 shows all electromagnetic waves alongside X-rays with their wavelength.

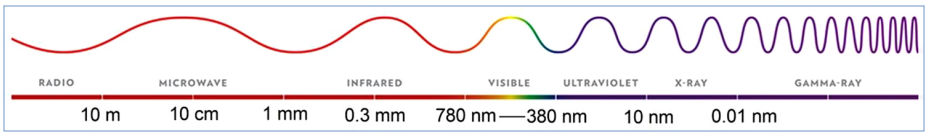


Fig. 2. Electromagnetic spectrum [19]

Depending on the density and atomic properties of matter, as well as the energy of the X-ray, X-rays can either pass through matter, be absorbed by it, or scattered upon impact.

Hard X-rays are the highest energy X-ray. They have the ability to penetrate materials and are used for screening objects (in medicine, for security screening, for quality control, etc.), whereas soft X-rays have less energy and therefore don't penetrate matter as easily as hard X-rays. Different materials absorb X-rays at different rates because of their density. Steel has a very high density and because of that only high-energy X-rays can permeate while lower-energy X-rays are absorbed. Organic materials like bread can be passed by hard and soft X-rays because of the low density [19].

2.2 Industrial X-Ray Imaging

Fig. 3. Example of a resulting image for a pistol scanned by an industrial X-ray scanner [13]

X-rays come in different wavelengths. The lower the density of a substance and the shorter the wavelength, the more transparent the substance is to the X-ray. Unlike in medical fluoroscopy, the lower the density and thickness of a material, the brighter it will be displayed. Higher density materials or really thick objects are displayed dark, very high-density materials like lead crystal, cement, and different metals will show up black [21]. This is because in industrial

scanners the positive - and not the negative - image is portrayed. By showing the silhouette of objects with high density, other items in front or behind that object cannot be recognized on a simple X-ray image [15]. An image of different shades of gray is the end result of an industrial standard X-ray screening as shown in Figure 3.

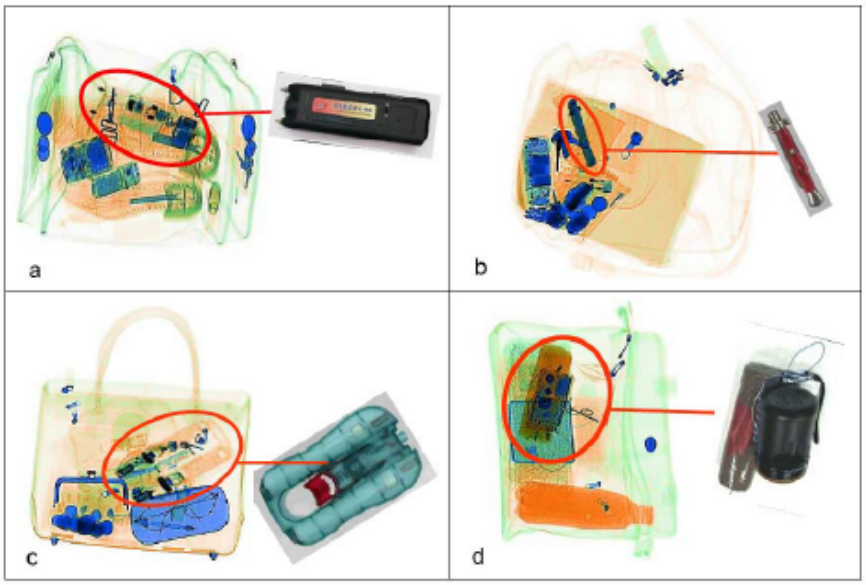


Fig. 4. a) Electric shock device, b) switch blade knife, c) self defense gas spray Guardian Angel, d) improvised explosive device (IED) [16]

A luggage inspection is a complicated task because objects, in reality, can look very different to those displayed on an X-ray image. People therefore need training and support to recognize objects on X-ray images [16]. A typical graphical support for this is to display different materials in different colors. Compared to grayscale images, color images are more pleasant to look at and areas can be recognized better because the human eye can distinguish many more colors than gray values [11] [9]. Such color images that display baggage with threat objects inside can be seen in Figure 4. A distinction is made between three main material groups:

Shades of Orange represent organic materials
Shades of Blue represent inorganic materials
Shades of Green represent mixed materials

3 State of the Art

X-ray scanners were not only developed for medical purposes but also for industrial applications. Many different approaches have been pursued over the years which led to a wide selection of products and advances. These various technologies can be divided into X-ray scanners that generate 2D images and computed tomography that generates 3D computer models. For both technologies, X-rays are used to obtain images. X-ray radiation can consist of numerous wavelengths. As a result, X-ray beams can have a different amount of energies and therefore the interaction with matter changes depending on the structure of the beam and the properties of the matter. Knowing how a matter behaves at two or more different energies provides more precise information about a material. Due to this principle, mono-energy, dual-energy and poly-energy X-ray images can be produced [15].

3.1 2D X-Ray Scanners

In a conventional 2D X-ray scanner, an image is created in a receptor positioned opposite the source for the X-ray and which detects penetrating X-rays. The items are represented by functions f(x,y,z) of the position. These functions are then imaged as projections P(x,y) onto the x,y plane of the receptor. Through that, the z-coordination is lost and the thickness of the items cannot be represented by the resulting image [24]. Four different systems, described in the following subsections, make use of this method.

Standard X-ray Technologies use single-energy X-rays, which mainly produce grayscale images because the differences between materials cannot be identified. Therefore, the image shows only how much X-rays were able to pass through the matter. This method depends highly on human operators. Without material discrimination, threat items like explosives, drugs or poison are hard to recognize and algorithms may only identify objects by their shape using pattern recognition [10].

Dual-Energy X-ray Technologies with Single-View as well as other poly-energy X-ray systems are using a range of different X-rays to gain more information about the scanned object. Additional filters and two or more receptors in dual-energy and poly-energy X-ray systems are used to capture different X-rays (see Figure 5). X-rays are picked up before and after each filter. The filters block out X-rays of certain wavelengths. Thereby a better representation of matter with different densities and attenuation properties can be achieved by comparing the resulting images. This leads to a better representation of the objects by using colors for different materials. The distinction of matters is not only useful for the human operator, it can also be used by automatic algorithms to detect and mark certain materials. Furthermore, by using X-rays with different energies, objects can be displayed that would be hidden by high-density materials

in mono-energy systems [22]. Single-View systems only present one view of the scanned object to the operator. Usually, this view is from the top of the object.

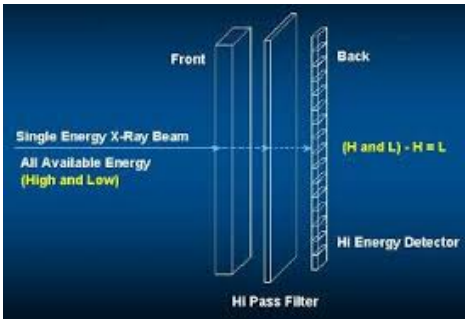


Fig. 5. Dual-energy X-ray system with two detectors in the front and the back. In between is a filter located [22].

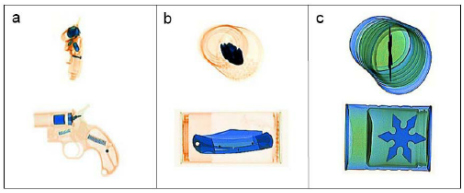


Fig. 6. Items are difficult to recognize when looked at from an unusual viewpoint. Dual-Views can be helpful [16].

Dual-Energy X-ray Technologies with Dual-View or systems with multi-view are using several X-rays and receptors to provide images from different perspectives. Most of the time, dual-view systems show a view from the top and one from the side. This gives operators more information about the content of the luggage [23].

Backscatter X-ray Technologies are often found in airport body scanners however, they can also be used for quick inspection of containers or luggage. In contrast to X-ray systems that use transmission methods, backscatter X-ray scanner detect the radiation that is reflected or rather backscattered from the object [3]. Low-density materials like organic ones tend to scatter X-rays and not many X-rays pass through. Therefore organics will not show up well on images of standard X-ray scanner but can be detected by backscatter X-rays

scanners [7]. In order to detect denser materials as well, which is important for baggage screening, backscatter X-ray technologies can be combined with a detector for penetrating X-rays. In this case, the system will generate two different images – one by the transmission and one by the backscattering (see in Figure 7) [10].

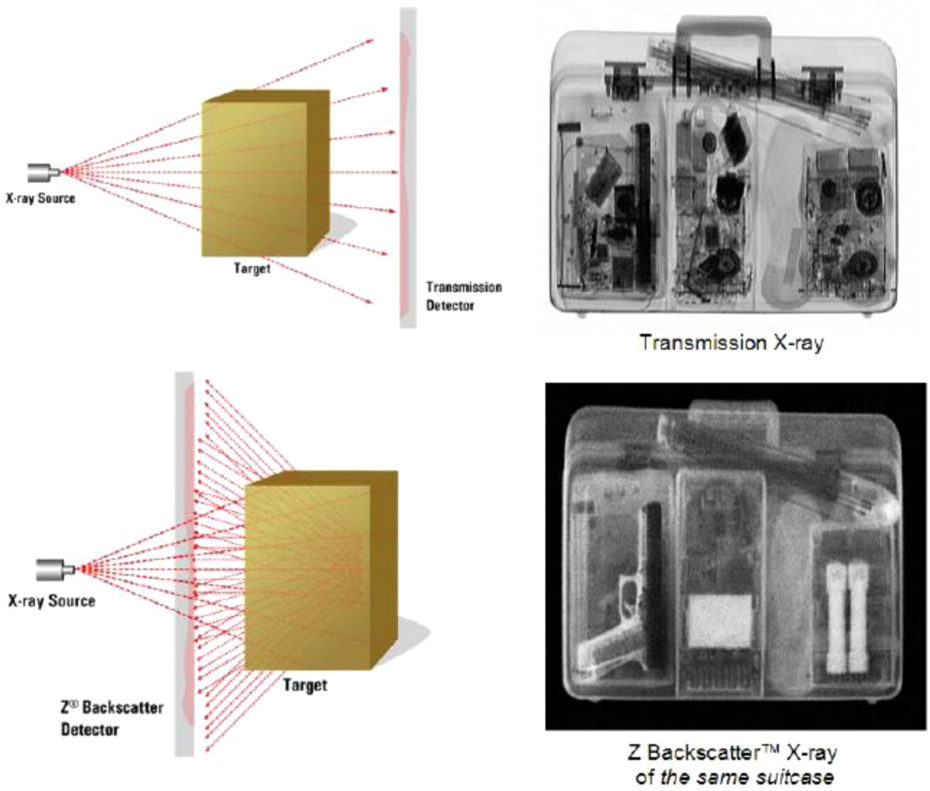


Fig. 7. X-ray transmitted (top row) and X-ray backscattered (bottom row) methods, with produced images [7]

3.2 3D Computed Tomography

X-ray **C**omputed **T**omography (CT) scan, also known as **C**omputerized **A**xial **T**omography (CAT) scan, produces cross-sectional (tomographic) images by generating different 2D X-ray images from various angles [6]. The external and internal structure of an object can be reconstructed by interpolating between those cross-sectional images to produce volumetric images in the form of 3D voxel(volume elements) images [1]. Therefore a three-dimensional representa-

tion of the luggage is generated. Some CT-systems display those 3D models in real time [8].

Just like 2D X-ray scanners, CT scanners can utilize distinct X-ray spectra to examine the different density properties of matter. **D**ual-**E**nergy **C**omputed **T**omography (DECT) uses this advantage by scanning an object at two distinct energies. By calculating the mass and density of an object, it can be determined from which material the object consists. Today, DECT Scanners are preferred over Single-Energy CT's because of their more precise and effective distinction of materials, which also enables automatically material-based detection of explosives [17].

CT scans provide much more information than conventional X-ray systems. However, to achieve images from different angles, the X-ray source has to be rotated around the baggage which makes the scanning-process more expensive and slower than the conventional one [20].

4 Difference to medical scanners

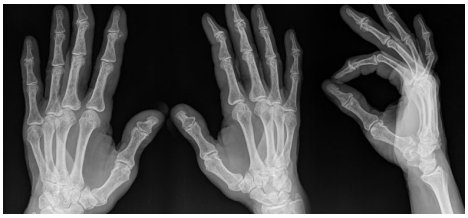


Fig. 8. Example of a resulting image of an medical X-ray scanner [18]

X-ray baggage scanners are very similar to medical X-ray systems with few exceptions. Newer baggage scanners make use of the different wavelengths of polychromatic X-rays which is not necessary for most medical X-ray systems. As already mentioned, X-rays that have a short wavelength and therefore high energy can pass through many materials while X-rays with a long wavelength can only penetrate low-density substances. Therefore, it depends on the density of the materials, which wavelengths can pass through and which cannot. By distinguishing which wavelengths passed the objects, X-ray scanners can resolve a much greater range of substances. The substances can then be colored differently which is another fact that differs from a medical system. To penetrate dense objects as well, baggage scanners use stronger X-rays which wouldn't be possible in a medical X-ray scanners because they can cause lasting damage to the human body. In contrast to medical CT's, in most industrial ones, the X-ray tube and the detector are at rest while the object rotates. Baggage screener are an exception – in their case, it's more practical to rotate around the baggage just like in medicine.

5 Limitations and Advantages of Different X-ray Systems

X-Ray systems have different advantages and limitations. They are therefore used in different contexts and situations. An overview of these advantages and limitations is given in Figure 9.

	STANDARD	DUAL-ENERGY	BACKSCATTER	CT
	SINGLE VIEW	SINGLE / DUAL VIEW	SINGLE VIEW	3D
Material discrimination	✗	✓	✓	✓
Threats detected				
Precision				
Throughput	FAST	FAST	VARIES	SLOW
Benefits	• Low cost	• Distinguish between materials and color-code for operators • Can detect explosives hidden behind object made of heavy materials	• Distinguish between low Z elements • Generates 2 images	• Great spatial resolution (up to a few millimeters) • Looks past/around high-density objects
Limitations	• Can't distinguish between various absorbing materials • Tend to have higher false alarm and overlooked threat rates	• Can't distinguish as effectively between lighter elements	• Reduced penetration (esp. through high density materials) • Expensive, esp. if 2-sided version needed.	• Slow throughput • High costs
Cost (estimated)	€	€ €	€ € €	€ € €

Fig. 9. Table comparing different X-ray technologies [2]

5.1 Standard X-ray Systems and Single-View

Standard X-ray systems have the drawback of not being material specific, which means that a distinction between organic matters like liquids and explosives is not possible [24]. In addition, overlying or underlying objects with high density may mask other potential threat items.

Single-View systems only show the shape of objects from one view which also means, that it can be really hard to recognize objects if they are rotated. An example for this is shown in Figure 10. This Figure shows (a) an example bag with a threat item in it, (b) shows a resulting image after a 2D X-ray scan from the top revealing the item of interest within the bag and (c) shows a different scan from the side. The item of interest in (c) has a different orientation in which the object is no longer clearly recognizable [8]. The difference of orientation is a huge limitation of 2D X-ray systems with only one view which makes the detection of threat items not only hard for human operators but also for automated algorithms. Moreover, no information about the depth of an object can be given. In exchange, standard X-ray systems have the advantage of being cheaper than other X-ray technologies and they have a high throughput, which means that they can scan fast [10].

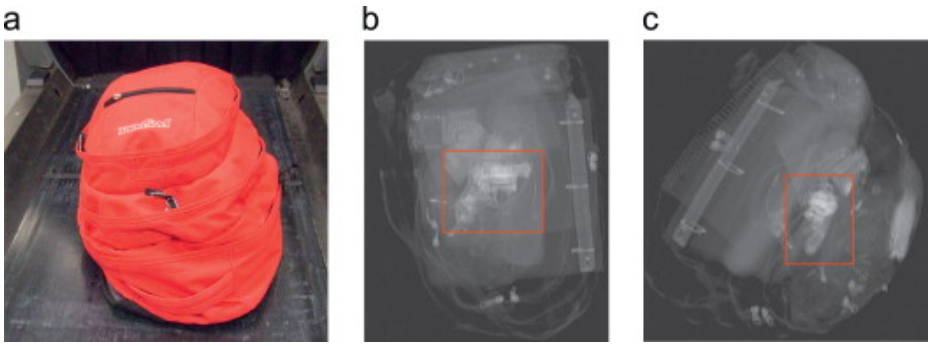


Fig. 10. a) Example bag with a threat object, b) First orientation of pistol, c) Second orientation of pistol [8]

5.2 Dual-Energy X-ray Systems and Multi-View

Contrary to standard X-ray systems, **Dual-Energy** systems have the advantage of distinction of materials [23]. Colors are applied to show the difference between the materials. This helps human operators to locate threat items but it can still be hard to distinguish between different organic ones. Therefore many dual-energy systems have automated algorithms which can flag suspicious organic materials that have a similar density to known threat materials [22]. Systems with **Multi-Views** have the additional advantage of displaying more views of rotation. Objects can be difficult to recognize using only one viewpoint, therefore a second view can help. Dual-Views may be an improvement, but sometimes even two perspectives do not suffice [14].

Dual-Energy X-ray systems cannot easily distinguish the shape between various lighter elements like organic materials. Although they are more expensive than standard X-ray systems, they are still cheaper than other technologies and have a high throughput [10].

5.3 Backscatter Systems

Backscatter systems can display two different images of the scanned object and are great for displaying organic materials but struggle with high-density objects. To make these systems efficient for baggage scanning, they need a second set of equipment with X-rays that can pass through dense objects. Therefore backscatter systems can be really expensive [10].

5.4 3D Computed Tomography

In combination with dual-energy X-rays – which is the state of the art of new CT baggage screeners – more features of the object are provided [24]. Because of their spatial resolution, CT's can determine the depth of an object and it can be viewed in every orientation. A human operator can just rotate the 3D

model to look around items, which means more items can be viewed [10]. One disatvantage of CT's is, that they are time consuming and therefore don't have such a high throughput as other systems [14]. This is because of the computation intensiveness of generating a 3D model. In addition, the cost of a CT system is higher than most other X-ray systems [10].

6 Conclusion

Threat items are very difficult to detect within baggage. They are closely packed, concealed by other objects or included within them. Sometimes, items are rotated in such way that they are unrecognizable since the scan only shows a 2D projection from a single perspective. Furthermore, threat items can show up in all different kinds of shapes and materials. X-ray systems are tools that enable security staff to identify such threat items without checking each bag by hand. In this paper, 2D and 3D X-ray technologies were discussed and compared to each other. All of them have some advantages and limitations, which means that none of the described systems fits every situation but all of them have their own use case.

Standard X-Ray Technologies cannot distinguish materials and only represent objects in a single view. Therefore, they cannot offer a precise security inspection but they are cheap and time-efficient. In places with high-traffic and a lower security, they can be used to detect weapons with a familiar shape.

Dual-Energy Sytems, on the other hand, are commonly used at airports [10]. They have a high throughput, can distinguish materials – which is important in locations were explosive matters and some liquids are forbidden – and allow the use of multi-views which helps to identify rotated objects. In general, they can be seen as good all-arounder for places with high-security measures and a lot of traffic. However, hidden objects and those unfavorably rotated in some particular ways cannot be detected.

Another described technology are **Backscatter** which are mostly used in body scanners but can also be useful for baggage screening. They can be valuable when the shape of organic objects matters. For example, if foods are banned in one place, these systems can be used for identification.

The last mentioned technology and the only one that produces a 3D model of the scanned object are **Computed Tomography Scans**. These systems can be powerful because of their high precision. By additional information such as depth and a full view, threat items are likelier to be detected and suspicious items can be better examined. CT's are slower than other X-ray technologies and need more space because of their additional equipment. This makes them less ideal for places with a high-traffic. If CT's are improved to faster technologies, maybe we will see more of them in airports instead of the current used 2D Dual-Energy X-ray scanner.

In all these systems **automatic algorithms** can be adapted to flag suspicious looking items or those with a material similar to known threat substances (except in the Standard X-ray system). They can also be used to hide informa-

tion such as specific materials like cloth which can come in handy in baggage screening because most of the time bags are filled with clothing and other objects are only in between. Considering the complexity of the detection of threat items, it's hardly possible to find every item that can be designated as a threat. In all these systems human operators are required, but algorithms can be of great help and will be improved. Nowadays, security measures are being intensified because of various events and this could lead to further development of baggage screening technology.

References

1. Ralph Benjamin. Object-based 3d x-ray imaging for second-line security screening. In *Conf. Publ*, number 408, pages 310–313. IET, 1995.
2. Promis Electro-Optics bv. Compare 4 x-ray technologies for baggage scanners. Website, 2016. peo Security, Available online at http://www.peo-security.com/en/geen-categorie-en/compare-4-x-ray-technologies-for-baggage-scanners/ last accessed on 23. January 2018.
3. Zongjian Cao. Optimization for the tradeoff of detection efficiency and absorbed dose in x-ray backscatter imaging. *Journal of Transportation Security*, 6(1):59–76, 2013. Springer.
4. Guillermo Avendao Cervantes. *Technical Fundamentals of Radiology and CT*. IOP Publishing, 2016. 1–2.
5. Christine Connolly. X-ray systems for security and industrial inspection. *Sensor Review*, 28(3):194–198, 2008. Emerald Group Publishing Limited.
6. Leonardo De Chiffre, Simone Carmignato, J-P Kruth, Robert Schmitt, and Albert Weckenmann. Industrial applications of computed tomography. *CIRP Annals-Manufacturing Technology*, 63(2):655–677, 2014. Elsevier.
7. Rahmani Faezeh, Azimi Sepideh Sadat, Bayat Esmaiel, and Dost Mohammadi Vahid. Comparison of the time behavior in the separation of light and heavy materials in x-ray backscattered method as a diagnostic tool in inspection. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 812:86–91, 2016. Elsevier.
8. Gregory T Flitton, Toby P Breckon, and Najla Megherbi Bouallagu. Object recognition using 3d sift in complex ct volumes. In *BMVC*, pages 1–12, 2010.
9. Dr. Thorsten Hansen. Seminar farbwahrnehmung - bersicht. Justus-Liebig-Universitt Giessen, Available online at http://www.allpsych.uni-giessen.de/hansen/teaching/SeminarFarbwahrnehmung/referate/ SeminarFarb-wahrnehmung_Overview.pdf last accessed on 04. March 2018.
10. Astrophysics Inc. 4 x-ray technologies that will make you a security expert. Blog, 2016. Astrophysics Inc., Available online at http://www.astrophysicsinc.com/4-x-ray-technologies-that-will-make-you-a-security-expert last accessed on 22. January 2018.
11. K Kase. Effective use of color in x-ray image enhancement for luggage inspection." vol. *Master Degree Knoxville: The University of Tennessee*, 2002.
12. Wayne D Kimura. *Electromagnetic Waves and Lasers*. Morgan & Claypool Publishers, 2017. 1–3.
13. Asylum Models & Effects Ltd. Website. Asylum Models & Effects Ltd, Available online at http://www.theknowledgeonline.com/profile/asylum-models-effects-ltd last accessed on 24. January 2018.

14. Domingo Mery. Computer vision for x-ray testing. *Switzerland: Springer International Publishing*, pages 1–5, 2015. Springer.
15. Domingo Mery, Erick Svec, Marco Arias, Vladimir Riffo, Jose M Saavedra, and Sandipan Banerjee. Modern computer vision techniques for x-ray testing in baggage inspection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4):682–692, 2017. IEEE.
16. Stefan Michel and Adrian Schwaninger. Human-machine interaction in x-ray screening. In *Security Technology, 2009. 43rd Annual 2009 International Carnahan Conference on*, pages 13–19. IEEE, 2009.
17. Andre Mouton and Toby P Breckon. A review of automated image understanding within 3d baggage computed tomography security screening. *Journal of X-ray Science and Technology*, 23(5):531–555, 2015. IOS Press.
18. Noosa Radiology. Website, 2015. Noosa Radiology, Available online at https://www.noosaradiology.com.au/services/x-ray last accessed on 24. January 2018.
19. Malcolm Sperrin and John Winder. *Scientific Basis of the Royal College of Radiologists Fellowship*. IOP Publishing, 2014.
20. Renful Premier Technologies. X-ray screener: Ct scanning. Website. Renful Premier Technologies, Available online at http://www.x-rayscreener.co.uk/?xray=ct-scanning last accessed on 22. January 2018.
21. Renful Premier Technologies. X-ray screener: X-ray imaging. Website. Renful Premier Technologies, Available online at http://www.x-rayscreener.co.uk/?xray=x-ray-imaging last accessed on 22. January 2018.
22. Jeff Tyson and Ed Grabianowski. How airport security works. Website. Renful Premier Technologies, Available online at https://science.howstuffworks.com/transport/flight/modern/airport-security4.htm last accessed on 22. January 2018.
23. Claudia Christina von Bastian, Adrian Schwaninger, and Stefan Michel. Do multi-view x-ray systems improve x-ray image interpretation in airport security screening? 2011.
24. K Wells and DA Bradley. A review of x-ray explosives detection techniques for checked baggage. *Applied Radiation and Isotopes*, 70(8):1729–1746, 2012. Elsevier.