A stochastic approach for automatic layout synthesis in interior design, using a learning-based scoring function

Emil Racec

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

I am grateful to all those who contributed towards the success of this project. I would like to express my deepest gratitude to my advisor, Dr. Alfredo Vellido, for his unwavering support, collegiality, and mentorship throughout this project. A sincere thank you!

I would like to thank you Marc Fernàndez Vanaclocha for always standing up for this project, providing a deep understanding of Interiorvista company goals and continuous involvement in managing the company involvement in project requirement.
Abstract

Despite the increasing capabilities of computers to master sophisticated human-like tasks and the recent explosive new wave of Machine Learning-based methods, the interior design field still remains a hard-to-master area, without robust, mature models that could compete with the expertise of humans in the field. This is an exciting new area for Artificial Intelligence in general, still in its very early stages of development, both in terms of models performance and in terms of specialized data availability. Most current applications of this type of models remain only in the area of virtual reality. Veering away from this trend, the current thesis proposes an end-to-end proof of concept for applying Machine Learning techniques to realistically assess the quality of professional and realistic room furniture layouts. We do so by proposing a learning-based scoring function comprising various interior design guidelines, ergonomics and plain common sense metrics. We further propose a stochastic optimization proof of concept based on Simulated Annealing techniques, aiming to generate new plausible and pleasant furniture layouts that obey the strict regulations of interior design. This proof of concept represents a first step towards the final goal of developing a software tool that would eventually demonstrate that real world, furniture layouts of professional quality can be obtained in an at least semi-automatic manner, using an energy function that analytically represents, as cost terms, various furniture functional and style interdependencies, common practices in relative furniture positioning in a room and other ergonomic factors that contribute to obtain a pleasant, livable room. Using machine learning to adapt the ranking function parameters across various types of rooms and sophisticated furniture objects, the method is supposed to scale in modeling complex interior design know-hows, hard to be modeled mathematically or learned directly by a purely data-oriented model.

Keywords. Interior design, Machine Learning, stochastic optimization, learning based scoring function, Data Mining
Contents

1 Introduction .......................................................................................................................... 7

2 Related work ....................................................................................................................... 11
   2.1 Design as a learning problem ......................................................................................... 11
   2.2 Non-learning algorithms ............................................................................................... 12
   2.3 Evolutionary computation ............................................................................................. 15
   2.4 Stochastic optimization ................................................................................................. 20

3 The Interiorvista company and the research project description .................................. 26
   3.1 Motivation ...................................................................................................................... 26
   3.2 The Interiorvista company and its products ................................................................. 26
   3.3 The software tool (planner) for virtual interior furnishing ........................................... 29
   3.4 Algorithms limitations and challenges ......................................................................... 36

4 CRISP approach for data mining ..................................................................................... 39
   4.1 Problem understanding ................................................................................................. 39
      4.1.1 Motivations for a data based, learning model ......................................................... 39
      4.1.2 Data challenges in the interior design field ............................................................. 41
   4.2 Data understanding ....................................................................................................... 42
      4.2.1 Main characteristics of initial data ......................................................................... 42
      4.2.2 Data description ..................................................................................................... 42
5.5.3 Scoring function implementation details ................................................. 101
5.5.4 Steps implementation .............................................................................. 102
5.6 Model performance and running measurements ........................................... 104
5.7 Obtained results ......................................................................................... 108
5.7.1 Only collision and available space constraints ......................................... 108
5.7.2 Adding furniture distance relation relative to the wall .............................. 110
5.7.3 Adding interdependency relation .............................................................. 112
5.7.4 Adding rotation angle relative to the walls – Complete scenario .......... 115
6 Conclusions and Future Work ....................................................................... 120
6.1 Conclusions ............................................................................................... 120
6.2 Future work ............................................................................................... 123
7 Publication related to the thesis .................................................................... 125
8 Bibliography .................................................................................................. 126
1 Introduction

As very explicitly stated by Ilčík and Wimmer [1] in recent years, “there is a lack of robust and efficient techniques for modeling the interior of buildings. In particular, [...] for the subdivision of the interior space into rooms and for placement of furniture in those rooms”.

It might come as a surprise, as this area is commonly thought as closely ascribed to the arts & crafts domain, but interior design has become, of late, a field in which the use and analysis of data using computer-based techniques and tools is steadily becoming commonplace.

This is perhaps not as surprising if we think of interior design companies as just another type of businesses fully immersed in the current and pervasive trend towards Big Data [2]. It could also be seen as a natural step forward stemming from the early attempts, 20 years ago, to standardize Data Mining procedures in business, coalescing with the ready availability of inexpensive computer facilities and the widespread Internet adoption.

Some interior design companies are in fact realizing that internet-based tools can provide means of interaction with their potential customers that are akin to those of social networks. And as with social networks, electronic interaction with customers means that potentially informative data are automatically generated and can subsequently be automatically analyzed for knowledge extraction. This in turns generates value as customers can take advantage of customized service, making them more amenable for contact with third party companies selling their products.

The current thesis has been developed within the fold of one such type of business, namely the Interiorvista (http://interiorvista.com) company, which specializes on the development of web tools that provide customers with the means to design, in a semi-guided manner, their own interiors, also facilitating subsequent contact with vendors that can provide them with the elements of furniture and decoration that are part of customers’ designs.

Data Mining in this context requires careful problem and data understanding, as well as
often painstaking procedures for data pre-processing, prior to data modeling itself. In turn, data modeling can resort to algorithmic and/or quantitative methods. Over the last two decades, significant effort has been made in the design and development of Artificial Intelligence and, in particular, Machine Learning techniques for problems of architectural and interior design [3]. Interestingly, much of this research can be found on just a few publication outlets, out of which the *ACM Transactions on Graphics* journal and the *ACM SIGGRAPH* conferences have played a central role.

In this thesis we explore the possibility of applying stochastic optimization techniques for automatic room furniture synthesis processes that obey complex, rigorous and compulsory interior design guidelines and ergonomics. They are integrated in our developed room layout learning-based scoring function, having as end goals both the generation of new, plausible and complete layouts that encode such natural affinities, and the scoring/ranking of existing interior scenes for problems of interior design. All of this is developed within the constraints of an in-company project for software tools development that is part of an agreement between the Interiorvista company and the *Facultat d'Informàtica de Barcelona* (FIB) at *Universitat Politècnica de Catalunya* (UPC BarcelonaTech).

The main goals of this thesis can be summarized as follows:

**Goals of the thesis**

- providing an overview of the data mining process involved in the development of machine learning (ML) models addressing interior design problems and challenges such as: data requirements understanding, data acquisition, cleaning and correction, pre-processing and feature extraction.
- designing a learning-based, interior layout ranking function for assessing bedroom scenes and associating them to a realistic scoring, based on both hard constraints such as: objects collision, room or furniture accessibility; and various soft and more complex ones, such as: professional interior design guidelines, ergonomics, styles, functional arrangement and grouping.
- designing a stochastic optimization model for automatic furniture synthesis in a
room scene, based on the previously defined scoring function, that would take into account numerous, complex, specific interior design regulations, ergonomics, user preferences, otherwise hard to express analytically or learned directly from data, all towards obtaining new, professional-looking room layouts.

Main results of the thesis

- a data mining framework required for the implementation of ML and computational intelligence (CI) models addressing interior design challenges, including data definition, acquisition, cleaning, preprocessing and features extraction.
- a learning based scoring function for assessing room furnishing options for ranking according to hard constraints and more abstract, soft criteria.
- a stochastic optimization approach - Simulated Annealing, based on the scoring function defined to generate new feasible, realistically arranged furniture layouts that obey complex ergonomics and design criteria, analytically encoded in the evaluation function.

Structure of the thesis

The remaining of the current thesis is structured into 6 chapters, as follows:

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; 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: Veering away from the current trend towards virtual appliance of interior design ML and CI based model, we present in chapter 3 the real-world, commercial environment details, aiming towards the end goal of the system whose proof-of-concept is presented in this thesis: the integration of the tools offered by Interiorvista to virtually display representative furnishing solutions for the clients that want to buy furniture in an online
environment.

**Chapter 4:** In this chapter, we detail the data modeling framework, structured according to CRISP Data Mining approach, involving all stages from data requirements understanding, acquisition, cleaning, preprocessing and features extraction.

**Chapter 5:** Two main contributions are presented in this chapter: a learning based scoring function, aiming to deeply assess a variety of aspects in a room furniture arrangement starting from hard constraints such as collision detection, furniture accessibility space and moving to the more abstract ones, that are a key factor in the overall sense of reality, such as: furniture fine affinity relative positioning, functional grouping, ergonomics, common furnishing practices. The second contribution is a stochastic gradient optimization procedure, based on Simulated Annealing addressing a still hard to master task for a machine: automatic furniture arrangement. The model is using the predefined evaluation function to generate new plausible, believable bedroom layouts, given a representative, basic subset of furniture types for a simple, yet functionally complete bedroom.

**Chapter 6:** We summarize the conclusions and outline some ideas for future work in the project in the closing chapter of the thesis.
2 Related work

In recent times, architecture has excelled in technological terms, from the points of view of research, innovation, development and production - undergoing continuous change, influencing and being influenced by modern technologies. One recent, radical change was the introduction of computer-based methods in architectural practice, which has changed the face of this field in all its subdomains, including materials, engineering and design.

Powerful software frameworks with underlying sophisticated methods for computer-aided design (CAD) and engineering (CAE) are widely used nowadays to assist professionals in the field to virtually model, in depth, every detail of the architectural process.

Being still a field in its very early stages in terms of AI-related development, performance, maturity and achieved robustness (despite much researched for almost three decades now) architectural design processes still remain hard-to-master tasks for computer-based models, when confronted against a professional human designer.

We further explain models approaches and evolution, focusing on their targeted domain, constraints, contextual strengths, limitations and specificity regarding the proposed solution for a particular problem.

2.1 Design as a learning problem

One of the earliest studies addressing the challenges imposed by the complexity of the design process through learning, was presented in [4]. This research analyzes the relationship between the design phenomenon and the possibility of learning in this context, in a way that would benefit the designer. The model is based on an analytical, formal representation of cognitive activities in design. The resulting system, *LinD*, tested various correlations and learning areas that could be further exploited in a learning fashion, including: abstraction / detailing, association / disassociation, derivation / randomization, generalization / specialization, and similarity measurement. Despite the fact that the
obtained model has certain limitations, including its specificity, given that it uses only a single subject (a professional designer recorder for two hours and 45 minutes while doing his work) and that the overall reported predictability results were poor [4], this study is especially relevant for it emphasizes that design implies learning hidden and often complex patterns, best practices and a process of evolution towards perfection, all of which have the potential of being further evolved in automated systems aiming for assistance and task refinement in design.

Interior design, coexisting only in more recent years with architectural design, present for almost three decades in the field of AI, naturally developed as an evolution from the latter, undergoing exploration in various directions and well-known paradigms in CI and ML, in the attempt to develop frameworks that would be more intelligent and helpful to their users by achieving goals such as: assisting a professional designer in a specialized manner (by learning common practices, suggesting relevant next steps, partially to fully automating of certain, complex tasks); and guiding and reinforcing an amateur user with rigorous design know-how information adapted to its current goal by suggesting specific next steps, corrections, suggestions. While the main common tasks addressed in the field of architectural design include the automatization of floor plan generation (deciding the layout of the spatial allocation) and exterior building layout design (The overall construction form and exterior façade), most important ML directions in interior design concern the following sub-fields: furniture selection and arrangement, furniture style assessment, and ornamental decoration.

In the following, we will briefly describe the most important paradigms that govern the AI-related field of design, within the context of architectural construction in its relation with interior design and with a focus on furniture selection and arrangement, the ultimate goals of our project.

2.2 Non-learning algorithms

Many classical algorithms have been applied to solve challenges in design, and their
drawbacks are the main justification for the need to develop learning models. In architectural design, an extremely difficult challenge if attempted in all its true complexity using a classical approach, is that of floor plan generation, also known in literature as the spatial allocation problem. In this scenario, the user would define a set of hard constraints, such as the desired number of rooms, the imposed adjacencies and the dimensions, in a way that the model would exhaustively generate all the possible combinations, which would validate the constraints. Because of the obvious limitations of classical approaches in a process that implies the exponential growth in number of solutions generated, manual design of the constraints for each context is required. Because of such inflexibility, these methods had very limited application, mostly to the design of buildings that almost always have known shapes and with their space distribution highly regulated by parameters and rules, such as schools, health facilities, etc. [5].

Even methods from very different fields were applied in the attempt to provide a solution to the problem of floor plan generation, as is the case of VLSI circuit layout design, in which an initially proposed, manually design floor plan is optimized locally, for a defined goal [6]. The main limitations of this type of methods include: the possibility of reaching local minima; the burden to encapsulate the goals in the optimization criterion due to the complexity of the objective; and the need of an architect to manually define the initial floor plan. Moreover, the architect might need to draw more than one layout for local minimum avoidance, a problem that can be address by starting from a different point in the search space of possible configurations.

Moving now the focus to interior design, recent non-learning based methods still coexists with the learning-based ones, because they offer a key advantage: they do not need to rely on the existence of a (large enough) dataset. Moreover, in this subfield, the availability of such complete, vast, professional datasets containing human-furnished rooms is very limited, because they are usually built as part of private companies’ undertakings, which does not make them publicly available for obvious commercial reasons. Furthermore, when dealing with a large dataset, automation of the data pre-processing phase is a decisive factor that is hard to accomplish due to the amount and nature of pre-processing involved: often dealing with the complexity of the 3D models in scenes, in-depth
knowledge of design and usage of dedicated frameworks to manipulate the graphic represented entities.

Therefore, in [7], the authors proposed a model in which one such big and complex dataset is not available, introducing a non-learning, heuristic-based approach to generating new room layouts from existing ones – which are “only several manual designed scenes”. The model is structured as a pipeline, comprising three stages: input scenes pre-processing, followed by new layouts generation (through shuffling) and finally filtering best candidates. Because of the main initial condition: a (very) small subset, complex and general rules regarding furniture arrangement (achieved mostly through learning in other cases) were not a plausible option, and neither was the choice of a probabilistic, data-oriented approach for the furniture arrangement. Therefore, and as also seen before in non-learning models applied to architectural building, the authors decided to drastically reduce the dimensionality and representation of the problem, in order to reduce overall complexity.

In more detail, the entities in the input scene and the relationships among them are grouped and merged, resulting in fewer types of entities, functionally defined, which preserved main functionality (e.g., the sound boxes besides a TV should be grouped in one unit; chairs around a big table remaining together converse the usage and usefulness of the corresponding table) linked through limited number (exactly three) of relationships between them (i.e. facing relationship – one unit in front of another: TV and sofa, aligned relationship – same type entities, placed adjacent: two chairs and supporting relationship – one entity is the physical support of the other: desk under a book). At the global level, the scene is represented as a graph, containing exactly one binary relationship between the adjacent units.

With this scene representation, the algorithm is based on iterative shuffling/reshuffling, one entity at a time, by replacing units or adding new ones. A visual example is presented in
Figure 1, where the layouts in the box are the original rooms. Evaluation metrics are applied to assess the results, the formula consisting of two factors: size variation and furniture distribution in the environment, summed using a method that encourages more reshuffles (i.e. diversity in outputted scenes). The assessment is followed naturally by filtering, which is present in the final step of the pipeline and between reshuffles in order to prevent entity generation explosion, resulting in mixture layouts based on the input scenes – controlled by the rigorous grouping of the furniture pieces in functional units and the crafted relations among these.

2.3 Evolutionary computation

Genetic algorithms have also been applied to solve design challenges, starting with architectural building, including the layout generation problem [8] and building façade definition according to some chosen optimization criteria, as daylighting [9], and later in interior design to generate room furnishing layouts [10] - being part of the models used for searching in domain space.
Figure 1. Furniture layouts obtained through reshuffle, in a non-learning fashion [7]
Evolutionary approaches have the strength to incorporate certain preferences and customizations - reinforced and better contoured through an iterative process, which optimizes a certain objective by maximizing the fitness value of individuals throughout generations. These advantages come at well-known costs, such as: heavy performance impact [9]; often getting stuck in local minima / maxima [9]; results being heavily influenced by the initial chosen individual [9]; or user manual feedback needed at each generation.

An interesting approach, tackling the automatic furniture layout optimization problem using genetic algorithms was presented in [10]. The authors focused on obtaining a unique
arrangement, in which the user’s tastes and personality were the main priority, rather than a more standard, common, professional created layout – in which interior design rigidness would dominate. This is always a tradeoff and comes at the cost of involving the user more intensively. In this case, each generation of individuals is evaluated by the user, maximizing both user preferences (i.e. from continuous feedback to the model) and ergonomics, by using a cost function encoding various related factors such as: accessibility, visibility, pairwise distances and angles.

Figure 2. Flowchart of the Genetic algorithm workflow

Maintaining the general flow of a classic genetic approach, as shown in Figure 2, the process of obtaining the final layout is an iterative one in this work, each stage involving user’s evaluation of current room layouts, followed by the creation of a new generation through selection, cross-over and mutation. Selection is carried out using a fitness function based on the defined ergonomics (accessibility, visibility, pairwise distance and pairwise angle) and user’s evaluation: a grade proposed by the user between 0 (MIN) and 1 (MAX),
which has a significant impact on the overall fitness of that individual (if the user likes the layout and scores it with a degree close to 1, the individual will likely participate in the next generation). Cross-over and mutation are realized in a classical manner, as in any genetic algorithm, in a process supported by the gene-like representation of the room furniture layout and containing all information (each piece’s coordinates and angles with their corresponding pairs) expressed binarily.

Figure 3 includes an example of a few layouts generated by the system, showing its ability to adjust pairwise distances and angles according to user preferences. Because the user evaluates each individual at each iteration, the system cannot scale beyond a certain threshold imposed by the time required to evaluate each entity. Moreover, because the model complexity increases heavily with the expansion of scene details, which is not an option in this case, the rooms included in the study had to be simplistic, including few and basic types of furniture, resulting in scene solutions being still far from real-world appliance.

Figure 3. Example of room layouts generated by an evolutionary based model [10]

This powerful paradigm has not yet shown promising enough results neither in the
architectural engineering field, nor in interior design, somehow failing at providing real-world inspired solutions [8] and requiring the user’s continuous feedback throughout the iteration process [10]. It has also yielded poor overall performance and generalization [9]. Having said that, evolutionary models are still worth considering for their strength in generating diverse, highly-customized and user preference-oriented results, as opposed to standard, designer-guided blueprints.

2.4 Stochastic optimization

Stochastic optimization methods have been successfully applied for various problems in which the defined space is usually very large and high-dimensional, not offering the possibility of being explored in an exhaustively manner. In contrast, the stochastic gradient methods define a heuristic based approach for searching the space, thus providing an approximation of the real solution in a timely manner. To achieve this, the category of methods usually implies the definition of 2 main components:

- **An energy function**: that evaluates the current state quality and proposes a score accordingly. The model uses this function in an attempt to find the optimal solution, associated with the minimum value of the energy function, by iteratively searching the local neighborhood of the current state towards new states that lower the energy. Because of the strategy employed, these methods are prone to get stuck in local minima, thus implementing a heuristic approach, usually by defining a special “reset” step that aims at restarting, in a probabilistically fashion, the gradient descending and avoid getting stuck in local minima by enlarging the search space - through exploring “further” neighborhoods.

- **A set of steps**: the set of steps defined are used to move the current solution in a new solution, towards achieving the minimum. The set of possible movements usually imply the existence of 2 types of steps:
  - **Convergence steps**: that are meant to drive the solution towards a local minimum, thus contributing to the algorithm convergence process
- **Local minima avoidance steps:** that aim at restating the algorithm by producing jumps, increasing the probability of exploring further search neighborhood, with possibly better minimum solution. An example of 2 different stochastic gradient convergence paths, achieving different minimum, is presented in Figure 4. In case of using this type of steps, which often produce states with higher energy, the cost function is adapted accordingly, to allow, in a probabilistically manner, transitions to “worse” states (i.e. higher energy ones).

In general, at each iteration a step, from the possible set of steps is randomly chosen to compute the new state value.

**Figure 4. Illustrative example of gradient descent converging to two different minima.**

Representing a popular choice for searching a high dimensional space, gradient based methods have been successfully applied also in architectural design, initially in exterior design – for representing the goodness of a layout [11]. Other approaches use Simulated Annealing (SA) [8] and Monte Carlo Markov Chain (MCMC) to explore the vast space of solutions and produce plausible yet diverse city scale results [12]. Stochastic optimization was successfully applied in furniture arrangement [13] and for the optimization of interior ergonomics [14].
More recently compared to exterior design, stochastic gradient methods have been successfully applied also in interior design, because of their power to model complex, abstract furniture fine affinity relations that are hard to model mathematically because of the complexity implied by the field and also proved hard to capture in a learning, purely data oriented fashion. For instance, although probabilistic models have been successfully used representing the search space in a probabilistic manner, thus capturing abstract patterns using 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 stylistic furnishing subtleties towards achieving homely, artistic, construction-ready solutions.

In the light of these challenges, other analytical approaches have emerged, aiming at a mathematical formulation of both hard constraints (e.g. related to architectural feasibility and regulations) and soft constraints (e.g. related to desirability, pleasant features). Usually these approaches are based on stochastic optimization models such as SA [13] or MCMC [14], defining an energy function that presumably incorporates defined constraints - which aim to encapsulate underlying rules and patterns that designers apply in their work. Typically include a research phase dedicated to determining such rules as well as ways to analytically translate and combine them in density functions, this group of models tackle the interior design challenge from a more structured, algorithmic angle, as opposed to the looseness allowed by data-driven approaches. These methods usually define a set of steps that aim at efficient, iterative, convergence, while having a “reset” mechanism to avoid getting stuck in local minima.

In our project, we draw inspiration from such methods, because of their ability to incorporate specific, important design guidelines that help fine tuning the potentially rough approximations obtained in the initial data-driven stage, through optimization.

Related work is presented in [13], where furniture arrangement is accomplished through an iterative optimization approach, given a complete specified room plan and a finite set of fixed entities, representing a similar, yet more restricted, context to the end goal we aim,
but an inspiration for the scope of the proof of concept developed in this model.

The system learns \textit{a priori} various features regarding the objects such as visibility, availability space, common usage, hierarchy and positioning (absolute and relative to other objects). In this work, SA is accompanied by a Metropolis-Hastings state-search step to minimize a cost function, which integrates the most relevant features claimed by authors to obtain a realistic, human-approved room layout, such as:

- accessibility – the ability for a person to use/access the object accordingly
- visibility – not obstructing the front of relevant objects (e.g., TV, paintings)
- pathways connecting doors
- pairwise constraint: applied to highly interdependent objects (e.g. bed and nightstand, TV in front of a sofa) – manually designed for each type or room

Designed to iterate through the furniture configurations solution space, a possible set of steps is defined in [13] as:

- small objects rotations and translations - contributes to convergence.
- swapping objects - used for diversity and avoidance of local minima.
- moving pathways control points - helps in adapting the paths in the room with object movement in the iterative process, towards minima.

Graphical results obtained for running the model are presented in second and third image, the first one showing the initial stage modeled.

\textbf{Figure 5. Results obtained using SA for automatic furniture synthesis}
Similar work was carried out in [14], where a software-guided interior synthesis system based on an MCMC sampler is presented. Such framework aims at guiding a user to furnish a room in a professional-like manner, by incorporating functional and visual criteria in the suggestions offered, as seen in the screenshots provided in Figure 6. Because of the underlying model type, the system has the capability to allow the user fix, at each step, any desired furniture piece, and receive positioning suggestions for the other ones accordingly. As in [13], the system incorporates pre-established interior guidelines, expressed analytically through independent terms and combined in a density function. The criteria used in this model are split in functional terms:

- **clearance** – e.g. objects are not blocked, are accessible and therefore can function as designed.
- **circulation** – ensures that the main flows and room utility are not affected by furniture positioning.
- **pairwise relationships** – defined between dependent (coupled) furniture pieces (e.g., table with chairs, TV with sofa)
- **conversation feature**— supports furniture arrangements that encourage social aspects: conversations and collective activities.

and visual ones:

- **balance** – room symmetry (e.g., furniture positioning at a global level that preserve aesthetics)
- **alignment** – exact positioning and orientation of furniture relative to close proximity.
- **emphasis** – this describes the point in the room towards which the users should usually be oriented (e.g., a fireplace, a TV, or a painting) that has a certain layout dominance.

Figure 6. Screenshot of the interactive system suggesting furnishing layout to a user.
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 [15] is an interior design company that aims to empower furniture selling companies, such as Roca [16], 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

3. Figure 9 we present a single, zoomed one.

**Figure 8. A gallery comprising various styles of chicken furnishing layouts**
Figure 9. An example of a 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
2. Figure 10.

Figure 10. Interiorvista planner: starting with gallery vs. style.
3. **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).
4. **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.

5. **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

6. 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 plumbing (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 computational intelligence 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 \textit{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.

- \textbf{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.

- \textbf{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.

- \textbf{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 patterns 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 [17], which were created in Google SketchUp [18], 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 [17], 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 advised 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.

- 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 centimeters 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 [19],
and Unity [20], 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 [20], 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 axis, 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 [19] 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 script 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.

<table>
<thead>
<tr>
<th>Furniture name</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>rot</th>
<th>FW</th>
<th>width</th>
<th>length</th>
<th>height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desk</td>
<td>86.4</td>
<td>15.72</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>Light_02</td>
<td>25.18</td>
<td>202.39</td>
<td>26.81</td>
<td>0</td>
<td>-90</td>
<td>1</td>
<td>0</td>
<td>11.41</td>
</tr>
<tr>
<td>Window_02</td>
<td>265.43</td>
<td>120.99</td>
<td>39.43</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>109.95</td>
</tr>
<tr>
<td>Window_01</td>
<td>181.5</td>
<td>-0.12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>91.04</td>
</tr>
<tr>
<td>TV Desk</td>
<td>257.07</td>
<td>211.22</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>74.44</td>
</tr>
<tr>
<td>Wardrobe</td>
<td>256.98</td>
<td>279.55</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>DVD</td>
<td>248.86</td>
<td>166.65</td>
<td>19.92</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>16.87</td>
</tr>
<tr>
<td>Phone</td>
<td>250.98</td>
<td>106</td>
<td>38.62</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>5.23</td>
</tr>
<tr>
<td>Frame</td>
<td>245.2</td>
<td>123.8</td>
<td>39</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>TV</td>
<td>248.53</td>
<td>215.01</td>
<td>26.96</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>76.65</td>
</tr>
<tr>
<td>Bed</td>
<td>15.28</td>
<td>113.77</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>82.45</td>
</tr>
<tr>
<td>Nightstand_01</td>
<td>16.13</td>
<td>89.59</td>
<td>12.5</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>23.33</td>
</tr>
<tr>
<td>Dresser</td>
<td>253.84</td>
<td>127.62</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>31.25</td>
</tr>
<tr>
<td>Nightstand_02</td>
<td>19.14</td>
<td>197.29</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>HDD MultiMedia</td>
<td>253.59</td>
<td>182.29</td>
<td>20.05</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>13.31</td>
</tr>
<tr>
<td>Light_01</td>
<td>24.54</td>
<td>96.29</td>
<td>27</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>11.41</td>
</tr>
<tr>
<td>TDT</td>
<td>244.94</td>
<td>198.14</td>
<td>19.93</td>
<td>0</td>
<td>90</td>
<td>-1</td>
<td>0</td>
<td>11.61</td>
</tr>
<tr>
<td>Floor</td>
<td>11.06</td>
<td>345.12</td>
<td>0</td>
<td>0</td>
<td>180</td>
<td>0</td>
<td>1</td>
<td>251.12</td>
</tr>
<tr>
<td>Walls</td>
<td>1.31</td>
<td>345.12</td>
<td>142.32</td>
<td>0</td>
<td>180</td>
<td>0</td>
<td>1</td>
<td>260.87</td>
</tr>
<tr>
<td>Door</td>
<td>1.31</td>
<td>298.19</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>1</td>
<td>0</td>
<td>41.25</td>
</tr>
</tbody>
</table>

Minimum distance between any 2 objects

In order to compute the minimum distance between any two objects, Unity [20] 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:

\[
\text{distance}(P_1, P_2, (x_0, y_0)) = \frac{|(y_2 - y_1) \cdot x_0 - (x_2 - x_1) \cdot y_0 + x_2 \cdot y_1 - y_2 \cdot x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}
\]

4.3.4 Further data integration for stochastic optimization methods

The data pre-processing steps presented above constitute an important, first stage towards transitioning the dataset representation from a rigorous specified, 3D model view to an analytical, feature-like one, adequate for serving as input to an ML model.

The system will tackle the furniture arrangement and selection problem from a stochastic optimization perspective, aiming toward professional interior design layout, achieved by analytically integrating rigorous, strict, complex interior design guidelines, acting as a fine-tune of a furniture layout. Although this approach would be “scalable enough” for the purpose of prototyping various strategies towards the goals described above, the final tool, oriented towards a real-world, customer scenario would make this solution unfeasible on its own – due to the infinitely large, high-dimensional space of possible configurations – generated by the numerous furniture piece options that could fit in a room and various types of clients’ floor plans.

Because of this bottleneck, the stochastic gradient strategy will be run on “almost” synthesized layouts, which could be crudely estimated through a CI model that natively deals with the challenge of very large search spaces. The abstract strategy pursued by
these probabilistic, generative models is to first learn, from data, underlying structures and important patterns that govern the data followed by a sampling stage, in which estimations of the layouts parameters are generated, close to the final solution, therefore limiting the space needed to be searched with the stochastic strategy proposed in this project. Accordingly, the main objective of this model is the ability to incorporate “soft”, interior design professional guidelines to drive a furnishing layout solution towards a finished, pleasant one, that often cannot be obtained with a full data-driven model, which is hardly able to capture very specific, human-related aspects that are key to the scene overall plausibility, such as: style (i.e. all chairs should have the same style), fine-tune of furniture rotations (i.e. most furniture pieces have right angles, being perpendicular on the walls), ergonomics aspects (i.e. daylight and artificial light positioning relative to the room working area), positioning and grouping of highly interdependent objects (i.e. bed and nightstands, sofa/bed with TV, table and chairs).

4.3.4.1 Selected furniture subset

The subset of furniture types that will be considered in developing the stochastic approach, and thus extracted from the dataset, represent a group of basic furniture pieces that could be present in a simple, but functional complete bedroom (and are present in most of the entities in our dataset). The types considered in the subset are: bed, nightstands, table / desk, working chair, TV and wardrobe, and the counting of each in every room in the dataset are presented in Table 2.
Table 2. Furniture occurrence, by type in the dataset.

<table>
<thead>
<tr>
<th>Room ID</th>
<th># bed</th>
<th># chair</th>
<th># table or desk</th>
<th># nightstand</th>
<th># wardrobe</th>
<th># TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

As inspired from similar works, using stochastic space search in the field of furniture layout synthesis [13, 14], a pre-learning stage on specific features that expose sensitive furniture relationships and their measurement details is required. This subchapter is dedicated to detail the process of extracting such features and the data obtained for the learning phase of the model prototype developed.

**4.3.4.2 Data features for learning furniture relationships**

We propose, inspired from [13, 14], the following details of the furniture layouts for extraction, which will be used for training the model and thus obtain realistic interior
guideline rules that govern the arrangements in our dataset:

- **Furniture positioning relative to room walls:** the distance $d_i$ and angle $\theta_i$ of an object to the “back” wall represents a key aspect of the furniture placement in general and in particular for the bedroom: most types of entities having strong rules for their positioning “against” a wall, such as: wardrobe, bed, nightstands, desks, TVs – usually implying a $0^\circ$ angle rotation to the “back wall” and very small distance. This information will contribute to optimize the positioning of certain furniture pieces which are *apriori* known to be usually placed near a wall in the bedroom, contributing to the overall sense of reality of the scene. Therefore, for each of the furniture pieces we selected before, we will extract a pair $(d_i, \theta_i)$ modeling the distance and angle of object $i$, as described above.

- **Interdependency relations:** Another compulsory set of rules that govern the overall arrangement of objects in a bedroom is represented by the interdependent furniture pieces, which achieve together certain functionalities that support human activities. For our scenario, we consider the following furniture relationships:
  - **TV – bed/sofa:** watching TV is supported by the correct alignment of a TV and a corresponding sofa / bed;
  - **desk – working chair:** a desk related activity (i.e. working at computer, reading, writing) is supported by the correct positioning of a chair and a corresponding desk (i.e. at feasible, short distance and facing each other),
  - **bed – nightstands:** the bed and the nightstands also have to be aligned – both for functionality and style, being sometimes two types of furniture so highly connected, that often results in a single piece of furniture (i.e. bed with nightstands). In our dataset, all instances have the nightstands decoupled from the bed and thus represent individual furniture pieces.

These pair relationships are representative for the considered types of objects (i.e. the furniture piece subset defined above) and will be used as prior knowledge in arranging the furniture, given the stochastic search proposed model described in detail in 5. Each relation will be represented through the relative distance and relative angle between the furniture pieces, $(d_i, \theta_i)_{\text{pair}}$

- **Hierarchy relations not represented:** the hierarchy relation, usually modeling a
parent-child link is used in interior design analytical approaches to model the “support” relation between 2 furniture pieces that are one on top of the other [13], as the case of a table with a plate, a desk with a TV. Because in the initial subset of selected furniture we do not have furniture objects placed elsewhere the ground (i.e. we note that for the case of the TV we assumed that its corresponding bounding box is projected on the ground, ignoring the shape of the supporting surface and thus behaving always as a floor-placed furniture piece).

These significant, furniture type oriented data measurements serve for learning important a priori knowledge about individual furniture pieces, compulsory for correct positioning in a bedroom scene and a final plausible layout – guided by human patterns and common sense practices.

4.3.4.3 The process and challenges of automatic furniture relations extraction

Dealing with a field still in its very early stages such as automatic interior design, no ultimate choice for a dataset comprising of a relevant set of features or 3D scenes used for CI modeling exist in the literature. Therefore, the requirement of having a dataset of bedrooms, that would be completely specified and secondly, be created (or approved) by persons, made us select a representative subset of 12 bedrooms from Google Warehouse [17], showing various personal related bedrooms (i.e. representing virtual representations of various real-world scenarios, such as: their own bedroom, a desired one, their friends, a guest room they seen, etc.).
Although the collection was depicted from a real-world scenario, which is very important for the purpose of the end-goal Interiorvista tool to be used to furnish real rooms for potential customers by presenting the client with furnishing options from various providers adapted to their room plan, the subset needed thorough corrections and augmentation because amateur-related mistakes represented blocking factors in correct extracting relevant features, as detailed in 4.3.2. Following the data cleaning and correcting stage, achieved with help and professional advice from Interiorvista, we used the new dataset of 3D files in an attempt to automatically extract the extra features needed, specific to the
model applied and the rules reinforced. Because of the complexity of the problem modeled – the scenes, depicting real world, detailed aspects of the furniture and overall room appearance, certain features for certain bedroom still needed adjustments from the standard developed routine, because of limitations such as:

- **Lack of a global reference point**: in interior design and 3D modeling it is hard to determine a global reference point in a scene. In particular, for a bedroom scene no ultimate corner can be decided, with objective, consistent criteria as the global point (i.e. in which the world coordinates should be placed in). This is also available for the furniture present in the room, to which a standard way of attributing local coordinates and express metadata is hard to define outside the interior design field. Thus, as concluded with expert in field from the Interiorvista company, each object coordinates were attached to the left side, top corner, as the object would be naturally viewed by a human person (i.e. in opposite direction to the forward vector), as shown in Figure 18. Same procedure was applied to attach the world coordinates to the scene.

- **Rotations and alignment of the furniture related to different axes**: in each case, the furniture alignment is relative to the global coordinates, implying complex analytical determination of furniture alignment. For instance, in order to compute the distance from the bed to the nightstands, given the previously extracted data, namely: object coordinates, bounding box sizes and room sizes the distances might be measured along OX or OY according to the alignment with the global coordinates.

- **Particularities and complexity of certain furniture pieces**: few examples in the dataset contain some furniture pieces that are very complex, which have to be taken into account separately (i.e. treated individually in the extraction method).

- **Uncertain bounding boxes**: due to the existing of very specific, furniture type related decorations of some furniture pieces that exceeded the size of the furniture piece, (i.e. sheets of a bed) the bounding box is hard to determine exact and no definite option exists – each having certain advantages (e.g., allowing specific measurement to be achieved correctly) and disadvantages (e.g., particular handle in the extraction algorithm to treat corner case).
In the following two sections, we present the process and particular, challenging corner cases we dealt with in the process of extracting the model-related features, in an attempt to show what degree of complexity, burden and techniques need to be involved when dealing with real world, human provided, interior design 3D scenes preprocessing for machine learning modeling.

Relative furniture positioning to the walls

As briefly described above, for each furniture piece of interest we extract the position and angle to the nearest “back” wall, (i.e. the wall opposite to the direction of the forward vector). In order to achieve this task, we consider the direction of the forward vector and its rotation relative to the world coordinates in order to determine the “back” wall.

In our scenario, given the initial dataset and the furniture pieces selected for our stochastic model, all the angles to the walls are right angles (i.e. 0°, 90°, 180°, -90°, etc.) or very close values. Therefore, the angle to the back wall would be, in every case, 0°. This is a common scenario for a bedroom and furthermore, besides pieces being perpendicular to walls, most are also positioned very close to the walls by default (bed, wardrobe, TV, sometimes desk).

In order to compute the distances, given the room coordinates can be positioned in any corner, we use the rotation of the forward vector relative to these coordinates in order to determine the correct method to compute the gap to the back wall. Four cases are represented, each for one right angle of the forward vector relative to the room coordinates:

- **0°**: if an object has the forward vector 0° rotated then the distance would be the projection on the OY, and thus the object coordinate on OY (note that each object coordinates are in the left, top corner)
- **270°**: if the object is rotated 270° (i.e. -90°) the distance would be its coordinate on OX, accordingly
- **90°**: if an object is rotated 90° this means that the “back” wall will be one that is not on the coordinate axes and thus, the wall coordinates are taken into account to
compute the distance, the formula would be:

\[ d = x_{\text{wall}} + width_{\text{wall}} - x_{\text{object}} \]

- **180°**: similar to the previous case, the wall measurements have to be taken in consideration in order to compute the distance, the formula is:

\[ d = y_{\text{wall}} - y_{\text{object}} \]

Although, at theoretical level, the data extracted in the previous phases should suffice and the extra features are supposed to be computed directly using the above formulations, a series of anomalies and corner cases occurred, resulting in the need to treat specific objects and scenes separately, or approximate the target values, by measuring directly in the 3D model, with dedicated software - Unity [20]. Such cases include:

- various amateur mistakes not detected/treated in the data correction phase
- the overall complexity of the bedroom scene and some furniture pieces – more sophisticated than the default scenario
- poor quality of the reality of room and furniture measurements: although the scenes have realistic ratios, the size of the furniture pieces are not adjusted to real ones. Such examples include: rooms of approximately 2 meters width and 1 meter length.

In the following, we detail most common anomalies and corner cases that occurred, together with the solution used, in an attempt to offer a better, detailed understanding of the effort and inability to fully automatize data extraction from 3D models that are not professionally made. We note that heavy preprocessing represents a common scenario for the computational intelligence and machine learning models applied in the interior design automation field, often resulting in manual annotation of the data (i.e. as part of the preprocessing) or user burden in feeding the model with thorough specifications:

- **Wrong positioning of world coordinates**: some rooms have the world coordinates not positioned in the corner of the room, but at a random point. Therefore, for these rooms the walls dimensions and other object coordinates were used in order to compute the correct distances. Examples are presented in Figure 19.
Not measured inner wall: A scene had an inner wall that was not measured (i.e. only the exterior walls were measured) resulting in an inability to measure the distance of the object near the wall (i.e. the TV) to the wall, Figure 20. Therefore, in order to estimate that distance, we used the positioning and dimensions of the lamp in the corner, aligned with the TV, to estimate the distance.
• **L-shape rooms, no measurements for “L” walls:** Two rooms have L-shape walls and for none measurements were done for the walls contributing to the “L”. In one case, the room does not have any relevant object near those walls and in the second case there was one drawer to which an exact approximation was hard to made, due to the fact that no other object would relate to the wardrobe position. Given that the room had two wardrobes, of which one could be measured, we considered that this wardrobe was positioned at the same distance to the wall. The scene is presented in Figure 21.
Thick walls – width cannot be subtracted: In the 3D models, the walls are represented as one rectangle object, as opposed to each wall one object, therefore their width and length corresponding to the room width and length. This has the advantage of determining directly the room layout features, but at a cost of not knowing individual measurements of the walls. In the case each individual wall has a certain width (i.e. we call it thick wall) their width cannot be extracted and therefore they influence the correctness of the results in the third and last case of the algorithm (i.e. for 180° and 270° rotation of the forward vector). The solution is to treat these cases separately and estimate the width of the wall (i.e. using coordinates of other objects near the walls, near the global co-ordinate axes, etc.) which will then get subtracted from the obtained results. One example of such walls is presented in Figure 22.
Figure 22. Thick walls: width cannot be subtracted.

The dimensions to the wall obtained for the selected furniture pieces and extracted semi-automatically (i.e. after treating the previously presented corner cases) are presented in Table 3. The “-1” marks the absence of a certain piece of furniture. As stated above, the table with angles is not displayed because every furniture piece has a 0° rotation relative to the “back” wall.

Table 3. Distance to wall feature.

<table>
<thead>
<tr>
<th>Room ID</th>
<th># bed</th>
<th># chair</th>
<th># table or desk</th>
<th># nightstand</th>
<th># wardrobe</th>
<th># TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.3571</td>
<td>-1</td>
<td>4.2658</td>
<td>4.3571</td>
<td>3.9368</td>
<td>3.9369</td>
</tr>
<tr>
<td></td>
<td>5.4657</td>
<td>121.0394</td>
<td>8.3602</td>
<td>9.0357</td>
<td>-1</td>
<td>14.25</td>
</tr>
<tr>
<td></td>
<td>5.2560</td>
<td>-1</td>
<td>18.2400</td>
<td>5.9620</td>
<td>4.5860</td>
<td>-1</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>----</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.2882</td>
<td>-1</td>
<td>15.7284</td>
<td>16.1355</td>
<td>5.2005</td>
<td>13.6495</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.7640</td>
<td>-1</td>
<td>0.82</td>
<td>9.3870</td>
<td>7.7982</td>
<td>3.5879</td>
</tr>
<tr>
<td></td>
<td>7.8740</td>
<td>-1</td>
<td>-1</td>
<td>7.8740</td>
<td>7.8740</td>
<td>7.8740</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.8739</td>
<td>-1</td>
<td>-1</td>
<td>7.8739</td>
<td>7.8739</td>
<td>7.8739</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.1830</td>
<td>-1</td>
<td>-1</td>
<td>7.6750</td>
<td>3.9797</td>
<td>19.8879</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.1499</td>
<td>48.1650</td>
<td>-1</td>
<td>4.7269</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.4870</td>
<td>73.3220</td>
<td>4.4189</td>
<td>6.4870</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13.0610</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>10.8529</td>
<td>10.853</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>5.5810</td>
<td>-1</td>
<td>-1</td>
<td>5.6869</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Functionally dependent relations

As in the case for furniture-to-wall measurements computation, we also developed an automatic procedure for extracting the pair relations between the consider furniture pieces. Referring to the same dataset, we encounter here similar particular cases which need special case handling, presented in the following.

In terms of the angle between the interdependent furniture pieces, we note that 2 relations are of type “face to face” (e.g. TV – bed/sofa and desk – working chair) the object being oriented one facing the other and vice versa (i.e. the forward vectors form a 180° angle) and one relation is of type “aligned to each other” – the case of the bed – nightstand. In this case the forward vectors are parallel, resulting in an angle of 0° between them. We note only one exception from the rules presented, where the nightstand is rotated 90°, facing the bed. We ignore this corner case, due to consistency issue and the purpose of the data collected: analytical modeling of interior design and furniture standard practices, as opposed to individual preferences or existing possibilities – thus aiming to restrict the space of solutions toward few that respect interior design strict guidelines and regulations rather than generating new feasible, plausible possibilities of room layouts.

We present the general process to extract the distances for the three types of relations, on top of the formulations presented in the previous case, describing also the limit scenarios and the correction made in each case:

1. **TV – bed/sofa:** in order to compute the relevant distance between the TV and the bed/sofa we consider the minimum distance between the two distances from the corners of the TV bounding box face towards the bed and the segment of the bed bounding box face towards the TV. We use the well-known formulation of the distance between a point and a segment defined by 2 points to compute the two distances. The reason behind this formulation relies in the ability to treat correct the two possible cases, given that for all the models in the dataset the objects are facing each other (i.e. the angle between them is exactly 180°). The first case, in which the TV and bed are also aligned, meaning the two facing segments can be intersected by a single perpendicular, the distance computed will be the minimum,
perpendicular one. In the other case, in which they are not one in front of the other, but still have parallel facing segments of their bounding boxes, the minimum distance will be from the closest TV corner to the closest point on the bed bounding box face (i.e. the closest corner) – a feasible distance – similar to the minimum one a human would see when watching TV from the bed.

2. desk – working chair: this scenario is only present in 2 rooms of our dataset. Moreover, one room represents a corner case, where the desk is not rectangular but in L-shape and therefore positioned in the corner. Due to its positioning relative to the chair and his shape, the bounding boxes (both rectangles) of the 2 interdependent objects collide, as can be seen in Figure 23. Moreover, the chair has a -50° rotation, being positioned in the center of the desk. Therefore, the only exception of the angle statement presented above is this case, which has an exact angle of 140°. We can argue that the chair, which is a rotating one, can be easily positioned perpendicular to the desk – making this case able to be included in the general rule, of having only angles of 180° between directly facing furniture pieces. We will therefore consider an approximation of the distance and an angle of 180° for this case, to align with the other example and the general observation.
3. **bed – nightstands**: in the case of this relation, which, in contrast to the other two, is an alignment relation, the challenge to compute the distance relies in correctly determining the coordinate axis along which they are displayed. We ignore the single case where one of the two nightstands is rotated because the distance in that case would reflect the functional space needed to properly operate the nightstand rather than an alignment to the bed - serving purposes such as close enough to represent a supporting surface for objects. The distance measurement is computed with a similar strategy as adopted in wall distances, analytically determining the axes and formulation according to the forward vector rotation relative to the room coordinates, depicting similar cases to the one presented above.
We present in Table 4 the values obtained for each type of relation, for all entities in the dataset, displayed as a pair of distances and angles.

Table 4. Furniture pair relations measurements.

<table>
<thead>
<tr>
<th>Room ID</th>
<th>TV – bed</th>
<th>desk – chair</th>
<th>bed – nightstand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>distance</td>
<td>angle</td>
<td>distance</td>
</tr>
<tr>
<td>12.7088</td>
<td>12.7088</td>
<td>180</td>
<td>-</td>
</tr>
<tr>
<td>58.1303</td>
<td>58.1303</td>
<td>180</td>
<td>9.0816</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>128.1811</td>
<td>128.1811</td>
<td>180</td>
<td>-</td>
</tr>
<tr>
<td>82.7455</td>
<td>82.7455</td>
<td>180</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>60.2764</td>
<td>60.2764</td>
<td>180</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5.3395</td>
</tr>
<tr>
<td>55.738</td>
<td>55.738</td>
<td>180</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
5 Stochastic optimization for furniture synthesis

Interior design automation, despite having been around for some time in the research literature, still remains an open field in terms of providing mature, robust, scalable CI models that are capable to address this profound problem in all its complexity and provide solutions that would compete with a human expert in field (which right now would be, in a way, the gold standard in day-to-day design practice)

In more detail, and focusing on the specific problem addressed in this thesis, which is furniture selection and arrangement for a bedroom, given a particular room plan, we can outline two main trends according to which the problem was addressed.

The first would be a probabilistic approach, dealing with the infinitely large search space of possible solutions through a data-oriented approach that would first model this space probabilistically, potentially capturing the underlying, hidden patterns and interdependencies governing the data and then sampling new solutions from the learned model – obtaining directly new furnishing layouts and thus proposing a solution to the challenge of searching an infinite, high-dimensional space.

The second approach would be based on stochastic optimization modeling and it is the core interest of the current thesis. It tackles the challenge of automatic furniture arrangement from a more analytical angle, by proposing an evaluation function (i.e. energy function) and a set of steps that are used to transition the current solution to a better one (i.e. with lower energy), iteratively converging to a minimum value for the cost function. One pitfall of this procedure is getting stuck in a local best result, that is, in a local minimum, a challenge that is usually treated by defining a “reset” step, which forces the solution to “jump” from the current, convergent path, causing a spike of energy.

Accordingly, the evaluation function needs to be design to allow such transitions, from a lower energy to a higher one, with a certain probability, thus increasing the chance to avoid local minima. However, a guarantee that the algorithm converges to a global minimum cannot exist with such methods, which do not exhaustively search the space. Yet
empirically, the obtained solution may represent an “acceptable” approximation of the real solution.

A usual cost function for a room furniture arrangement is designed to assess the scene against strict, complex ergonomics and professional interior design guidelines, usually mathematically modeled on the basis of in-depth studies and rigorous principles, or learned a priori from a specialized dataset. Accordingly, a set of steps in an interior design related model, meant to drive the solution towards a best configuration, corresponding to the lowest energy (i.e. value of the evaluation function), usually defines converging steps, such as: small rotations and translations and reset steps, such as swapping furniture pieces. Remaining in the context of an infinite, high-dimensional space, an exhaustive search for the best configuration is an unfeasible task and, therefore, a common approach to optimize a current interior furniture setup against strict, a priori determined rules and regulations is achieved through a stochastic gradient based method.

Such models applied to interior design challenges are, for instance, Simulated Annealing [13] - used for automatic optimization of furniture arrangement or MCMC [14], used for suggesting options of room furnishing arrangements to amateur users along the design process, aiming towards achieving professional-like results.

As opposed to the probabilistic approach, by designing a function which combines the analytical representation of well-known interior rules and regulations and specific common sense practices with the learning of the ones that are more challenging to formulate, the results obtained from minimizing it are fine-tuned and stylish, professional looking layouts, as opposed to crude estimations, sampled probabilistically, that often yield unrealistic and “strange” solutions that need manual adjustment before presenting to the user the final version. However, this advantage of obtaining results that obey interior design criteria comes at a noticeable performance price, as stochastic methods are known to be computational-intensive, when dealing with a large search space.

In this thesis, which is an initial proof of concept for the developed methods and where we define a narrowed context (i.e., with few types of furniture pieces and only rectangular-shaped floor plans), searching this limited space is feasible. The solution cannot scale on
its own for the end goal of the product: a tool for real-world, customer usage that would deal with a large, high-dimensional space, defined by a library of numerous, varied furniture pieces and diverse shapes of bedroom plans. Therefore, in the real-world scenario, the stochastic optimization approach would only optimize “almost ready” initial furniture layouts that were roughly approximated using a probabilistic model, dealing with the infinitely large space by directly sampling the data-oriented, learnt distribution.

The initial results obtained from the probabilistic stage might often be ambiguous and unable to capture sensitive, but important aspects of furnishing specific guidelines. They could then be reinforced with strong interior design guidelines, through an iterative stochastic optimization process, which now becomes feasible, given the drastic reduction of searchable space, due to the initial probabilistic approximation of the solution.

5.1 Learning furniture arrangement relations

One step towards achieving realistic rules that govern the furniture arrangement is to obtain the measurement details regarding such arrangement. Each furniture piece has its positioning particularities in a bedroom scene, which will be modeled in this approach through two types of relations, as inspired from related work in the field [13]:

- **Furniture positioning relative to the walls**: represented through a pair of distance and angle relative to the wall, for each type of object modeled. According to related interior design case studies and as discussed with Interiorvista experts for the particular scene of a bedroom, the furniture positioning relative to the walls is very important, and hard rules apply, such as: a bed will always be near a wall; the same applies to the wardrobe: usually the angle towards the furniture is a right-angle and particular furniture pieces need to be parallel to the walls in order to appear “properly arranged” – as viewed by a person.

- **Functionally dependent relationships**: also represented through a pair of distance and angle, this time relative to each other, this relation is important to model for furniture pieces that should be grouped together from a functional or style
perspective. Some of the modeled relations between the furniture pieces that we considered include: a bed with its corresponding nightstands – a style and functional grouping; a TV with corresponding bed/sofa – functional grouping and desk with working chair – supporting specific activities such as writing, operating a computer, reading, etc.

In order to provide a general idea and comparison metrics for the obtained relations measurements, we present in Table 5 the results of an interior design case study, depicting the ideal intervals of common space between certain common furniture types and estimated accessible space around them (i.e. named “clearance space”), modeled also in our project through the functionally dependent relations. We note that these measurements do not take into account any other information regarding the furniture type (i.e. size, style, doors type, etc.), thus lacking the ability to estimate realistic measurements to more specific furniture pieces. A main advantage of a model that automatically learnt these measurements would be the ability to obtain multiple measurements for different subtypes of same type of furniture pieces, by providing the model a representative training subset of scenes containing that specific category (e.g., wardrobes with sliding doors vs. wardrobes with normal doors, needing obviously different accessible space in front).

Table 5. Interior design standard metrics for common bedroom furniture relations [21, 14]

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Distance in/cm</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bedside</td>
<td>36</td>
<td>91.44</td>
</tr>
<tr>
<td>Seat</td>
<td>30</td>
<td>76.2</td>
</tr>
<tr>
<td>Cabinet and shelving</td>
<td>24</td>
<td>60.96</td>
</tr>
<tr>
<td>Dining table</td>
<td>36</td>
<td>91.44</td>
</tr>
<tr>
<td>Coffee table to seat</td>
<td>16-18</td>
<td>40.64 - 45.72</td>
</tr>
<tr>
<td>End table to seat</td>
<td>0-12</td>
<td>0 - 30.48</td>
</tr>
<tr>
<td>Nightstand to bed</td>
<td>0-12</td>
<td>0 - 30.48</td>
</tr>
</tbody>
</table>
5.1.1 Wall relative positioning estimation

Although numerous statistics exist for our particular case of limited available subset of furniture that precompute the ideal distance and angle for certain furniture pieces, we employ a learning method, based on clustering, to estimate, in a data-oriented manner, the best wall-relative metrics for each object. By doing this, we contribute a method for automatically synthesizing the rules for other new furniture objects that might lack certain such guidelines, or that were too complex to fit in an easy-to-determine category.

Moreover, and according to experts in field, walls represent a strong guideline for a user to position and think of furnishing options. Therefore, strong correlation between same type of furniture pieces relative positioning can be captured in a learning approach, and estimated accordingly. Following the model estimation, the pair values for each type of furniture learned are then included as factors in a cost function, aiming to reinforce the measurements in the scene currently optimized – towards expert arrangements. We will explore this further in the part of the thesis concerned with the design of the cost function.

5.1.1.1 Clustering

In order to correctly predict the value of the distance of each furniture piece relative to its back wall, we apply a clustering algorithm and we consider the centroid of the formed clusters as the estimations of the wall relative measurements for each furniture piece. Using the clustering approach, we first hope to better understand the consistency and structure of the measurements both – for each furniture piece and overall.

Table 3 depicts the measurements obtained for the subset of selected furniture types in our prototype (i.e. the name of the columns). We note that a value of -1 marks the absence of certain furniture in a bedroom scene and thus the lack of the corresponding measurement. A graphical view in 2D of the points, each color representing a type of furniture is shown in Figure 24. We note that a random deviation on the OY coordinate was introduced for each point in order to better view the points in the plane. This serves
only to ease the visualization. In the algorithms, the points were represented in 1D (as distance measurements).

**Figure 24. Measurements relative to the wall for the subset of selected furniture**

We observe that the measurements of furniture distances relative to the back wall are very similar across various types of objects. This indicates that the setting of a furniture piece relative to the wall is highly dependent on the type of action (e.g., a user positioning the furniture near the wall) rather than on the type of objects positioned (e.g., a user positions in the same manner a nightstand, a bed or a wardrobe relative to the wall).

We notice an outlier furniture piece that represents the working chair measurement to the back wall. We expected it to be an outlier because in our model the chairs are positioned relative to the desk and not to the wall. In these particular cases, the system can indicate
when a certain dependency is not appropriate for the linked furniture object – as is the case of measuring the back wall relative positioning for a chair, by measuring the mean and standard deviation of the measurements performed and comparing the standard deviation against a predetermined “consistency threshold” – which can be computed as the mean (or max) of the well-known dependencies, or by applying an outlier detection strategy.

in Figure 25, we represent the mean (values on the OX axis) and standard deviation (values on the OY axis) of the groups of measurements taken in this scenario. We notice that the standard deviation in the case of the chair relative to the wall, a non-existing dependency, has a much higher value, indicating the inconsistency of the measurement. This option can be used to trigger warnings for the tool administrator for new added dependencies that are not correlated to the data.

Figure 25. Mean and standard deviation of the relative to wall measurements
K-means with K-means++ initialization

In order to obtain the best estimate value for the relative measurements we applied a clustering strategy inspired on the work in [13]. In contrast to this study, we analyze *apriori* if the measurements are correlated with certain (i.e., few-to-all) types of furniture pieces, or are dependent on the type of action, as in this case, where the back to wall measurement is similar across different types of objects to which this constraint is applied. In consequence, we will generate the default measurements for the respective terms of the scoring function based on this criterion.

We apply a variation of the well-known, simple clustering algorithm K-means, with a KMeans++ initialization, called *MiniBatchKMeans* that improves the overall performance of the model by using batches of samples. The implementation used in our scenario is taken from *Scikit-learn* [22], a *Python* library.

In order to further prove the correlations of the measurements across different types of furniture pieces, we run the model using a number of clusters equal to the number of furniture types, presented in Figure 26. This provides us with a visual representation of the obtained clusters, and we can observe that the formed ones are rather unnatural.
We consolidate this hypothesis further by running an algorithm that estimates the correct number of clusters based on the Silhouette Score [23] – a popular measurement of a cluster structure scoring, based on inner, cluster density and overall, exterior density. The function used to run and plot the number of clusters analysis is inspired from Scikit-learn collection [24], with small adaptations to accept 1D points. A visual interpretation of the cluster scoring is provided in Figure 27. As anticipated, the best Silhouette Score is obtained for the case of 2 clusters, with a value of 0.90 compared to alternative splits consisting of 3, 4, and 5 clusters respectively, which achieve a maximum of 0.68. We also note the decreasing tendency of the scoring, which shows that the points rather form one
big cluster, integrating all the correct measurements of relative wall positioning and a second clusters containing only points from the chair relative positioning to the wall – a dependency that is not present in the data, but used here to outline the power of the model to detect such cases.

Figure 27. Cluster performance for different nr. of clusters for wall relative measurements

('For n_clusters =', 2, 'The average silhouette_score is :', 0.904)

('For n_clusters =', 3, 'The average silhouette_score is :', 0.685)
The results obtained by the model for the best number of clusters (two) are presented in Figure 28. We remind that in terms of angles, the measurements were consistent 0° to the “back” wall so no further analysis was needed. In case of variations, the same procedure for the angles would be applied to estimate the default angle to sue in the scoring function.
Therefore, in our prototype we will consider the value of “8.62” for the default measurement between the furniture and the walls, representing the centroid for the large cluster – corresponding to the values obtained for all furniture pieces from the subset of furniture selected except chair – the only furniture piece that does not show such dependency.

5.1.2 Interdependent positioning estimation

Although rigorous measurements and defaults between common, interdependent furniture
pieces, such as the ones proposed in this prototype, namely: TV – bed, desk – working chair, bed – nightstands, have been widely investigated and used in interior design, the advantage of proposing a model that would automatically learn and approximate such measurement directly from a dataset (i.e. made from professional designed bedroom scenes) presents important advantages, such as:

- **easily obtaining new interdependencies**: For new added dependencies, either between already existing furniture pieces or new ones, the system can automatically learn and predict the new measurement for the desired new relationship. In more detail, the model can automatically provide the best measurements, learned from data, by training on an appropriate dataset that contains the new furniture pieces between which the pair measurements need to be taken. Moreover, the model is able to outline if such dependency would really exist by outputting the degree of data correlation – based on inner representation.

- **adjusting measurements for a specific scenario**: another advantage of learning the furniture linkage details is the ability to address certain scenarios separately, in contrast to being forced to manually approximate a known, general, interior design metric, as the ones presented in Table 5. For instance, if certain information about the rooms or the furniture pieces were known, the model could learn specific measurement for all or a part of the rules to emulate that exact scenario, such as: luxury bedrooms, a house bedroom versus an apartment bedroom – which can be larger, or contain more lighting sources, etc.

- **automatically treating special cases of furniture types**: as presented in the case of a corner desk Figure 23, in certain furniture pieces have unusual, different properties (shape, functionality, size, positioning restrictions) than the standard shapes. Therefore, the interior guidelines should adapt the rules for such objects by providing adapted measurements. A ML system would deal with this task in a natural manner, by providing an appropriate dataset with various scenarios of usage of such certain subtype and compute the specific measurements, resulting at a global level in more robust, highly customizable furniture rules and regulations, while maintaining the overall rigorousness of interior design protocols.
5.1.2.1 Clustering

As briefly described before, we model a representative subset of relations between the furniture pieces considered in this initial prototype, represented through distance and angle between the pairing objects, namely:

- **Bed – nightstands dependency**: composing of the estimated angle and distance between the bed and the nightstands
- **TV – bed / sofa relation**: the angle and the distance between the two furniture pieces
- **Desk – working chair**: same measurements as above, in order to enable the functional activity of the group (i.e. working, reading, writing).

The measurements extracted for these furniture links are represented in Table 4. We take each pair in part in order to further analyze it, arguing the accuracy and trust of each given the limited available dataset for this initial scenario and moreover, the limited number of specific furniture units present in the bedrooms. This set of binary dependencies is a representative one, proposed for the modeled scenario, which can be further extended when applied in the end goal case of a real world, commercial oriented tool for interior scene synthesis.

The model key advantage over a static, manual design one is the ability to automatically depict, in a data-driven manner, various measurement for new interdependencies or for the already existing ones between new types of furniture. These new rules can be then integrating in the scoring function, thus increasing the accuracy of evaluating the layouts and thus providing first a powerful ranking tool for professional interior design layouts and second better accuracy of the stochastic gradient approach for automatic furnishing layouts synthesis, resulting in nicer, more complex, specific and exact bedroom scenes.

**Bed – nightstands**

In order to analyze the dependency and the way it variates (i.e. across nightstands or rooms) we plot the measurements obtained in Figure 29. As in the previous plotting
scenario, we introduce a random variation in the $OY$ coordinates thus extending the points from 1D to 2D for the purpose of clearer view of the point values. These $OY$ coordinates are only used for easing visualization, thus not contributing in any way to the model. As depicted in Figure 29, where the red color represents the bed right-hand side nightstands and the blue the opposite ones, there is no measurement correlation between a certain type of position (i.e. left of right) and the bed thus we will treat all the nightstands as similar objects.

**Figure 29. Bed - nightstands distance measurements**

As in the previous case, we run the Silhouette score-based algorithm to analyze the clustering performance, and its results are outlined in
Figure 30. Given only one type of furniture piece included and a limited number of points for it, our aim would be obtaining a lower bound for the preferred number of clusters, ideally containing a main one – with common used distance measurements and others (a few) containing atypical cases or outliers.

Figure 30. Cluster performance for different no. of clusters for bed – nightstand dependency

('For n_clusters =', 2, ', 'The average silhouette_score is :', 0.732)

Silhouette analysis for KMeans clustering on sample data with n_clusters = 2

The visualization of the clustered data.

('For n_clusters =', 3, ', 'The average silhouette_score is :', 0.685)

Silhouette analysis for KMeans clustering on sample data with n_clusters = 3

The visualization of the clustered data.
As the analysis indicates, we used 2 clusters and extracted the centroid value for the most populated one, resulting in a value of 0.72 for the default distance between the nightstands and the bed. The obtained clusters in this best case are presented in Figure 31, outlining a dominant cluster, and a possible outlier one.

**Figure 31.** Best clustering results for the nightstand–bed dependency
TV – bed / sofa

In our dataset, the TV is correlated only with beds, as can be extracted from the relative positioning and facing angle between the two corresponding furniture pieces. From the 12 available scenes, only 6 have a TV in the room, thus resulting in a very limited subset of points, presented in Figure 32, using the same trick as before to represent 1D points in 2D. This does not limit the purpose of the proof-of-concept detailed in this thesis because the analysis and modeling of any type of binary relation would be similar to the ones presented above (i.e. the bed – nightstands and the relative positioning) – all these relations being represented as a pair of angle and distance, between two entities. Therefore, we consider the mean of the values for this pair, namely 66.29. As depicted in Table 4, the angles are equal, in all cases, to $180^\circ$ - representing the facing position.

Figure 32. TV – bed/sofa distance measurements
Desk – working chair

Similar to the previous case, our limited dataset only contains 2 desks associated with chairs, one of which being the corner case of the L-shape scenario, as seen in Figure 23. Therefore, in order to further use this term in defining the scoring function, we will take the second measurement, namely: 9.08 distance and 180° angle.

5.2 Measurements estimations review

As a summary of the previous section, we present in Table 6 the default measurement values we concluded to use in the bedroom interior layout evaluation function, and we compare our results, based on a data-oriented automatic procedure with the values obtained in the interior design case study, presented in Table 6.
Table 6. Learnt metrics comparison against an interior design case study values [21]

<table>
<thead>
<tr>
<th>Relation ID</th>
<th>Learned metrics (in, °)</th>
<th>Case study metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall positioning</td>
<td>8.62</td>
<td>0</td>
</tr>
<tr>
<td>Bed – nightstands</td>
<td>0.72</td>
<td>0</td>
</tr>
<tr>
<td>TV – bed</td>
<td>66.29</td>
<td>180</td>
</tr>
<tr>
<td>Desk – chair</td>
<td>9.08</td>
<td>180</td>
</tr>
</tbody>
</table>

As a verification criterion, we note that the learned values are situated within the interval provided by the authors in the interior design study. Moreover, in our case, the learned value accurately emulates the practices and rules used in furnishing the bedroom scenes in the dataset. When extended to the end goal scenario of the tool, addressing various types of furniture I various styles of rooms, the learning phase of the scoring function will be able to adapt to such scenario providing accurate scoring specific to certain types of variations (i.e. furnishing functionality, sophisticated room plan, corner case furniture types).

5.3 Simulated annealing

The stochastic optimization model we use for furniture arrangement is Simulated Annealing [25, 26], an algorithm that is inspired on the thermodynamics of the process of heating a metal, followed by a slow cooling down. In more detail, the algorithm mimics the process of slow metal cooldown, in which case the atoms start slowly settling down, moving more towards stable positions than to new, chaotic ones as the temperature decreases. Using the same technique, the Simulated Annealing algorithm starts at a “high temperature”, where the algorithm has a high probability of accepting new states that have greater energy than the current one, thus doing more exploration of the solution space in the initial part that contributes, probabilistically, to a lower chance of the method getting stuck in local minima. As the temperature decreases, the probability of accepting worse
states (i.e. with higher energy) also decreases and the previously chaotic jumps transform into smoother ones, towards lower energy states, thus converging towards a local minimum.

Enforcing this strategy, with dominant exploration in early iterations substituted in time with more exploitation and conversion as the temperature decreases, Simulated Annealing has proved to be a reliable general approximation technique when dealing with a large search space, finding “acceptable” approximations of the global minimum in a timely manner. In general, accepting worse states for an optimization model is a fundamental property that drives the process towards a more extensive search of the solution space and thus helping avoid local minima. Because of these qualities, this strategy was applied in literature to problems that provide the exact solution in exponential time, like the Traveling Salesman Problem (TSP) – known to be NP-complete and obtained “good enough” approximations of the global optimum.

Later the algorithm has also been applied to the field of design, first to architectural design [11] for floor plan generation and then to interior design for furniture synthesis [13].

5.3.1 Metropolis criterion

Inspired by the early Metropolis-Hasting algorithm [27], Simulated Annealing is often used together with the Metropolis criterion formulation for the acceptance probability. In the Metropolis algorithm, also used in our Simulated Annealing implementation, a new state with lower or equal probability is always accepted and a “wilder” one, having a higher energy is accepted with the probability:

$$P(e, e', T) = \begin{cases} 
1 & \text{if } e' \leq e \\
\exp\left(-\frac{(e' - e)}{T}\right) & \text{otherwise}
\end{cases}$$

We used the following implementation of Simulated Annealing with Metropolis criterion, taken from [28], which was provided as a Python library, suitable to the framework 90.
(iPython [29]) used in our implementation. The acceptance probability for “worse” states was modeled in our scenario, in Python, as follows:

\[
\text{math. exp}(-dE / T) < \text{random.random()}
\]

### 5.3.2 Steps proposed

As with any stochastic gradient method, a set of steps needs to be proposed, that would move the current solution to a new solution. This should contain two types of steps: convergence steps – that are usually small variations of the current solution, aiming towards driving it to the local minima; and “reset” steps – that move the solution to a new, usually higher state, often causing a “spike” of energy. At a more abstract level, the “restart” steps are meant to increase the searched space trying to avoid getting stuck in local minima by producing new solutions further away from the current solution (i.e. often seen as “jumps” in the search space).

As inspired by related work tacking furniture arrangement automatization using gradient-based methods [13, 14], we define the following set of steps for our prototype:

- **2D translation**: this move modifies the position of a certain furniture piece in scene with a random pair of values \((\delta x, \delta y)\), sampled from a normal Gaussian distribution. The furniture piece to which the move is applied is chosen randomly in the scene, at each iteration. The move can be formally described as: \((x_i, y_i) \rightarrow (x_i + \delta x, y_i + \delta y)\)

- **Right angle rotation**: given the initial modeled scenario – a simplistic, complete furnished bedroom with basic furniture pieces, all furniture types included in the scenario modeled (as is also the case of most of the ones in the dataset considered) are perpendicularly aligned with the walls. This is a common scenario for a bedroom, where most furniture types are usually placed against the walls (i.e. bed, desk, wardrobe, nightstands). Therefore, the angles learned from the dataset are, accordingly, only right angles, relative to the global, room coordinates (e.g. 0°, 90°, 180°, 270°/-90°) and we model the possible rotation in terms or right angle rotations (i.e. +90° or -90°) relative to the current position.

- **Swapping objects**: this “wild” step is commonly used in related works [13] as the
“reset” step, for driving the solution away from its current convergence path and explore new (further) areas of the search space, thus encouraging rapid exploration if the large space. We also define this step in our implementation, which simply exchanges the position coordinates of two objects. We do not impose any restriction of this move (i.e. required available space for the swap to be done without collision with other objects, swap only similar objects, no wall collision) because all these are further resolved in the following exploitation process, in which the energy is lowered and thus the objects are repositioned in a correct manner – reinforced by the cost function factors.

5.4 Evaluation function (learning-based scoring function)

The cost function is used to attribute a score to a certain furniture layout reflecting how well the furniture pieces are arranged in the room. Our scoring function, representing one of the contributions of this thesis, encapsulates rigorous interior design guidelines and regulations that helps drive a furniture arrangement layout towards a plausible, pleasant, professional-looking one.

5.4.1 Advantages of automatic parametrization using machine learning

In order to provide a highly customizable, automatically adaptive furniture layout evaluation function we incorporate in the mathematical representation of the factors contributing to the evaluation automatically learned measurements for variously defined relations among interdependent entities and general guidelines of their relative positioning in the room (i.e. distance and angle to the “back” wall). Being learned from an adequate dataset – composed by interior room layouts designed by experts in field, these measurements can model specific scenarios – resulting in highly specialized rules that adapt to various, complex scenarios – offering significant advantages over the traditional approach, in which these measurements, computed in interior design related study cases, were static, addressing only general types of furniture objects (i.e. bed, wardrobe, desk, etc.), as
shown in Table 5.

Among the most important advantaged of an energy function parametrized automatically through ML, we include:

- **easily adding new furniture relations**: aiming towards the end goal scenario of the tool – showing and selling furniture in a real world, customer oriented scenario, the model will deal with a very large collection of furniture pieces. Although, in the context of this prototype few, representative relations are enforced for the considered entities, the ability to scale up, in an automatically manner is of key importance. Therefore, the learning of various measurements and interdependencies between new furniture pieces, which can be integrated in new created furniture relationships formulations, represent a leading advantage of the learning based evaluation function over a classic approach one – providing an automatic way to depict, directly from data, measurements consistent to rigorous, up-to-data interior guidelines and.

- **learning measurements for a specific set of room scenes**: Another important aspect of the Interiorvista tool relies in its appliance to furnish various types of rooms, given clients specific floor plan. In this scenario, scaling to significant number of rooms, we can depict categories of bedrooms, such as: luxury bedrooms, apartment bedroom, house bedroom, child bedroom, couple bedroom, which might influence the values of parameters learned across all categories of bedrooms. For instance, a child room might usually have no space between the nightstand and the bed in order to not fall in-between when playing, a small room might have object more close to each other as opposed to a luxury room enlarging spaces between furniture. Therefore, in order to maintain the accuracy of the interior furnishing scoring function to various categories of rooms the parameter learning phase can be applied to a subset of a certain category of rooms in the dataset, thus providing customized measurements for the target type of bedroom and increase the scoring function accuracy, in an automatic manner. Moreover, a higher accuracy of the scoring function would result in a boost of performance of the gradient optimization approach, producing better, category-specific results.
• automatically adapting to sophisticated furniture types: given the overall complexity of the furniture objects in general, customization of the scoring function is needed also at an object level, providing specific scoring for meaningful subcategories of furniture pieces. A relevant example, present in our dataset, is the L-shape desk represented in Figure 23 – which is totally different to a normal desk, thus implying other ergonomic and guidelines measurements (i.e. should always be placed in a corner). In case of real world furnishing solutions, considering only the basic types of categories is not enough, and therefore, subcategories of the main categories need to be addressed accordingly (i.e. an L-shape wardrobe will not obey the same guideline constraints as a normal, wall-side wardrobe). The automatic manner in which the measurements are computed for various defined guidelines would provide a direct link to deal with these subcategories – the accuracy of the layout evaluation function can be increased by learning the sophisticated furniture relations details a priori for the desired, representative categories.

5.4.2 Proposed furniture guidelines

For the prototype implemented in the model, we propose a subset of interior design guidelines, inspiring from related work of applying stochastic modeling to the automatic furniture synthesis challenge [13, 14]. These guidelines, representative for the modeled case: a simplistic, but complete functional furnished bedroom with basic furnishing, can easily be extended once the complexity increases to real world appliance scenario.

Adding new guidelines

Adding new guidelines or enforcing existing ones between new types of furniture will only imply analytically representing the new guideline (as a term, in the cost function), or defining the objects between it, would imply the task of estimating the equation parameter automatically by training the parameter estimation model on a representative subset of bedroom solutions that would contain the furniture pieces involved in the new designed guideline.

In the following subsections, we detail the subset of guidelines modeled in our prototype.
implementation. The final cost function is defined as a weighted sum, each guideline representing a weighted factor. The weights are determined empirically in our project and their values are presented in Table 7.

Table 7. Weights associated with the cost factors, in the scoring function, determined empirically.

<table>
<thead>
<tr>
<th>Cost term ID</th>
<th>Weight value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall relative positioning</td>
<td>0.03</td>
</tr>
<tr>
<td>Wall relative angles</td>
<td>10</td>
</tr>
<tr>
<td>Bed – nightstands</td>
<td>0.1</td>
</tr>
<tr>
<td>Bed – TV</td>
<td>0.1</td>
</tr>
<tr>
<td>Desk – working chair</td>
<td>0.1</td>
</tr>
<tr>
<td>Collision (with walls and other objects)</td>
<td>15</td>
</tr>
</tbody>
</table>

5.4.2.1 Room walls collision

The cost term measuring the room walls collision is given by the following formula:

\[
\text{cost}_{\text{wall},i}(\emptyset) = \begin{cases} 
0 & \text{if } p_i \in \text{space}_{\text{room}} \\
\text{dist}_{\text{min}}(p_i, \text{wall}_\text{space}) & \text{if } p_i \notin \text{space}_{\text{room}}
\end{cases}
\]

\[
\text{cost}_{\text{wall}}(\emptyset) = \sum_i \text{cost}_{\text{wall},i}(\emptyset)
\]

The wall collision cost is given by the minimum distance between the object coordinates and the room space coordinates if the object is not inside the room and 0 otherwise. We note that the object coordinates are represented as the upper left corner of the object, identical to the case of the handling the objects in the dataset.

5.4.2.2 Accessibility distance

The accessibility distance cost term penalizes furniture objects that are too close to each
other and thus limiting their accessibility and functionality. Inspired from [13], we predefine in our experiments an accessibility space for each object, according to its size and its functionality in a basic scenario. In contrast – when addressing a real world scenario, the accessibility cost will be a well-known value specific to each object, provided by the furniture piece manufacturer, a default term being unable to model specific particularities that would heavily influence the accessibility space of each object, such as: a wardrobe with a sliding door needing less accessibility space than a wardrobe with a normal swing door. However, the predefined values do not influence the correctness or validity of the proof of concept in this project, the cost term for the accessibility between 2 objects, $i$ and $j$, having the following formulation:

$$
cost_{acc,i}(\emptyset) = \sum_j \max(0, 1 - \frac{|ca_i - c_j|}{ad_i + b_j})
$$

Where $ca_i$ is the center of the accessibility box of the object $i$ (i.e. bounding box plus accessibility space around it), $c_j$ is the center of object $j$, $ad_i$ is the semi-diagonal of the accessibility box of object $i$ and $b_j$ is the semi-diagonal of the bounding box of object $j$.

$$
cost_{acc}(\emptyset) = \sum_i cost_{acc,i}(\emptyset)
$$

We include also the collision with the walls in the accessibility term – thus assuring the objects are not functionally obstructed by any of the environment entities (i.e. other furniture pieces and room walls).

**5.4.2.3 Wall relative positioning**

Inspired on [13], the enforcement of the *a priori* computed furniture relationships represents a cost factor in the scoring function as well. As described in the previous sections, each inter-dependency is represented through a pair of (angle, distance) that is learnt for each object in particular. The only exception is made for the case of the wall relative positioning, where we determined that this relation is highly dependent on the concept itself (i.e. putting a furniture piece near a wall) more than on the furniture type, the
latter being unrelated to the constraint and thus having only one pair of \((d_{\text{wall}}, \theta_{\text{wall}})\) for all furniture pieces. The cost term is expressed mathematically as follows:

\[
cost_{\text{dist}}^\varepsilon(\emptyset) = \sum_i ||d_i - d_{\text{wall}}||
\]

\[
cost_{\text{acc}}^\theta(\emptyset) = ||\theta_i - \theta_{\text{wall}}||
\]

### 5.4.2.4 Furniture interdependencies

The furniture interdependencies encode the natural, fine tuning positioning between objects that form functional groups, thus supporting human activities, such as: a TV – bed/sofa pair supports watching television, a desk – chair pair supports working activities such as writing, reading, using a laptop, etc. These pair-defined relations are compulsory for obtaining well arranged, plausible furnished bedrooms and are often hard to capture in a full data-driven manner. They are thus usually present in a stochastic optimization refinement step, following the crude sampling of the learned probabilistic distribution.

In our model, these relations are all represented through the same mathematical formulation. For instance, considering the defined formulation above, one can replace the “wall” object with the pair corresponding object of a certain furniture piece. We define in the following the analytical representation of the bed – nightstands dependency, following that the others can be defined in the same manner:

\[
cost_{\text{bn}}^\varepsilon(\emptyset) = ||d_b - d_n||
\]

\[
cost_{\text{bn}}^\theta(\emptyset) = ||\theta_b - \theta_n||
\]

In this transparent manner, the definition of new furniture linkage is carried out semi-automatically, enforced by the pre-learning stage of a best estimation of the parameters used in the new cost terms – all towards obtaining, in an easy manner, a more accurate, highly customizable and specific scoring function that would realistically grade various
types of scenarios, containing diverse furniture types. These terms grouping, each enforcing particular aspects in a complex, professional designed scene, contributes towards enforcing strong and varied natural affinities from various areas such as ergonomics, style, functionality, etc.

5.4.2.5 Main pathway clearance

We introduce a term in the cost function to define the clearance of the main access corridors in the rooms that will enable rapid, direct access to the main furniture pieces. This is different from the minimum access distance of each piece of furniture in the sense that it reinforces that each object is accessed fast, and that the path from any part of the room to it is in general a straight one. Also, this particular cost term aims at maximizing the doors clearance and the space nearby.

In our model, the pathway clearance is statically defined as a certain surface in the room which should remain, in principle, not covered by furniture pieces, thus having a small weight associated in the energy function (as compared to the other more relevant factors). The area expands near the door (or doors, in more complex scenes) and all the way through the center of the room, which, according to numerous studies and as advised by experts in interior design at Interiorvista company, tends to remain unfurnished, if possible, to maximize the free movement aspects of a bedroom.

We note that most (all but the chair) furniture pieces are enforced to converge to a position close to the wall, through the relative-to-wall relation. Therefore, we neglect this factor in this implementation, because its purpose represents, in this simplistic scenario, a consequence of the combination of the other cost terms.

5.5 Model implementation

5.5.1 Framework used

In our implementation, we used Python2.7 [30] to implement the specific model related features extraction and to implement the prototype. On top, we run the model as a Jupyter
Notebook [29]. Along the implementation, we used Python popular choice libraries, such as Numpy [31], SciPy [32], math for their already implemented basic functions and routines and Matplotlib [33] for plotting the charts and graphics.

Pygame for drawing the scene

In order to obtain a 2D graphical interpretation of the obtained results we used Pygame [34], a popular Python choice for implementing simplistic games, which provides a library of functions for graphics manipulation such as: scale, rotate and transition. We used this library only in static mode, by feeding it a state to be plotted so as to analyze the furniture arrangement. An online mechanism could also be easily coded to view the furniture room scene updates in real-time.

5.5.2 Context details

For a basic bedroom represented in 2D, in which common furniture pieces are included, the used measurements and details are presented in Table 8.

Table 8. Furniture size details used in the implemented scenario

<table>
<thead>
<tr>
<th>Furniture type</th>
<th>Width</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>Wardrobe</td>
<td>300</td>
<td>150</td>
</tr>
<tr>
<td>Bed</td>
<td>300</td>
<td>250</td>
</tr>
<tr>
<td>Nightstand</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Desk</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Chair</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>TV</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

In the system, each furniture piece is represented as an entity through its ID, coordinates, size and rotation, as shown in the following formula:

\[ obj_i = (ID, x_i, y_i, w_i, l_i, \theta_i) \]
In order to appropriately plot the furniture on a 2D scene, aside from the actual floor space it occupies, we use real pictures of the furniture objects, as presented in Figure 33.

Figure 33. Images of the objects used in showing the stochastic procedure results
5.5.3 Scoring function implementation details

**Minimum distance between two rectangles**

In order to compute the distance between two rectangles (i.e. the 2D floor projections of the furniture pieces), we take advantage of the constraint that all furniture pieces can be rotated with only right angles, thus remaining parallel with the walls throughout the implementation. Therefore, we compute the minimum distance between two rectangles using the well-known formula of the distance between a point and a segment, and apply it between all the vertices of an object and the edges of the other object, followed by the same procedure for the other object.

We apply the same procedure for computing the distance of an object to the walls, representing the room walls as an exterior, larger object “supposedly” incorporating the one to which the measurement is made. We note that this is not always the case, given the fact that during the optimization process the object move freely in the 2D space, often
ending up in collision (or outside) of the room walls.

5.5.4 Steps implementation

In order to implement the steps, namely the moving / positioning of a furniture piece in the scene, or the rotation of a certain furniture piece, we apply the following procedure:

At each step:

1. Select a furniture piece at random (from the furniture pieces in the scene), by drawing from a “discrete uniform” distribution, using the \texttt{randint} function from the random package in Python [35]

2. Probabilistically choose the type of move (e.g., positioning vs. rotation, each having a 50% chance of being selected)

3. a) if a positioning step is selected: draw from a normal distribution, with zero mean and a standard deviation of 10. We set the mean to zero in order to draw positive and negative values to move the furniture in all directions with equal probability. The chosen standard deviation was decided empirically given the ratios and sizes of the furniture pieces in the room and the bedroom size.

   b) if an angle rotation step is drawn, the furniture piece that got selected is rotated by 90°, clockwise or counterclockwise with equal probability

**Energy function formulation**

In the implemented stochastic gradient procedure, the value of the energy of a state is given by the cost function defined in section 5.4, defined as a weighted sum over the factors presented. We set the weight values empirically, according to the context and magnitude of the cost factors – in order to obtain an optimum balance between the enforced guidelines. The set of weights used in our scenario were presented in Table 7. Given that the stochastic optimization procedure using the Metropolis criterion requires a Boltzmann-like energy function, the energy term is integrated together with the temperature in the Simulated Annealing implementation into the final formulation, as theoretical detailed in 5.3.1.
Model parameters

The Simulated Annealing algorithm, as most stochastic optimization methods, has a set of parameters that provide the option of fine-tuning the model to provide more accurate results for the specific problem modeled. In our implementation we varied the following set of parameters:

- *temperature interval* – given through the maximum and minimum values, which indicates the starting and stopping decreasing values throughout a convergence process.
- *number of steps* – equal to the number of model iterations.
- *number of updates printed* – the total number of steps outputted during an iterative process, indicating model performance, including: acceptance rate, current temperature, current energy value, time elapsed in the current iteration, solution improvement, and estimated remaining time

The values of the parameters used in our implementation are detailed in Table 9.

Table 9. Simulated Annealing parameter values in our implementation.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum temperature</td>
<td>3500</td>
</tr>
<tr>
<td>Maximum temperature</td>
<td>1</td>
</tr>
<tr>
<td>Number of steps</td>
<td>100000</td>
</tr>
<tr>
<td>Number of updates printed</td>
<td>100</td>
</tr>
</tbody>
</table>

State representation and initial preprocessing

As described above, the state in the model is represented through a list of furniture pieces,
represented through:

- \((x, y)\): the 2D position in the room, taken as in the dataset in the left upper corner of the object.
- \((\text{width, length})\): the 2D dimensions of the object (i.e. the floor space occupied in the room)
- \(\theta\): the rotation angle, relative to the walls. The possible values in the scenario were only based on right angle steps, namely: \(0^\circ, 90^\circ, 180^\circ\) and \(270^\circ\) (i.e. furniture rotation that would move the objects only in positions parallel to the walls)

Room dimensions were also provided as an input to the model.

No manual pre-processing was needed in the implementation, except for the common practice of randomly initializing the furniture-related values, in this case the positions and angles, contributing to a faster convergence and initial exploration of the searched space.

### 5.6 Model performance and running measurements

Total running time of the model, for a fixed 100,000 iterations is presented in Table 10. The system was run on a laptop with Intel Core i5 @ 2.53GHz with 4GB of available RAM.

<table>
<thead>
<tr>
<th>Simulation description</th>
<th>No. of iterations</th>
<th>No of furniture pieces</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only collision constraints</td>
<td>100000</td>
<td>7</td>
<td>1:03 minutes</td>
</tr>
<tr>
<td>Collision + Furniture interdependencies</td>
<td>100000</td>
<td>7</td>
<td>1:48 minutes</td>
</tr>
<tr>
<td>All above + Wall</td>
<td>100000</td>
<td>7</td>
<td>4:01 minutes</td>
</tr>
</tbody>
</table>
Across multiple runs, with various scenarios for the energy function, the model parameters, as well as for the room scenario, we observed that the model converges well overall. A representative case, corresponding to the most complex performed one, involving all the cost terms, is displayed in Figure 34, representing the energy value and temperature decrease across the iterative process.

**Figure 34.** Temperature and energy value decrease in Simulated Annealing – the complete scenario.
We also analyzed the variance of the acceptance ratio and improvement measurement over time, for multiple scenarios, as represented in Figure 35. We again limited the number of maximum iterations to 100,000, according to the complexity of the bedroom scene modeled and the degree of freedom of the state transitions. We also modeled the temperature interval for various scenarios, allowing more exploration in favor of exploitation and vice versa. We noticed that one important factor that influences the acceptance rate and improvement rate is the complexity of the scoring function in terms of the factors that contribute to it and their respective weights. This can be explained by the fact that some criteria are easily and rapidly modeled by the algorithm, as is the case of the angles relative to the walls, while some others are continuously optimized, representing a threshold in the optimization process (i.e. optimizing in one direction might
increase the score of other terms in the energy function). One such scenario would be the positioning of the TV relative to the bed, bed relative to the back wall and TV relative to the corresponding back wall. In this case, if some other positioning dependencies do not allow a perfect alignment, the bed and TV positioning are continuously optimized back and forth, only superficially improving the score and thus remaining stuck in a local minimum neighborhood - if the temperature already decreased enough to hardly accept “wild” states.

Figure 35: Acceptance and Improve rate in Simulated Annealing – the complete scenario
5.7 Obtained results

5.7.1 Only collision and available space constraints

In the following snapshots, in Figure 36 – containing a 2D occupied space view of the furniture objects and Figure 37 – containing rectangle, representative images for the furniture, we present a result obtained by employing only the available space and collision costs in the energy function. We can note that the objects, although displayed in the room in a feasible configuration, do not form a plausible bedroom scene.

We note that for this case, the algorithm converges faster, easily finding a 0 value energy
state.

Figure 36. Only collisions and available space – 2D space occupied
5.7.2 Adding furniture distance relation relative to the wall

We add the furniture relation of positioning most (i.e. all without chair) models near the wall to the collision and available space constraints. We can review this exact improvement using similar style screenshots as those previously presented, in Figure 38 and Figure 39. Still, the scene is not quite realistic, with the objects not being rotated to fit the wall nor linked in functional grouping (as expected, due to the lack of respective optimization criteria). Although a slightly more time was employed in this run, the algorithm convergence rate was acceptable, lowering the energy from a mean of approximately
6,000 to 2.13.

We note that in this case the furniture pieces “stick” to the wall, as a consequence of the new added scoring factor.

**Figure 38.** Relative to the wall constraints and availability space case – 2D space occupied
5.7.3 Adding interdependency relation

We follow in the same manner and add the interdependency relations enforced in the scene. The improved results can be appreciated in Figure 40 and Figure 41. As compared to the previous two cases, which were run only for 50,000 iterations and a temperature interval of [2.5, 2500] we increase for this and the next (last) case the number of iterations
to 100,000 and the temperature interval to \([1, 3500]\), as described also in the model running parameters (stated for the final, most complex scenario implemented).

We note that the results improve in the expected manner, with furniture starting to be more grouped in functional relations, and the rooms consequently becoming more plausible, with a relative distance and positioning of the TV versus the bed, the nightstands are close to the bed, the desk and chair are positioned together in the room.

*Figure 40. All distance furniture relations and constraints and availability space case – 2D space occupied*
Figure 41. All distance furniture relations and constraints and availability space case – 2D space occupied
5.7.4 Adding rotation angle relative to the walls – Complete scenario

We present now some results of the obtained arrangement for the last and more complex scenario model, comprising of the cost factors detailed in the previous sections. This time, we noticed that the furniture pieces have a correct rotation to the wall and according to the facing relation, often being one in front of the other. Given the scenario and its differentiating purpose (i.e. introducing angle costs) we will plot only the screenshots showing that. In this way, the correct rotations can be analyzed. For this complex scene it can be seen that the energy function very rarely reaches a zero value in any case, the model having to threshold certain preferences in favor of others in the stage of optimizing towards local minima.

The models we show in the following images are the ones closer to a realistic scenario, as per this implementation. Further fine-tuning and factor weight changing can be possible, but is not the scope of this proof of concept, which aims to show general, straight through obtained results as opposed to corner cases, best parameter guesses for this scenario. In Figure 42 and Figure 43 a scene run for 150,000 iterations for a temperature in the range [1, 3500] is depicted. Other results obtained with similar runs (in which the only change is the number of iterations: 100,000) are presented in Figure 44.
Figure 42. In scene furniture arrangement for all enforced cost terms
Figure 43. In scene furniture arrangement for all enforced cost terms
**Intermediate states towards convergence**

Furniture intermediate layout states, taken from the convergence process after 0%, 25% 50% and 75% of the total iterations, are presented in Figure 45. We notice that various parameters are optimized and continuously changed, and that the movement in the scene is allows any change, not considering any type of collisions or space restriction – these being optimized and resolved by the components of the scoring function, through the iterative optimization process.
Figure 45. Furniture layout in the optimization process, taken after 0%, 25% 50% and 75% respectively of the iterations
6 Conclusions and Future Work

Main results of the thesis

- a data mining framework required for the implementation of ML and computational intelligence (CI) models addressing interior design challenges, including data definition, acquisition, cleaning, preprocessing and features extraction.
- a learning based scoring function for assessing room furnishing options for ranking according to hard constraints and more abstract, soft criteria.
- a stochastic optimization approach - Simulated Annealing, based on the scoring function defined to generate new feasible, realistically arranged furniture layouts that obey complex ergonomics and design criteria, analytically encoded in the evaluation function.

6.1 Conclusions

As a general overview, in this thesis we have delivered a proof of concept of the end-to-end process required to integrate and apply ML and CI methods in the interior design sensitive problem of automatic furniture synthesis, starting with a proposed initial framework for data mining, and following with defining a learning based scoring function for bedroom interior layouts ranking and a prototype of a stochastic approach for automatic furniture arrangement, based on the previous defined room evaluation method.

Our first contribution, a proposed data mining framework for implementation of ML and CI models addressing interior design challenges, is of key relevance for the success of the designed intelligent systems and comes at a time in which no mature, robust, ultimate choice of a dataset or an option to obtain such sensitive data is available. As seen in the recent scientific literature in the field, we are still at a very early stage in terms of providing reliable solutions in this sensitive, highly complex field of automatic interior design, where high level of model-specific data pre-processing is still done individually, often implying a case study in which users manually decorate room scenes or use internet to gather scenes from dedicated communities.
In light of these challenges, we have proposed a data mining process for extracting representative features of room furniture layouts, providing a detailed, end-to-end overview of each step involved, namely: data requirements understanding and definition, data acquisition, cleaning and correction, pre-processing and feature extraction, using the CRISP data mining approach.

The data mining framework, decisive in the ability of applying intelligent models in a field with few solid foundations, contributed as the first step towards the end goal of the project: an automatic model to assess the quality and rank accordingly expert-made room interior furnishing and, second, generate new stylish, professional-looking ones appropriate for exposing various options to clients willing to buy new furniture. In contrast to the recent trend towards the development of virtual environment appliances, the tool addresses a real world, customer oriented scenario, where the data quality and quantity plays a crucial role in achieving model robustness, completeness and abstraction through learning, thus meeting the required expectations imposed by the new context.

The dataset used in this prototype, comprising of 3D models of furnished bedrooms of various styles depicts real layouts created by persons and inspired by the real-world, often modeling their own or friends’ bedrooms. Containing amateur design 3D scenes, the models required heavy manual corrections and reconstruction. Following the data acquisition and cleaning process, we proposed a representative set of features that would analytically represent the complex dependencies and positioning of furniture in a common bedroom and present the process and complications involved in extracting such features – the automatic feature extraction process often encountering various types of corner cases, furniture particularities, scene representation errors, room shape constraints, etc.

Obtain model-specific features contributed in our project to developing both: a learning based scoring function for interior bedroom layouts and second a stochastic optimization based procedure for generating new furnishing solution, based on the scoring function designed. A learning-based scoring function has been proposed to realistically evaluate and rank the interior furnishing of a room, in our particular scenario: the bedroom. The method is designed to assess hard constraints, in our case measuring object collision with
each other or the walls and accessibility space in order to correctly distance each type of furniture piece and soft, more complex ones usually related to the overall functionality and aesthetics of the furniture arrangement. Soft constraints are in general harder to represent mathematically, because of the field complexity and numerous possibilities, often caused by abstract hidden patterns.

On the other hand, a learning based, pure data-oriented approach hardly captures fine affinity furniture relations. The scoring function proposed in this proof of concept comprises various energy factors that reward specific, “nice” features of the layout while penalizing the “ugly” ones thus providing a heuristic solution to realistically rank various, experts made interior designs. In detail, we model furniture functional or style interdependencies, representative in our case for a bedroom scene proof of concept, being the TV and bed/sofa link, bed – nightstands relationship and desk chair; relative positioning to the wall, measuring and modeling the positioning and rotation of adequate objects relative to the wall. Involving a learning stage of the parameters, the scoring tool can adapt in an automatically manner to predict layout quality in complex scenario, being able to incorporate subtle details even for sophisticated, non-standard, subtypes of furniture pieces – by training on a representative subset. Besides, through training, the scoring function can adapt to learn specialized metrics and measurement details in special types of rooms (i.e. smaller rooms, luxury rooms, L shape rooms, etc.) in which default, interior design default rules might not apply (i.e. a smaller room might imply smaller distances between furniture pieces, in an attempt to conserve space.

Our third contribution is an automatic furniture synthesis model, using Simulated Annealing, based on the layout evaluation function defined aiming to generate new plausible, fine-tuned, realistically furnishing options. The obtained results, comprising of a furnished bedroom using a preselected subset of furniture objects represents a proof of concept that subtle, fine affinity furniture relations can be learned and applied automatically to obtain new furnishing solutions, our model being able to produce plausible, realistically looking layouts that incorporate functional and general positioning guidelines, namely the ones modeled by the scoring function: close to wall positioning of appropriate furniture, with “back” against the wall; functional grouping of specific furniture
pieces, in terms of relative distance and relative forward direction, in order to support daily activities and routines, such as: TV-sofa/bed for watching television, desk-chair for working or leisure activities and bed and nightstands style based grouping.

6.2 Future work

The final goal of this project is a software tool, aiming to a real world, customer-oriented scenario for presenting possible furnishing options to the client who wants to buy furniture for its personal room. Given the described scenario, the tool needs to be scalable – i.e. given the numerous types and subtypes of furniture pieces available on the market and adaptive, robust, mature to generate furnishing layouts adapted for the personal room plan, and thus having the ability to address numerous shapes and sizes of rooms.

In the project described, we propose the development of a complete prototype to tackle the challenge end-to-end: starting with a data mining technique to obtain specific model features; proposing a learning-based interior ranking function for a bedroom scene and developing a stochastic approach towards generating new pleasant, fine-tune furnishing layouts – incorporating specific style, positioning and functional furniture relations. One challenge imposed in the project, motivating us to consider the CRISP data mining framework, was the absence of a large, expert made dataset needed to address such a complex problem from a new perspective, a real world, customer oriented manner one, as opposed to the recent tendency towards limiting to virtual appliance. The dataset used for validating the initial proof of concept, comprised a very limited number of instances: 12 real world scenario bedrooms, designed as 3D models by authors. This represents the main challenge to be solved in our next steps: addressing scalability, robustness, customization possibilities in order to make our models amenable to be used as a real, one-stop solution towards furnishing people homes.

The data used for modeling a first version of a complex, realistic room layout scoring function and designing a stochastic model for generating new furniture layouts was acceptable to validate a proof of concept, demonstrating the ability to enforce soft
guidelines, strict interior design rules and regulations, fine affinity furniture relations in obtaining pleasant, plausible furnishing options in a bedroom scenario. However, the possibilities unleashed by obtaining a significant amount of high-quality, representative data, would expand the current limitation of the prototype beyond the current application boundaries of such methods only in virtual reality, making them capable to conform a system to automatically synthesize fine-tuned, realistic, professional-level room layouts in a comparable manner to an expert in field.
7 Publication related to the thesis

8 Bibliography


design of urban procedural models, ACM Transactions on Graphics (TOG) -

it Home: Automatic Optimization of Furniture Arrangement," ACM Transaction on


http://www.sketchup.com/products/sketchup-make

max/overview.


[23] P. J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of

http://scikit-
learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html.


[26] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient


[42] P. Muller, P. Wonka, S. Haegler, A. Ulmer and L. V. Gool, Procedural modeling of


