This Thesis is submitted as a Final Project for the Master in Artificial Intelligence (UPC)

Probabilistic methods for furnishing bedrooms in interior design: Bayesian Networks for occurrence modeling and GMMs for furnishing arrangement

Ștefania Budulan

Advisor: Alfredo Vellido, *PhD*

Supervisor: Marc Fernàndez Vanaclocha (Interiorvista Company)

Erasmus Plus Agreement Coordinator:

Adina-Magda Florea, PhD

University Politehnica of Bucharest

July 6th, 2016

Acknowledgements

The authors would like to thank Interiorvista company for their continued expert support in interior design guidelines, practices and 3D modeling, articulated through a collaboration agreement with the *Facultat d'Informàtica de Barcelona* at *Universitat Politècnica de Catalunya*, BarcelonaTech.

This thesis would not have been successful without the guidance and support of several individuals who contributed to the process of finalizing this study. I am grateful to my advisor and former professor, Dr. Alfredo Vellido, for sharing his knowledge throughout the evolution of this project, remaining encouraging, patient and reliable.

Another person who believed in the importance of this project and was personally involved in facilitating data understanding and processing, as well as sharing insight about the company's work, was my supervisor within Interiorvista, Marc Fernàndez Vanaclocha. I thank him and our coworkers for their involvement in providing materials and use their experience for helping, especially, with the dataset preparation.

Abstract

Tasks in architectural and interior design range from defining the building floor plans and ensuring desired functionality, to deciding furnishing styles and arrangement choices; all to best fit certain pre-established purposes. The process of design, as a whole, has remained hard to master for computer-based optimization in general and for computational intelligence approaches in particular. Some attempts to tackle different subfields of this problem in a machine learning fashion have emerged over the last few years, aiming to offer partial automatization of human tasks, personalized support for specialists in the field and professional guidance for amateurs. In this thesis, we first present an overview of current advances of computational intelligence in architectural science with a focus on interior design. We describe various learning models applied to interior design challenges such as furniture type selection, style compatibility, furniture arrangement, or ornamental decoration. The core of the thesis is devoted to report ongoing research towards the development of a commercial, robust and scalable solution for automatic furniture arrangement, given a room plan. We propose two probabilistic models to be used in the complex problem of furnishing bedrooms. The first resides in a Bayesian Network based approach for the automatic generation of the number and types of furniture entities to occupy the new space, namely the occurrence model. The second one, called arrangement model, deals with learning different commonly met sets of items interconnected within the same space and estimating their relative positions with GMMs. Both models heavily contribute to the main goal of achieving a 3D planner for bedrooms, but their genericity allows other types of interiors to be modeled through the same process.

Keywords. Architectural design, interior design, computational intelligence, probabilistic models

Contents

1 Introduction

Over the last thirty-odd years, our societies have steadily become more and more aware of the importance of information and data at large as a tool for knowledge extraction and expert decision support.

What started as a mostly business-related endeavor in the form of Data Mining or Knowledge Discovery from Data, has become an almost all-encompassing endeavor under the umbrella term of Big Data and, more formally, under the conceptual framework of Data Science.

The increasingly common availability of large and often complex databases has changed many operating paradigms mostly in IT-oriented companies but, interestingly, it has also become the stalwart companion of many fields of science, from astronomy [1] to high-energy particle physics [2] and neuroscience [3]. Arguably, though, no bigger effort has been made to tame the Big Data beast than in Biology [4].

Importantly, this concern for data and their uses has gone well beyond IT companies, big businesses and science to permeate far smaller enterprises. This thesis is an example of that, as it deals with data science as applied to customer-generated information gathered by an interior design small enterprise.

This company, namely Interiorvista (url: http://interiorvista.com) focuses on the design and development of software tools to accompany their customers in a seamless process of designing interior spaces, facilitating the process of choosing the right elements of furniture and decoration and the subsequent contact with vendors.

Architecture has been an endeavor at the heart of human societies throughout history. This well-established field has undergone continuous progress in different fronts, from the use of materials to engineering and design. Modern architecture can be said to have undergone a further radical change through the adoption of computer-based methods for both engineering and design. Interestingly, the pervasive introduction of increasingly sophisticated methods of computer-aided design (CAD) and engineering (CAE) temporarily displaced mathematics from its central role in architectural practice. This seems to be a trend currently in the process of reversion, paradoxically due to the formal complexity allowed by CAD systems, which begs for the use of parsimonious data models that can be used to transform the available information into actionable knowledge.

This need of data modelling has naturally led to a more recent step forward in the use of computers in architecture, which is the integration and application of computational intelligence (CI) approaches in design processes, so as to answer an older question posed by MacCallum [5]: does intelligent CAD exist?

Note that research on the use of Machine Learning (ML) in design was already quite active in the 1990's [6]. In the architectural engineering field, artificial intelligence (AI) is currently present in subfields such as building design, interior furniture organization, ornamental decoration and style, aiming to offer automatization of human tasks, personalized support for specialists in field, professional guidance for amateurs, etc.

This thesis concerns the use of CI probabilistic approaches in the specific field of interior design and within the constraints of a project that involves an agreement between the Interiorvista company and the *Facultat d'Informàtica de Barcelona* (FIB) at *Universitat Politècnica de Catalunya* (UPC).

The specific goals of the thesis are listed next.

Basic goals of the thesis

1. Implementing a Bayesian Network model for the automatic generation of the number bedroom furniture pieces that will be aggregated within the given layout.
2. Developing a GMM-based furniture arrangement system for three types of furniture categories that can be found within a bedroom: bed and nightstands, bed and TV, and desk and chair.

We now summarily list the main results of the thesis.

Main results of the thesis

1. Achieved a functional furniture occurrence generator through the help of Bayesian Networks.
2. The occurrence system makes the task of sampling unseen solutions easy to use and to improve.
3. Creating furniture arrangements with GMMs for objects that can be found within the context of a bedroom.
4. Particularizing the arrangement system for three type categories: bed and nightstands, bed and TV, desk and chair.
5. Obtaining general use models that learn and sample solutions in under one second.
6. Providing generic approaches and implementations for each model that can be easily improved upon.

Structure of the thesis

Beyond this introduction, the current thesis is structured according to the following chapters:

Chapter 2: In this chapter, we review the present capabilities of AI, with a focus on CI and ML, in the field of architectural design and, more specifically, in interior design, including an examination of the underlying data models. We expose the models' strengths and limitations, as well as their dependency on the specific context, their similarities and evolution across design main subfields and the possibilities opened by the potential combinations of methods.

Chapter 3: This part offers details about the Interiorvista Company, their current achievements, vision and goals. It also reveals a motivation of our project's existence within the company, research prospects and future employment of Artificial Intelligence techniques to contribute to the company's tools to attain its goals.

Chapter 4: description of the CRISP approach to Data Mining for the dataset included in this research. It includes the tools and frameworks needed, decision motivation and examples relevant to the issue of usability. This chapter also provides specific information regarding data integration and further processing for each of the engaged models.

Chapter 5: The overview across fields of application of CI in interior design provided in Chapter 2 is a first contribution of our ongoing research leading to the development of a real-world, commercial solution to offer automatic custom room plan-adapted furnishing suggestions to clients. Besides, the models behind the solution must be robust and scalable in order to incorporate the numerous existing furniture options and to fit various room layouts. Furnishing options must comply not only with user preferences, but must also be able to incorporate professional design expert knowledge-learned and incorporated by the model, resulting in suggestions as close to professional design as possible. This chapter includes the details of the implementation of several probabilistic CI methods and their use in experiments for the analysis of the available data.

Chapter 6: This last chapter of the thesis includes, first, an overview of the project, results and the knowledge gained through experimental investigation. Secondly, we detail future development of potential interest, including future dataset requirements for advancement of this research and an overview of the ultimate goal we are striving for, continuing the difficult and untrodden path, started with our proof of concept research, towards achieving a fast, professional and user-friendly 3D planner software tool.

2 Related work

Architecture, a science, art and craft that has been at the core of human societies throughout history, has always represented one of the most intriguing and continuously evolving field, with feeding influences and contributions from exact sciences such as mathematics and geometry that helped pushing its boundaries towards bigger and more, complex and accomplished constructions. Architecture can be traced back to ancient times, in cultures in India, Egypt and Greece [7]. Besides technical influence, architecture was heavily modeled by arts and personal taste, becoming a field at the heart of human communities.

With the ever increasing complexity of architectural models, practices and knowledge, and with the advances on computer-based technology providing more powerful and end-to-end architectural solutions, the progress in the field now faces the challenge of integrating computational intelligence (CI) and human expert knowledge to provide integrated automation and personalized assistance in using these tools.

Machine Learning (ML), in particular has been used in the sub-field of architectural design research since the early1990's, [8]. Numerous artificial intelligence (AI) and ML techniques have emerged, in many architectural sub-fields including building design, interior furniture arrangement, style assessment, artifacts decorations. They aim to offer at least partial automatization of human-specific tasks; personalized experience based on user practices and interaction with the software; professional guidelines and practices integrated with the current stage of the projects, amongst other features.

In order to provide an in depth understanding of the current positioning and capabilities of CI and ML models to address the challenges imposed by our project, which include automatic furniture organization provided a personalized room plan, we review in this chapter the noticeable advances in architectural design, which inspired and directly influenced the research and models present in our target subfield – interior design automation. In doing so, we expose in our analysis the underlying models used, their contextual strengths, imposed constraints, limitations and dependency on the specific problem they aim to solve.

2.1 Early attempts to model the human design phenomena

Among the earliest efforts to develop intelligent systems for building design, a multi-criteria based model was presented in [9], building upon two ongoing efforts to algorithmically represent the human design process. The first, that we could place in the area of cognition science, tries to understand the details of the mental process performed by a human in a design task, while the second puts the computer first and tries to express the process only analytically, abstracting the human and cognition processes and translating them all in the form of a rigorous, algorithmic process.

The work in [9], based on these two lines, analyzes the role of multi-criteria problem solving in modeling the design process as done by a human mental process. Although the system implementing the model proposed, namely $K_LD^E_0$, reports poor performance due to the inability to express requirements that are ideal for the model, often causing ambiguity, imprecision, incompleteness and inconsistency, this work still provides preliminary results of interest, in an early attempt to develop appropriate computational intelligent systems to achieve architectural design goals.

2.2 Probabilistic approach

As a response to the challenges regarding the solution space dimensionality, problem complexity, and the difficulty and rigidness of completely specifying the layout rules, various probabilistic models have emerged in recent times.

The main advantages provided by this category of approaches include:

- a first possibility to partially model, in a probabilistic manner, a space that is impossible to be represented completely through a deterministic approach, due to its vastly complexity, both in terms of possible solutions and space dimensionality.

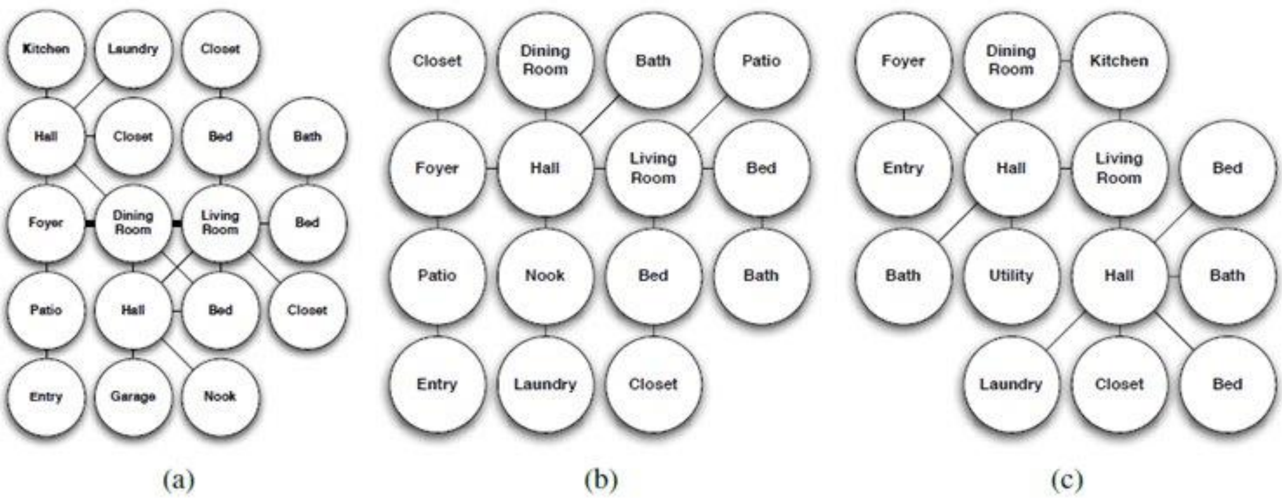
- the ability to learn meaningful relations and parameter values directly from real-world data, followed by sampling reasonable layouts from the learned probability distribution model, thus avoiding, first, the need to model a tremendously complex and highly multi-dimensional space of possibilities, and, second, the need to search this vast space for optimal, plausible configurations.
- the option of incompletely specifying the problem requirements: compared to previous models explored, in which a rigorous definition followed by an analytical representation of the desired goals and problem parameters was required. In this case, a probabilistic model trained on real-world data would be able to estimate, in a feasible manner, the other unknown parameters, resulting in a great flexibility in the specification of the input and consequently allowing diverse results for a more loosely-specified input.
- the choice to fix arbitrary constraints and sample from the probabilistic network the rest of the parameters: this feature is very useful when the degree of freedom of certain parameters is known, forcing the model, in appropriate cases, to keep some parameters at desired values while sampling compatible values of the other parameters within their respective constraints. Besides the ability provided by this property to interact with the model and constraint it, thus reducing the search space and guiding it towards a desirable solution, the model can also be used at a more abstract level, to define a limited, narrow, local search space of feasible solutions, in which a stochastic optimization method could be feasibly applied to obtain the best solution according to a defined energy function.

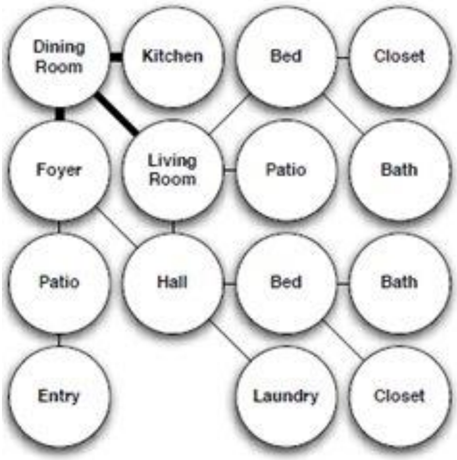
Because of these advantages, probabilistic models have also been applied in the design field, both in architectural design for the automatic generation of floor plans that would lead to the ability to generate 3D buildings [10]; to generate (much more) new, feasible objects from components, starting from few, diverse ones [11] and, more recently, in interior design for furniture sampling and organization [12].

Bayesian networks for floor plan generation

An interesting study, which uses Bayesian Networks to model complex, underlying patterns and dependencies among the entities of a floor plan was presented in [10]. The probabilistic model is trained on a real-world dataset in order to learn meaningful relationships among the room sizes, aspect and adjectives. By implying a learning model, the authors claim to overcome previous limitations and failures to correctly capture deep, underlying dependencies, in a manner that would lead to feasible room generation. The learning process is done at a global level, thus avoiding the possibility of learning only local dependencies which would fail to merge once the overall floor plan is created.

Figure 1. Bayesian Network trained for (a) 10 iterations; (b) 100 iterations; (c) 1000 iterations and (d) manual designed structure.





(d)

Figure 1, a visualization of a Bayesian Network trained for 10, 100 and 1,000 iterations is presented, the last model being handcrafted by a professional interior designer. After training the model, samples are drawn from the probability distribution over the space of solutions and an optimization phase occurs, to fine tune the previously obtained result. A local space exploration is possible at this stage, when starting from a feasible estimation, as compared to other models in which the initial sample was randomly initialized.

The optimization is accomplished using a simple Metropolis algorithm with a set of steps compatible to possible floor plan changes, as would be done by an architect, and including:

- **sliding a wall**: where wall is defined by the authors as a contiguous set of wall segments, which are randomly split and moved in a controlled manner, in order to fine-explore the neighborhood of solutions and find a lower energy point in the search space.
- **swapping rooms**: this move, in opposition to the previous one, aiming towards convergence, is intended to create spikes of energy, commonly used in order to restart the optimization process (i.e. not get stuck in a local minimum) and faster explore the search space.

Accordingly, the cost function optimizes the following:

- **accessibility**: which and how rooms have to be connected – via open space, doors, etc.
- **dimensions**: as overall space area and aspect ratio for different types of rooms, specific rooms, etc.
- **floors**: a feasibility measurement, imposing that a certain room, at a certain floor, has to be inside the space defined by any lower levels floor plans.
- **shapes**: a design expert metric, rewarding “nice” room shapes – preferred by professional architects when designing a floor plan (e.g. usually near convex shapes, large enough spaces, etc.)

Bayesian Networks for shape synthesis from components

In [10], a Bayesian based approach for floor automatic generation was introduced, defining a model through training on real-world data, which was used for sampling reasonable layout characteristics, such as adjacency relations, number, types and sizes of rooms. This model represented an inspiration to our work for developing the Bayesian component, trained on real world data, for sampling / generating furniture pieces that would fit in a bedroom.

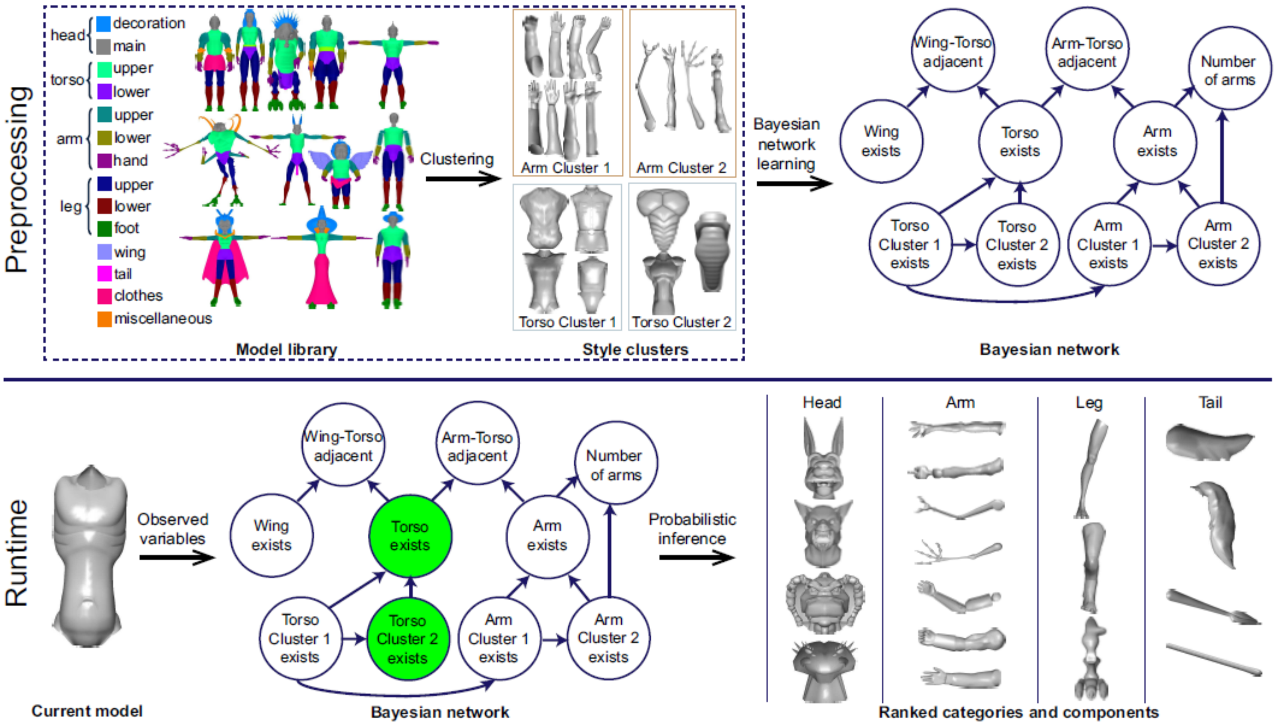
Other similar works, which use a probabilistic approach to model the complex, underlying, hidden relations among components of complex 3D objects (i.e. vessels, planes, furniture pieces) can be found in [13] and [11].

In [13], the authors develop a system that uses a Bayesian Network to learn patterns and dependencies between component styles, including overall shape, aspect, contextual usability and functionality, in order to support the user in the task of assembling various types of objects from components – the system suggesting only “compatible” ones. The Bayesian Network learns from data, both in terms of structure and parameters. The structure learning, which is known to be an exponential complex problem, is learnt using a heuristic that searches the local space of network structures with steps that vary the current configuration through adding, removing and flipping edges.

The pipeline of the model is presented in Figure 2. Apart from the probabilistic component, the system has a pre-processing stage, in which the dataset, comprising various 3D models gets “semi-automatically” segmented into sub-components (e.g. arms, legs, head, etc.) which are tagged accordingly, in a hierarchical fashion (i.e. a component may contain sub-components). These segmented shapes are then clustered by style, resulting in groupings of the same type consisting of components with similar style. The Bayesian model is trained on top of the obtained clusters, with the result of a system capable of suggesting best style ranked components according to the current design object.

This model provides inspiration to the approach we have developed for the furniture occurrence model in a bedroom, where furniture pieces that fit a certain bedroom can be compared to the shape components assembling a complex object. In our case, a Bayesian Network was also learnt, both in terms of structure and parameter values, for capturing underlying dependencies between the furniture pieces that would fit in a plausible bedroom.

Figure 2. A Bayesian Network based system for capturing dependencies among components of shapes [13].



Another successful instance of usage of probabilistic models for shape synthesis focused on automatic new object creation [11], compared to a user-assisting tool [13], at a massive scale, generating much more plausible objects from only a few ones. As stated by the authors, the system addresses certain limitations of its predecessor [13], including the use of the Bayesian Information Criterion (BIC), the probability tables in the Bayesian network and the lack of encapsulation of latent causalities in the model (i.e., through latent variables) – which are used in the model to capture abstract, high-level concepts regarding the overall object, as opposed to only individual component linkage.

Top observations in the paper include the necessity to develop a hierarchical probabilistic model, as opposed to a “flat” one, because the diversity and the deep, underlying differences among similar types of objects (e.g. working desk chair, child chair, dining chair) are hard to capture at a more concrete level - as in the various linking properties among different components. Similarly to the previous model, the structure and all parameters (including the latent variables) are learnt from data, allowing the model to better capture the trends and dependencies imposed for the various types of objects used.

In Figure 3, we present the results of this system in generating new, many more (870), plausible, furniture objects (chairs), starting only from a few (88), displayed in the image in green.

Figure 3. Probabilistic model [11] results for generating new chairs (in blue) from a few starting ones (in green).

Although an indirect link between assembling shapes from components and assembling bedroom furnishing sets from individual pieces can be made, we highlight certain key differences between the two, making the latter a hard to master challenge only using a Bayesian Networks:

- **Infinite search space of positioning furniture**: compared to components of objects, which have a clear positioning in the objects (i.e. chair legs for a chair, wings of an airplane, etc.) the furniture placement is a very challenging task to master, furniture pieces only being loosely coupled, with various types of relationships between them in a real-world scenario, which influence positioning, angles, relative distance (i.e. a TV and a sofa, a table with chairs, etc.), resulting in infinite space possibilities, with only very few plausible ones.

- **Deeper underlying causalities for furnishing configurations**: besides similar compatibility aspects to the 3D object modeling, such as objects style, functionality, coupling and room aspect, a bedroom layout is heavily influenced by patterns and rules far beyond this scope, such as: user personality, tastes, daily routines, activities, marital status, background, age, etc.
- **Numerous furniture types and variations**: in a real world scenario, furniture objects are complex and numerous variations do exist: in style, color, functionality, dimensions, all contributing to a very large collection of possible entities with complex, multi-dimensional representation, as compared to virtual 3D shape modeling.

Mostly data-driven oriented, able to capture underlying, hidden properties of the entities and their relation to each other and the context, Bayesian models have successfully been used in architectural design related research for representing learnable relations between complex objects components [11, 13], learning plausible room sizes and arrangement dependencies [10]. One key advantage they offer in this field is the process of discretely sampling from the probabilistic model, which overcomes, in a heuristic manner, the performance impact caused by searching or exploring the large, complex potential solution space.

Probabilistic models in interior design

Focusing now specifically in interior design, we must bear in mind that this subfield is, first and foremost, characterized by complexity. This complexity is often characterized by a very large solution space, which can be efficiently explored or estimated, as previously mentioned, through learning probabilistic models [11, 13] (at well-known costs, such as ambiguity, crude estimations of parameters for complex scenarios, or failure to capture the problem in all its dimensions).

In particular, when dealing with furnishing a room, including both objects selection and arrangement, most data-driven approaches include a probabilistic learning model such as Bayesian Networks [12], or a mixture of models; for instance, using a Bayesian Network for furniture occurrence task and Gaussian Mixture Models (GMM) for the arrangement task, as in [12] – in every case aiming to capture certain aspects of the data, hard to express in an analytical manner. This approach is similar to the one for exterior architecture building, in which the solutions' space of construction shapes can efficiently be captured in a probabilistic manner, as shown in [10] and previously detailed.

In [12], Fisher *et al*. proposed a system for generating new, plausible scenes (i.e. "environments composed of arrangements of 3D objects") by starting from a few user-provided ones. The system deals the problem that is also a challenge in this thesis, namely furniture selection and arrangement, by using a Bayesian model for learning the furniture occurrence in different types of rooms and a GMM for capturing arrangement patterns.

In [12], the probabilistic generation of new scenes starts from only few examples, which represents a potential limitation given the usual complexity of a classic furnishing scenario, in which realistic, plausible layouts are governed by numerous types of objects and many placement possibilities. To address this issue, the system uses an "external" large database of scenes, which is used for both introducing diversity in the new scenes created (by inter-changing objects) and automatically enlarging the dataset (by incorporating similar scenes to the examples given by the user).

The second component of the model proposes a clustering algorithm to obtain the object categories that can be swapped with each other; it aims at identifying objects surrounded by similar types of entities (i.e. present in a comparable context). Creating these "contextual categories" of objects, which can contain a variety of types of objects (e.g., objects that have a desk as support could be TV, lamp, computer, book, etc.), the model is able to introduce much variety in the synthesized scenes, not only in terms of similar objects but also in terms of objects of different type, due to these groupings.

The third component addresses the limitation of training the furniture synthesis model on only the limited set of examples provided, by proposing a model to crawl the vast content of the dataset for similar scenes to the one provided, resulting in an automatically expanded and representative training set to the initial input. Moreover, the degree of freedom of the synthesized scenes can be adjusted by the user, who is able to decide between more diversity and consequently less similarity of solutions to the initial context, or *vice versa*, this always representing a tradeoff.

Figure 4. Results (right hand side) of scenes generated for given input examples (left hand side) in [12].

The furniture selection and arrangement model (representing the first part of the model) uses, as briefly mentioned before, a Bayesian Network for learning the furniture occurrence and a GMM for capturing arrangement patterns in various scenarios.

The probabilistic model in this thesis is inspired on this work, dividing it accordingly: in a mixture of a Bayesian Network addressing the generation of plausible furniture sets and a GMM for arranging such furniture. However, as presented in Figure 4, showing the obtained scenes for different values of the occurrence parameter, given an input example (left), the synthesized solutions rather emphasize complex ornamentals and decorations rather that full room space furniture synthesis, as addressed in our case – where the goal is rather deciding the main furniture pieces towards a complete, fully functional bedroom, than plausibly decorating a certain portion of it, or a functional group of few furniture pieces (e.g. a desk with a chair, with various decorations and artifacts). Moreover, the performance of the GMM arrangement model is limited to the scenario context addressed – modeling only direct, usually binary relations among few furniture pieces (i.e. a table and a desk) and adding variations in terms of interchanging objects – which are usually rather "scene accessories" compared to the most relevant furniture. These artifacts, being inter-changed with others that are present in similar contexts, rather have a fixed, low-variation position, determined from the context itself (for instance, an apple on a plate, or a TV on a desk can be placed in a limited, deterministic, known set of positions, mostly maintaining the arrangement integrity). In contrast, furniture pieces, which are rather loosely coupled, allowing a variety of possible positions in an initial unfurnished room, permit only few plausible final configurations, making the placement crucial for the overall sense of reality of the scene (a chair facing a wall is unlikely, and so are a bed blocking the door or the access to a wardrobe, a TV placed correctly but rotated away from a possible corresponding sofa or bed).

A key aspect regarding arrangements in furnishing of realistic bedrooms, is related to complexity: furniture positions, angles, direct dependencies can be highly influenced by "soft constraints" such as human supported activities, daily routines, user preferences, rather than "hard ones", which are more rigorous and thus easier to model in a more deterministic, algorithmic approach, such as: available space, specifics of placement (e.g., a glass should always be on a table or on a desk).

Another novel data-driven approach to capture furniture grouping functionality can be found in [14] and is based on the concept of “Wall Grid Structure” (WGS), which addresses a similar problem to the one in this thesis, namely furnishing an empty room, represented through type, shape, dimensions, doors and windows positions.

The system was used to furnish three types of interior spaces, namely: conference rooms, living rooms and bedrooms. The goal of the research was to advice potential clients in ways of designing their interior property, showing how their method could be used to furnish apartments interiors, and also help professionals in the field by enlarging their vision and easing the interior design process. The system was developed based on scenes taken from Google Warehouse dataset, consisting of 52 scenes and 1,111 furniture pieces for conference rooms; 45 scenes and 741 objects for bedrooms, and 48 scenes and 875 models for living rooms. The algorithm consists of two main stages: the “learning stage” – in the initial phase of the algorithm, using a database of same-type rooms - and the “synthesize stage”, which consists of furnishing the desired room.

The learning stage occurs only once, at the onset, and involves creating the basic structures (patterns of furniture structures) to be used in the second phase. This stage does not consider the room details, focusing on extracting the information from the dataset. It has the following main steps:

- **objects’ front direction computation**: which influences the orientation and positioning of the object in the room (e.g., a TV or screen would usually be with the back near a wall and therefore facing the room interior, compared to a big desk in a conference room occupying a center position). A novel approach to tackle the detection of forward direction of the furniture pieces in an automatic manner is presented; it is based on a series of general observations around the relation between object positioning and deducted functionality, such as: front direction of models near a wall usually have their forward direction towards room, perpendicular to the wall (wardrobe, bed); functional groups of furniture usually face the center of the group, (chairs around a table); objects usually face wider open areas, containing more details in the room than the other faces (probabilistically, a person would usually use the object from that side of the room). Accuracy reported by the authors for this step is encouragingly high (85% to 90%), reporting poor detection of forward vectors mostly for objects hard to figure out, even for human expertise, such as bottles, bonsai, or a “cube style cabinet”. Also, for some objects like a symmetric table, the surrounding objects and walls represent a decisive factor in determining the front direction.

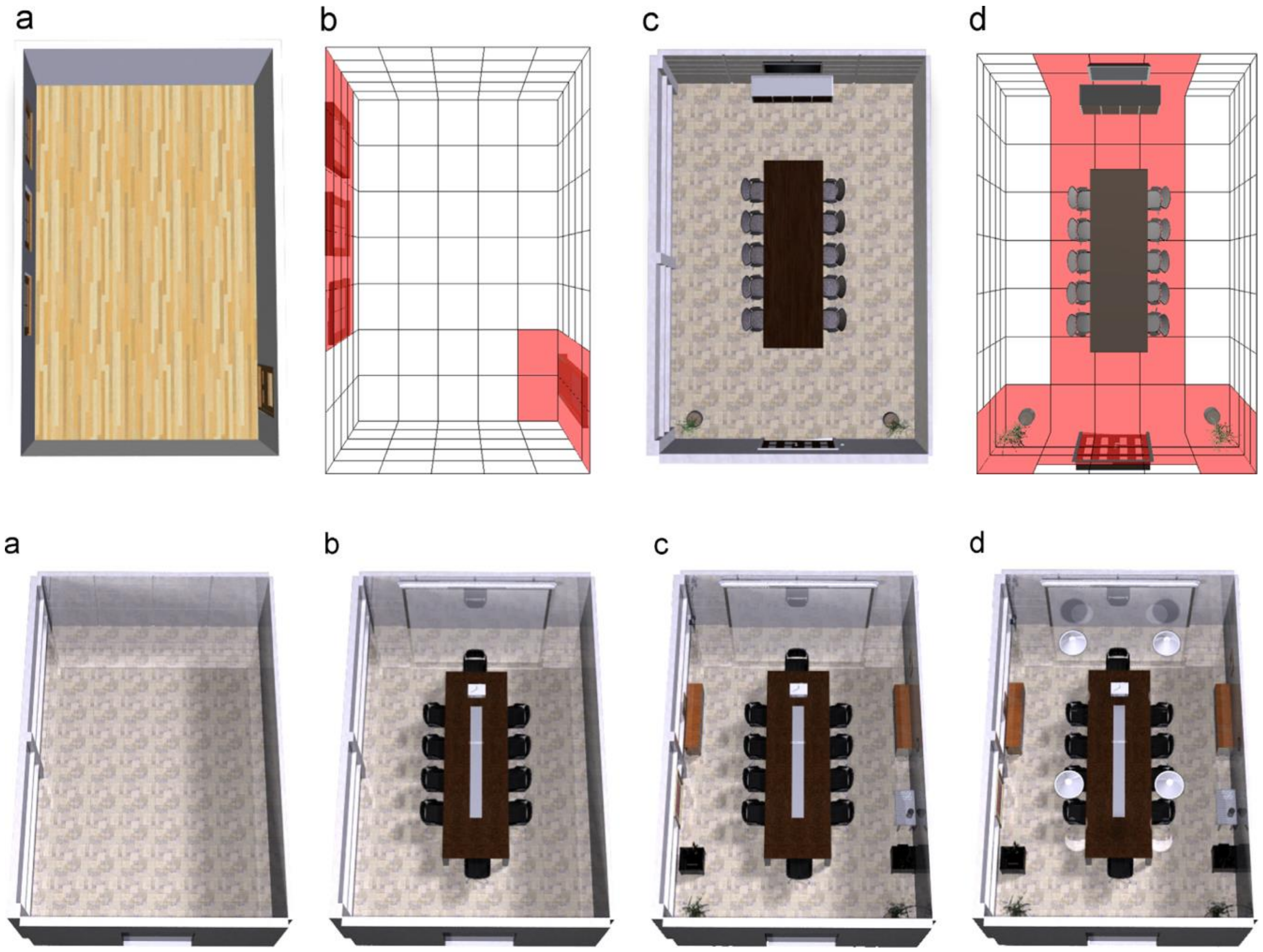
- **Functional Groups (FGs)**: A new concept, the functional object grouping is introduced. It is necessary to obtain unique, compatible, artistic style of the object collections rather than putting objects together only by functionality. As an example: a table should have all the chairs of the same model, and the overall look and feel should be pleasant. An FG is represented in the system as a graph: nodes are the objects and edges the relationships between then. Two types of relations are defined, namely: “center element” – in which an object points towards the center object of the FG it belongs to (i.e. each chair is linked to the corresponding table) - and “supporting relationship” – in which an object represents the supporting surface for another one (e.g., a table and a PC). The FGs are extracted automatically from the scenes in the dataset, based on manually-added relations between furniture components. In the following steps, these abstract grouping in functional units will represent furnishing options, given the functionality and user’s desires. One important aspect is that FGs only contain information about the functional relations between furniture pieces (i.e. groping in favor of supporting certain human activities, such as: conference table with chair is a communication / socializing FG, a desk with a chair is a working FG, etc.) and not spatial and positioning information (relative to the room). These FGs, treated as single entities in the next steps, will be used to generate the room layouts (in contrast to the individual furniture pieces) preserving in this way hard-to-learn anomalies such as different types of chairs for the same table, asymmetric decorative objects, or different type nightstands, therefore preserving the overall style and professional aspect of the resulting layouts.
- **Wall Grid Structures (WGS)**: The main novel concept introduced in this paper is

represented by WGS which are the “template” for learning the room overall appearance, at a very abstract level. The WGSs are used for learning the scene arrangement information using a set of grid cells along the walls and symmetry axes. In the first phase, these are used for a detailed analysis of the arrangements of objects at an abstract level, including symmetry, probabilities of models to be placed at certain positions in the grid and *vice versa*: for models, the best positions can be determined. The WGS will model a probabilistic placement of the types of previously computed FGs in various types of room, for different space ratios.

The second part, the “synthesizing stage” is the one in which a furnishing option is suggested for a given room. This is done by automatically determining the WGS of the given room, followed by suggesting appropriate FGs. The system will start with a main FG, called “seed” that is selected among the most frequent items in that type of room (80% and more), e.g. a table with chairs in a room conference, a bed in a bedroom, etc. Following the placement of the main seed, secondary “supplementing” FGs are added (e.g. a visual FG – that might contain a TV or a projector, accordingly, etc.). On the first row of Figure 8, the room occupancy information of an initially empty room and after adding few FGs is presented, while the second row presents an empty room to which the main FGs are added followed by secondary ones towards a fully furnishing layout.

The system presented in [14] is thus a novel approach for generating valid, unique, artistic furniture configurations based on probabilistic learning of furniture placement, using WGS, and a library of models, grouped in FGs in order to conserve style and general appearance, as well as to reduce overall scenario complexity by removing certain degrees of freedom, hard to capture in all their completeness in the learning phase. However, moving the focus to the context considered in our case – a real-world, customer-oriented scenario, the model shows a number of limitations, such as: feasibility of furniture grouping in FGs – resulting in exponential feasible combinations when a large dataset of many compatible furniture pieces is present, common case in a furniture selling company comprising a vast library of objects; restriction of the room plan to general cubical form – which is enough for most conference rooms but not always the case for personal bedrooms and living rooms.

Figure 5. First row: Occupancy information of an empty room to which main FGs are added. Second row: Steps in the process of adding main and secondary FGs towards a fully furnished room.



Although probabilistic models have been successfully used in capturing abstract patterns in data driven approaches, synthesizing real-world-ready scenes, with application to commercial scenarios such as furniture sales, or automatic furnishing of houses, remains an open problem. Moreover, a pure data-oriented model is still far from capable of capturing rigorous and stylish furnishing subtleties towards achieving homely, artistic, construction-ready solutions.

2.3 Generative models

The problem of generating diverse building models, from floor plan generation and building façade to interior design furnishings and including realistic decorations in a scalable fashion, has to some extent been addressed through methods that aim at learning from existing, real-world entities to generate new feasible ones.

Briefly presented as a probabilistic model in section 2.2, the study reported in [11], for instance, focused on generating various new plausible object models from a few, real ones by evolving new compatible components. This is accomplished through a probabilistic model linking properties of the components' shapes and learning the plausible variations within a context. In [15], the authors created a model able to, first, determine the space of plausible, local variations of building layout and, second, merge such local derivations through a linked transitions graph with valid pathways at a global level, enabling easy transition in the building space neighborhood.

Compared to the other explored trends, in which the focus was place on iterative optimization of certain goals towards a desired solution (or a few similar options narrowed manually to one), the models described in this chapter aim at obtaining new valid models which are "inspired" from real ones and are built from compatible components. The goal here is *generating* considerably more models while preserving overall consistency and diversity, starting from only a few ones. The realistic generation capabilities and cross-components similarity learning of these models also have application to interior design for ornamental decoration of rooms [16], where new artifacts need to be generated and well-placed in the room, preserving user preferences and the overall style [17], as well as the functionality of the space, while maintaining diversity.

Another area related to interior design and room layout generation is that of automatic ornamental decoration. Although it may be seen as more of an option, this feature is particularly important for transforming an empty room into a livable one. Specifically, the aim of this process is to populate the empty furniture pieces (such as shelves, wardrobes, tables, walls, etc.) with adequate artifacts, preserving the style and overall arrangement while providing overall utility and functionality. In the scenario described in this thesis: a commercial tool for selling furniture, offering clients the possibility to view online few, representative virtually furnishing layouts adjusted to personal room plan, this feature would be of key importance, making the client presented layouts more desirable and adding a “ready to be lived in” component of the mixture.

A data-driven, ML approach in this area was presented in [16]. As argued there, the challenges imposed by generating furniture decorations are slightly different from the ones of furniture arrangement optimization in the following aspects:

- Existence of a much more diverse set of artifacts compared to the furniture types that can be present in a certain room.
- The model for generating artifacts arrangements has to be more scalable, ensuring diversity and variability according to the magnitude of the decorative space, style and functionality. In comparison, furniture variance is more restricted due to the room space / objects dimensionality ratio.
- The arrangement of artifacts is a more subjective matter than design itself. For instance, two different persons might both agree on the same professional designed kitchen furniture, but might as well have different opinion on how artifacts should populate the furniture. Decoration is, thus, a rather more sensitive, personal matter. In more detail, and as stated by the authors, personal preferences affect: overall artifacts symmetry inside a shelf or vertically aligned, density of objects, grouping style - similarity based versus functionally / activity oriented, overall aspect and organization (e.g. tidy, messy, strict).

Another challenge addressed by the research presented in [16] was obtaining diversity in artifacts types and forms, something that was unfeasible through a stochastic optimization procedure, resulting in a unique solution or in few ones with very similar decoration results. Therefore, compared to furniture synthesis models, a valid space of solutions is defined here, using a set of inequality constraints, and the optimization process aims at bringing the solution within this valid space, as opposed to a point in the search space. Optimization steps such as object addition, deletion and inter-change are defined and applied in random order sequences in order to drive the solution towards the goal and assure diversity.

The work in [16] proposes a solution for the challenge of considering the personal arrangement style of artifacts in generating personalized decorative solutions based on an image or a 3D model provided by the user with an already decorated “cabinet”. Some examples of decorations generated based on user input (“Input Exemplar”, shown in the left side) are displayed in Figure 6. In this prototype, the image is manually annotated with the user’s help. The model then analyzes the personal style properties at two levels: an *object level*, consisting of: distributions on shelfs, adjacent relationships and placement in furniture; and a global/general level, consisting of: object density, grouping of similar objects (e.g. plates, glasses), variability, and overall symmetry. After understanding the user style and ordering preferences, the model is able to automatically fulfill furniture in a scalable fashion, preserving the style properties described above**.**

Figure 6. Decorating results according to personal style preferences, as reported in [16].

3 The Interiorvista Company and the research project description

3.1 Motivation

For a non-expert, properly furnishing its personal room is usually a difficult task. Some common factors contributing to it include:

1. a high variety of options for furniture pieces available, plus the ability to highly customize each of them (including coloring, type of wood, handles, doors, etc.)
2. the difficulty of doing proper measurements and choosing appropriate size pieces of furniture to fit their personal room. Often, clients buy furniture that looks great in the shop galleries but does not necessarily fit well in their rooms.
3. large, very detailed shopping lists: especially when buying furniture pieces that are custom-made (i.e., from parts), or highly customized.
4. the lack of professional expertise and guidance for matching personal preferences, furniture style and available room space with the best available furnishing options.

3.2 The Interiorvista Company and its products

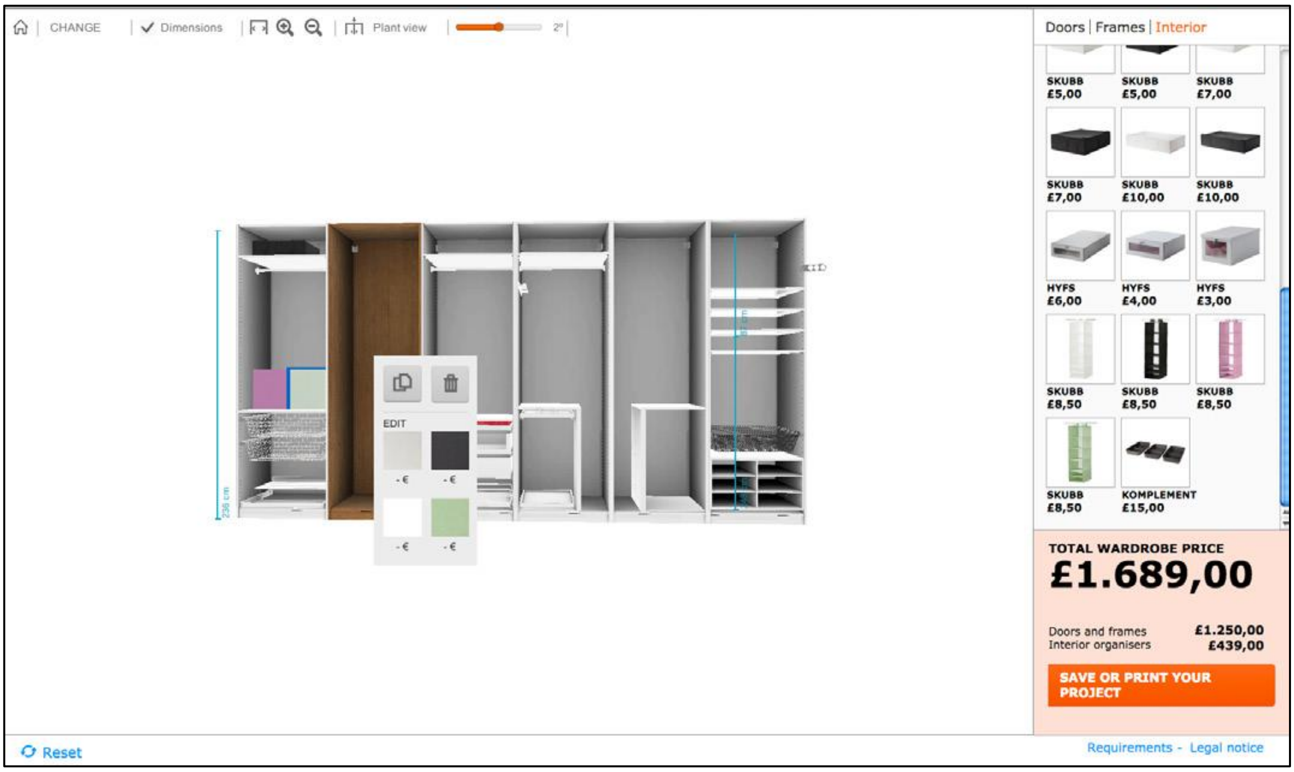
Interiorvista [18] is an interior design company that aims to empower furniture selling companies, such as Roca [19], to sell “more, better, fast”, their main products. Through its products, Interiorvista’s target is reducing the burden and complexity involved in furnishing an empty room, while inspiring and guiding the customer with a high-quality images catalogue of available products, captured in various, expert designed scenes. For that, the company provides:

1. a software tool, named planner, meant to assist users online in virtually furnishing different types of rooms (i.e. kitchen, bathroom, bedroom), given their specific room

plan. We will present more details in the following section, where we correlate the real world challenges imposed with the main objectives of this research project.

2. a furniture piece configuration tool – meant to help the client ease the process of building a customized furniture piece, by choosing each component type out of a variety of shapes and colors. The tool is meant to be integrated in the existing selling software of the furniture provider, in such a way that a final price and a complete shopping list is generated once the customization is accomplished. A snapshot of the tool is presented in Figure 7, showing the construction of a custom wardrobe, adapted to the desired room space and shape and with personalized interior partitioning. The tool also provides a total price for the wardrobe and can be purchased directly at the final step, by sending to the furniture provider the list of all pieces. The interface is simple enough to allow the user focus only on the personal important aspects, such as shape, choice of components according to their features (i.e. sliding door or glass door, coloring, internal shelving), etc., leaving the burden of computing the total price and the detailed list of components to the tool.

Figure 7. Snapshot of a custom wardrobe built in the tool



virtual galleries and paper catalogues with professional scenes - meant to expose, in an artistic style, numerous possible combinations of available furniture options in a variety of styles and scenarios. Figure 8 emphasizes a gallery comprising various styles of chicken furnishing layouts and in Figure 9 we present a single, zoomed one.

Figure 8. A gallery comprising various styles of chicken furnishing layouts

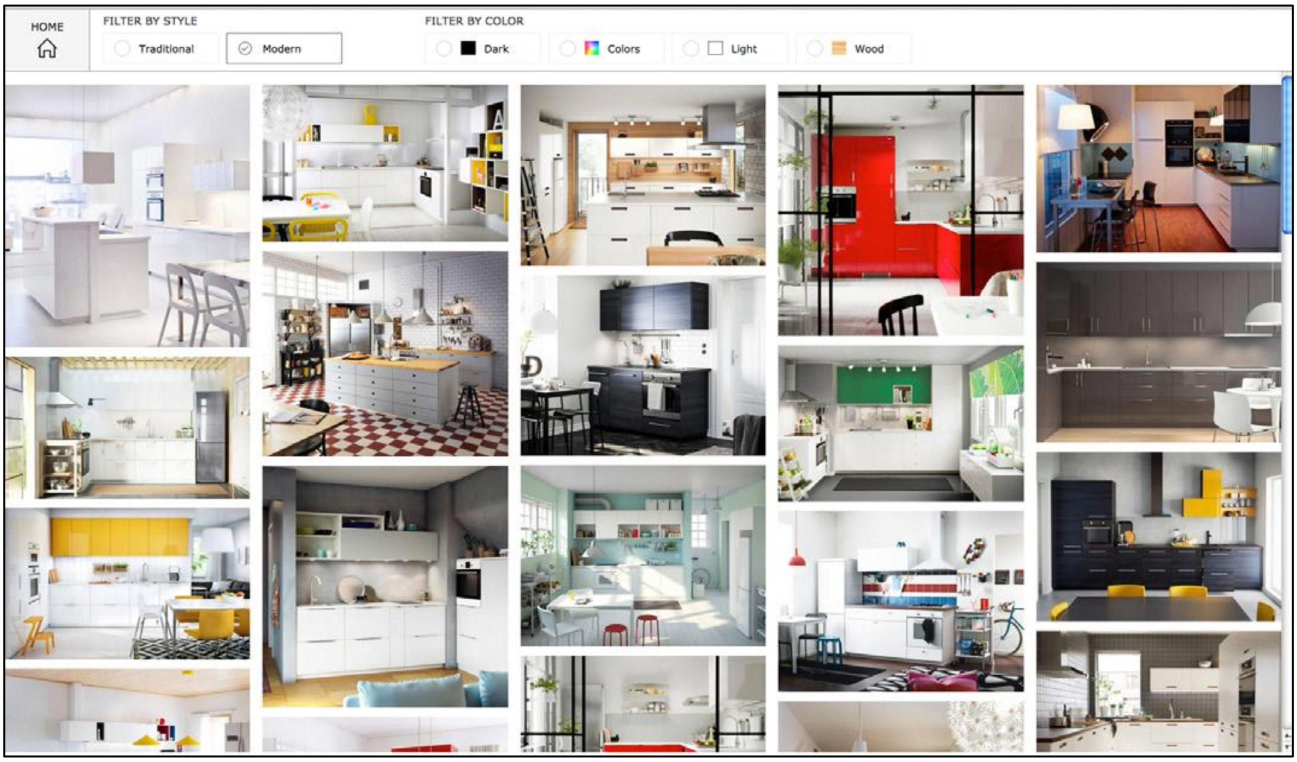
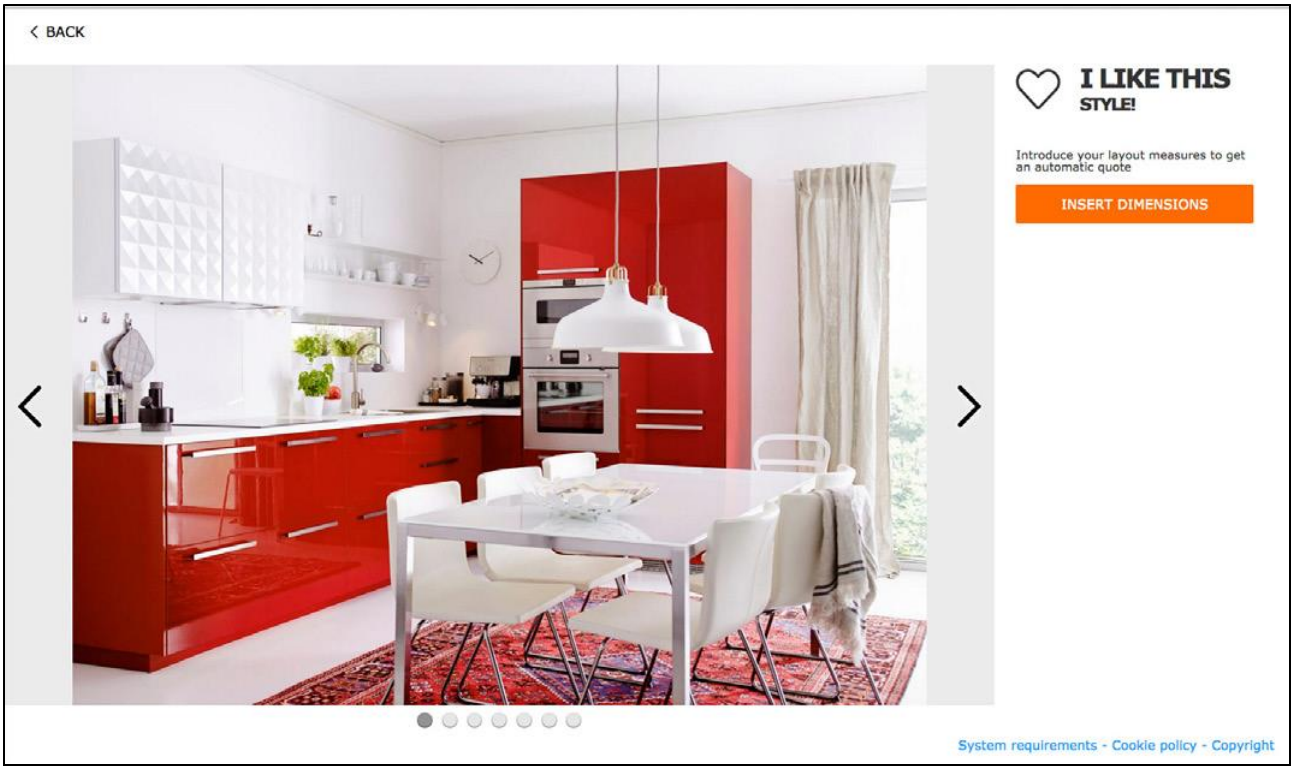


Figure 9. An example of furnished kitchen



3.3 The software tool (planner) for virtual interior furnishing

Representing one of the main software tools at Interiorvista, the planner main goal is helping the clients to virtually furnish their rooms with the desired, chosen furniture style. This gives the potential customers a free, easy, and online way of viewing their personal room (i.e. kitchen or bedroom) decorated in a certain style or with certain furniture pieces desired. In this way, the customer can have a better understanding and 3D visualization of their room and furniture that goes inside before making the purchase, thus helping with a faster decision and less unpleasant surprises after buying, such as incompatible sizes, or shapes, unusable free space, inaccessibility or improper usability of the room, etc.

The planner also comprises a library of furniture pieces available to buy from a certain provider, as 3D graphical models, together with metadata including: price, available customization (shape, dimension, positioning, color, material) and usage constraints (i.e.

suitable type of rooms, space needed for proper operating the object, safety constraints and handling, etc.)

We present, in the following, a scenario case for a potential customer furnishing a kitchen, outlining the stages and interactions with the Interiorvista planner:

1. **Starting point: gallery *vs*. style.** In order to simplify the process as much as possible and guide the user towards professional furnishing solutions, the customer is presented with two options in the first step of the planner wizard: the "start by gallery" and "start with kitchen sets/styles". This allows the user, in the first case, to select a preferred option among various kitchens that were designed by experts, using only furniture provided by the company (e.g. Roca). The furniture IDs in each image are known to the planner, resulting in an automatic furnishing of the kitchen with that furniture set, completed in a later stage. The second option allows the user for directly selecting a pre-made furniture set, with a certain style and functionality, that will be adjusted by the planner to fit their personal kitchen. The screen seen by user at this step is displayed in Figure 10.
2. **Room plan insertion.** In this step, shown in Figure 11, the user has to input the room plan (i.e., the kitchen), through simple drag & drop actions, and resize and positioning commands. In the planner, this is defined through:
   1. the floor plan: namely the walls, windows and doors, including dimensions and positioning.
   2. the "gas point": marked with a gas icon, which is needed in order to know the positioning of the gas-dependent furniture pieces (e.g. oven).
   3. the "water point": marked with a water icon, is used to position the water dependent pieces (e.g., sink, dish washer).

   In Figure 11, we show different user interactions with the planner at this step, such as: resize / reposition walls and windows; positioning the heat and the water points. In the case collisions appear, the planner will highlight the items in red and the kitchen creation process cannot be completed (as shown in the last snapshot of Figure 11).

Figure 10. Interiorvista planner: starting with gallery vs. style.

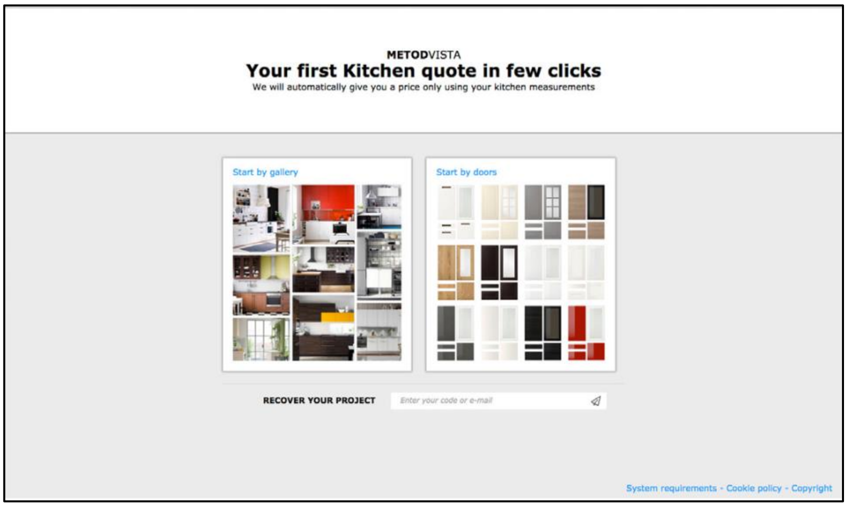
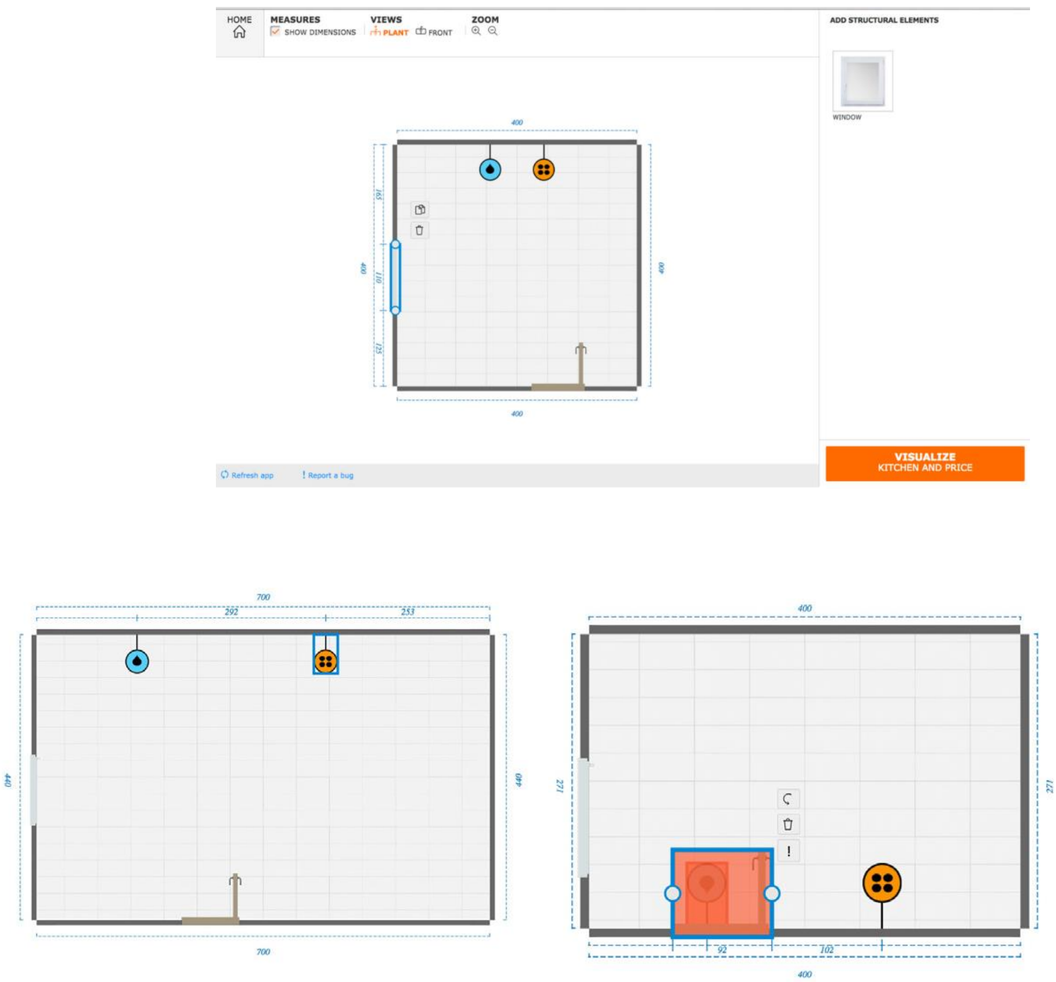


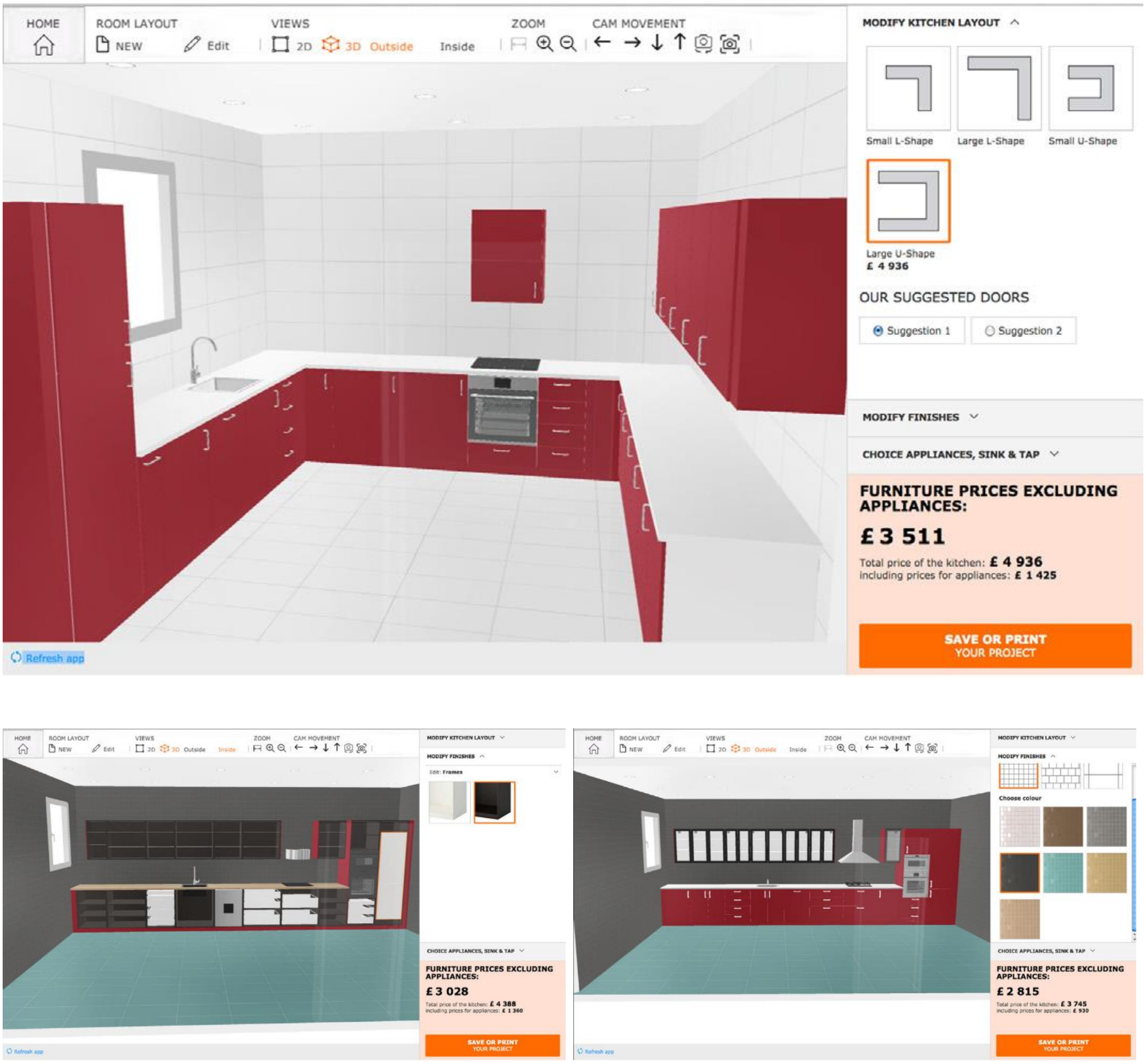
Figure 11. Interiorvista planner: kitchen plan drawing.



3. **Furniture generation.** After the floor plan is complete, the customer can advance to the next step in the planner wizard, namely, visualization of the furnished room, which is available in both 2D – front and top and 3D. This step is the most important and also the hardest, because furniture generation is a complex task. The current process, in the case of kitchens, is formally described according to the following outline (details of the algorithms used being private):
    a. *Input*: this consists of the following:
        - furniture set to use for furnishing - synthesized in the first step when the user selected a picture of their preferred furnished kitchen or a style.
        - the user's kitchen floor plan – drawn in the second step.
        - the kitchen furnishing shape; current options include: along a single wall, *L* shape, *U* shape – with customizations on the furniture occupation dimensions.
    b. *Possibilities generation*: using these and static, predefined templates and rules, the engine generates the maximum plausible furniture combinations. These patterns, manually created, aim at massively reducing the number of total possibilities (e.g., putting furniture only along the walls, certain alignments and order of different types of furniture pieces, etc.).
    c. *Filtering*: invalid combinations, according to extra, static defined rules are tested. Some examples of rules are:
        - distance from the door to the closest furniture has to be X.
        - fire point-to-water point minimum distance is Y.
        - an object cannot be placed in front of a window (to allow the light to enter the room).
    d. *Scoring*: Other manual, empirical designed rules and functions are applied in order to score each remaining furniture configuration. These are highly related to common sense (i.e., what a person would value and would avoid). Therefore, they penalize bad aspects (e.g., unusable free space, small distance between similar furniture pieces, natural light blocking, fridge is between two lower, equal-size furniture pieces), and rewarding positive ones (e.g., a larger area of working space, alignment, compatible functionality grouping, etc.).

e. *Output best result*: Based on scoring, the best furnishing option gets selected and, according to customer selection, the 2D or 3D view is rendered. A few examples are presented in Figure 12. As seen, the user also has the option of viewing the inside of the furniture, including internal space partitioning in drawers, fridge, oven, dish washer, etc.

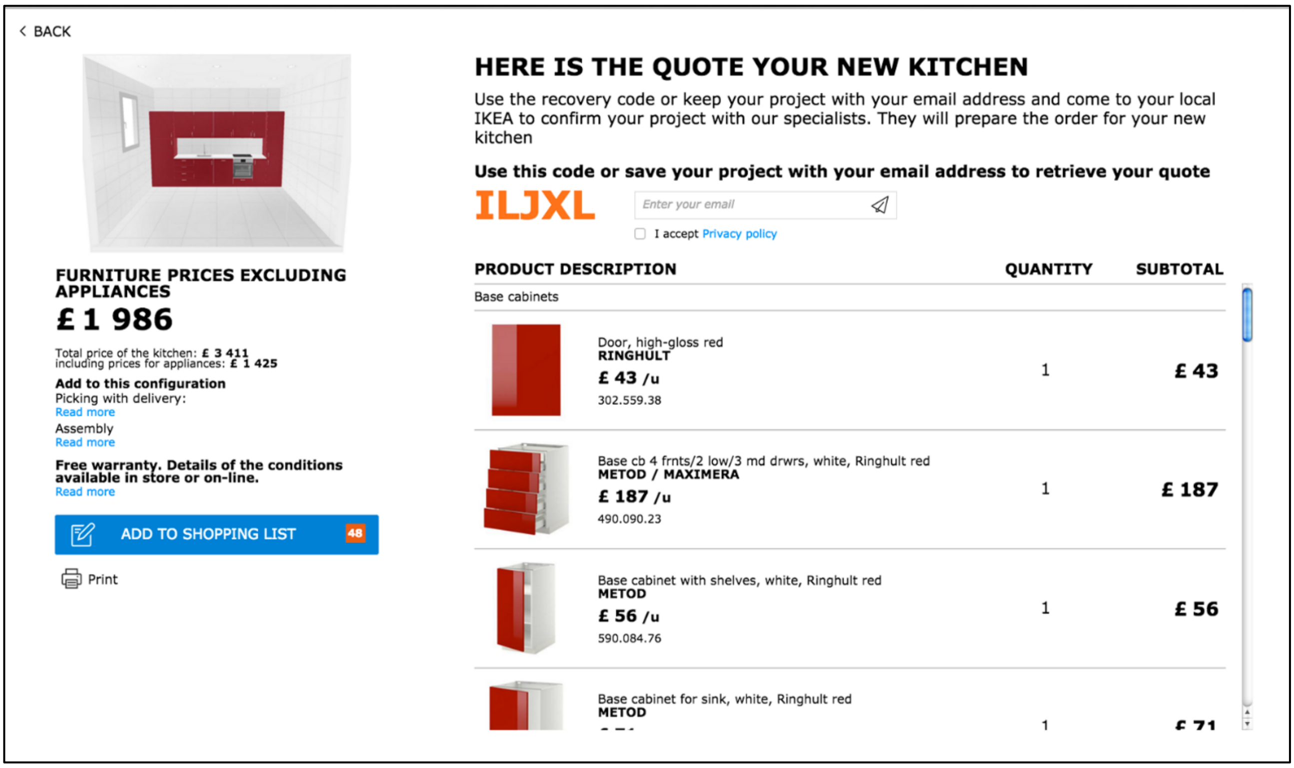
Figure 12. Interiorvista planner: kitchen furniture generation.



4. **Save and print.** The final step of the planner allows the user to save the project online, giving a recovery code to allow her or him to return to the website and resume the work

(i.e. restore the project), as shown in Figure 13. Besides, the tool generates the shopping list to be submitted to the furniture provider, containing all the furniture pieces, together with all the customization and components details needed and including prices. The list can represent a potential burden for a customer when manually built, having to contain in-depth details about the furniture pieces, customized components, etc., written in a compatible terminology with the furniture provider. The planner removes this burden by automatizing the overall process, providing the shopping list and the final price once the planner riches this step.

Figure 13. Interiorvista planner: Project save and shopping list



3.4 Algorithms limitations and challenges

Automatic furniture arrangement is an open-ended research topic, being too complex to yet have intelligent software to achieve results compared to a professional interior designer. The Interiorvista planner represents a first step into automatizing furniture arrangement in order to ease and enrich the customer shopping experience in this area. The software, currently able to furnish kitchens and bathrooms, provided with a specific floor plan, uses static patterns and manual scoring measurements in order to generate a complete furnished room.

Specific for the kitchen and bathroom, the diversity and complexity of the scenarios are not as challenging as in more complex types of rooms, such as the living room and bedrooms: the furniture is usually placed along the walls or in certain positions for specific furniture pieces that depend on wall pluming (for instance, the sink near the water point, the oven near the gas point, the toilet near the drain, etc.).

Another reason for a classical, hand-crafted algorithm to work (as compared to a computational intelligence model) is that, besides usually lacking much complexity and variations, the kitchen and bathroom furniture is also constrained by many safety regulations and best practices, both being rather rare in the case of other types of rooms, such as bedrooms and living rooms, which, in contrast, allow more freedom, personal style reflection, various ways of placing and grouping furniture according to specific desires, functionality and daily routines. Moreover, for the latter case, almost each furniture piece can be present in a room in almost any position, in various and complex surrounding scenes, strongly related to the human personality, including daily activities, job, marital status, culture, background, preferences. All these contribute to deep, underlying causes for a furniture layout to be chosen by a customer and are key selling points for any player in interior furniture selling business.

Another drawback of the current planner is the lack of an option to easily allow for diversity and variation: after the user selects the preferred furniture set, either by choosing a picture or a kitchen predefined set, the algorithm populates the room according to some static templates – which results in very similar styles of furnishing, with small variations of object positioning. This aspect is not acceptable for the more complex room types, such as bedrooms, where there is a great variation in arrangement.

Therefore, as briefly outlined in the previous sections, our project aims to define a CI model that would address the limitations of furniture arrangements by combining a data-driven learning approach, aiming to generate furniture arrangement that emulates customer preferences, with rigorous professional interior design guidelines, incorporated analytically towards achieving more complex, diverse, human oriented room layouts, with a professional outlook.

4 CRISP approach for data mining

4.1 Problem understanding

Being a complex, human expert dependent field, automatization of room interior furnishing is better tackled from a computational intelligence modeling perspective, because of the numerous factors interacting in complex ways towards the generation of an output. This approach can in turn be addressed as part of the *conceptual umbrella* of Data Mining, which goes beyond data analysis to cover a wider range of issues such as the understanding of requirements in order to successful tackle the challenges imposed by interior design automation. We dedicate this section for analyzing the dependency, requirements and expected impact of data in developing an intelligent system for automatic room furnishing.

4.1.1 Motivations for a data based, learning model

In this project we deal with a problem of learning interior design practices to provide complete furnishing options to potential customers. As in any ML-based approach, data plays a crucial role in the possible success of the modeling process. In the following, we present the most important criteria for deciding upon an intelligent system, in contrast to a classic, strictly algorithmic approach, and the data role and requirements for providing an answer to this challenge:

- ***The lack of a clear choice of a deterministic algorithm to solve the challenge of interior design automation***. Although numerous attempts to tackle the problem from a non-learning perspective can be found in the existing literature, none can stand as the definitive choice, given the impossibility to formally benchmark it against human experts in field.
  These algorithms often drastically reduce the complexity of the problem with unfeasible constraints for a customer-ready, real world appliance software, often involving: very limited library of furniture pieces that can be present in a room; grid representation of the room floor with few options for furniture positioning – usually

determined *a priori* and only analytical; oversimplification of the furniture style, functionality and representation; limited room shapes (i.e., often only rectangular, within some dimension constraints). All of these result in oversimplified, improbable room layouts, with unrealistic furniture that cannot be used in our complex scenario, where the software tool is bound to use real-world furniture (i.e., properly modeled as 3D objects, with consistent metadata including detailed specification and safety guidance, various customization possibilities, available sizes, functionality available, price, etc.), which should be used to design client specific room plans with various shapes and sizes.

- ***The complexity of the problem being modeled:*** because of a potentially infinite, highly multidimensional search space of possible bedroom interior layouts, deterministic algorithms, which would generate, combine, filter or search this vast space are not feasible. Alternatively, intelligent, data-oriented models that can learn and capture significant patterns and trends, resulting in a probabilistically representation of the problem, would be advisable.
- ***From a user perspective, the final interior layout is highly influenced by its personality and life style, which depend on numerous personal factors such as: age, marital status, gender, social position, background, daily routines***. These key factors, contributing to an appealing interior design for the potential customer, can be modeled only through a data-driven learning model and are paramount in the development of a successful, customer-oriented commercial tool for the design company.
- ***The native capacity of a learning-based model to adapt, evolve and increase the captured complexity through training***. Another characteristic, adding to the overall complexity of the field, is what we could call “trendiness”, which is perceived in numerous aspects, such as: continuous changes in design and style, furniture evolution (i.e. in style, design and capabilities), daily routines and global life style adaptation, overall room’s shape and dimensions variation.

  Interior design is a field that is continuous evolving in all its components and perspectives. A purely analytical representation of the current “rules” that govern interior design might not be the best choice for the near future because of the volatile, ever-evolving nature of the problem. Addressing such a sensitive, real world scenario – a commercial tool for selling furniture, by presenting the customer various options for furnishing their real bedrooms, being in line with change and recent trends is a must.

Therefore, a purely analytical representation of the current "rules" that govern interior design might not be as accurate to represent the near future trends. In contrast, a data-oriented model, designed to combine an analytical approach (i.e., one that captures strong interior design analysis and regulations) with learned underlying patters and templates would natively remain up-to-date when trained regularly with recent data.

4.1.2 Data challenges in the interior design field

As in any Data Mining stage-based approach, the data modeling stage plays a crucial role in the overall performance of the system, representing a main component of the general architecture. Moreover, interior design automation is still an open-ended research area, still in its very early stages, at least as compared to other real-world application fields.

Models that can make sense of data through learning can only be applied after detailed pre-processing of the 3D models involved. Because of the complexity of such representations, numerous options may emerge, usually resulting in features that are highly dependent on the model applied and often only modeled in a semi-automatic manner.

Currently, no dedicated model or paradigm can be successfully applied to solve true design challenges (i.e. as tackled by a professional designer) and intelligent models performance heavily depends on thorough data structuring and representation. As the results of so many constraints, well-defined, complex, representative and research-oriented datasets are extremely hard to come by. Moreover, in our case, many professionally designed rooms are not made public by interior design companies, for obvious commercial reasons, making the process of data modeling including data acquisition, enriching, and preprocessing even harder than usual.

4.2 Data understanding

4.2.1 Main characteristics of initial data

Given the complexity and nature of the problem we tackle, automatic furniture arrangement for a personalized room plan, an ideal dataset would have to contain a variety of professionally-designed furniture arrangements that clients agreed / purchased to furnish their personal bedrooms. This would assure that the bedroom's interior designs were professional (i.e. would agree with style consistency, ergonomics, desired functionality but also were preferred, feasible to normal persons furnishing their rooms (i.e. had a "fair" price, meet their expectations, obey their routines, life style, etc.).

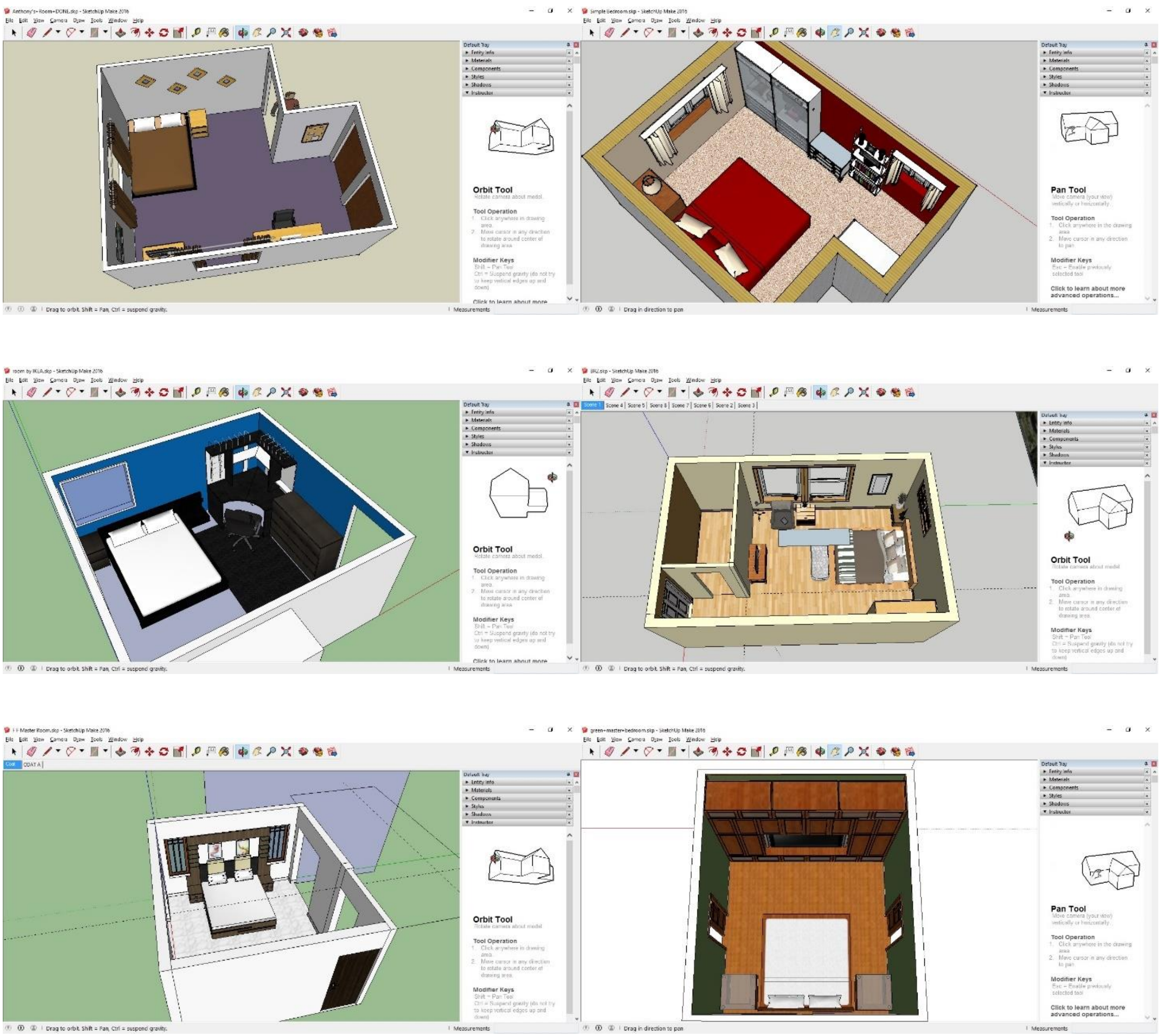
Moreover, each entity in the dataset should contain enough metadata in order to extract numerous details, which will represent the features for our learning model, about various aspects such as: the furniture type, functionality, positioning - both global (i.e. relative to the room coordinates) and local (i.e. relative placement in its neighborhood, distance to surrounding objects, coupling for different functionalities, etc.).

4.2.2 Data description

Because of the required complexity and details needed, we considered only entities represented in 3D model files, which had to be designed by humans. Taking a look to the existent state of the art, no definite choice for a publicly available dataset that would meet these requirements was found. Moreover, Interiorvista did not have any kind of database comprising already furnished bedrooms (neither by professionals nor amateurs) that would have been validated (i.e. purchased by clients or designed by professionals).

Therefore, in the described context, we considered a subset of 12 representatives, manual designed bedrooms from Google 3D Warehouse [20], which were created in Google SketchUp [21], by amateur designers. Some example screenshots of these rooms are presented in Figure 14. These were selected manually by us, based on the online comments, purpose of the designed bedrooms and room "popularity" (i.e. agreed / liked by others). Such room aspects we considered, which aimed towards a real world scenario, included: personal bedroom (or of a friend, etc.), desired bedrooms, guest rooms.

Figure 14. Example snapshots of the bedrooms in the dataset described in the main text.

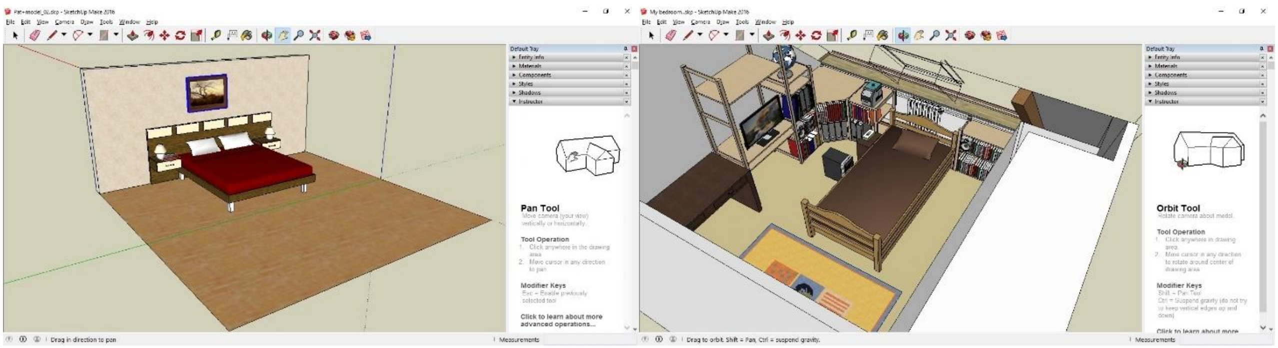


Plenty of diversity can be seen in the rooms on display in Figure 14, and different underlying purposes for each choice of furnishing can be easily depicted from the title of the file (added by the owner / designer) but also from the general outline. For instance, the 5th bedroom in Figure 14 addresses a guest scenario in which there is no need of working or activity related activities – most focus being on short term stay and sleep. In contrast, the 2nd bedroom depicts a very large bedroom, furnished with three wardrobes and two chest of drawers which is oriented on long term leaving, for a person disposing of plenty of depositing space. There are also bedrooms that combine default furniture (i.e. dedicated for a bedroom) with furniture for leisure and working activities, for day routines, as seen in 1st and 3rd bedroom, which have a desk / table and chair.

Although a lot of variation in style, functionality, personal taste can be depicted from the data, we excluded from this set extreme cases, such as: too complex bedrooms in terms of the shape of the room or the furniture involved and too simple ones that would not include a representation of the walls and windows or would rather model in detail a certain piece of furniture rather than a bedroom scene.

One example of each are presented in Figure 15, where the first image depicts a room without walls and windows which outlines a certain style of bed rather than a plausible, complete bedroom and the second illustrates a complex bedroom, in an attic, with dedicated, complex furniture to maximize the usability of the limited space available.

Figure 15. Example snapshots of bad bedrooms, not included in the dataset described in the main text.



4.3 Data preparation

4.3.1 Feature selection

As seen in the related literature, machine learning models dealing with furniture sampling and arrangement usually need training data with numerous features, which are meant to deal with the high complexity of the problem. Being still a research field in its early stages, there not exists a preferred, widely used set for features to be extracted, nor some compulsory, ever-present one. Moreover, most related work models rely on heavy dataset preprocessing and annotations / tagging, which can be automated only to some degree. These pre-processing of the training data is usually highly correlated to the model designed and to the specific subset of the problem to be solved (i.e. adding furniture decoration to a limited subset of furniture pieces; synthesizing room layouts from a limited set of furniture, arranging furniture based on a genetic model, with continuous user feedback).

Therefore, choosing a set of features to be both necessary and sufficient for the addressed problem and the models designed was a challenging task. We considered the following set of initial features:

- **ID** – the (unique) id of each furniture piece.
- **(obj_x, obj_y, obj_z)** – 3D coordinates of a furniture piece, relative to the room (i.e. global world coordinates).
- **(alpha_x, alpha_y, alpha_z)** – rotation of the object relative to the room (i.e. on each ax of the 3D space).
- **(fwd_x, fwd_y, fwd_z)** – the forward vector of each object, which will first get extracted analytically from the 3D scenes and then approximated / represented from a human-like perspective (e.g. a bed will have its origin in the left-top corner and the fwd vector will point from head to toes on its longitude).
- **(length, width, height)** – of the object bounding box.
- **min_dist** – the minimum distance between every 2 furniture pieces, computed on the floor plan space (i.e. 2D space). Only main furniture pieces are of interest for

this initial model, which focuses on main furniture pieces composing a bedroom, disregarding decorative or artifact objects such as: nightstand lamps, paintings, pictures, books, bed pillows, etc.

4.3.2 Data cleaning, correction and reconstruction

As previously presented, in the lack of a best-choice dataset comprising of professional designed bedrooms (i.e. by a company or experts in field), or bedroom furniture layouts which would have been approved (i.e. purchased or liked by customers), the dataset agreed with Interiorvista to be used in the project was collected from Google Warehouse [20], and comprises a selected subset of 12 good bedrooms, available as 3D design files, some examples of which are shown in Figure 14.

Being highly related to interior design, and therefore implying the use of dedicated architectural software tools and knowledge, the processing of the dataset was carried out with professional help from Interiorvista experts, which contributed and adviced with model correction and with the extraction of the features chosen by us.

3D models manual cleaning and correction

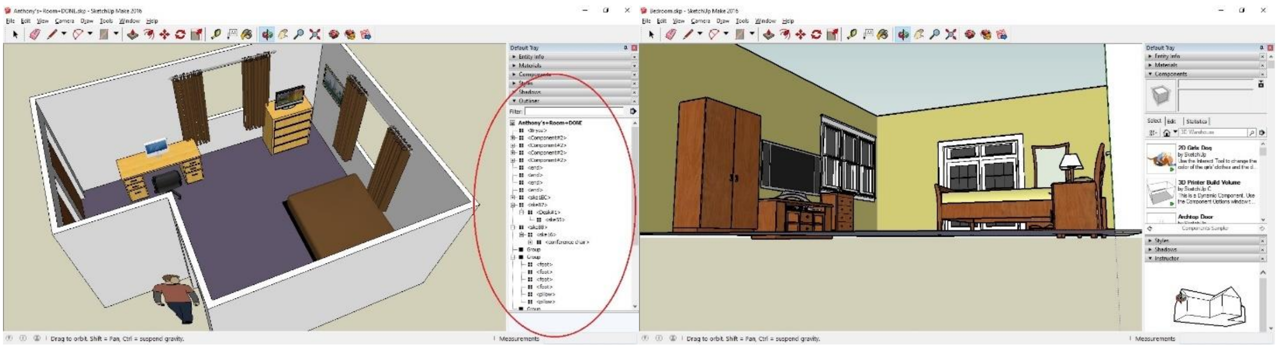
The use of a dataset of models entirely designed by amateurs, who lacked any prior knowledge about best practices and correct architectural modeling, containing bedroom scenes, entailed the problem that, even though looking correct and coherent, some of the scenes had in fact to be corrected manually for each scene and for each object in order to be able to extract the correct features.

Most common problems, some of which are visually presented in Figure 16, included:

- wrong segmentation of the furniture pieces, subcomponents, and other bedroom entities. Some of them, entirely missing structure and model hierarchy (i.e. coupling the low level details, as edges, surfaces, into entity subcomponents which would then be grouped and named into the respective furniture piece). Such example, to which an exhaustive manual correction was required, is presented in the first image in Figure 16.

FIX: manual correction: involving **segmentation, tagging and rebuilding hierarchy** (i.e. regrouping of subcomponent into whole objects). Besides, local coordinates have been attached to each object, independently of the scene and other models in it, which emulates the object orientation as viewed by a human. Therefore, the forward vector, represented with green in Figure 17 will always have the direction “back of the object” to “front of the object”, in a natural, human agreed, functional manner. For instance, a wardrobe would have the forward vector oriented from back to the door, on horizontal; likewise, a bed from a person head to toes (e.g. on its longitude) etc.

Figure 16. Mistakes in building the 3D models of bedrooms



- incorrect positioning of the global world / room coordinates, not being aligned with the room orientation.

  FIX: **reposition the global coordinates, to align the scene** 3D axes with a room corner, floor and walls. Although a definite choice does not exist, because the forward direction of a bedroom can be ambiguous, even for a human (i.e. because of the variety and nature of the bedrooms), the axes should be at least aligned with a chosen corner of the room, floor and walls.
- scale (i.e. dimensions): in some cases, the furniture pieces did not have real dimensions. The author only focused on the overall aspect and furniture design, not on the scale of the project. Therefore, when extracting the features, we were

confronted with anomalies such as: a double bed of 83 cm long and 58 cm width.

FIX: no fix for the features extraction part, (i.e. from the 3D design files), but taken into account in the latter stages of data preprocessing, which involve, **scaling and normalization of data**.

- the overall design process was not analytically validated by the authors (i.e. probably because of the lack of background), resulting in small errors and non-correlations in positioning, both for the furniture in the room and the entities' components, as shown in the second image of Figure 16. Some examples of these human-made errors would be: slight rotations of the furniture pieces, such as: bed is not aligned with the corresponding wall, but rotated at an insignificant small angle, often being few cms. from the wall, or colliding with it; some furniture pieces going through the floor or walls, as opposed to being positioned exactly on the floor; non-coherent positioning of the walls and windows and doors, which should be aligned accordingly, etc. In consequence, the features extracted have small variations in terms of angles (i.e. close to instead of exact 0°, 90° or 180°), floor relative position close to 0, sometimes negative distances, etc.

FIX: no fix for the features extraction part, (i.e. from the 3D design files), but taken into account in the latter stages of data preprocessing, through according **approximations**.

4.3.3 Feature extraction

After the manual correction and adequate tagging of the objects in scenes, the chosen features, described in 4.3.1, were extracted from the updated bedroom files. The feature extraction process, to which Interiorvista also contributed, was realized in 3DS MAX [22], and Unity [23], using scripts that were ran for each scene. The resulted output was converted into features and exported to MS Excel format.

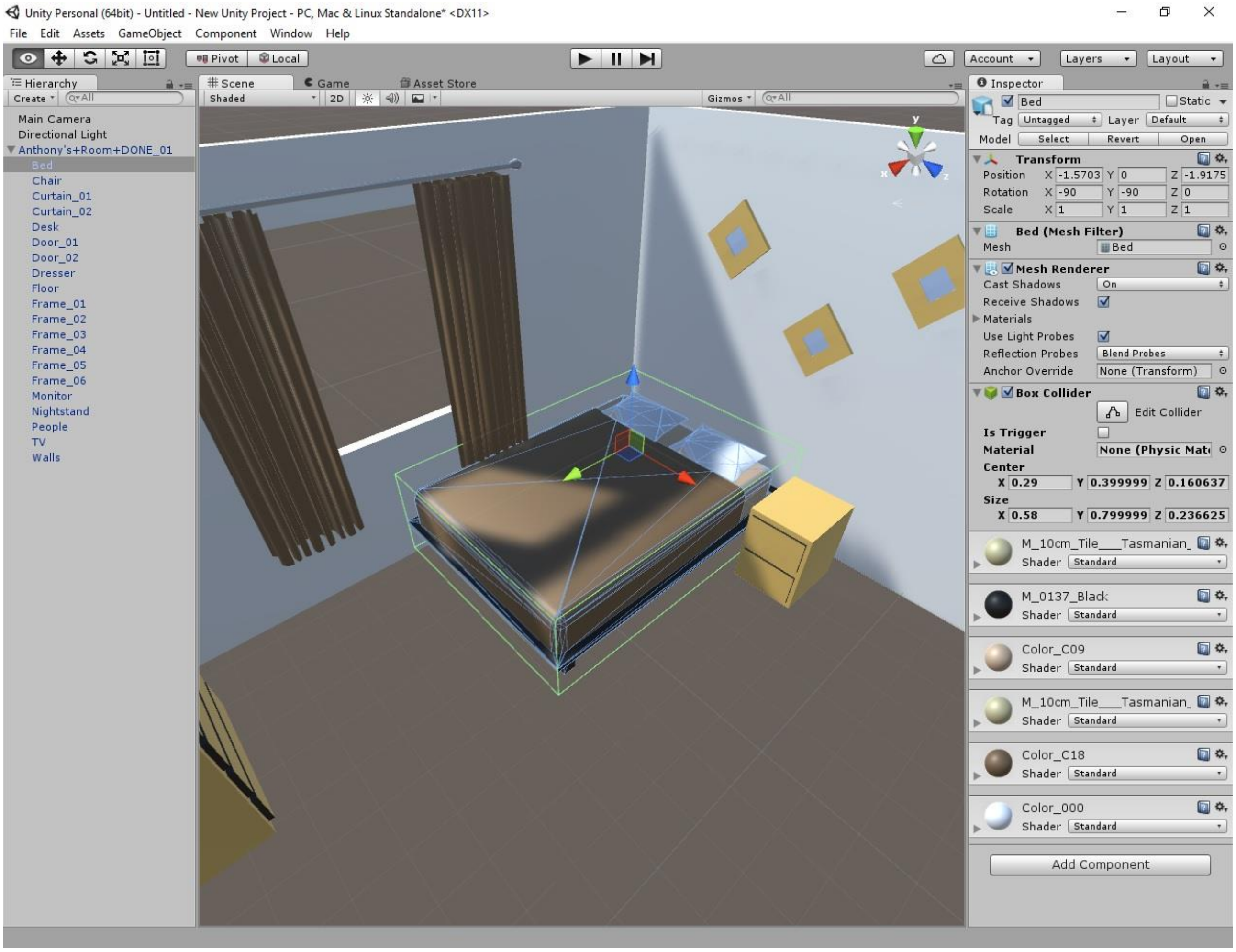
The scripts outputs required measurements for each object, which were used to create the features directly. A visual representation of such details is presented in Figure 17. The screenshot is taken in Unity [23], after the manually fixing stage (i.e. with correct segmentation and object hierarchy, which can be seen in the left side of

Figure 17). We can observe that each model in the scene is now represented as a unit, in contrast to the initial scenes without hierarchy - Figure 16, moreover, attaches a correct local coordinate system to each entity, independently. This object's coordinates reflect the orientation of the object, as seen by a person, always having the forward vector (i.e. the green ax, corresponding to OX) oriented from the back of the object to the front, on horizontal, as briefly described in 4.3.2. This manual correction of each model in each scene was needed in order to ensure a correct, automatized process of extraction of the required measurements, which was needed to build the dataset features.

Two main types of design software and corresponding scripts were used, depending on the team which helped us in extracting the relevant data for the features creation. Using 3ds MAX [22] we obtained most of the data required (i.e. all besides the minimum distances between any objects), outlined in the following:

- object bounding box and its dimensions: represented in Figure 17 as a rectangular, green box.
- the positioning of each object in scene. For this, the scrip measures the distances between the global coordinates (i.e. room ones) and the object's bounding box coordinates (i.e. manually placed for each object in part, to reflect functionality and common sense from a human perspective).
- the rotation of each object (on the three axes) relative to the room. As in the previous case, this is done also using the two coordinates systems: the global and each object local one.
- the forward vector and upward vector details. This data is embedded in the local coordinates, as described before and seen in Figure 17: the forward vector is the green arrow and the upward vector is always the blue one.

Figure 17. Visualization in Unity of the features extracted.



An example of extracted features for a room, using the 3ds MAX scripts without the minimum distances between any two furniture pieces is presented in Table 1. For the purpose of display, the values were rounded to 2 digits. Although the values reproduce with accuracy the measurements in the 3D model, manual changes had to be done for increased accuracy and consistency, such as to decide appropriate approximations to achieve coherence (i.e. all objects do not collide with walls, furniture pieces are at 0 distance from the floor, etc.). These changes have been carried out in the case presented in Table 1.

Table 1. Extracted features for a given room.

Furniture name	x	y	z	rot			FW			width	length	height
Desk	86.4	15.72	0	0	0	0	0	1	0	46	28.25	88
Light_02	25.18	202.39	26.81	0	0	-90	1	0	0	11.41	11.41	16.48
Window_02	265.43	120.99	39.43	0	0	90	-1	0	0	109.95	9.25	76.28
Window_01	181.5	-0.12	0	0	0	0	0	1	0	91.04	8	103.24
TV Desk	257.07	211.22	0	0	0	90	-1	0	0	74.44	24.92	29.72
Wardrobe	256.98	279.55	0	0	0	90	-1	0	0	48	25.49	84
DVD	248.86	166.65	19.92	0	0	90	-1	0	0	16.87	12.37	2.14
Phone	250.98	106	38.62	0	0	90	-1	0	0	5.23	3.9	7.33
Frame	245.2	123.8	39	0	0	90	-1	0	0	10	7	0.25
TV	248.53	215.01	26.96	0	0	90	-1	0	0	76.65	2.78	47.27
Bed	15.28	113.77	0	0	0	-90	1	0	0	82.45	102.28	50.75
Nightstand_01	16.13	89.59	12.5	0	0	-90	1	0	0	23.33	24.63	29
Dresser	253.84	127.62	0	0	0	90	-1	0	0	31.25	19	39
Nightstand_02	19.14	197.29	0	0	0	-90	1	0	0	22	23.87	29
HDD MultiMedia	253.59	182.29	20.05	0	0	90	-1	0	0	13.31	16.39	3.41
Light_01	24.54	96.29	27	0	0	-90	1	0	0	11.41	11.41	16.48
TDT	244.94	198.14	19.93	0	0	90	-1	0	0	11.61	9.29	1.61
Floor	11.06	345.12	0	0	0	180	0	-1	0	251.12	344	0
Walls	1.31	345.12	142.32	0	0	180	0	-1	0	260.87	344	142.32
Door	1.31	298.19	0	0	0	-90	1	0	0	41.25	11.32	75.25

Minimum distance between any 2 objects

In order to compute the minimum distance between any two objects, Unity [23] was used with a C# script, run for each scene. The algorithm applied was a simple approach, building on the following two main observations:

- each object was represented as a rectangle (i.e. through its bounding box), which is a convex shape
- all furniture types used in this initial model are positioned on the floor, simplifying the problem to a 2D plan minimum distance computation (i.e. minimum distance measured on the floor)

Therefore, the algorithm computes, for each of the two objects, the minimum distance between any vertex with any edge of the opposite object (more precisely, its bounding box). The geometric formula applied is the well-known minimum distance between a point and a segment defined by two points, calculated as:

$$distance\big(P1, P2, (x_0, y_0)\big) = \frac{|(y_2 - y_1) * x_0 - (x_2 - x_1) * y_0 + x_2 * y_1 - y_2 * x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$$

4.3.4 Further data integration for probabilistic models

Applying both models (Bayesian Network and GMMs) required further processing of data, on top of the ones being computed out of the graphical 3D models. The considerations were the inability to accurately predict every measurement, coordinate, angle or dimension that is going to be useful when applying the models, due to the lack of sufficient information in state-of-the-art literature; the ability to augment the model by engaging further-processed data and sometimes the ease of computing the additional information, instead of trying to extract it from the 3D arrangements.

Moreover, the existence of corner-cases and particularities of the objects and layouts, the mistakes made by inexperienced graphical designers, uncaptured in the early stages of pre-processing, or purely misalignment among the rooms in the example set (e.g. walls of different thickness) demanded auxiliary stages of data preparation.

The first (furniture occurrence) model required manual extraction of the number of targeted objects within each room from their corresponding file and correlation with the graphical view, in order to make sure that the objects having similar/identical names also had the same functional properties. For example, there are three rooms containing one chair each, but one of the chairs is meant for relax and leisure activities, while the others are strongly linked to a desk and are meant for working activities. This extraction contributed forwards to highlight the sets of objects that tend to be linked together and the number of said examples in order to obtain a rough estimate of their relevance, if it were decided to apply

Gaussian mixtures for those sets.

As for the second model (furniture arrangement), it implied heavy processing, starting from furniture alignment with respect to a certain point, and also including rotation and translation, to manually evaluate and adjust, when necessary, corner cases that were present in the examples (e.g. a nightstand was rotated with a 90 degrees angle to correspond the room's needs, while the other one was straight, facing the same direction as the bed).

The data transformations that GMM [12] demands target the overlaying of all the examples such that the objects of the same chosen category would (naturally) group together. Investigating this matter at a closer range, adapting the objects' coordinates from inside the rooms to fit a global positioning is not a trivial undertaking. Although exact description of the modeling part can be found in the corresponding chapter, in order to better clarify the exploration part, we will define specific cases.

In order to be able to apply GMM, the same set of objects needs to obey the same coordinate system. Considering the coordinates for the bed items, for instance, with all the different sizes, translations and rotations, would make it impossible for the model to yield a proper estimation, since data would be strongly inconsistent. Having the furniture's coordinates relative to the origin of the coordinate system, which is situated ideally in one of the corners of the room, as derived from the initial processing, the main pieces of furniture can be found in all four possible rotations (0, 90, -90 or 180/-180 degrees), needing particular treatment for each of the cases.

Bed and nightstands processing

For furniture arrangement, GMM operates on sets of items from two categories: the item(s) that tend to appear within the context of existence of other item(s). For instance, the *bed* would usually have around it one or two *nightstands*; in this case, the sets consisting of the bed and nightstand(s) for every room will be gathered to model their concurrent appearance inside a bedroom. The first major operation required would be to translate the objects from their world coordinate system (of each room) to the current (working) coordinate system. We found at least one nightstand present in 11 out of 12 possible rooms, imprinting a strong interest of modelling this type of causal link of furnishings with GMMs.

In the case of the bed and nightstand(s), it is crucial that the order is kept: if we look at the 2D blueprint with all the items in the room, the nightstand on the right-hand side of the bed has to remain always on its right after moving it to the new system, as well as the left-hand side nightstand. In order to meaningfully overlay all the desired objects in the set, we decided to align the left-top corner of the bed (the origin of object's coordinate system) to the origin of the new system, computing, therefore, the corresponding coordinates for the nightstand(s) according to the new system. The main cases to be analyzed occur when the bed and (usually) implicitly the nightstands are rotated with a certain angle, with respect to their upward (normal) vector. The dataset considers the forward direction of each object as being rotated with 0° when this vector points in the positive direction of the *0y* axis of the room's coordinate system. Given that the rotation of the needed objects present in the examples always leads to making their forward vectors parallel with the room's *0x* and *0y* axes, we can distinguish among the following four cases:

- **0° -** In this case, we only have to translate the nightstands to match our system in which the bed has its origin in the (0, 0) origin of co-ordinates.

$$x_{nstd\_right} = x\_old_{bed} - x\_old_{nstd\_right} \qquad x_{nstd\_left} = x\_old_{bed} - x\_old_{nstd\_left}$$

$$y_{nstd\_right} = y\_old_{bed} - y\_old_{nstd\_right} \qquad y_{nstd\_left} = y\_old_{bed} - y\_old_{nstd\_left}$$

- **90°** - The bed has its length along the *0x* axis, which we need to process accordingly

$$x_{nstd\_right} = y\_old_{bed} - y\_old_{nstd\_right} \qquad x_{nstd\_left} = y\_old_{bed} - y\_old_{nstd\_left}$$

$$y_{nstd\_right} = x\_old_{nstd\_right} - x\_old_{bed} \qquad y_{nstd\_left} = x\_old_{nstd\_left} - x\_old_{bed}$$

- **-90°** - Not only they are rotated, but we need to pay special attention to the order of the elements, so that we correctly align them with the others

$$x_{nstd\_right} = y\_old_{nstd\_right} - y\_old_{bed} \qquad x_{nstd\_left} = y\_old_{nstd\_left} - y\_old_{bed}$$

$$y_{nstd\_right} = x\_old_{bed} - x\_old_{nstd\_right} \qquad y_{nstd\_left} = x\_old_{bed} - x\_old_{nstd\_left}$$

- **-180°/180°** - Both values are present in the dataset, since they were extracted automatically; the forward vectors of the objects are pointing in the negative direction of *0y* axis.

$$x_{nstd\_right} = x\_old_{nstd\_right} - x\_old_{bed} \qquad x_{nstd\_left} = x\_old_{nstd\_left} - x\_old_{bed}$$

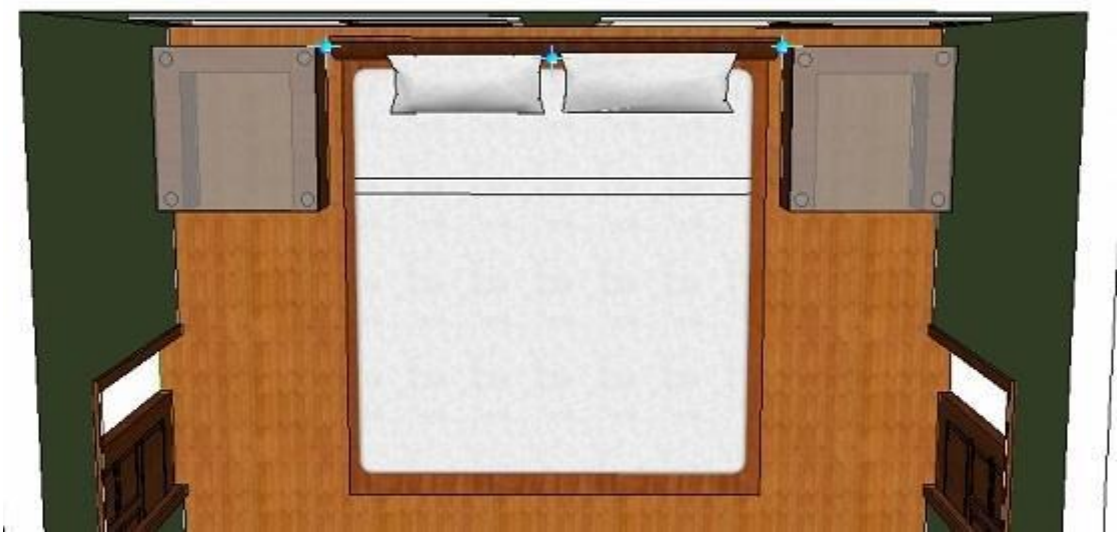
$$y_{nstd\_right} = y\_old_{nstd\_right} - y\_old_{bed} \qquad y_{nstd\_left} = y\_old_{nstd\_left} - y\_old_{bed}$$

The evolution of the model imposed some variations of these formulas, at first considering the center of the top segment of the bed being the origin of co-ordinates. The nightstands' positions will then move with half of the bed's width to the right (positive *0x*). In order to be consistent with the points chosen to represent each object, we then marked the center of the top segments of the nightstands, instead of the left-top corner, to match the bed's, adding, therefore, to each nightstand's representation half of its width along the *0x* axis.

Despite the rooms being scaled to comparable sizes, before extracting the measurements, the various sizes of the beds available in the dataset had an impact on the GMM process, leading to lower accuracy. Additional improvements consist of considering a universal size of bed (the highest found within data), given that the relation that is emphasized through Gaussian mixtures refers to the distances and positioning between nightstands and bed, desirably being independent from the size of the furniture pieces, which, in these cases, may vary widely even for the same style.

A solution found to be effective in this particular situation, was to keep the center of the top segment of the bed, moving the right and left nightstand according to the new, global, width of the bed. The operation involved subtracting the original width of the bed, adding the new width and center the objects in the current system. Furthermore, in order to preserve in a more accurate manner the distances between each nightstand and the bed, for the left nightstand to consider its top-right corner and for the right one, the top-left corner, both having the quality of being the closest to the bed, but preserving their alignment against the wall, as seen in Figure 18.

Figure 18. Outlining the representative Cartesian points for bed and nightstands for GMM.



Bed and TV

The occurrence table from Bayesian model is enlarged to contain the TV counting as well, since the bed-TV co-occurrence is typically present in modern bedrooms. This link is present in 6 rooms, half of the rooms existing in the initial set, leading to investing efforts towards adapting a mixture of Gaussian to express likely positioning for this type of relation's objects.

In all of the examples, no matter the rotation of the bed inside the scene, the orientation of the forward vector of the TV points towards the opposite direction (180°). Using common sense from a functionality perspective, we can state that this arrangement should be obvious, since a person would establish the TV's position so that he/she would be able to watch TV programs from the comfort of their own bed. We point out that the TV's position is dependent on the bed's position and not the other way around, since the bed is a much larger item with additional constraints and the TV does not necessarily require a standing surface whereas it can be fixed to the wall, making it a very manageable object, no matter how small or crowded the room can be.

Based on the learnings of the previous case, we consider the center of the top segment of the bed to be homologized with the origin of the Cartesian system engaged in running the GMM, and the TV representative point to be as well the center of its top segment (or the one at the bottom of the image, as we look at the system). The rest of the distances and/or ratios will remain faithful to their initial disposition, being translated and rotated to fit our needs. The particular computations are disseminated in conformity with the bed's rotation about the normal of its surface, as explained below:

- **0° -** The coordinates of the center of the upper segment (in 2D) of the TV are computed such that they remain relative to the center of the upper segment of the bed, which now overlays the origin of the axes.

$$x_{TV} = \left(x_{old_{bed}} - \frac{width_{bed}}{2}\right) - \left(x_{old_{TV}} + \frac{width_{TV}}{2}\right)$$

$$y_{TV} = y_{old_{bed}} - y_{old_{TV}}$$

- **90° -** The initial axes interchange

$$x_{TV} = \left(y_{old_{bed}} - \frac{width_{bed}}{2}\right) - \left(y_{old_{TV}} + \frac{width_{TV}}{2}\right)$$

$$y_{TV} = x_{old_{TV}} - x_{old_{bed}}$$

- **-90°**

$$x_{TV} = \left(y_{old_{TV}} + \frac{width_{TV}}{2}\right) - \left(y_{old_{bed}} - \frac{width_{bed}}{2}\right)$$

$$y_{TV} = x_{old_{bed}} - x_{old_{TV}}$$

- **-180°/180°**

$$x_{TV} = \left(x_{old_{TV}} + \frac{width_{TV}}{2}\right) - \left(x_{old_{bed}} - \frac{width_{bed}}{2}\right)$$

$$y_{TV} = y_{old_{TV}} - y_{old_{bed}}$$

An attempt to consider a universal size of bed, precisely its length, would not perform better in this case, since the placing of the TV at a certain distance from the bed involves harder constraints than the bed’s dimensions, such as the distance to the wall that in front of the bed, the nearest supporting surface or possibility of wall attachment etc. The angle of the two forward vectors of the fittings is highly important in this case and is computed as an absolute value of the difference between the two rotation angles:

$$\theta_{bed,TV} = abs(rot_{bed} - rot_{TV})$$

Desk and chair

Many bedrooms have often been designed with a desk and a chair meant to serve working purposes, thus being a requisite especially met for, but not limited to, people who lack other areas in the house that they can use for that purpose. From a functional perspective, as detailed in previous work in the field [24, 25], a chair obeys the setting of the desk in an almost strict relation, making the collocation of the two items highly feasible using GMM.

However, the size of our examples set comes as a shortcoming yet again, since we can find only two examples containing both chair and desk objects. In addition, the examples cannot even be considered to address the same situation, since one of the desks is pushed against the wall, having a rectangular shape, while the other is placed in one of the corners of the room, having an ‘L’ shape, with the chair pointing towards that corner, as presented in Figure 19.

Figure 19. The two examples of rooms having chair and desk.

If this set of items is needed to be integrated within a bedroom, our suggested approach is to use one of the examples (depending on the layout so far) to generate other similar positioning of the chair in the context of having the desk already in place. The extraction of the objects' relative coordinates and translation to a new Cartesian system is kindred to the previously detailed relation between bed and TV, because both sets have the same relative orientation among the items they contain. In the particular case of the corner desk, instead of considering the medium of one segment as the salient point, we will consider its corner, leaving the chair's representation to be the middle of the top segment, as before.

Corner cases and shortages

Unfortunately, the variety of the room layouts space and the dataset being made up from 3D models of common and many times unexperienced users often leads to come across situations of unique cases which stray from the regular processing and require special attention in order to depict and treat them individually. Such cases may be processed thereupon, enhanced or even removed from subsequent considerations, if they represent an isolate, hence marginally important event; they fail to fully represent a particular class or solution; or there is not enough information to adequately transform them for future use.

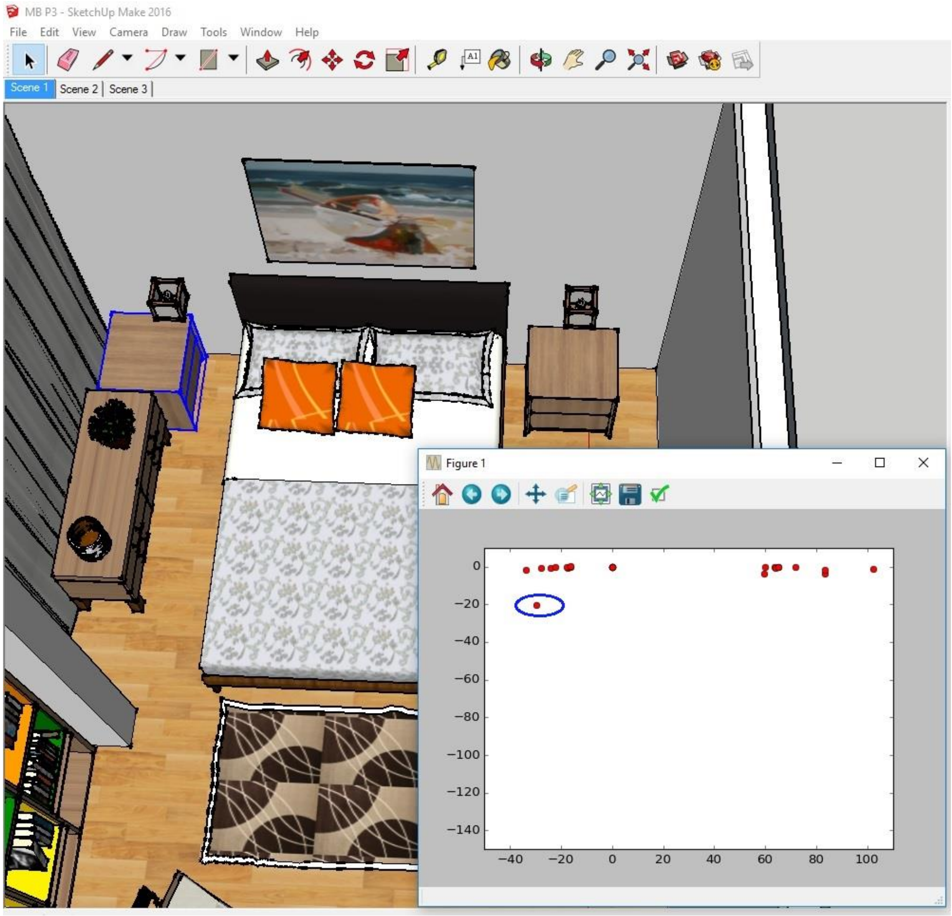
These cases will be mentioned separately in the consequent paragraphs and/or in the chapter dedicated to GMMs, depending on where their impact will be more significant.

While processing furniture pieces for the GMM evolution, it is critical to make sure that every item and piece of data matches the pattern and does not introduce false variations into the system. Considering that, in most cases, the nightstand(s) are positioned sideways from the bed, keeping the same orientation of the forward vector as the bed, the event of one of the nightstands being rotated with a 90° angle from the bed's orientation is considered isolated and does not increase learning possibilities towards our goal. Moreover, it affects accuracy due to the different type of coordinates, induced by the fact that it obeys different convenience rules and also by the particular layout of the room.

Figure 20 depicts the bedroom which contains the differently-arranged left-hand side nightstand, which is highlighted in blue, as a 3D graphic object. The window overlaid on top of the image represents the coordinate system comprising all the representative points for every bed and nightstand found in the set of examples, after the suitable calculations considering the first approach, as described above. Each of the beds is centered in (0, 0), having the nightstands properly placed on its sideways. It is evident to the human eye that the circled point, symbolizing the left nightstand in the picture, represents an outlier instance, making it difficult for the algorithm to properly estimate the variance of the data.

The attitude taken with respect to similar occurrences may differ, depending on the size of the dataset, the frequency of such examples within the dataset and the target aimed by the scientists involved in the project, leading to removing the sample(s), modify it to fit the other pieces of data or analyze the case individually. In our case, given that we aimed for obtaining a proof-of-concept for rather un-complex or particular arrangements and that this particular state of the object appear only once in the entire set of examples, we chose to discard it, since it would not bring significant improvements in accordance with our goals.

Figure 20 Corner case: left nightstand is rotated with 90° angle as against the bed.



5 Probabilistic models for furniture synthesis

By putting into practice some of the best approaches for furnishing interior spaces in the state-of-the-art in the field (while bearing in mind that this field is still in its very early stages of development), we aimed in this thesis to achieve at least a proof of concept that laid the grounds for the future development of an operational software tool for Interiorvista company.

It is meant to offer, both to the company and to ourselves, some insight regarding: the world-wide accomplishments in the field attained so far; the difficulties associated to the gathering, understanding and processing of datasets of use to accomplished our goals; the feasibility of implementing analytical methods that are reasonably successful on real-world examples; and the potential directions for further mining research perspectives for of these data.

After thorough investigation, I decided to implement two probabilistic methods that solve crucial parts in the endeavor that, in the future, will help us to put together a complete tool for automatic production of furnishing layouts within a given perimeter. The first method, described in section 5.1, implements a **furniture occurrence generator** which learns from the examples in the database, but manages to output novel, unseen sets of items for a bedroom, based on Bayesian Networks. The second part of the development, available in section 5.2, handles **fittings arrangements** for items sets in which a causal relation exists, via GMMs, striving for human-approved placements of objects within the set that are easily translatable to the new space to be furnished.

5.1 A Bayesian Network for furniture occurrence

An initial step towards obtaining new furnishing arrangements consists on generating sets of furniture pieces to be present in the room. This stage focuses only on selecting what types of furniture would be suitable for a certain room type in our initial model (that being the bedroom). This task is similar to selecting compatible components for a model in order to generate new shapes from existing ones, while preserving the “realistic factor” and plausibility of the model. In [13, 11], probabilistic models were used for synthesizing new objects from components, and were applied for various types of objects such as planes, boats, or chairs. These methods fully learn compatibility aspects and placement possibilities from training the available data and are used to generate new objects by sampling the obtained model.

Furniture pieces’ selection for automatic room decoration can be viewed as a similar problem, in which the components of the shapes represent the furniture pieces and the new shape, based on these parts, is the set of furniture pieces in that room. However, important differences have to be taken in account, such as the loose coupling of objects in a room, the fact that various valid arrangements for the same set of furniture pieces maybe possible, or the high diversity of each type of furniture.

Our initial model for generating the set of furniture that would be suited to a given room (a bedroom, in our case) consists of a Bayesian model in which each node represents the number of one certain type of furniture such as: table, bed, or nightstand.

5.1.1 Training data

For the initial model, we consider a representative subset of possible furniture types, containing pieces that are common in various bedrooms and which would, in certain subsets combinations, favor various types of activities in the designed bedroom. This ensures the predisposition of the model to learn, besides naïve independent furniture frequencies, the underlying structure and functional dependencies/grouping of different types of objects. For instance, the presence of a desk-and-chairs group would add features such as: *study*, *writing*, or *social activities*; a sofa and a TV would suggest leisure activities such as *watching movies* or *playing video games*, whereas the presence of a table without chairs should, in principle be uncommon.

Therefore, the types of furniture pieces considered in the initial model are: bed, chair, table or desk, nightstand, wardrobe, and chest of drawers. The nodes in the Bayesian network represent the counts of these objects. For the purpose of our model, and in order to be able to learn from the initial, small, dataset, we also limit the range of values each node can take to the discrete set: {0, 1, 2, 3+}, where 3+ represents 3 or more pieces of furniture. The selected furniture pieces, together with the pieces counts in the bedroom dataset are presented in Table 2.

No ultimate choice exists in modeling this challenge in a probabilistic, data oriented manner, this being still an open-ended research area, with a focus towards virtual environments representation. Veering away from this field, our model addresses a real world, commercial scenario, aiming towards scalability – both in terms of furniture pieces and types of bedrooms, robustness and real-time performance. Because no *a priori* knowledge can be imposed about the furniture dependencies in a bedroom, existing numerous styles, mentalities, preferences among customers, one of the main objectives is the model ability to adapt to change while capturing hidden patterns, underlying trends and structure diversity from the data. Therefore, both learning the Bayesian model structure and parameters is compulsory and we apply well-known techniques to deal with this, which are described in the following section.

5.1.2 Analyzing furniture dependencies

In order to build the Bayesian Network structure by applying various learning techniques and reasonability judgement, we conducted a pre-analysis to gather evidence by discussing with people with prior experience in furnishing their bedroom and experienced, professional interior designers from Interiorvista company. The hierarchical dependencies concluded have a variety of causes and are highly linked to the context– given by available space in the room, general shape of the room plan, available money for furniture purchase and personal factors such as lifestyle, age, marital status, daily routines, background, gender, etc. Besides this wide context, the dimensionality of the problem is also complex, dependencies being highly influenced, besides by the type of furniture pieces, by their style, individual functionality and features (e.g. if a wardrobe has the particular possibility to place a TV, it can replace a desk/table and a sofa can, in this case, be dependent directly with the wardrobe).

Table 2. Number of objects of each furniture piece type in the dataset.

Room ID	# bed	# chairs	# table or desk	# nightstand	# wardrobe	# chest of drawers
	1	0	1	1	1	0
	1	1	1	1	1	0
	1	0	1	2	1	0
	1	0	1	2	1	0
	1	0	1	1	1	0
	1	0	0	2	0	0
	1	0	0	2	0	1
	1	0	0	2	1	0
	1	0	0	2	1	1
	1	1	1	2	1	1
	1	0	0	0	3	1
	1	0	0	2	0	1

This results, for some cases, in strange groupings and arrangements, due to the high variations in functionality and purpose of some types of furniture. Reducing the analysis to our initial scenario of the problem (i.e. count of different types of furniture pieces, initially without considering style and with a general usage) we depict dependencies such as:

- **bed – nightstand**: the bed has a direct influence on the presence and number of nightstands. For example, if no bed is present, nightstands might not make sense in a room, being replaced by a person by a chest of drawers, if personal space is needed, or a desk/table if a supporting surface is desired. Usually, and depending on the general arrangement of furniture and space available, a bed might have one or two nightstands

- **wardrobe – chest of drawers – nightstand**: these three pieces of furniture, although very different from a human perspective, share a significant amount of functionality in terms of storage possibilities, accessibility and supporting surface. Depending on the particular style, size and positioning in the room, the chest of drawers and the nightstand(s) can be both used for storage and "object placement" and their presence and positioning is highly influenced by personal style. A chest of drawers might offer a wider range of storage functionality, expanding in our dataset from day working space for books, documents, to clothing depositing option and also room decorations such as artifacts, pictures, personal collections. The presence of the wardrobe, usually occupying much space in the room, can influence the presence of nightstands and chest of drawers from a space point of view, storage capacity of the room (if the wardrobe is large enough, it might compensate the need of nightstands and chest of drawers).
- **table/desk – chair:** the number of chairs is directly influenced by the presence of a table in a bedroom, leading to the case of having chairs without a table to be rare. This still can happen, and is present in our manual build dataset; reasons for this might include the lack of space, supporting social interactions and "seated activities". Also, other indirect influences can be seen between the desk size and the functionalities it offers and the number of chairs (this aspect not being explored further in this initial model).

5.1.3 Framework used

In our implementation, we used the Bayes Net Toolbox for Matlab (BNT) framework [26] for designing the Bayesian model and apply the learning techniques. Some of the most important features it offers, together with a brief overview of what they include, depicted from the BNT references [26], are:

- Numerous types of conditional probability distributions are supported, such as:

multinomial, Gaussian, logistic/sigmoid, deterministic, etc, offering great flexibility in designing the Bayesian Network.

- Various types of nodes, such as: decision, utility, chance.
- Various methods for model regularization are included, offering features such as: parameter clamming, parameter sharing among nodes, etc.
- Numerous algorithms for exact and approximate inference are implemented in the toolbox, for both static and dynamic Bayesian Networks.
- Learning methods for parameters, such as: maximum likelihood estimation for a fully observed network; maximum likelihood / maximum a posteriori probability estimation (MAP), using the batch expected maximization for partial observations, for both static and dynamic models.
- Learning methods for structure, including: the greedy search K2 algorithm – requiring a fixed node ordering, Hill-Climbing, and the Markov Chain Monte Carlo (MCMC) algorithm.

This toolbox offered the ability to rapidly implement and validate the proof of concept against the data, while maintaining the possibility to research further - increasing model complexity and problem/data dimensionality.

5.1.4 Bayesian Network Structure Learning

It is known that structure learning is NP-hard and exact algorithms have super-exponential complexity – e.g. the number of Directed Acyclic Graphs (DAGs) to be searched in. Because no feasible possibility exists to exhaustively iterate through all possibilities and score each, we use two algorithms for learning the structure: the K2 greedy search algorithm and MCMC – a global search algorithm.

K2 algorithm

The K2 algorithm is an efficient, well-suited approach, if a total ordering of the nodes is known *a priori*. The initial problem of finding the best DAG then resumes to finding for each node the corresponding set of parents. In our case, the dependencies of the furniture pieces cannot be known and can depend on various factors such as bedroom main (usual) activities, human preferences, or furniture functionality. Moreover, no ultimate choice for a model to successfully outline and correlate furniture interdependencies with personal style, life habits, and routine activities exists in the related literature.

We manually designed some common sense, plausible, total ordering of furniture sets, presented in Table 3, over which we apply K2 algorithm, in order to form an initial idea about the types of dependencies that form. These will provide various insights of the dataset such as: potential functional groups that form - supporting specific activities in the room, variations of types of bedrooms that comprise the dataset, etc. The orderings that we fed the K2 algorithm are presented in Table 3. The first 3 arrays represent variations that attempt to encapsulate the possible dependencies that were concluded in 5.1.2, after analyzing personal opinions of people furnishing their bedroom and discussing with professional interior designers from Interiorvista company.

Table 3. Orderings set for K2 algorithm.

ID	Order	Observations
1.	n_bed, n_wardrobe, n_table_desk, n_chair, n_nightstand, n_chest_drwrs	-
2.	n_wardrobe, n_chest_drwrs, n_bed, n_nightstand, n_table_desk, n_chair	-
3.	n_chest_drwrs, n_wardrobe, n_nightstand, n_bed, n_table_desk, n_chair	-
4.	n_chair, n_nightstand, n_chest_drwrs, n_table_desk, n_wardrobe, n_bed	reverse of 1
5.	n_chair, n_table_desk, n_nightstand, n_bed, n_chest_drwrs, n_wardrobe	reverse of 2
6.	n_chair, n_table_desk, n_bed, n_nightstand, n_wardrobe, n_chest_drwrs	reverse of 3

The parent-child relationships resulting through a data-driven learning, on top of these established relationships, will represent a better insight into the complex relations between the furniture entities from both perspectives: an analytical one – provided by experts in interior design field and a data one – obtained from the greedy K2 algorithm run on the dataset, and using as the scoring function the maximum likelihood of the model (e.g. Bayesian Score in BNT [26]). The last three ordering arrays are the symmetric of the first 3. We include these in order to permit the algorithm to build parent-child dependencies in both directions, given that we initially created this ordering array set from bidirectional dependencies. Moreover, starting from the established dependencies described in section 5.1.2, the learned DAGs should have the chance to incorporate and extend these relations in both directions.

The results of running the K2 algorithm are presented in Figure 21 and Figure 22. Although significant variations have been introduced in the first three cases of total ordering, the same structure was learned from the data. This shows a strong correlation of the established parent-child dependencies, the results being invariant to changes in hierarchy if these modifications preserve the possibility of building the resulted DAG. Intuitively, the set of three symmetric orderings also provided only one DAG.

Another aspect worth mentioning is the simplistic structure obtained, with at least 50% of the nodes being independent, this being another possible reason of the stability of the results. The lack of more complex dependencies in the DAGs can also be explained by the algorithm itself, trying to match each child with its list of parents and by the simplifications in the hypothesis, reducing the furniture pieces set to main types, without style and with general functionality.

Figure 21. Bayesian Network DAG obtained by K2 model on first half of the total orders set

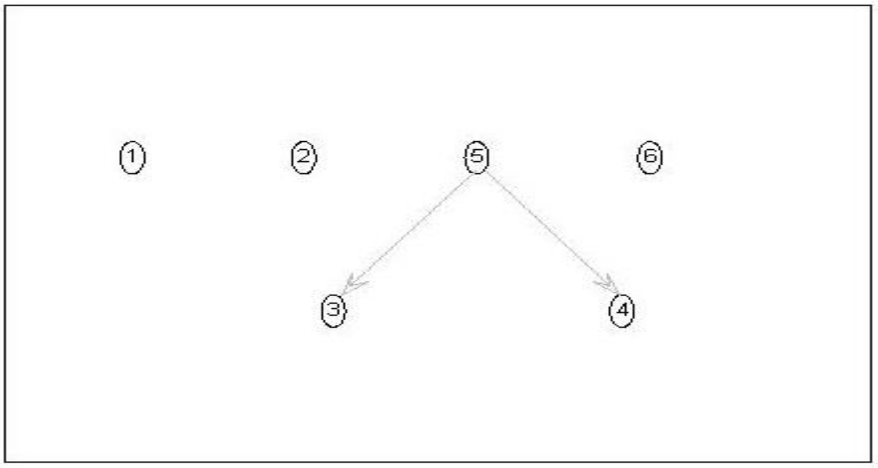
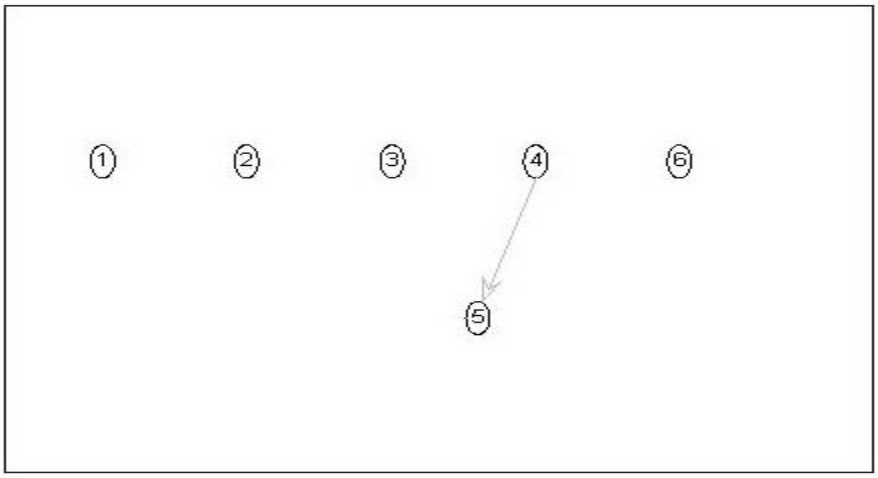


Figure 22. Bayesian Network DAG obtained by K2 model on second half of the total orders set.



In more detail, the relations in Figure 21 present a dependency between the number of wardrobes and the number of nightstands and a dependency of the number of wardrobes with the number of tables and desks. While the first dependency is intuitive and was also depicted in our furniture dependency study within Interiorvista Company, the second is a rather interesting, new one.

Although no direct, intuitive relation can be depicted between the wardrobe presence and the table/desk entities in a bedroom, indirect ones can be further observed in the dataset: usually the presence of a wardrobe implies the presence of usually one or more desks/tables. One reason for this is that a bedroom with a wardrobe shows that a person is long-term leaving in it (has her or his personal objects, clothing, etc.) as opposed to, for instance, a guest room, that can lack a wardrobe (i.e. just have a smaller chest of drawers) and consequently lack a table, because of short term usage. This is common in bedrooms for rent and was observed by us, for instance on the Airbnb company website [27], where many real private hosts provided rooms with neither a wardrobe nor desks/tables, usually because the lack of space and the strict purposes of the room for short-stay and limited activity (i.e. only sleep).

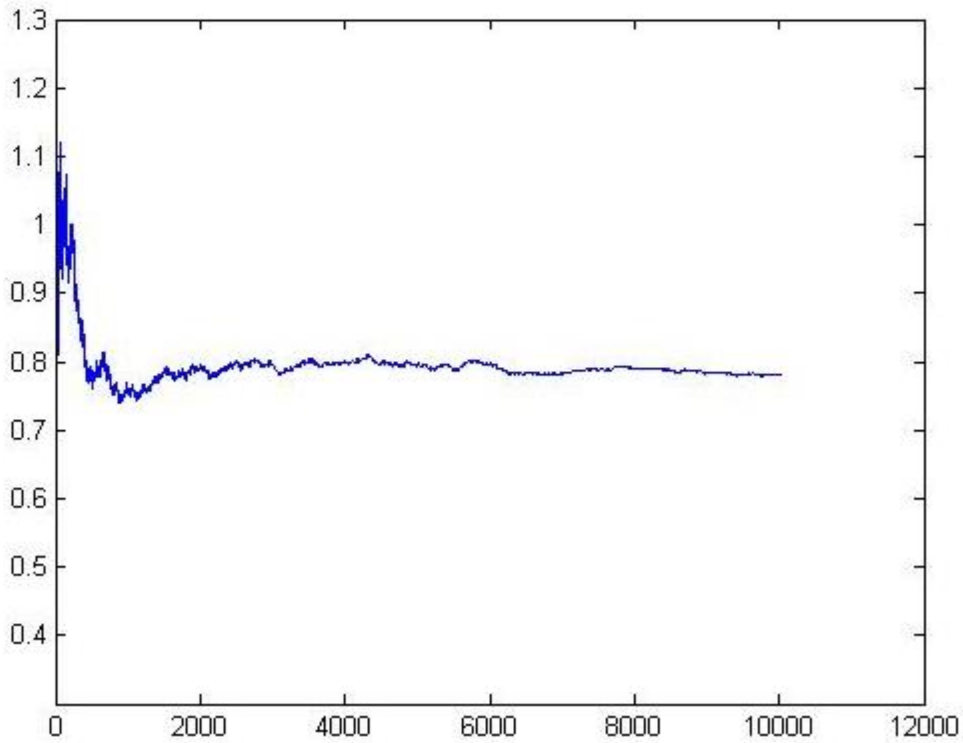
The DAG in Figure 22 only contains one parent-child link, between the number of nightstands and the number of wardrobes. This DAG is also loosely coupled, but compared to the previous one, the link learned does not represent a significant relation. This is probably caused by the counter-intuitive dependencies provided in the total orderings in the last half of Table 3 and proves that the analytical relationships depicted in our case study are as intuited, in the order of the first half.

MCMC algorithm

We use MCMC for learning the possible DAGs structures for the Bayesian Network. Compared to the K2 algorithm, the model does not rely on any *a priori* ordering of the nodes, exploring the space of all possible DAGs. The scoring used in the model is, similar to the previous case, model likelihood maximization (Bayesian Score). The BNT [26] method takes two model-related parameters, namely the number of samples to be taken and the number of steps to take before drawing samples.

Given the size (in nodes) of our Bayesian network, we run the algorithm initially for a high number of iterations, in our case 10,000 iterations. In order to understand the model convergence and the accuracy of the obtained results, we used a crude measurement, called the “accept ratio”, which is computed, at each iteration, as the ratio between the number of accepted samples and the number of rejected ones, as shown in Figure 23. From this figure, it is clear that the value of the acceptance ratio stabilizes at around 0.75, indicating model convergence and stability.

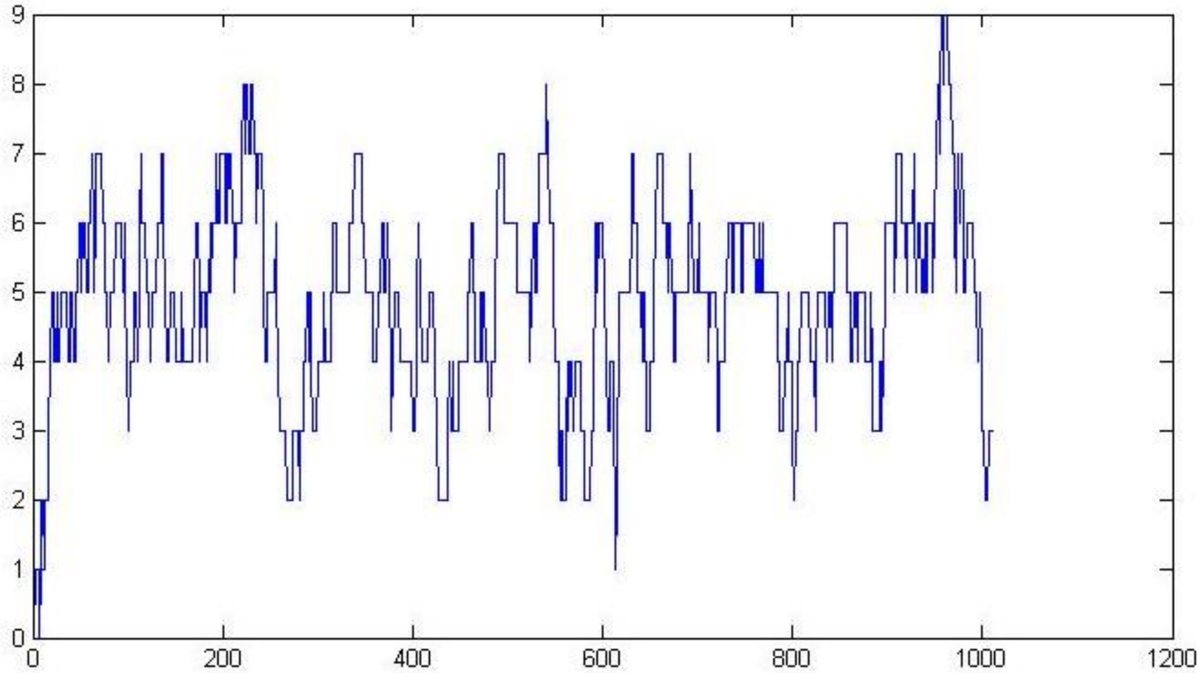
Figure 23. Acceptance ratio for the MCMC algorithm.



Because the Bayesian score and not the BIC (which would also penalize the DAGs complexity) was used, we also plot the number of edges for the last 1000 iterations (i.e. DAGs) in Figure 24 as a control of the model complexity and thus overfitting, preferring simpler models than complex ones. It is shown that the number of edges constantly varies, usually between 3 and 7 edges per DAG, thus an ultimate choice for the Bayesian structure cannot be concluded.

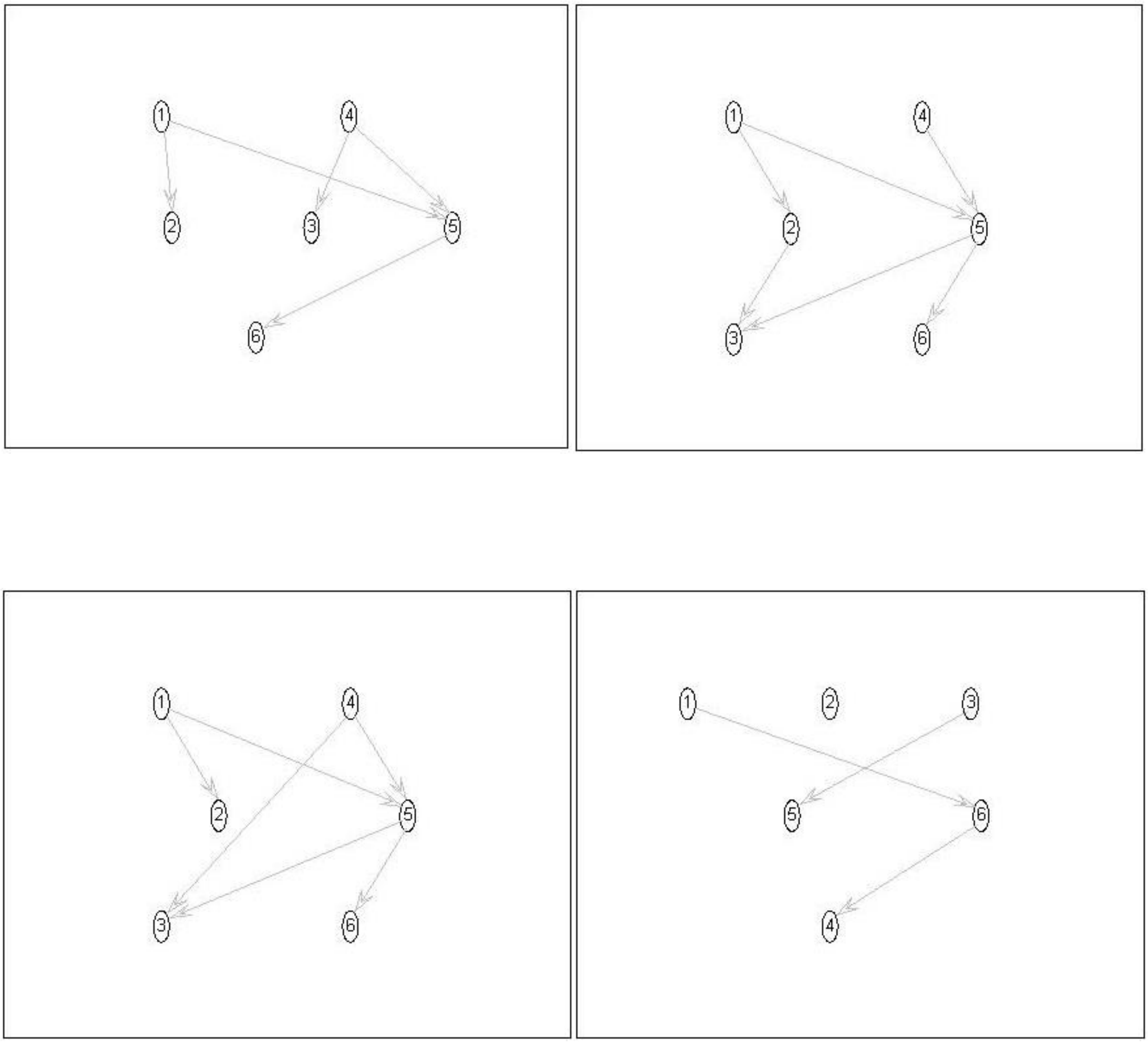
The model fluctuation might also be caused by the fact that the dataset is relatively small to reproduce the problem complexity and learn best structure by applying a global search algorithm. Although the model is intended to only capture relations between the occurrences of different types of furniture, more complex underlying patterns influence the data, which inevitably try to be captured by the Bayesian model. Still, insightful parent-child relations learnt through the MCMC approach can be observed in Figure 25, which presents the resulting structure variation through six randomly chosen samples, with very different DAG structures (i.e. dependencies, number of edges).

Figure 24. Number of edges in DAGs for the MCMC algorithm.



This analysis constitutes proof that learning exact, unique structure is hard and many valid configurations that would validate the dataset can be obtained. The challenge becomes even harder when dealing with an open-ended research problem such as interior furniture synthesis for a given room plan. Because no ultimate choices for a probabilistic model exists in the recent literature and the results obtained with learning the structure can be manually improved and did not provide a definite, best candidate, we will use the information gained and results from learning the structure and will design manually the Bayesian Network structure, with respect to the analysis of furniture dependencies done with Interiorvista interior designers and amateurs decorating their bedrooms.

Figure 25. Different DAGs produced by the MCMC algorithm.

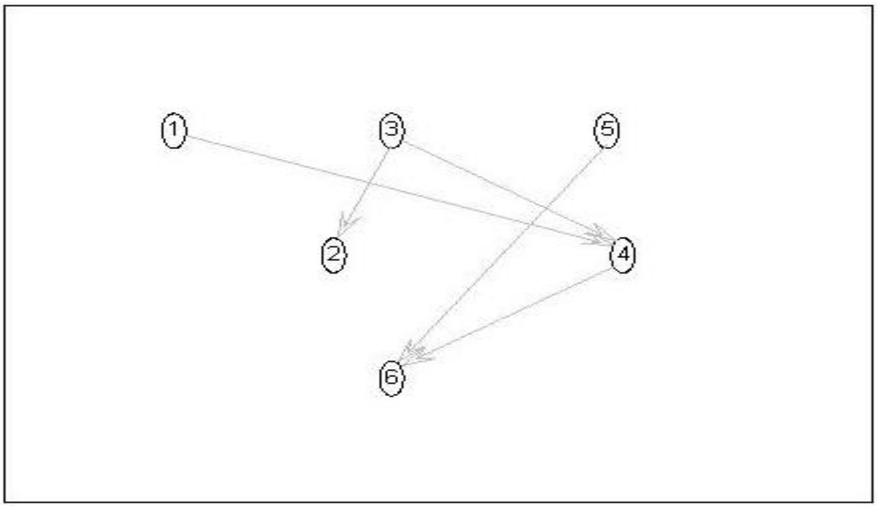


Manually designed DAG

The manually designed Bayesian structure, presented graphically in Figure 26, will contribute to the overall proof of concept with an analytical, human analysis of furniture pieces' dependencies, reinforced with strong correlations depicted from data, through structure learning. A broad palette of reasons for these correlation have been taken into account, expanding from general furniture functional groupings (e.g. table with chairs, bed with nightstand) to more abstract, general ones such as bedroom purpose (e.g. long stay *vs*. short stay), size and shape variations (i.e. allowing certain pieces of furniture to be (or not) of usable support surface (i.e. total area of surfaces that can be used to put daily common objects such as table, nightstand, and more rare chest of drawers).

Although the considered complexity represents a valid starting point for POC, integrating significant complexity in order to achieve insightful analysis, important aspects are currently left out, but will need to be included in further work, towards achieving robustness, scalable, production-ready results – as needed for this real-world, customer-oriented company project. These aspects include: style compatibility, personality analysis (e.g. life style, age, marital status, gender, background, preferences), furniture set enlargement, detailed functionality adding, compatibility specs as of the furniture providers.

Figure 26. Manual designed Bayesian Network DAG.



In more detail, such direct parent-child relations are:

- **wardrobes – chest of drawers; nightstand – chest of drawers; table – chair; bed – nightstand**: these dependencies were added as a result of the study analysis, detailed in 5.1.2 and partially correlating to obtained learned structures

- **wardrobe – table/desk**: as depicted from the learned structure using the K2 model, this indirect caused relation is a strong candidate to be included in the designed structure. However, this link will be removed from this initial model because of the bad integration with the model: resulting in strange, non-representative samples of furniture sets.
- **table / desk – nightstand**: this correlation, outlined in some cases of the MCMC learning, related the number of tables / desks to the number of nightstands. Having main different functionalities, namely social/work activities enabling and small deposit space, horizontal near bed surface respectively the influence is probably related to the space challenge and usage of the room in daily activities.

  Analyzing further together with Interiorvista experts and amateur people who furnished their bedrooms, we presumed that a person, with limited space room, would include a table/desk in favor of a nightstand if day light and social activities would be done in the bedroom and opposite (i.e. a nightstand instead) if the room main purpose would be for sleep and activities close to sleep time (e.g. morning of evening routines) – the nightstands usually being used adequately: support surface for smartphones, books, watch/alarm, lamp that favor before sleeping activities and waking ones as: waking up with alarms, reading a book, watching a movie before sleep, hygienic routines, etc.

5.1.5 Bayesian Network Parameter Learning

Using the built Bayesian structure in sub-section 5.1.4, we move forward to parameter estimation through learning. In our case, each node is a discrete one, having one of the following 4 values: {0, 1, 2, 3+}, representing the number of each furniture piece (i.e. for each node). Also, because the furniture generation problem is currently represented only through furniture counting nodes, we deal with a fully observable Bayesian Network. We use the maximum likelihood estimation algorithm for learning the parameters and we keep the same learning data as previous (i.e. counts of selected furniture pieces in the bedroom dataset). After learning the parameters, we use the Bayesian Network for sampling new possible sets of furniture that would go in an unfurnished bedroom.

In order to better understand and compare the power of the Bayesian Network and the power of the model to learn and reproduce furniture interdependencies, we will compare the generated results (i.e. through sampling) of the last Bayesian Network (i.e. the one with the had crafted DAG) with other three representative structures, resulting the following scenarios:

- No connections (i.e. no dependency between the nodes)
- K2 structure
- MCMC structure (chose randomly among the learned ones)
- Manually built

In Table 4, we present 10 samples from each of these cases. We start with the simplest, basic case with a Bayesian Network with no connections following with the appliance of the presented methods for learning the structure and finally the manual design one, integrating significant dependencies observed both analytically in the study case and data oriented, from the previous Bayesian Networks.

Table 4. Furniture samples for different BN structures and learned parameters.

	# bed	# chairs	# table or desk	# nightstand	# wardrobe	# chest of drawers	Obs.
Null DAG	1	0	1	2	1	0	- no linkage between number of furniture pieces - sampled according to occurrence frequency
	1	0	1	2	1	0	
	1	0	0	2	1	0	
	1	0	0	2	1	1	
	1	0	0	1	1	0	
	1	0	0	0	1	1	
	1	0	1	1	1	0	
	1	0	1	2	0	0	
	1	0	1	2	1	0	
	1	0	1	2	0	0	
K2	1	0	0	0	3	0	- local

DAG	1	0	0	0	3	1	constraints reinforcement, acc. to BN structure
	1	0	1	1	1	0	
	1	0	1	1	1	1	
	1	1	0	2	1	0	
	1	0	0	2	1	1	
	1	0	0	2	0	0	
	1	0	1	1	1	0	
	1	1	1	1	1	0	
	1	0	1	2	1	1	
MCMC DAG	1	1	1	1	1	0	- complex local dependencies enforced: - support surface balance: - deposit space ratio
	1	0	0	0	3	1	
	1	1	0	2	1	1	
	1	0	0	2	0	1	
	1	0	1	1	1	0	
	1	0	0	0	3	1	
	1	0	0	2	1	0	
	1	0	1	1	1	1	
	1	1	0	2	1	0	
	1	1	0	2	1	1	
Manual DAG	1	0	0	0	0	3	- incorporates all the above relations, plus: - general, global abstraction, resulting in different types of bedrooms, oriented to human life style
	1	1	1	1	1	0	
	1	0	0	2	1	1	
	1	0	0	2	1	0	
	1	0	0	2	0	1	
	1	0	0	2	1	0	
	1	0	1	2	1	0	
	1	0	1	1	1	0	
	1	0	1	2	1	1	
	1	0	0	2	3	3	

No connection DAG

The first and also the simplest case – the DAG with no connections - produces samples that do not relate the number of entities with the context, given by the other existent entities. The generated values for each type of furniture piece emulate the frequency of furniture pieces in our dataset. Therefore, we can easily depict many types of anomalies that are not representative of our data, such as: tables usually do not have chairs, the values usually do not vary because the context in which these values are generated is represented by the furniture piece itself, no balance or correlation of furniture groups' functionality can be depicted, etc. This case serves in our scenario as the baseline, to which we compare the performance of the other results.

K2 model DAG

Moving to the next case, namely the structure learned by the K2 approach, we start seeing some variations in the samples generated in this case, such as: number of chairs in rooms is not always 0, we can have more than 1 wardrobe in some cases, and some data correlations can be seen, given by the Bayesian Network structure learned by the model for this dataset. We recall that this algorithm requested an *a priori* known total order of the nodes, and therefore the results are highly correlated by the imposed order. However, a certain degree of model stability was demonstrated by the fact that variations in the total ordering of the nodes, which did not interfere with the learned, direct parent-child dependencies, did not introduce any kind of variations in the learned model. Moreover, by learning structures given all the symmetric ordering of the previous variations also resulted in a single, stable Bayesian Network structure.

In more detail, the learned structure dependencies, in this case between the wardrobe and the table, and nightstand, over which the parameter learning was applied is presented in Table 5, which outlines the Conditional Probability tables (CPDs) for the table/desk and nightstand nodes, respectively. Interesting hypothesis can be drawn directly from them, and indirectly from the sampled instances, such as:

- if no wardrobe is present, there will be no table/desk (possibly related to the lack of space in the room: a person would prefer a wardrobe over a table/desk for a bedroom and, therefore, if no wardrobe is possible, no table will be either)

- if one wardrobe: there is 75% chance for a table to be in the room and 25% chance for a table to not be in the room and 62% for 2 nightstands and 38% for one.
- for 3+ wardrobes, no tables, nor nightstands should be present in the room (because of lack of space, or the practical use of the bedroom as a personal storage site – e.g., for clothes)
- observation: there is 0 probability for 2 wardrobes in the "number of wardrobes" node, thus 0 sum probability in that case for each of the child nodes.

Table 5. Conditional probabilities in K2 Bayesian Network.

	# Wardrobe	0	1	2	3+
# Table / desk node	0	1	0	0	0
	1	0.25	0.75	0	0
	2	0	0	0	0
	3+	1	0	0	0
# Nightstand node	0	0	0	1	0
	1	0	0.38	0.62	0
	2	0	0	0	0
	3+	1	0	0	0

MCMC model DAG

A more complex DAG structure for the Bayesian Network is learned by the MCMC algorithm, which also outlines some possible linkage and dependencies of the furniture sampling. Being a model learned only from data, first the structure and then the parameters, we analyze the results, as before, from the perspective of how well the architecture was able to capture significant, underlying patterns in the furniture interdependency.

As indirectly depicted from the samples generated, the model has a better, deeper understanding of the data, taking into account more abstract patters for bedrooms such as total support surface available for depositing common object upon – through the dependency between the nightstand and the table/desk and total deposit space (e.g. for clothes and other personal objects), through dependencies between all the furniture pieces that have this functionality in our dataset, namely wardrobe, chest of drawers and nightstand.

We can view the results of these main relations both directly, as presented in conditional probability tables in

Table 6, Table 7, Table 8, and also indirectly in the data generated, shown in Table 4. The presence of the nightstands influences directly the presence and number of the wardrobes and indirectly the number of the chest of drawers (i.e. if no nightstand is present, a chest of drawers is usually sampled. Also, in the wardrobe – chest of drawers dependency we can see that the chance of a chest of drawers existence decreases if a wardrobe is present in the room: 66% for the existence of a chest of drawers if no wardrobe is in the room and only 25% if one wardrobe is in the room.

Because of the limited dataset available for analysis from Interiorvista, and the complete orientation of the model towards data learning, less plausible dependencies are also learned, containing the conditional probabilities for 3 significant nodes, namely wardrobe, table/desk and chest of drawers. Such case would be: the 100% chance for 3+ wardrobes if the room has 1 bed and no nightstands; the compulsory (e.g. 100% chance) of having one table if the room has one nightstand and the existence of exact one chest of drawers if the room has 3+ wardrobes.

In general, the observed atypical cases are caused by overfitting, due to the limited number of entities in our training set, and the complexity of the model developed. Another sign that these few cases are produced by the lack of similar examples (i.e. corner cases for our dataset) is the probability distribution: usually 100% for one case – showing that the case was learned from only one example – because of missing other similar, containing small variations examples. These should be straight forward avoided with a larger dataset, containing a greater variety of bedroom scenes.

For the purpose of this initial model and dataset presented in the project, in the light of an open ended research field, we focus more on what aspects can be learned and which model can be used, as a proof of concept, leaving the robustness and scalability of the system for future development.

Table 6. Conditional probabilities for # Table/Desk in MCMC Bayesian Network.

	# Nightstands	0	1	2	3+
# Table / desk node	0	1	0	0	0
	1	0	1	0	0
	2	0.62	0.38	0	0
	3+	0	0	0	0

Table 7. Conditional probabilities for # Chest of drawers in MCMC Bayesian Network

	# Wardrobe	0	1	2	3+
# Chest of drawers	0	0.34	0.66	0	0
	1	0.75	0.25	0	0
	2	0	0	0	0
	3+	0	1	0	0

Table 8. Conditional probabilities for # Wardrobe in MCMC Bayesian Network

	# Bed	#Nightstand	0	1	2	3+
# Wardrobe	1	0	0	0	0	1
	1	1	0	1	0	0
	1	2	0.38	0.62	0	0
	Other	Other	0	0	0	0

Observation: for the other cases the probability is 0 because of *apriori* 0 probability.

Manual DAG

Therefore, in order to combine the power of learning with the limitation of the dataset size and the knowledge gained through the expert case study done with Interiorvista, we developed a manual DAG and learned only the parameters from data.

Ten samples of the resulting Bayesian Network are presented in Table 4. Also, examples of relevant conditional probabilities learned are presented in Table 9 and

Table 10 for “number of chairs” and “number of nightstands”. Combining the strengths of the previous obtained models, such as the support surface balance and the deposit space ratio, this Bayesian Network adds also functional constraints for individual furniture pieces, such as: a table/desk should almost always have at least one chair and vice versa: a chair without a table in the room should be fairly rare.

This human-judged fact is also inferred from data, as the conditional probability table for “Number of chairs”, Table 9, outlines that if no table is present neither is a chair and a table is usually (i.e. in 66% cases) accompanied by a chair and the rest of 34% cases, by two chairs – which is plausible for a bedroom. The model also takes into account subtle, specific furniture grouping, for the types of furniture included in the dataset, according to general style and overall aspect of rooms, such as the bed with usually two nightstands, conversing overall symmetry of the interior design.

This is also outlined in the conditional probability table for “number of nightstands” node,

Table 10, which shows that for one bed and no tables in the room (e.g. no other supporting surface in the bedroom, and possible space left – due to the absence of table and chairs group), there are either two nightstands (with a probability of 83%) or none. If one table / desk is present in the room (i.e. some support space is available in the bedroom, and no information about the free space remaining for furnishing purpose), there is equal chance for having one or two nightstands. Further, such correlations can be depicted from the conditional tables of other nodes, mostly around the hidden patterns of available horizontal surface and depositing space in the room, main features in supporting bedroom specific activities and common routines.

Table 9. Conditional probabilities for # Chair in Manual Bayesian Network.

	# Table / Desk	0	1	2	3+
# Chairs	0	1	0	0	0
	1	0.66	0.34	0	0
	2	0	0	0	0
	3+	0	0	0	0

Table 10. Conditional probabilities for # Nightstands in Manual Bayesian Network.

	# Bed	#Table / Desk	0	1	2	3+
# Nightstand	1	0	0.16	0	0.84	0
	1	1	0	0.5	0.5	0
	Other	Other	0	0	0	0

Observation: for the other cases the probability is 0 because of *a priori* 0 probability.

Besides the direct observations, these hidden structures are also nicely outlined in the samples generated in this case, Table 4, which enforce the qualities of this latter model. Such examples include: a chair has a desk/table pair, the number (or presence) of the wardrobe influences the chest of drawers and nightstands. Because of more abstraction is involved in this model, we can depict different types of bedrooms, oriented towards certain functionalities and supporting certain styles. For instance, the last sample shows a bedroom, in which the main focus is on personal belongings and not on working or daily activities, only having, besides the bedroom, 3+ wardrobes, 3+ chest of drawers and 2 nightstands. This type of room could be appropriate for persons having plenty of clothing to store but also artifacts and decorations. In opposition to this room, the first instance depicts a room with only a bed and 3+ chest of drawers, the absence of the wardrobe and table/desk outlines the possible orientation of this room towards guests or short term renting. Most bedrooms, focusing on standard furnishing (most present in our dataset, too) have a bed with 2 nightstands (most common cases in IKEA bedroom furniture) and a wardrobe, with almost half of the cases presenting a table/desk to support daily activities.

Considering the achievements and limitations of the obtained models, it is possible to see that, with the increase of the model complexity and by combining expert interior design analysis with dependencies learned from data, we should be able to model abstract, hidden patterns, expanding from various local correlations such as functional, available deposit space, support surface to global ones resulting in different types of bedrooms, oriented to multiple life styles and supporting certain categories of activities.

Moreover, for this initial proof of concept, the model demonstrates that, with the right amount and type of data, we can capture trends and preferences of users and types of bedrooms from a person point of view, as opposed to a designer one. These results will offer Interiorvista the advantage to address the user preferences from a client side perspective, being able to learn, from real data generated by people furnishing their bedrooms, patters deeply present in customers' familiarities, rather than only imposed by nicely, well though arrangements provided by experts in field.

5.2 Mixture Models for furniture arrangements

A Gaussian Mixtures Model is meant to offer a way of arranging certain strongly-connected groups of furnishings, learnt from previous examples, therefore reducing the dimensionality of the searching space for placing the specific types of furniture produced in earlier stages with Bayesian networks. The intention is not to output a completely furnished bedroom, but rather gain some insight regarding the associations that can be made among the objects within a room, given by their coordinates and a few other attributes, automatically, and creating plausible arrangements that can be placed in the context of a new space.

The positioning of a complete set of furnishing layout would include, at a large scale, operations applied for arranging the pieces of furniture output by the Bayesian network, inside a given perimeter of a room, keeping in mind their dimension and functional dependencies from other objects. In the latter case, GMM would be used to draw samples from the resulted Gaussian for each object. The hard constraints for fitting everything inside the walls, without overlapping, would have to be satisfied by conditional formulas or more advances methods in the future (e.g. a learning system using many more examples).

5.2.1 Framework, technologies and models engaged

The arrangement component for furnishing layouts was developed with Python 2.7, installation included in the Anaconda platform [28]. The open-source distribution offers pre-installed packages which were heavily used in this project, such as NumPy, Pandas, SciPy, Matplotlib, and IPython. The IPython notebook offers an interface for code development and fast prototyping, making generating and saving modified versions of the same model and their results easier.

Scikit-Learn [29] is an open-source library, developed on the basis of these previously mentioned packages, offering stable and efficient tools for machine learning, data mining and data analysis. The GMM [30] and Dirichlet Process GMM [31] are available through the package, being completely reliable and ready to use.

The results were obtained using a GMM with maximum-likelihood estimation for the parameters, and a Dirichlet Process GMM (DPGMM), offering a prior distribution for the number of clusters through a Dirichlet process. The plots for GMM and DPGMM show the confidence ellipsoids of each of the centers, after trained with a '*full*' covariance type. For evaluating the performance of the GMM Model according to the number of components engaged, we used the Akaike Information Criterion (AIC) [32] and Bayesian Information Criterion (BIC) [33].

5.2.2 Data: usage evolution and selection motivation

The data engaged to model Gaussian mixtures must have a tendency to cluster around relevant points of interest. In our case, we look for groups of pieces of furniture which appear to be present in the same environment simultaneously, but yet one category would be the dominant one, whilst the objects in the other category would make more sense usually in the context of existence of the first one. For instance, for the chairs around a dining table would be more reasonably for the dining table to exist prior to adding the chairs into the room. Normally people arrange the table keeping in mind its functionality and the space that has to have around it for the usability of chairs and only afterwards set the chairs, almost instinctively.

Other than **the causality among the set**, a very important characteristic to selecting the type of objects that are going to be a part of the same group is **the rationality of the spatial distribution** in practice, for the objects within the room. In particular, although one can fairly assume that, in most cases, a wardrobe tends to occur in the same cases as the bed occurs within a bedroom, the placement of a wardrobe is very little influenced by its relation with the bed, instead having to comply with strongest, more restricting rules given by the dimensions of the room, for example, being itself a quite large entity.

The groups which are worth considering for the GMM are selected based on previous work in the field, the company's specialists expertize, the specifics of the rooms selected as examples, and the common sense any human can use in order to reject or accept a given layout. These groups are constituted having functionality as a common denominator: bed and nightstands, bed and TV, desk and chair. More about the data preparation for these models can be found in the dedicated section (4.3.4) and during the following chapters additional information will clarify both the implementation and the key decisions with respect to the involved data.

Based on previous research [12], and confirmed after testing with both higher and lower values, we generated other **200 examples** starting from each instance obtained after processing all rooms, drawing jittered samples from a normal distribution with the point as mean and the variance depending on the type of the link being highlighted.

5.2.3 Learning arrangements of furniture sets

In the following sub-sections we present the process of learning through GMMs and

DPGMMs three sets of items commonly found in typical bedrooms: bed and nightstands, bed and TV, and desk and chair, where placing the objects named first, in each example, influence the object(s) which follow.

5.2.3.1 Arrangements of bed and nightstands

The tuple used to train the GMMs for this case has the form of: $(x,\ y)$; the z-coordinate is unnecessary since all the objects have the floor as their supporting surface and the θ angle representing the item's rotation around the Oz axis remains the same for all the objects involved in the current set.

Initially, we considered all the instances of beds and nightstands aligned and overlaid within the same coordinate system, the bed having its top-left corner being anchored by the origin and the nightstands being placed sideways from the bed without any alterations to their size or distance to the bed from the initial context, marked in the working coordinate system through their top-left corner as well. We then removed the bed coordinates (all of them being (0, 0)), allowing the nightstands to be represented with better in the mixture. Figure 27 depicts the instances after translating and rotating them from their initial environment to the common coordinate system, representing the top-left corner of all nightstands. Each of the 18 instances of nightstands, extracted from bedrooms, was used to generate another 200 points, the new positions being altered by $\alpha \sim \mathcal{N}(0.25 \cdot \mathbb{I}_2)$, as shown in Figure 28.

Figure 27. Initial nightstands instances, each point representing their top-left corner.

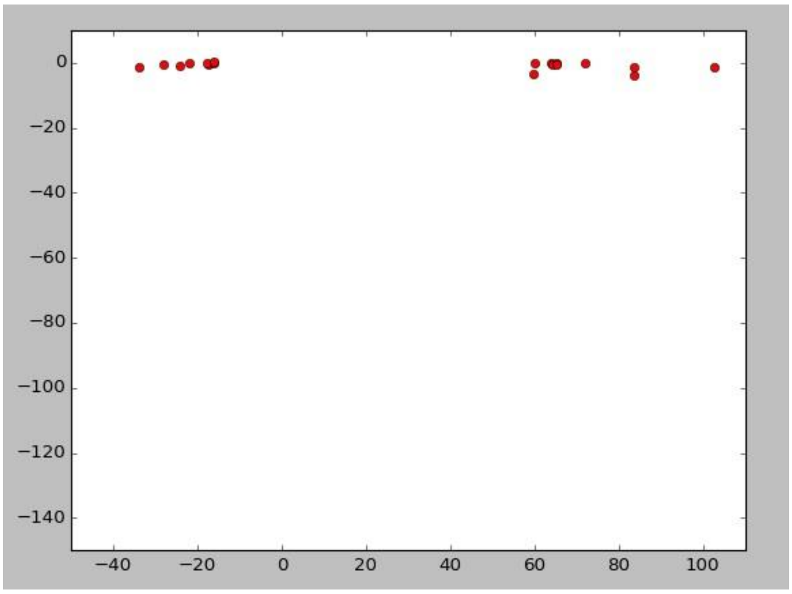
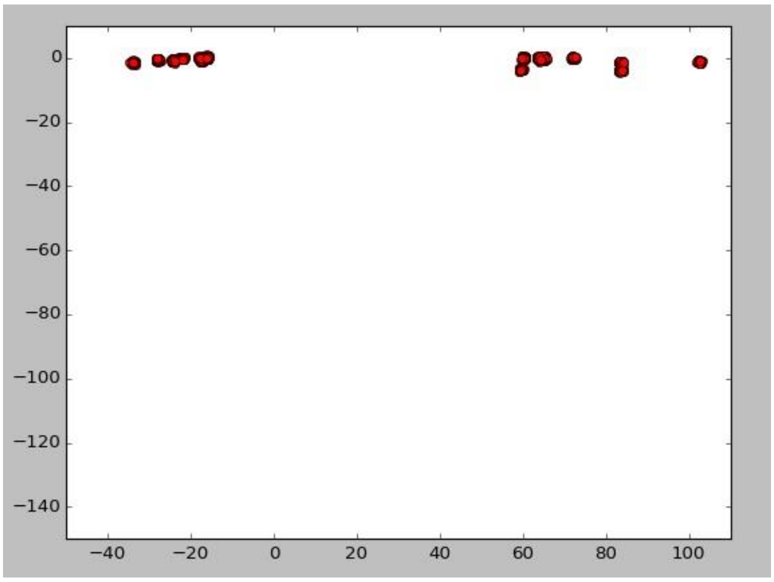


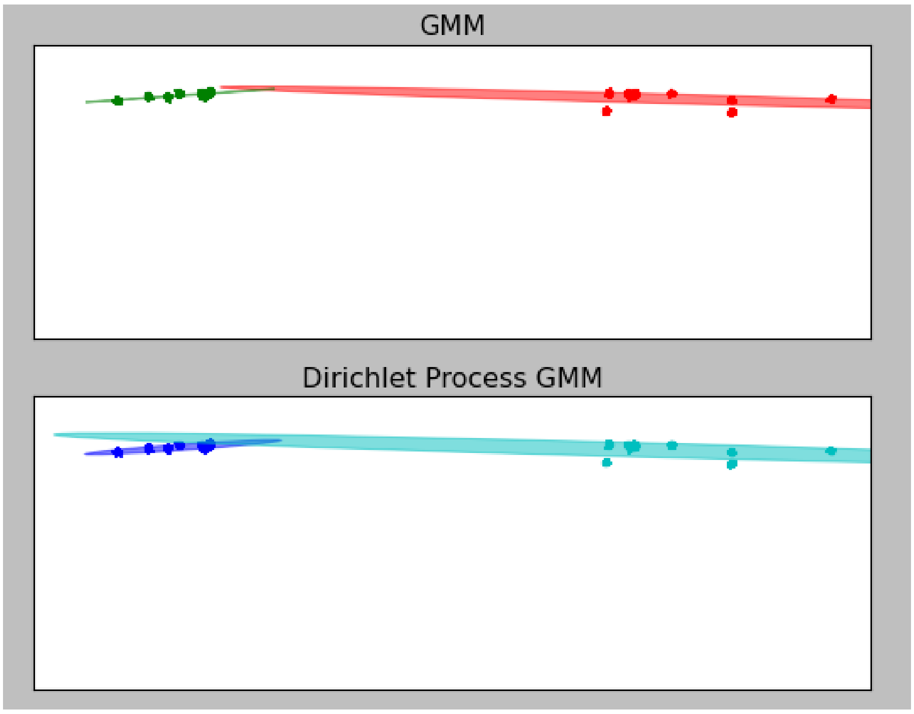
Figure 28. Sampling nightstands instances, 200 examples for each initial point

$\alpha \sim \mathcal{N}(0.25 \cdot \mathbb{I}_2)$



Forcing the GMM and DPGMM to have only 2 components, the results (Figure 29), does not yield promising results. Although the points are correctly identified as being in the left-hand or right-hand clusters, the ellipsoids are very elongated and overlap each other.

Figure 29. GMM and DPGMM applied for 2 components, initial assessment $\alpha \sim \mathcal{N}(0.25 \cdot \mathbb{I}_2)$



After increasing the noise for sampling, considering $\alpha \sim \mathcal{N}(1 \cdot \mathbb{I}_2)$, we observe that the confidence ellipsoids tend to remain the same as before (Figure 30). Further increasing the variance of the normal distribution for sampling forces the ellipsoids to thicken, but their long radius remains the same. The conclusion which can be drawn so far is that the right-hand nightstand examples are too far apart from each-other to improve the model through the help of parameters.

Using Akaike and Bayesian criteria in order to observe the optimal number of clusters to encapsulate our data (Figure 31), it is easily perceivable that the minimum scores are achieved for a number of clusters greater than 13, very close to the number of the points in the initial dataset, which could have been predicted.

Figure 30. GMM and DPGMM applied for 2 components $\alpha \sim \mathcal{N}(1 \cdot \mathbb{I}_2)$

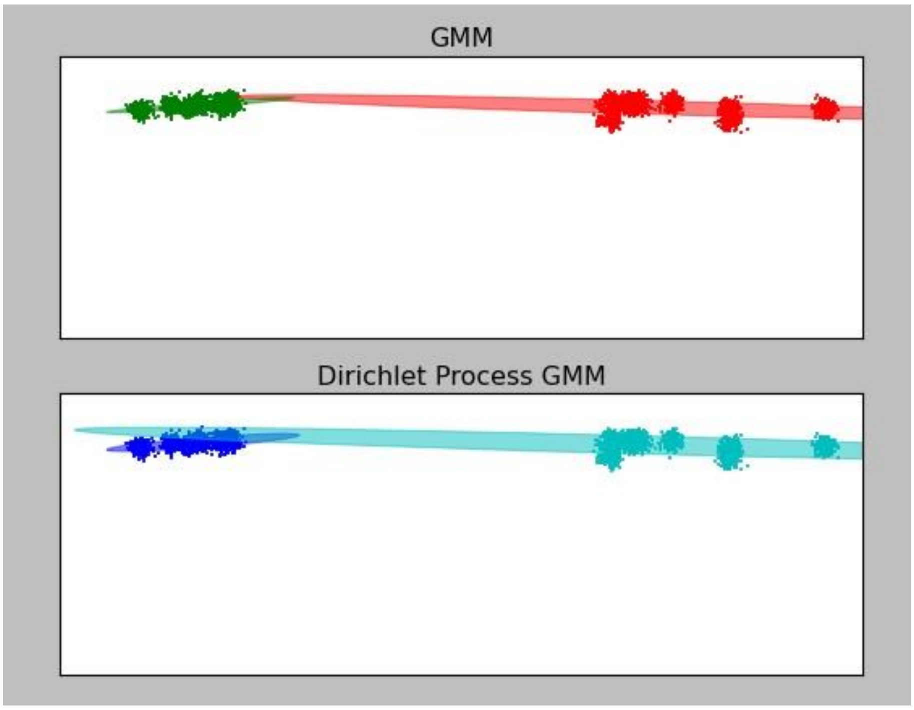
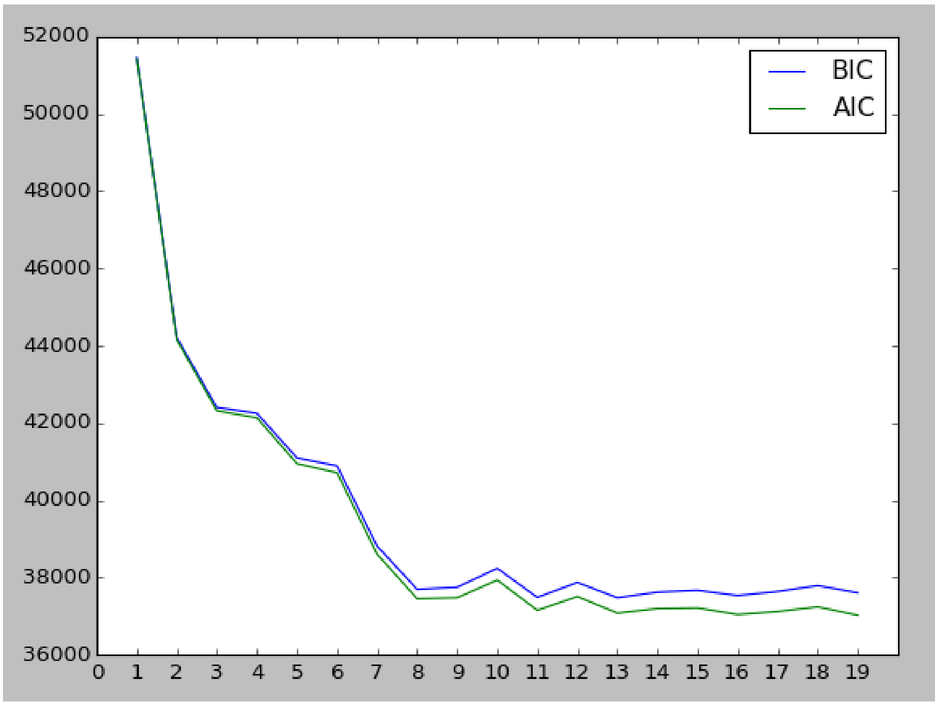


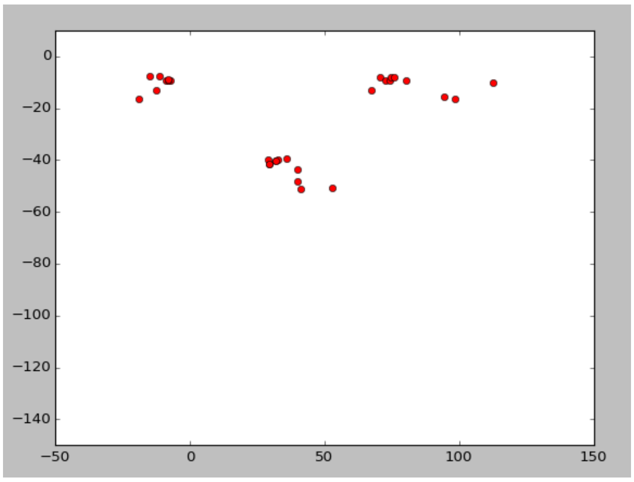
Figure 31. AIC and BIC evolution for GMM; nightstands points $\alpha \sim \mathcal{N}(1 \cdot \mathbb{I}_2)$



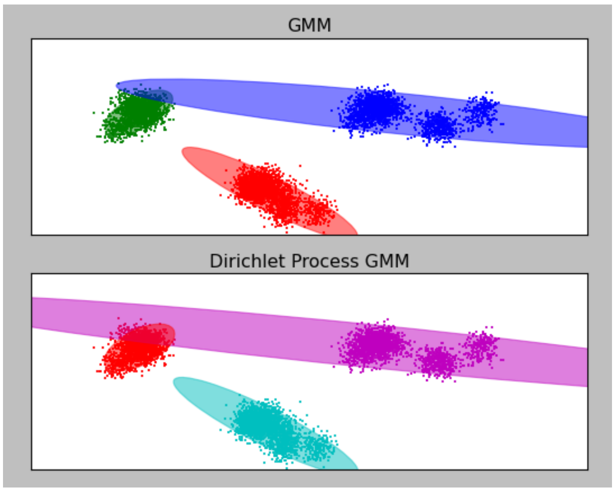
Trying to tackle the problem differently, we further considered the centers of the objects to bear more information than the top-left corners and we chose to include the bed instances in the set (Figure 32, a), to be modeled in the same process, given that now these instances are not all equal to (0, 0), becoming more relevant for the arrangement we are trying to learn, and also more believable.

Figure 32. Analyzing GMM performance with 3 components in the mixture for bed and nightstands given by their center points.

a) Nightstands and bed instances



b) Clustering augmented dataset: $\alpha \sim \mathcal{N}(3 \cdot \mathbb{I}_2)$



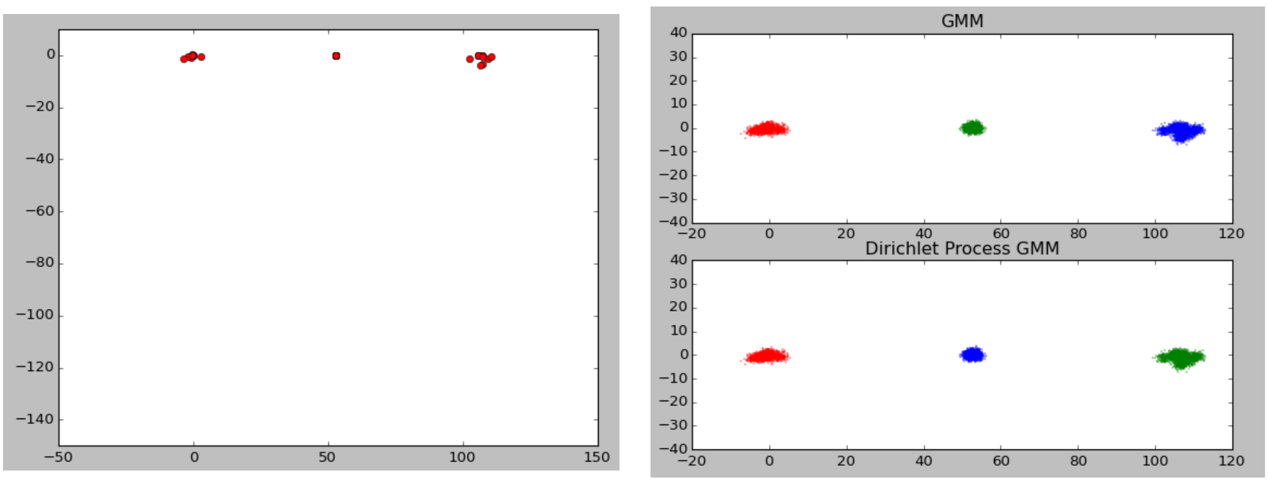
While the left-hand nightstand and the bed seem to be approximated reasonable, the right-hand nightstand continues to have problems for its large positioning variance, in Figure 32, b). AIC and BIC show that the optimal number of Gaussians for the mixture is at least 8, the almost-detached clusters being easily identifiable in the figure. Although it seems illogical at first as to why the left one has better results, since they usually are symmetrical with respect to the bed, the explanation comes from the fact that when doing the calculations for obtaining their coordinates, we considered the top-left corner of the bed to be the center of the axes and then obtain the nightstands' coordinates in accordance with that reference.

To solve the problem concerning different-sized beds which evidently influences especially the right-hand nightstand's final position, we consider all the beds to have the same width (the maximum found in the dataset), leaving all the other relative dimensions unchanged. Moreover, in order to balance the positions of the nightstands and convey their symmetry towards the bed, to the representative points in the working system, we chose as pivots: for the bed, the top-median point, for the left hand-side item, the top right corner, and for the right hand-side item, the top-left corner, respectively, as explained using Figure 18.

A plot for the initial points used for clustering is available in Figure 33, a). Now that the bed size is fixed, there is not much variance among the felt and right groups of nightstands. We decide to keep the points for the bed, serving as a clear demarcation for the objects. Besides, even though they have the same coordinates now, after over-sampling and drawing a location of the bed to fill in the given layout of the new bedroom, this operations might introduce a favorable variation, conferring an even more human-approved aspect of the desired environment to be furnished.

Figure 33. Analyzing GMM performance with 3 components in the mixture for bed and nightstands given by their top-right, top-median, and top-left, respective points

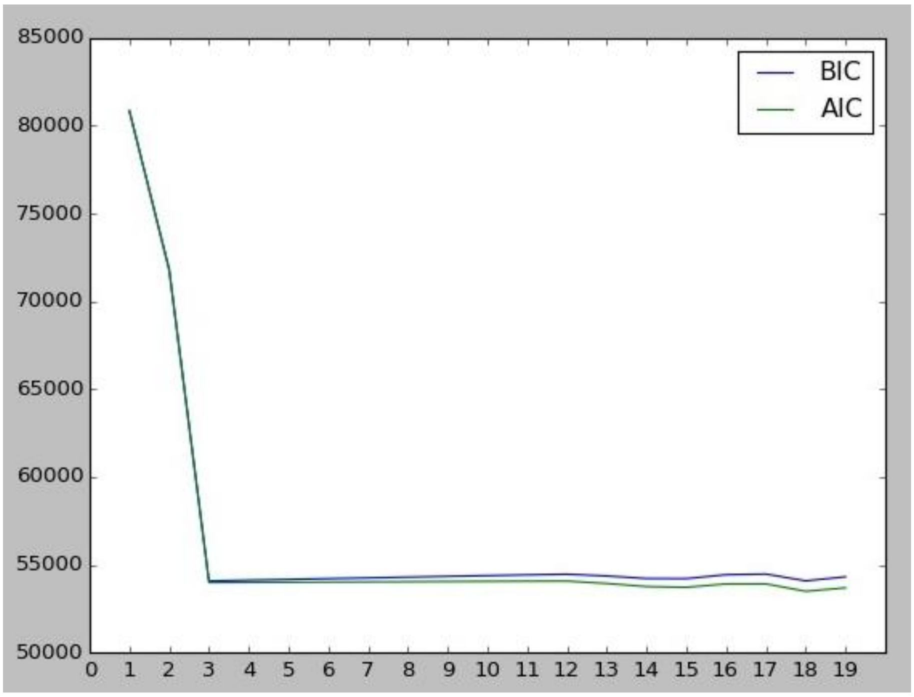
a) Nightstands and bed instances **b) Clustering augmented dataset: $\alpha \sim \mathcal{N}(1 \cdot \mathbb{I}_2)$**



Both GMM and DPGMM manage to approximate the data with ideally-shaped ellipsoids, Figure 33, b), reinforcing the ability to automatically extract an arrangement of the bed with one or two nightstands, depending on the needs.

Evaluating AIC and BIC scores for the latter experiment, Figure 34 confirms the expectations regarding the outstanding results for this group of items, capturing their interrelated functional and aesthetically pleasant properties. Both information criteria provide a steep descend until reaching the value 3, inferring that 3 components for GMM are exactly the right number to safely model all points in the dataset.

Figure 34. BIC and AIC scores for GMM over bed and nightstands positioning; $\alpha \sim \mathcal{N}(1 \cdot \mathbb{I}_2)$



The process of placing actual objects in the scene is achievable by reversing the operation we did for considering a universal size of bed, now we will subtract its width and add instead the width of the desired bed, the group being, thus, ready to be placed in the required room.

5.2.3.2 Arrangements of bed and TV

Taking the learning from the previous group, we had more insight when facing choices regarding implementation decisions for bed and TV. The link is usually created when a bedroom is provided with a TV. Another possible example from the same category would be sofa/fauteuil and TV, thus the TV being dependent on the sofa, instead of the bed, but maintaining the same functional link. However, in our dataset all the TVs that were present within the room were clearly related to the bed, their position and orientation being directly influenced by it.

The general rules someone complies with when positioning a TV inside their room, after the bed has been placed, regard the orientation and other objects interdependency constraints. The orientation has to provide a good angle so that watching TV from bed would be comfortable enough. In our dataset every TV has its forward vector pointing in the other direction as the bed, creating a 180° angle between the 2 forward vectors. The distance regards saving as much space as possible, while keeping the utility of the TV and the livability of the bedroom as a whole. In our examples, as in general, TV has either a supporting surface – usually a chest of drawers in our case – or is attached to a wall. It is desirable to have it as close to the wall as possible, if the room is not too big and the other furniture pieces allow it.

Given that, this time, the orientation is very important, the instances that are going to be learned from GMM now have 3 dimensions: $(x,\ y,\ \theta)$. We kept $(x,\ y)$ representing the top-middle point of each object, in a canonical form, as before, while θ stands for the orientation of the forward vector of each object, also modified from the original room when translated, so that the data would be consistent. Figure 35 includes the pivot points for all instances of bed, centered in $(0,\ 0)$, and the computed top centers for all 6 instances of TV.

We initially altered only the coordinates in order to generate more samples, with $\alpha \sim \mathcal{N}(5 \cdot \mathbb{I}_2)$, leaving the θ orientation identical to the one in the instance used to over-sample, resulting, again, 200 more examples for each of the initial instances.

Figure 35. Representative coordinates for bed and TV, the bed having the top segment centered in *(0, 0)*

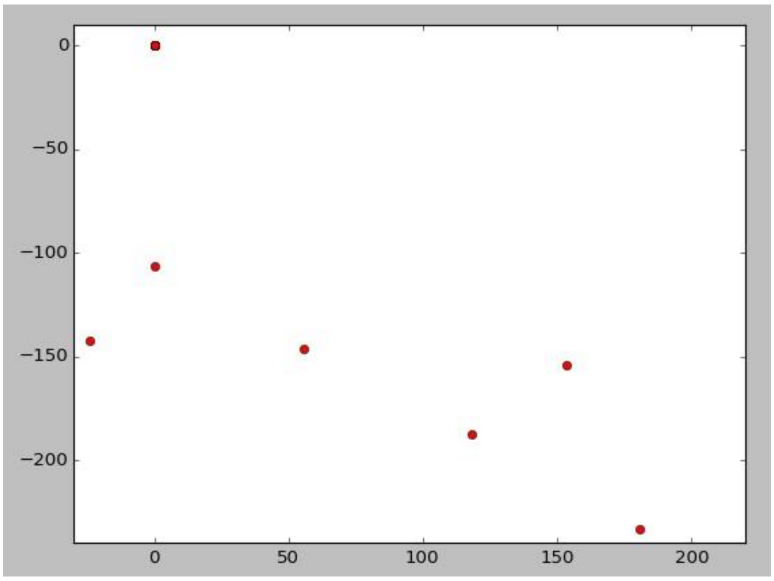
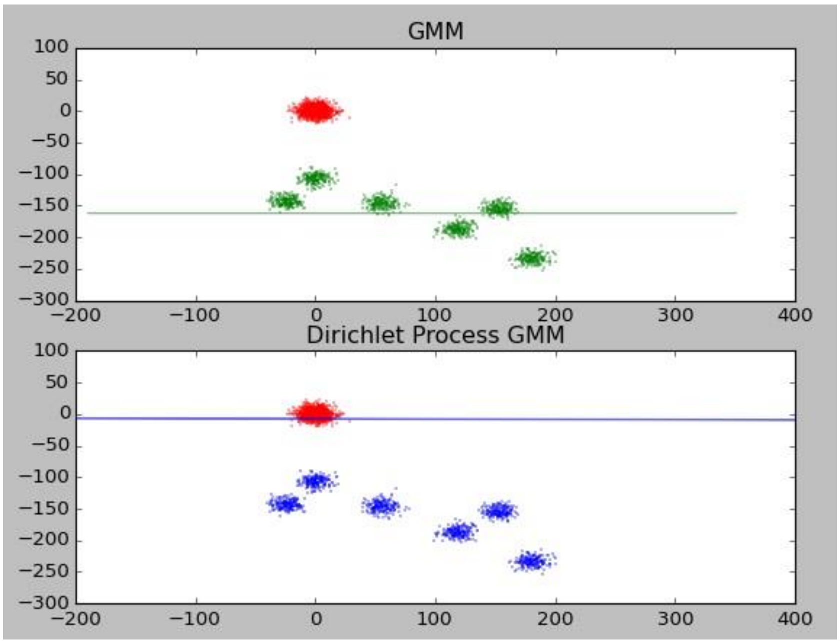


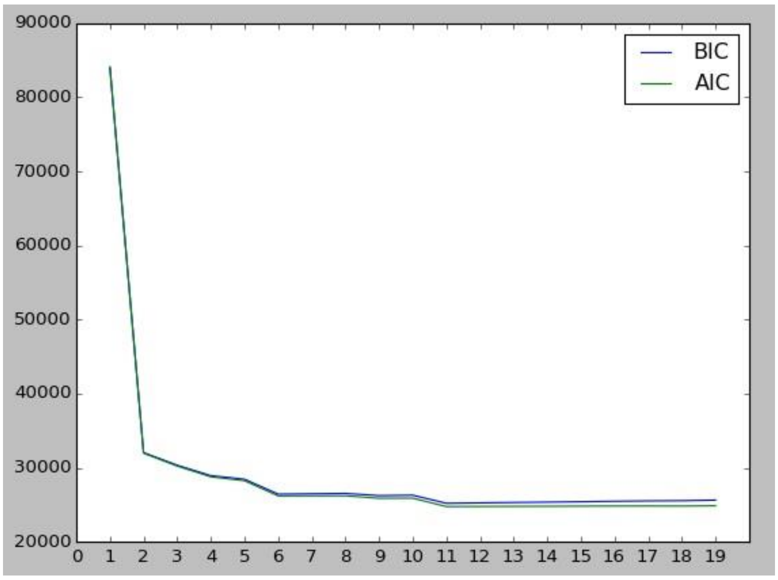
Figure 36 holds the results for learning GMs and DPGMs with 2 components over the data. The angle is not shown in the image, the plot is made according to the coordinates, but it is taken into account for training the models. DPGMM does not seem to capture the underlying properties of the data, drawing the TV's ellipsoid too high. On the other hand, GMM brings out a plausible Gaussian over the TV's set. The elongation of the ellipsoid is not necessarily a problem, since in real-life usually is near the opposite wall and does not have to be aligned on the *0y* axis with the bed. And with the bed being placed near a corner of the bedroom, it leaves the feasible amount of space for the TV; we just have to make sure that when we draw samples for new furnishings, the will be on the right side of the bed.

Figure 36. GMM and DPGMM over jittered coordinates with $\alpha \sim \mathcal{N}(5 \cdot \mathbb{I}_2)$, for bed and TV



As we assumed, AIC and BIC achieve minimum for more than 7 components, which may be accounted for the beds cluster, plus one cluster for each of the generative instances of TV. However, Figure 37 logs a high drop for 2 components, followed by rather minor improvements, as the number of components increases.

Figure 37. BIC and AIC scores for GMM over jittered coordinates with $\alpha \sim \mathcal{N}(5 \cdot \mathbb{I}_2)$, for bed and TV.

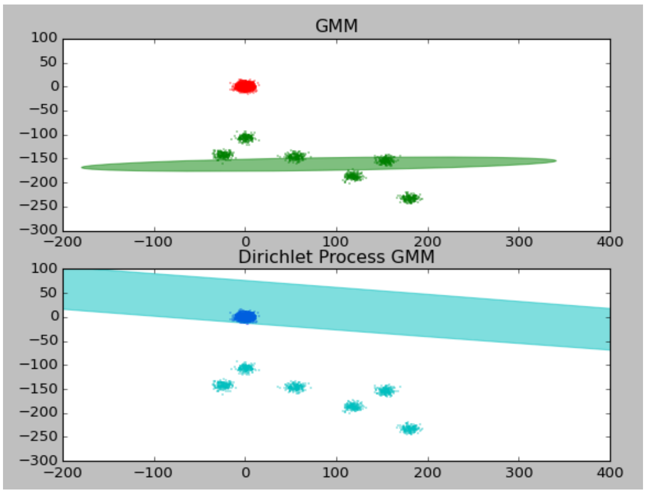


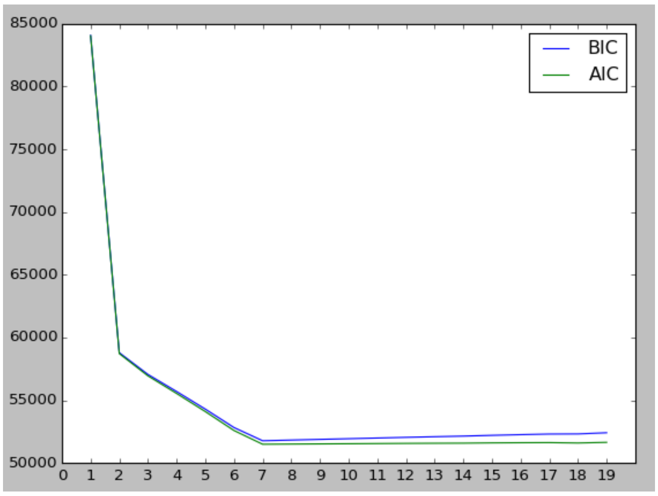
The length of the bed is not a problem this time because the room's dimensions diversity is far wider than the bed's length and the proximity of the TV to the bed, along the *0y* axis, comes from its proximity to the wall, instead of the beds' length differences. For a more human-approved appearance of the final arrangement, we trained the GMM after jittering the samples and adding noise to the orientation component as well. This procedure added thickness to the ellipsoid, making new potential candidates being prone to slight changes of position, as well as angle (Figure 38, a).

Figure 38. GMM and DPGMM over jittered $(x,\ y,\ \theta)$ tuples with $\alpha \sim \mathcal{N}(5 \cdot \mathbb{I}_3)$, for bed and TV

a) GMM and DPGMM results

b) BIC and AIC scores





In this example, Figure 38, b), it is more evident that a major improvement on the information criteria is obtained for 2 components, but for the minimum to be achieved are necessary 7 components, corresponding to the bed's cluster, plus 6 clusters for each of the TV instances, which does not come as a surprise.

5.2.3.3 Arrangements of desk and chair

For bedrooms designed with a working functionality purpose, especially inhabited by children or young people and others requiring their own space to work, a desk and a chair is often mandatory. The items are usually modeled together, but only two examples were met within our dataset, one of which containing a particular case of desk, custom-made for the wall. In general, the approach consists in gathering all examples and translating them to a reference system, enriching the dataset with jittered examples and, afterwards, twitching the parameters of a GMM for an optimal solution, as we did in previous cases.

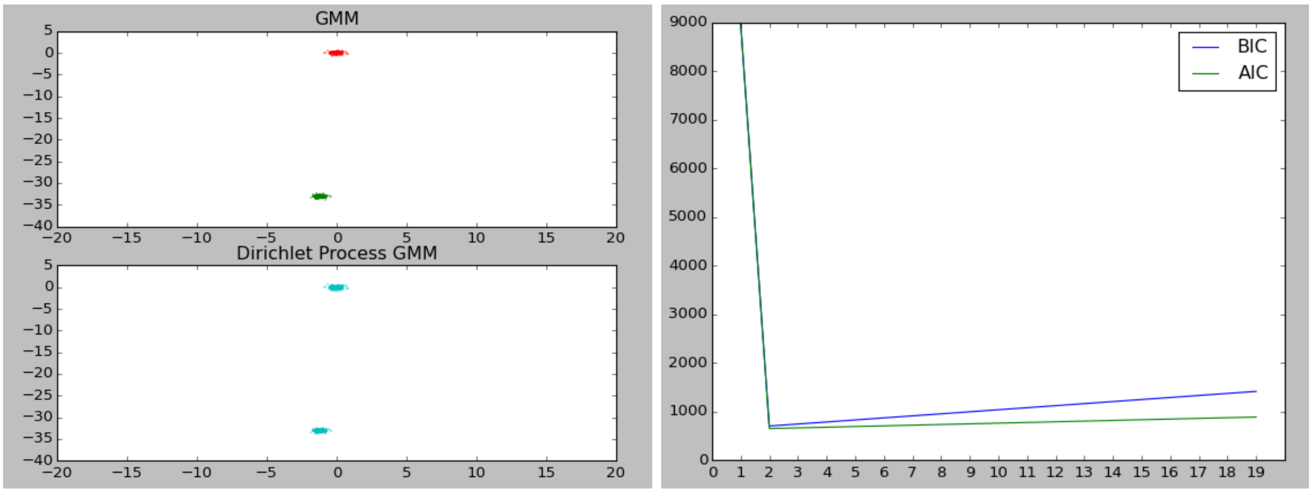
Having only one example, we can still model a Gaussian mixture after over-sampling so that it can be used in future layouts. Desk is the furniture piece that influences the chair's position and, therefore, we chose to center the top segment of the desk in *(0, 0)*, while repositioning the chair in the corresponding place. As well as the case of bed and TV, here we will model the GMM training $(x,\ y,\ \theta)$ tuples, as the orientation of the desk vs. chair is extremely important and should not be neglected.

Figure 39, a) presents the solution obtained after training both GMM and DPGMM, after jittering the instance tuple with $\alpha \sim \mathcal{N}(0.25 \cdot \mathbb{I}_3)$, thus obtaining 200 more samples. We can observe the DPGMM does not perform according to our intentions, since the third dimension, the rotation angle, is preventing the model to estimate 2 clusters, instead of one. In the next picture, Figure 39, b), we can observe the values of AIC and BIC scores. It is only logical that 2 Gaussians would be perfectly fit to model 2 clusters whose data were generated by 2 Gaussians.

Although it may seem artificial to model furniture arrangement from only one raw example, drawing samples for using them in a real-world furnishing problem may outperform rigid methods such as using a formula to indicate the position of the chair. This method will also provide a pleasant, human-approved arrangement, which comes from the mild variations of the rotation angles and positions.

Figure 39 . GMM and DPGMM over jittered $(x,\ y,\ \theta)$ tuples with $\alpha \sim \mathcal{N}(0.25 \cdot \mathbb{I}_3)$, for desk and chair.

a) GMM and DPGMM results **b) BIC and AIC scores**



6 Conclusions and Future Work

6.1 Conclusions

Artificial Intelligence has been present in design matters for over 20 years now [6], and is currently opening research paths within architectural engineering towards 3D building automatic planning, organizing interior spaces, furnishing layouts, introducing ornaments and matching styles. Our topic of interest in this thesis, populating a given floor plan with appropriate furniture pieces, raises concerns regarding feasibility and ensuring functionality while providing results that are compatible with professional guidelines and, above all, results that respect humans' common knowledge related to their environment, something that is very difficult to grasp and impossible to achieve via mathematical constraints.

Advancement in this field is, therefore, arduous due to the almost intractable search space of plausible solutions and the complexity of gathering enough examples as to capture the essence of the purpose of a particular space, which is, quite often, inaccurately branched into homogeneous or heterogeneous categories, based on the inhabitant's lifestyle, background, culture, and so on.

In the context of the design of a room planner, the final objective of the ongoing research that includes current results reflected in this thesis, is the development of a fast solution in the form of a software tool, offering customers an immediate visual result for furnishing their "dream room", with a given perimeter, in a form that complies with the person's needs, but also obeys interior design guidelines.

The personalized solution the user is looking for is not a matter of "dragging & dropping" items inside a formal layout. On the contrary, we deal with a possibility space that cannot be represented in a finite manner, ingraining a highly non-polynomial aspect in any method attempting to search throughout that space on account of finding an even simplistic solution.

In this thesis, the answer to those obstacles takes the form of a probabilistic approach, applied for bedroom fittings. Firstly, finding a way to approximate probabilities of objects occurrences within a room and sampling from that model to produce plausible and, sometimes, unseen types and number of components. We have achieved it through our Bayesian Network system, able to learn from few examples of manually designed layouts, probabilities for the apparition of each type of object in a bedroom, enabling us to get a grip on which objects will have to appear in our space.

Secondly, we propose a ready-to-use arrangement model, based on GMMs, able to learn the interdependencies function in the context of different bedrooms and, without demanding too many examples, output ready-to-be-placed items arrangements. The arrangements we unfold in the current project meet two requirements: they are suitable to be fit inside a bedroom and are strongly inter-connected. The sets of colligated furniture pieces, emerged from our work, fall under one of the three following categories: bed and nightstand(s), bed and TV, desk and chair. Any of these groups can be directly sampled within milliseconds and then naturally placed in the space to be furnished.

Evidently, the quality of both undertakings would improve enormously if a larger database was available, empowering specific oriented needs to be fulfilled through the gathering of enough examples to exhibit a particular feature evolution within distinctive contexts. The next section explains in a more detailed manner the future requirements and necessary tools to allow these improvements, thus significantly contributing to the ultimate goal of creating a fast, professional and user-specific 3D room planner.

6.2 Future Work

The plans for future developments can be partitioned into different-duration targets, all leading towards achieving the main prospect of putting at the disposal of customers a fast and complete tool for setting their floor plan and preferences, retrieving a custom-made, professional-approved furnishing result, visualizing it with the ability to change some components, and lastly obtaining an evaluation of the price for implementing it into his/her own room.

All the experiments is this thesis were carried out with an extremely parsimonious set of data, so that conclusions have to be taken not as definitive but only as a proof-of-concept of what the modeling can provide to the final software tool for the company. In order to create a full-featured planner, with seamless flow from one component to another, it is imperative to benefit from a wide dataset, counting thousands or more examples, as well as a tool for automatic extraction of object position, relative to a standard world-coordinate origin, rotation, orientation and dimensions.

A crucial requisite for successfully learning different types of interdependent fitting arrangements would be having access to many kinds of measurements and canonical positions, regarding placing the items within the same coordinate system. Other measurements include distances from every object to every other object, to the walls and the center of the room. The walls need not to be subdued to different levels of thickness and, certainly, the objects inside the room need to accurately refer their positions towards world-system's origin, placed in the usable space of the room, disregarding the walls thickness.

All of the objects need to respect the same naming pattern, in order to be easily retrieved, as well as traceable throughout their processing including data mining, data analysis and models' implementation. The number for each type of object needs to be automatically counted for all items within the room and added to the global counter. It would be useful to split the room instances into separate stacks, according to different criteria: shape of the room (e.g. rectangle, 'L' shaped), purpose of the room (e.g. working, children area) or other criteria a users' feedback form would reveal.

Bayesian Networks can be enhanced to take into account context features and user requirements. For instance, adding a node in the network for allowing the inhabitant to express the particular future use of the space, such as working bedroom or children's bedroom.

A tool for visualizing and scoring certain furniture patterns obtained by the system will allow the professional interior designer to rate a certain arrangement or the entire layout, in a practical manner, so that the current solution's evaluation would feed back to the system and, eventually, enrich the database.

Lastly, the software tool capable of integrating all the learned knowledge would allow specifying main socio-economic characteristics of the user (self-employed, young person etc.), retrieving similar patterns from the dataset, applying the corresponding models for occurrence, arrangement sets and furniture placement and rendering the result, with the price-tag associated, enabling, at last, to ask for feedback and re-integrate it in the system, as a new learning process.

7 Publication related to the thesis

Racec, E., Budulan, S. and Vellido, A. Computational Intelligence in architectural and interior design: a state-of-the-art and outlook on the field. *19th International Conference of the Catalan Association of Artificial Intelligence* (CCIA 2016), accepted and to be published in the Artificial Intelligence Research and Development book series, IOS Press. 2016.

8 Bibliography

[1] E. D. Feigelson and G. J. Babu, Big data in astronomy, 4 ed., vol. 9, Significance, 2012, pp. 22-25.

[2] J. Martín-Guerrero, J. Lisboa and A. Vellido, "Physics and Machine Learning: Emerging Paradigms," in *The 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016)*, Bruges, Belgium, 2016.

[3] T. J. Sejnowski, P. S. Churchland and J. A. Movshon, "Putting big data to good use in neuroscience," vol. 17, no. 11, pp. 1440-1441, 2014.

[4] V. Marx, "Biology: The big challenges of big data," *Nature,* vol. 498, no. 7453, pp. 255-260, 2013.

[5] K. J. MacCallum, Does intelligent CAD exist?, vol. 5, Artificial Intelligence in Engineering, 1990, p. 55–64.

[6] M. L. Maher, D. C. Brown and A. Duffy, Special issue: Machine learning in design, 2 ed., vol. 8, Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, 1994, p. 55–64.

[7] K. Williams and M. Ostwald, Architecture and Mathematics from Antiquity to the Future, Volume I: from Antiquity to the 1500s (chapter 1), pp.1-24, Birkhäuser, 2015.

[8] M. Maher, D. Brown and A. Duffy, Special issue: Machine learning in design, Artificial Intelligence in Engineering Design Anal Manuf 8(2), 1994.

[9] L. Mandow and J. L. P. D. L. Cruz, "The Role of Multicriteria Problem Solving in Design," Vols. Artificial Intelligence in Design '00, pp 23-41, 2000.

[10] P. Merrell, E. Schkufza and V. Koltun, "Computer-Generated Residential Building Layouts," *ACM Transactions on Graphics,* 2010.

[11] E. Kalogerakis, S. Chaudhuri, D. Koller and V. Koltun, A Probabilistic Model for Component-Based Shape Synthesis, ACM Transactions on Graphics (TOG) -

Proceedings of ACM SIGGRAPH 2012, 2012.

[12] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser and P. Hanrahan, Example-based Synthesis of 3D Object Arrangements, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2012, 2012.

[13] S. Chaudhuri, E. Kalogerakis, L. Guibas and V. Koltun, Probabilistic Reasoning for Assembly-Based 3D Modeling, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2011, 2011.

[14] W. Xu, B. Wang and D.-M. Yan, "Wall grid structure for interior scene synthesis," *Computers & Graphics,* pp. 231-243, 2014.

[15] F. Bao, D.-M. Yan, N. J. Mitra and P. Wonka, Generating and Exploring Good Building Layouts, ACM Transactions on Graphics (TOG) - SIGGRAPH 2013, 2013.

[16] L. Majerowicz, A. Shamir, A. Sheffer and H. H. Hoos, Filling Your Shelves: Synthesizing Diverse Style-Preserving Artifact Arrangements, IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 20, NO. 11,, 2014.

[17] T. Liu, A. Hertzmann, W. Li and T. Funkhouser, Style Compatibility for 3D Furniture Models, ACM Transactions on Graphics (Proc. SIGGRAPH), 2015.

[18] Interiorvista, "Interiorvista," Interiorvista, [Online]. Available: http://interiorvista.com/.

[19] ROCA, "ROCA," ROCA, [Online]. Available: http://www.roca.com/.

[20] Google, "3D Warehouse," [Online]. Available: https://3dwarehouse.sketchup.com/?hl=en.

[21] Google, "SketchUp Make," [Online]. Available: http://www.sketchup.com/products/sketchup-make.

[22] AUTODESK, "AUTODESK - 3DS MAX," [Online]. Available: http://www.autodesk.com/products/3ds-max/overview.

[23] Unity, "Unity 3D," [Online]. Available: https://unity3d.com/.

[24] P. Merrell, E. Schkufza, Z. Li, M. Agrawala and V. Koltun, Interactive Furniture Layout

] Using Interior Design Guidelines, ACM Transactions on Graphics, 2011.

[25] S.-K. Y. C.-K. T. D. T. T. F. C. S. J. O. Lap-Fai Yu, "Make it Home: Automatic Optimization of Furniture Arrangement," *ACM Trans. Graph. 30, 4, Article 86 ,* p. 11, 2011.

[26] K. Murphy, "Bayes Net Toolbox for Matlab," October 2007. [Online]. Available: https://github.com/bayesnet/bnt.

[27] B. Chesky, "Airbnb," Airbnb, Inc., [Online]. Available: https://www.airbnb.es/.

[28] "Anaconda platform powered by Python," [Online]. Available: https://www.continuum.io/why-anaconda. [Accessed June 2016].

[29] "About Scikit-Learn," [Online]. Available: http://scikit-learn.org/stable/. [Accessed June 2016].

[30] "Gaussian Mixture Model API," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.mixture.GMM.html#sklearn.mixture.GMM. [Accessed June 2016].

[31] " Dirichlet Process Gaussian Mixture Model API," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.mixture.DPGMM.html#sklearn.mixture.DPGMM. [Accessed June 2016].

[32] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*, 1973.

[33] G. E. Schwarz, "Estimating the dimension of a model," in *Annals of Statistics*, 1978.

[34] M. Zawidzki, K. Tateyama and I. Nishikawa, "The constraints satisfaction problem approach in the design of an architectural functional layout," *Engineering Optimization,* pp. 943-966, 2011.

[35] H. Xie, W. Xu and B. Wang, "Reshuffle-Based Interior Scene Synthesis," *ACM 978-1-4503-2590-5/13/11,* p. Hong Kong, 2013.

[36] P. WONKA, M. WIMMER, F. X. SILLION and W. RIBARSKY, Instant architecture, ACM TOG (SIGGRAPH) 22, 3, 669–677, 2003.

[37] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes and P. Waddell, Inverse design of urban procedural models, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2012, 2012.

[38] A. H. B. D. Siang Kok Sim, "Evolving a model of learning in design," *Research in Engineering Design,* pp. 40-61, 2004.

[39] F. Scheurer and H. Stehling, Lost in Parameter Space?, Architectural Design, 81(4), 70-79, 2011.

[40] Y. I. H. PARISH and P. MULLER, Procedural modeling of cities., Proc. SIGGRAPH, 301–308., 2001.

[41] P. MULLER, P. WONKA, S. HAEGLER, A. ULMER and L. V. GOOL, Procedural modeling of buildings, Proc. SIGGRAPH, ACM., 2006.

[42] J. J. MICHALEK, R. CHOUDHARY and P. Y. PAPALAMBROS, ARCHITECTURAL LAYOUT DESIGN OPTIMIZATION, Engineering optimization, 34(5), 461-484, 2002.

[43] K. J. MacCallum, Does intelligent CAD exist?, Artificial Intelligence in Engineering 5:55–64, 1990.

[44] Y. E. Kalay, "Architecture's New Media Principles, Theories, and Methods of Computer-Aided Design," 2004.

[45] J. Gagne and M. Andersen, Multi-Objective Facade Optimization for Daylighting Design Using a Genetic Algorithm, SimBuild 2010, 2010.

[46] R. AKASE and Y. OKADA, Automatic 3D Furniture Layout Based on Interactive Evolutionary Computation, Seventh International Conference on Complex, Intelligent, and Software Intensive Systems, 2013.

[47] S. N. A, Algorithms for VLSI Physcial Design Automation, Kluwer Academic Publisher, 1998.

[48] IKEA, "IKEA," IKEA, [Online]. Available: http://www.ikea.com/.

[49] J. Panero and N. Repetto, "ANATOMY FOR INTERIOR DESIGNERS," vol. 3rd ed., no. Whitnew Library of Design, 1975.