DressUp! Outfit Synthesis Through Automatic Optimization

Lap-Fai Yu[1] *   Sai-Kit Yeung[2]   Demetri Terzopoulos[1]   Tony F. Chan[3]

[1]University of California, Los Angeles

[2]Singapore University of Technology and Design   [3]Hong Kong University of Science and Technology
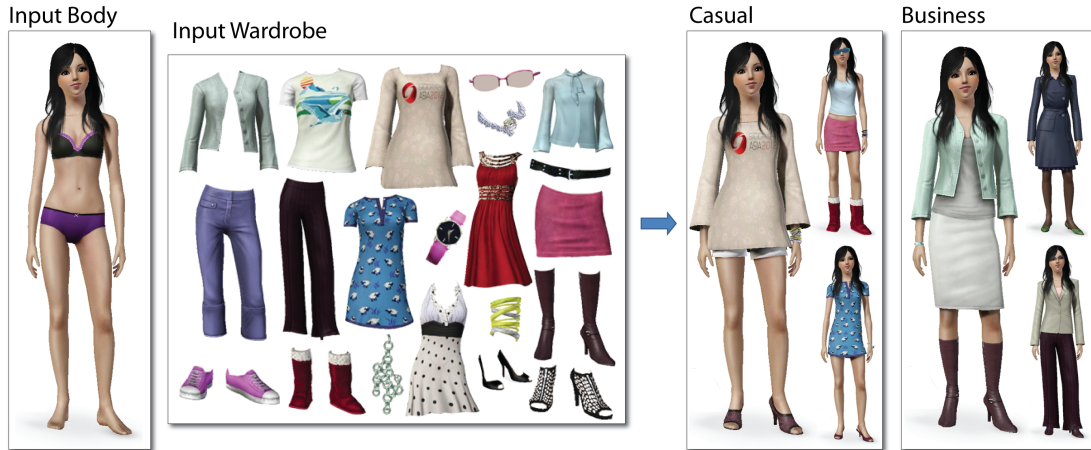
Figure 1: *Outfit optimization with different input dress codes. Left: An input human body with hair color, eye color, and skin color specified, plus a wardrobe of clothing items. Right: Optimized outfits for dress code* Casual *and* Business.

Abstract

We present an automatic optimization approach to outfit synthesis. Given the hair color, eye color, and skin color of the input body, plus a wardrobe of clothing items, our outfit synthesis system suggests a set of outfits subject to a particular dress code. We introduce a probabilistic framework for modeling and applying dress codes that exploits a Bayesian network trained on example images of real-world outfits. Suitable outfits are then obtained by optimizing a cost function that guides the selection of clothing items to maximize the color compatibility and dress code suitability. We demonstrate our approach on the four most common dress codes: *Casual*, *Sportswear*, *Business-Casual*, and *Business*. A perceptual study validated on multiple resultant outfits demonstrates the efficacy of our framework.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism;

Keywords: Clothing combination, perception, variety, functionally realistic, virtual world modeling, fashion, color matching, optimization, procedural modeling

Links: ◆DL 🗋PDF

*Part of the work was done when Lap-Fai was visiting SUTD.

1 Introduction

As you awaken each day, there is a simple question that you may need to answer: How should I dress today? Your wardrobe contains various kinds of clothes, such as dress shirts, dress pants, jeans, sweaters, suits, and different types of shoes. What combination of clothing will have you most appropriately dressed for the day's activities, thereby making you most visually appealing? Perhaps you would like multiple suggestions that best coordinate with the new tie that you received from your daughter as a birthday gift. The outfit selection problem also occurs in computer graphics modeling, especially in movie and game production: How should one appropriately dress a large number of human characters with an eye to functionality while avoiding visual awkwardness and repetitiveness? The manual specification of clothing is obviously tedious and it may be prohibitive on a large scale.

We demonstrate that the task at hand, of selecting appropriate subsets of clothing items from a wardrobe, can be addressed formally as a combinatorial optimization problem. A suitable outfit requires jointly combining a variety of clothing items to satisfy functional and certain visual criteria. We do not generally wear a pair of sandals with dress pants to the office, nor do we wear a red dress shirt with a green suit for a business meeting. In addition, to put a wardrobe into full use, we would like to explore as many good solutions as possible, so that we can exhibit sartorial variety. A similar, but much larger-scale problem comes up with regard to online boutique websites, where shoppers can select among many clothing items. Usually it is not difficult for a shopper to locate a desired clothing item; the nontrivial question is how this clothing item should be matched in terms of style and color with other clothing items from the same or different shops or from one's wardrobe at home.

There is no single universal rule that satisfies both the relevant *functional* and *visual* criteria. People generally categorize outfits into *dress codes*, which represent different functionalities. These can

range from strictly regulated ones such as *White Tie*, suitable for formal events, to relatively unrestricted ones such as *Casual*, suitable for many everyday activities. Without restriction, one can define a particular clothing requirement for an event and consider it a dress code. Different religions, societies, and cultural practices adhere to different dress codes; for example, in some formal occasions, Scottish men wear a kilt, a form of dress not commonly worn by men elsewhere. The visual criteria involve numerous factors, from human body attributes such as skin color, eye color, hair color, and body shape, which are model-specific, to aspects of the clothing items such as the clothing color, cutting, style, and fabric texture. The rules vary across national and cultural boundaries and historic timelines. Even when one has satisfied all the applicable rules, whether one is dressed in a visually pleasing manner is still a rather subjective question.

In tackling the clothes matching problem, we enforce functional and visual criteria through the two most important factors—*dress code* and *color*. While color is an obvious visual factor [Jackson 1987; Zyla 2010; Nicholson 2003], to a certain extent it is also related to functionality, which in turn depends on culture. For example, people in China usually dress in red for festivals and in white for funerals. On the other hand, the dress code is a broader guideline that pertains more to the combination of clothing items. Some dress codes also have strict requirement for the colors of particular items, but how different colors coordinate is not their main concern.

Given a specific dress code favoring various combinations of clothing items and a human body, our outfit synthesis framework optimizes the color compatibility between the human body and the suggested items in order to realize both the functional and visual criteria. We employ four of the most common dress codes *Sportswear*, *Casual*, *Business-Casual*, and *Business*, which cover the main functionalities of daily life in much of the world. These dress codes are encoded in our system within a probabilistic framework, via a Bayesian network. The Bayesian network is trained on real image data and it associates any particular clothing item combination with an observed probability distribution under any specific dress code. Additional dress codes and other matching criteria can be trained and included in the same manner. As is common practice in the fashion industry [Jackson and Lulow 1984; Jackson 1987; Henderson and Henshaw 2008], our system classifies the color type of the human subject as 'warm' or 'cool' based on his or her skin, hair, and eye colors. This is automatically accomplished by a classifier that is pre-trained on a database of images of people. After assigning the user color type, our system will suggest a preferable color palette for the subject and this color palette will serve as a soft constraint during the optimization, which automatically searches for clothing items guided by the dress code while satisfying color compatibility criteria subject to the suggested color palette.

In summary, outfit selection is a common everyday problem; however, the nature of this problem is very broad and it involves a considerable amount of visual and social factors that can be implicit and abstract. Our main contributions in this paper are as follows:

1. The introduction of a novel topic area to computer graphics and a first attempt to tackle the automatic outfit synthesis problem through a data-driven approach.

2. The encoding of implicit, probabilistic clothing matching relationships on real-world data through Bayesian Networks that support conditional queries and incorporate a Support Vector Machine classifier of body color tone that applies subjective evaluation criteria common in the fashion industry.

3. The formulation of outfit synthesis as an optimization problem that takes into account the style and color compatibility of clothing combinations, and that is flexible and easily extensible through the modification of the Bayesian networks and

cost terms of the formulation.

4. The application of our novel approach in different practical scenarios; e.g., as an outfit advisor, as a suggestion engine in shopping/boutique websites, or as part of the character modeling engine for games/virtual world applications.

5. The validation of the efficacy of our approach through a gender-specific perceptual study.

2 Related Work

We will begin by discussing related work on clothing and virtual character modeling in computer graphics and then review other relevant work on fashion and color.

2.1 Clothing in Computer Graphics

Modeling, animating, and rendering visually realistic clothing has been an area of interest in computer graphics for decades [Terzopoulos et al. 1987; Terzopoulos and Fleischer 1988; Provot 1995; Baraff and Witkin 1998] and it has received much attention in recent years in movies and games especially for dressing large numbers of human characters. Researchers have been putting significant effort into the realistic modeling [Kaldor et al. 2008; Volino et al. 2009; Kavan et al. 2011; Wang et al. 2011; Umetani et al. 2011; Guan et al. 2012] and/or animation [Kaldor et al. 2010; de Aguiar et al. 2010; Wang et al. 2010; Feng et al. 2010] of clothing, and their efforts have enabled computer animated clothing to blend seamlessly with the clothing worn by real actors.

Tools are now available to help artists interactively design virtual garments, which is adequate for highly-detailed, small scale production, e.g., for motion pictures. However, manual approaches become too tedious on a large scale, such as when there is a need to clothe numerous virtual humans in a virtual city. While our work does not concern the physically-realistic deformation of clothes meshes over virtual bodies, we are not aware of any research on automatic outfit synthesis in computer graphics; i.e., given a set of clothing items and a human body model, automatically suggest a clothing combination for a general or particular scenario.

Tsujita et al. [2010] conducted a user survey that pointed out the difficulty that people have in selecting suitable outfits from their wardrobe. They proposed the simple heuristic of not repeating outfits on consecutive days, and installed a camera system in a user's wardrobe that can acquire and upload pictures of clothes to the internet so that the user can solicit outfit selection advice from friends. Our data-driven approach captures clothing combination "advice" implicit in example fashion images, but we automate the suggestion and synthesis process.

2.2 Human Modeling

Human characters are an important aspect of creating virtual worlds [McDonnell et al. 2006; Dobbyn et al. 2006; McDonnell et al. 2008; McDonnell et al. 2009; O'Sullivan 2009]. While realistic human animation and rendering can be critical [Tecchia et al. 2002], variety in human appearance is equally important when considering a large group of people. Ulicny et al. [2004] describe a system that enables the interactive creation of virtual humans with variety. The importance of appearance variation in realistic human perception is nicely summarized with an extensive perceptual study in the work of O'Sullivan et al. [McDonnell et al. 2008; McDonnell et al. 2009; O'Sullivan 2009].

For the most part, existing human modeling software requires substantial manual intervention. However, researchers have proposed
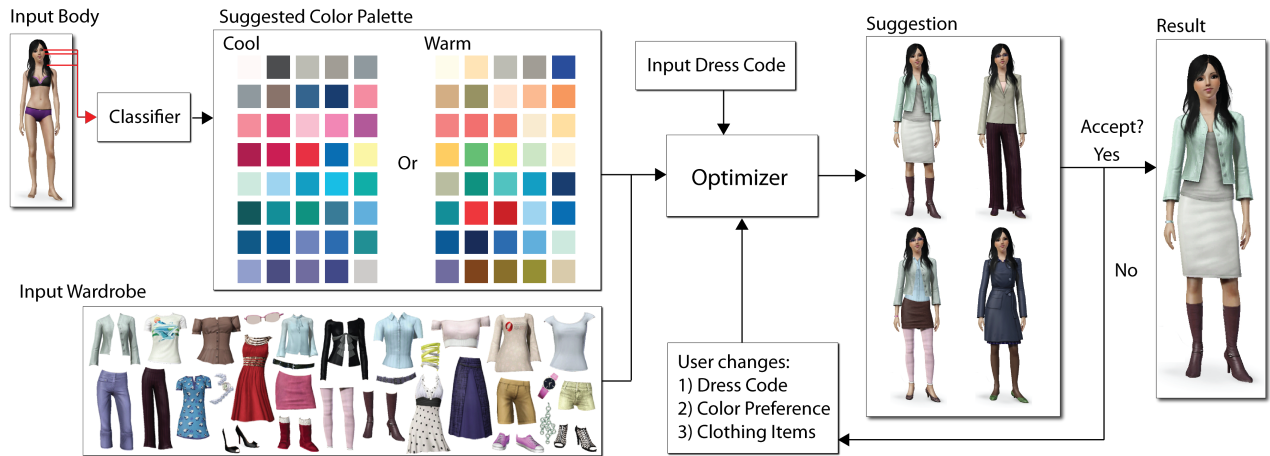
Figure 2: *Our optimization framework. Inputs include body color attributes, an input dress code, and a wardrobe of clothing items. The optimizer generates optimal suggestions according to cost terms defined by the dress code, the suggested color palette, and color compatibility. In response to a user's change of style, color preference, or specification of particular clothing items, our system automatically synthesizes new outfits.*

approaches to mass-produce various characters by automatically modifying the texture, color, and geometry of different body parts in order to create crowds that exhibit some natural variation [Mc-Donnell et al. 2006; Dobbyn et al. 2006; Thalmann et al. 2007]. However, the goal of prior approaches is to enhance the realism of the crowd as a whole, rather than specific concern as to whether any individual in the crowd is dressed properly or in a visually pleasing manner. The lack of a fast, highly automated approach to this problem limits variation in the style of human characters, leading, in particular, to repetitive sartorial patterns that greatly reduce realism.

2.3 Color in Clothing

Recently, techniques for combining colors in a scene to make it look, say, "harmonious" or "peaceful" have been gaining interest [Cohen-Or et al. 2006; O'Donovan et al. 2011]. Color coordination is a core consideration in clothes matching [Zyla 2010; Gilchrist 2011; Nicholson 2003]. Fashion and make-up professionals usually regard color coordination as person-specific, mostly dependent on the person's intrinsic color tones, in particular, the skin, eye, and hair colors [Jackson 1987]. A basic approach is to first classify individuals as suited to a 'warm' or 'cool' color palette, from which they should choose the colors for their clothes. As there is no definitive classification rule, subjective evaluation is usually performed, and a common test is to have observers evaluate whether the individual looks best wearing gold or silver accessories, respectively [Jackson 1987]. There are other variations of classification which are more subtle and abstract—e.g., in accordance with the season [Jackson and Lulow 1984], or according to "light/deep/clear/soft" [Henderson and Henshaw 2008]. However, the basic principle is still the same—suggesting a color palette for clothing items based on the classification result.

2.4 Dress Codes

A dress code is a set of rules governing what garments may be worn together and in what setting. Such rules are commonly agreed upon among people, usually dependent on events and occasions. Common dress codes nowadays include *Sportswear*, *Casual*, *Business-Casual*, *Business*, and *Formal*. Figure 3 shows typical example images. Some of the aforementioned dress codes also constrain

Casual   Sportswear   Business-Casual   Business

Figure 3: *Example images of dress codes from Dreamstime Images.*

the color of the items; for example, *Business* clothing tends to be darker, while there is not much restriction in *Casual* or *Sportswear*. Pattern, fabric weight, and texture are also relevant to the dress code [Gilchrist 2011].

The dress code is important in governing the functionality of the clothing [Schoeffler and Gale 1973; Fischer-Mirkin 1995; Flusser 2002; Sondag 2011; Gilchrist 2011]. However, the main objective of a dress code is to convey a message through the combination of various clothing items. For example, dressing without a tie for a job interview will convey a less formal and more relaxed impression, while donning a suit, dress shirt, and tie to the beach will create an unusual scene. Without a strict definition, the perception of some dress codes can be ambiguous and personal; e.g., some *Business Casual* outfits may be regarded as *Business* or *Casual*.

3 Data-driven Approaches

Figure 2 shows an overview of our optimization approach for automatic outfit suggestion. The inputs comprise a human body model, a specific dress code, and a predefined wardrobe. The output is one or more optimized outfit suggestions. Before presenting the technical details of our optimization framework and developing the objective function in the next section, let us consider the information required to define our cost functions.

There are two preprocessing steps before the optimization process—encoding the clothing relationship and classifying the color tone of the subject body. We must quantify the relationships among different clothing items so that we can define compatible

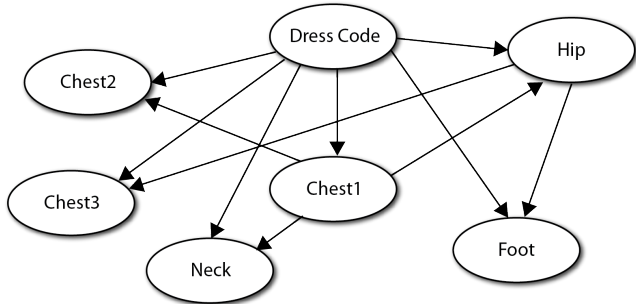| Example node	Example state
Dress Code	Casual, Sportswear, Business-Casual, Business
Chest 1	t-shirt, dress shirt, sleeveless
Chest 2	tank, sweater, vest, long t-shirt
Chest 3	suit jacket, jacket, hoodie, open sweater
Hip	jeans, shorts, dress pants
Foot	slippers, dress shoes, boots
Neck	necklace, scarf, tie, bow tie

Figure 4: *Representing distributions of clothing items combinations with a Bayesian network. Top: A table showing the major example nodes with some of their states. Bottom: A part of an example Bayesian network for men, trained using labeled fashion images. Refer to our supplementary materials for the complete graphs of the Bayesian networks. Note that each node, except for the dress code node, has a state 'none'.*

costs among them; for example, what should or should not be worn based on the selected dress code and some already selected clothing items. As we have discussed, the dress code involves various factors and can change from time to time. For example, a dress shirt usually goes with dress shoes if dress pants are worn, but there could be more flexibility if jeans are worn.

An expedient way to generate outfit variety is to randomly select among predefined rules to combine clothing items. However, the question of how to define the rules, which is critical to synthesis quality, is susceptible to subjective bias. It is difficult to consider all possible combinations, and the rules quickly become intractable to maintain as the types of clothing items grow. Restricting to a small subset of possible outfits may avoid awkward synthesis, but it will result in limited variety and common artifacts such as "repeated" characters that are noticeable in virtual scenes. The lack of conditional query support has also prohibited the use of such approaches in practical scenarios (e.g., shopping websites).

One possibility to encoding various relationships and defining compatible costs between clothing items is to adopt a data-driven approach based on observational data. Data driven approaches have recently proven to be successful in problems involving abstract semantic relationships; for example, in architectural design, furniture layout, assembly-based 3D modeling, and color compatibility applications [Merrell et al. 2010; Yu et al. 2011; Chaudhuri et al. 2011; O'Donovan et al. 2011]. Since our goal is to match different clothing items in a sensible manner, and with natural variety conforming to real world observations, a probabilistic machine learning framework trained by real world data is appropriate to encode the matching cost, such that the higher the probability of a particular clothes combination, the lower is its matching cost.

An important issue in establishing the probabilistic relationships between different clothing items relates to their *conditional dependencies*. For example, the frequent occurrence in the data of a jeans-sandals combination and a dress shirt-jeans combination could lead

to a dress shirt-jeans-sandals combination style being generated, which should have very low likelihood. Therefore, simply encoding the observed probability of a clothing item and any combination between it with other items is prone to error.

Probabilistic graphical models, in particular, Bayesian networks, are an elegant and efficient choice [Pearl 1988; Koller and Friedman 2009] for learning the implicit relationships among different clothing items consistent with their conditional dependencies. Our trained Bayesian networks effectively encode the probability distributions in the space of clothing combinations. An important feature of the Bayesian network is its ability to support conditional query, which is frequently needed in clothes matching. The values of any subset of a clothing combination can be fixed and the probabilities of the remaining clothes can be calculated. For example, given the *Business-Casual* dress code, one may constrain the upper body to be clothed in a t-shirt and blazer and query the probability of the lower body being clothed in jeans according to the trained distribution. This allows better flexibility to recommend clothing items under different user-specified conditions or scenarios.

3.1 Bayesian Networks for Clothing Relationships

To make the scope of our problem tractable, we train separate Bayesian networks for men's and women's wear and exclude color from the training process. In our current system, we train these networks on four dress codes: *Sportswear*, *Casual*, *Business-Casual*, and *Business*. Figure 4 shows part of the Bayesian network for men. The network for women is similar, with differences in some of the node states; e.g., having state *dress* in node *Chest 1*. The complete networks can be found in our supplementary materials.

The nodes of the Bayesian networks correspond to different body regions on which a clothing item can be worn, and each node state represents the type of clothing item being worn. For example, the node *foot* has states *dress shoes*, *slippers*, *boots*, and so on. Except for the node *dress code*, each node also has a state *none*, which is used when the node does not carry any clothing item; e.g., *foot = none* when no shoe is worn. While state choices can be easily modified to suit specific domain needs, as a general case, we follow common classification in boutique websites such as "H&M" and "eBay".

To enable us to handle more complicated situations where there is layering of clothing, we permit a body region to be represented by multiple nodes that correspond to multiple clothing layers. For example, the chest has nodes *Chest 1*, *Chest 2*, and *Chest 3*, with *Chest 1* corresponding to the innermost layer (e.g., a t-shirt), *Chest 2* to the middle layer (e.g., a vest), and *Chest 3* to the outer layer (e.g., a jacket).

Usually a reasonable quantity of input training data is required. For example, 120 architecture programs were used to train the networks in [Merrell et al. 2010]. In our case, we downloaded around 3000 images for the four dress codes for men and women from Google Images[1]. Since some of the downloaded images are not useful, and determining whether the images belong to the dress code is a subjective process, we hired three fashion school students to manually label the attributes of each instance in the network, who used their judgment to disregard inappropriate images. In total, around 2000 labeled data sets for men and women were used to train the Bayesian networks. Labeling each image took about 15–20 sec, and the whole labeling process took 4 hours. Example training images for *Business* and *Sportswear* are shown in Figure 5. Variety arises when multiple item combinations occur under the same dress code.

---

[1]Example keywords we used for the image search: 'Casual wear for men', 'Sportswear for men', 'Business-Casual wear for men', Business wear for men', and similarly for women.

Figure 5: *The top row shows typical example images from Dreamstime Images. The bottom row shows the corresponding labeled data used for Bayesian network training. Note that some images may have occluded items (e.g., shoes are not visible in the second image), but partially labeled data is still usable in training the Bayesian network.*

(a) *Conditional probabilities*:	
P(bracelet | *dress code* = Business)	0.0691
P(bracelet | *dress code* = Business Casual)	0.2456
P(bracelet | *dress code* = Casual)	0.4591
P(bracelet | *dress code* = Sportswear)	0.1023
(b) *Joint conditional probabilities*:	
P(dress | *dress code* = Business Casual)	0.1706
P(dress | *dress code* = Business Casual, *Foot* = legging)	0.6710
P(sweater | *dress code* = Business Casual)	0.0504
P(sweater | *dress code* = Business Casual, *Chest 1* = dress shirt)	0.1095
(c) *Conditional joint probabilities*:	
P(sport pants, sport shoes, tank | *dress code* = Sportswear)	0.1181
P(sport pants, sport shoes, long t-shirt | *dress code* = Sportswear)	0.0353
P(shorts, sport shoes, sweater | *dress code* = Sportswear)	0.0007

Table 1: *Example probabilistic queries supported by Bayesian Networks for women. (a) Simple queries by conditional probabilities. (b) Increased joint conditional probabilities effectively reflect common matching styles such as "legging, dress", "dress shirt, sweater" from the training data. (c) Conditional joint probabilities for more complicated combinations. The major advantage of Bayesian Networks is that they support instant, arbitrary queries.*

The images, labeling program, and labeled data are included in the supplementary materials.

Two attributes should be assigned to two different nodes if the corresponding two clothing items can coexist; e.g., shirt and suit jacket. Otherwise, they should be put under the same node; e.g., sandals and lace-up shoes, since it is not possible to wear both at the same time. The important point here is to capture the relationships among different clothing items and their conditional dependencies. Using the labeled data, we learnt the Bayesian network structures for men and women respectively, by the Tree Augmented Naive Bayes method [Friedman et al. 1997] which maximizes conditional mutual information between attributes. The conditional probability tables are trained by the Expectation-Maximization algorithm, which can learn the probabilities even if some training data are only partially labeled. Notice that other methods such as maximum likelihood estimation could also be adopted. We found that the results generated using the learnt networks faithfully reproduce our human perceptual requirement for the four dress codes considered. Table 1 shows some example queries based on the probabilities captured.

Figure 6: *Top: An example fashion image and its corresponding 5-color palette. (Image courtesy of COLOURLovers.)*

Body Attribute Description	Classification
(a) teal blue eyes, dirty blonde hair, peach skin	Autumn
(b) dark brown eyes, oyster white hair, ivory skin	Autumn
(c) black brown eyes, white hair, beige skin	Winter

Table 2: *Example classification guidelines for four-season body color tone. To obtain a classification result, the user must first determine his/her body attributes according to the description. The description and classification can be obscure to interpret: (a) and (b) have different descriptions, but are classified as the same, while (b) and (c) have similar descriptions, but are classified as different. (Courtesy of AskAndyAboutClothes.com; refer to website for the full table.)*

3.2 Body Color Tone Classifier

After encoding the probabilistic relationships among the clothing items, the next step is to inform the optimization process of a color guide. It is a common practice in fashion to first classify a person's body color tone and then suggest a suitable color palette for matching clothes for them. There are multiple ways for color tone classification such as subjective evaluation tests [Jackson 1987] or by "guidelines" or "rules"[2]. However, as shown in Table 2, the classification "guidelines" can be very obscure and cumbersome, arguably uninterpretable by a general user. There is obviously no unique one-to-one correspondence between body color attributes and color tone classification for users to follow.

To this end, we train a classifier to predict the body color tone of a target person consistent with human preferences. This has two major advantages. First, we integrate subjective evaluation tests commonly adopted in fashion [Jackson 1987] into a machine learning framework by capturing the subjective evaluation from a number of people. Second, after the classifier is trained, it is intuitive at the user's end—a user simply inputs his/her body colors (e.g., by a few clicks on his/her face photo) and automatically obtains a color tone classification result, instead of interpreting obscure descriptions.

We acquired from Google Images a training dataset comprising 1000 facial images after discarding images with strong illumination effect, including both males and females. For each image, we manually extracted the RGB values of the eye, skin, and two locations in the hair (to encode hair color variation). In accordance with common practice in the fashion industry, we matched each image with a set of silver accessories and then with a set of gold accessories, from which a test subject was asked to choose which one they preferred, thus indicating 'cool' and 'warm' color tone, respectively. We recruited 40 volunteer participants, including 20 males and 20

---

[2] http://www.askandyaboutclothes.com/Tutorials/CindyBuschColorAnalysis.htm

females whose ages ranged from 20 to 60, to evaluate the 1000 face images. Evaluation took about 5-10 sec per image.

We trained a Support Vector Machine (SVM) classifier [Cristianini and Shawe-Taylor 2000] and performed cross-validation by randomly choosing 900 data for training and 100 data for testing, achieving a prediction rate of about 77%. Given a previously unseen human body model with specific skin, hair, and eye colors, the trained classifier predicts the body color tone, thereby recommending either a 'cool' or 'warm' color palette to be used in the optimization. Each suggested color palette consists of 40 colors, as in [Jackson 1987]. While the evaluation is by its nature subjective and ambiguous, we find that in general people with brownish/reddish hair and brownish/greenish eyes are usually classified as 'warm', whereas those with light-colored hair and dark/bluish eyes are classified as 'cool'. The labeled training data and labeling program are included in the supplementary materials.

3.3 Color Compatibility Predictor

Figure 6 shows example images from fashion websites such as "Wear Palettes" and "COLOURLovers", which are usually accompanied by a representative 5-color palette that supports the color matching idea. Akin to this practice, at each iteration of the optimization, our optimizer extracts a representative 5-color palette from an outfit and evaluates the color compatibility of the palette based on the regression model from [O'Donovan et al. 2011], which is trained by a large number of user-rated color palettes. The trained regression model can take a 5-color palette as input, and predict a user preference rating (see [O'Donovan et al. 2011] for the details of the training dataset, prediction result evaluation, and analysis).

4 Outfit Optimization

In performing the optimization phase, our system exploits the trained Bayesian networks, body color tone classifier and color compatibility predictor described in the above sections. Given a human model, a wardrobe of clothing items and a dress code as inputs, our system suggests multiple outfits whose colors are adjusted desirably such that they are compatible to each other guided by the color palette.

To achieve our goals we must solve a combinatorial optimization problem. Denoting the wardrobe as W, which is a set containing all clothing items, the state of our system is a subset of W, which we refer to as an *outfit*, $\phi = \{\theta_i | i = 1, \ldots, T\}$, where each $\theta_i = (c_i, n_i, s_i)$ is a 5-value tuple representing a selected clothing item. The term $c_i = (r_i, g_i, b_i)$ contains the RGB values of the clothing item, which are quantized from 0 to 255, n_i is the node of the Bayesian network to which this clothing item belongs, and s_i is the corresponding node state. For example, $n_i = foot$ and $s_i = dress\ shoes$ means that selecting the clothing item θ_i corresponds to setting the node *foot* of the Bayesian network to state *dress shoes*.

Note that the total number T of selected clothing items is a variable that can be changed. Thus, the dimension of the input space is a variable. Our goal is to obtain an outfit ϕ that minimizes an objective function described in the next section.

4.1 Objective Function

We now describe the cost terms constituting our overall objective function.

Style Cost: In order to obtain the matching cost between different clothing items, at each iteration, we must determine every node state of the Bayesian network. Suppose the network has N nodes
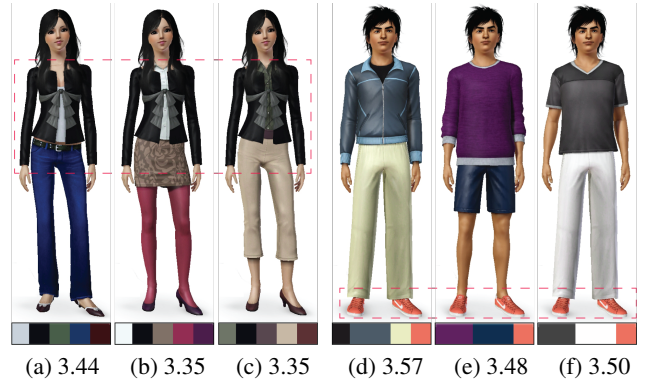

(a) 3.44   (b) 3.35   (c) 3.35   (d) 3.57   (e) 3.48   (f) 3.50

Figure 7: *Results with specific clothing items being fixed. (a)–(c) fixed black sweater. (d)–(e) fixed orange shoes. Color ratings are shown at the bottom.*

(excluding the root node *dress code*) denoted by $x_1, \ldots, x_N$. Given an outfit ϕ, every node x_k is instantiated to state $S(x_k)$ by:

$$S(x_k) = \begin{cases} s_i & x_k = n_i \\ \text{none} & x_k \neq n_i, \forall i \end{cases} \quad (1)$$

The style cost term has two components $C_{\text{style}}^{\text{indv}}$ and $C_{\text{style}}^{\text{joint}}$. Given *dress code* $= d \in \{Casual, Sportswear, Business-Casual, Business\}$, then $C_{\text{style}}^{\text{indv}}$ encodes the conditional probability of each clothing item. It guides the optimizer by penalizing the selection of clothing items that do not fit dress code d. On the other hand, $C_{\text{style}}^{\text{joint}}$ defines the conditional joint probability of the clothing item combination:

$$C_{\text{style}}^{\text{indv}}(\phi) = 1 - \frac{1}{N} \sum_k P(S(x_k)|dress\ code = d). \quad (2)$$

$$C_{\text{style}}^{\text{joint}}(\phi) = 1 - P(S(x_1), \ldots, S(x_N)|dress\ code = d). \quad (3)$$

To evaluate these costs, our framework makes queries over the Bayesian network to provide the conditional and conditional joint probabilities in (2) and (3).[3] In case the user fixes one or multiple node states, the fixed node states will become the given conditions. Figure 7 shows two examples with specific items being fixed.

Color Rating Cost: Similar to the convention in fashion images, we use a 5-color palette to represent a clothing combination ϕ which comprises T selected clothing items, based on a heuristic:

1. Each clothing item is represented by the color of its largest surface area.

2. Select 5 colors:

> If $T = 5$, select colors from all clothing items.
>
> If $T > 5$, sort clothing items by their surface areas. Select colors from the 5 clothing items with the largest surface areas.
>
> If $T < 5$, sort clothing items by their surface areas. Duplicate colors of the $5 - T$ clothing items with the largest surface areas. Select the $5 - T$ duplicated colors and the colors of the T clothing items.

---

[3]To illustrate the effectiveness of $C_{\text{style}}^{\text{indv}}$, suppose the dress code is *Business*, the initialized outfit is "shirt, jeans, slippers" and another outfit "shirt, dress pants, slippers" is sampled. Although $C_{\text{style}}^{\text{joint}}$ will evaluate both outfits as unlikely, $C_{\text{style}}^{\text{indv}}$ will favor the latter, hence effectively guiding the synthesis towards a *Business* outfit.
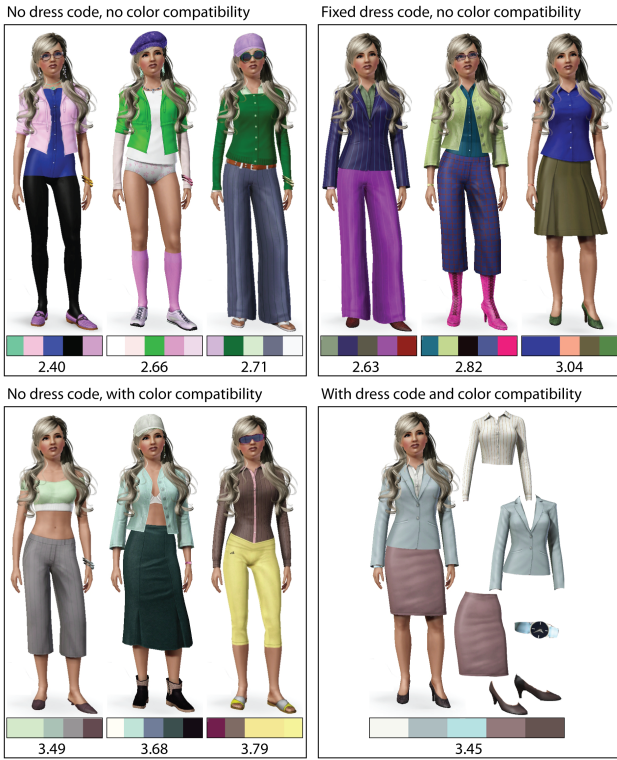
No dress code, no color compatibility

| 2.40 | 2.66 | 2.71 |

Fixed dress code, no color compatibility

| 2.63 | 2.82 | 3.04 |

No dress code, with color compatibility

| 3.49 | 3.68 | 3.79 |

With dress code and color compatibility

| 3.45 |

Figure 8: *Effects of omitting individual cost terms. Top left: No dress code and no color optimization. Top right: Fixed dress code, no color optimization. Bottom left: No dress code but with color optimization. Bottom right: Fixed dress code with color optimization. The color ratings are shown at the bottom.*

3. Sort the 5 selected colors according to their physical position on the body, from top to bottom.

In practice, we assume the outfit comprises at least 2 clothing items, i.e., $T >= 2$. Denoting these ordered 5 colors as $\lambda_1, \ldots, \lambda_5$, this is the 5-color palette representing outfit ϕ. The color compatibility cost is

$$C_{\text{color}}^{\text{comp}}(\phi) = 1 - [R(\lambda_1, \ldots, \lambda_5) - 1]/4. \qquad (4)$$

In (4), $R \in [1, 5]$ is the regression model from [O'Donovan et al. 2011], which predicts the user rating of a 5-color palette, with a higher rating implying higher user preference. The cost in (4) is normalized accordingly.

Color Palette Cost: To keep the clothing item colors close to the suggested color palette, the system calculates the distance of each clothing item's color c_i to each color c_j in the suggested color palette, and penalizes it if the nearest distance is larger than a threshold h. The color palette cost term is defined as

$$C_{\text{color}}^{\text{palette}}(\phi) = \frac{1}{T\sqrt{3Z^2}} \sum_i \max(\min_j \|c_i - c_j\| - h, 0) \quad (5)$$

,where $Z = 255$ is the maximum quantized RGB value.

The total cost function is the weighted sum of the above cost terms:

$$\begin{aligned} C(\phi) = w_{\text{style}}^{\text{indv}} C_{\text{style}}^{\text{indv}}(\phi) + w_{\text{style}}^{\text{joint}} C_{\text{style}}^{\text{joint}}(\phi) + \\ w_{\text{color}}^{\text{comp}} C_{\text{color}}^{\text{comp}}(\phi) + w_{\text{color}}^{\text{palette}} C_{\text{color}}^{\text{palette}}(\phi) \end{aligned} \qquad (6)$$

The w coefficients determine the relative weighting between the cost terms; in practice, we set $w_{\text{style}}^{\text{indv}} = 1.0$, $w_{\text{style}}^{\text{joint}} = [5.0, 10.0]$,

Figure 9: *Results with two different color palettes.*

$w_{\text{color}}^{\text{comp}} = 1.0$, and $w_{\text{color}}^{\text{palette}} = 1.0$. Figure 8 shows the effect of omitting the style and color cost terms. Figure 9 illustrates the effect of using different color palettes.

4.2 Reversible Jump Markov Chain Monte Carlo

Since our optimization problem is combinatorial and the number of combination items can vary (e.g., a jacket can be added or removed), it is difficult to define a closed-form solution. In fact, as in the real world, we wish to obtain multiple optimal solutions (outfits) from the same wardrobe instead of a single global optimum. This motivates the generation of candidate solutions by sampling a density function defined over the space of possible outfits. The density function is defined using idealized analytical formulations. Sampling is performed using a Markov chain Monte Carlo sampler. Figure 10 shows multiple optimal outfits generated.

One of the difficulties for our optimization problem is that its dimensionality may change; i.e., the number of clothing items may be altered during the optimization process. To deal with this complication, we adopt the Reversible Jump MCMC (RJMCMC) framework [Green 1995] which can be considered a generalization of the original Metropolis-Hastings (MH) algorithm [Metropolis et al. 1953; Hastings 1970]. RJMCMC works by supplementing the parameter-changing diffusion moves of MH with an additional set of dimension-altering jump moves, which allow the chain to move between subspaces of different dimension. RJMCMC has been successfully applied to other graphics and vision problems such as procedural modeling [Talton et al. 2011] and image segmentation [Tu and Zhu 2002].

4.3 Annealing

To efficiently explore the solution space, we apply the simulated annealing technique [Schneider and Kirkpatrick 2006] in the optimization process. We define a Boltzmann-like objective function:

$$f(\phi) = \exp(-\beta C(\phi)), \qquad (7)$$

where β is a constant inversely proportional to the temperature of the annealing process. At the beginning of optimization, β is set to a low value, equivalent to setting a high temperature, which allows the sampler to more aggressively explore the solution space. Then β is gradually increased throughout the optimization. Near the end, β attains a large value, equivalent to setting the temperature near zero, thereby allowing the sampler to refine the solution. Figure 11 shows the iterative optimization process.

Figure 10: *Multiple outfit recommendations. The dressed models and the corresponding items. Top:* Sportswear. *Bottom:* Business-Casual. *The recommendations from left to right are arranged in descending matching cost value.*

4.4 Proposed Move

We adopt the *dimension matching* strategy to allow reversible jumps across subspaces of different dimension or within the same subspace. At each iteration of our optimization, a move $m' \in \{m_a, m_r, m_s, m_m\}$ is chosen with probability $p_{m'}$. Associated with the move is a move-specific proposal distribution $q_{m'}(.)$, which is a function of an auxiliary variable U'. As move m' is chosen, a sample of the auxiliary variable U' is drawn from $q_{m'}(U')$, which modifies the current outfit ϕ to a proposed new outfit ϕ' by a deterministic function $\phi' = h(\phi, U')$. We also need to compute the reverse move m, which reverts ϕ' back to ϕ, by sampling U from $q_m(U)$ such that $\phi = h^*(\phi', U)$. The proposed outfit ϕ' is then accepted with probability

$$\alpha(\phi'|\phi) = \min(1, \frac{p_m}{p_{m'}} \frac{q_m(U)}{q_{m'}(U')} \left| \frac{\partial(\phi', U)}{\partial(\phi, U')} \right| \frac{f(\phi')}{f(\phi)}), \quad (8)$$

where $|\partial(\phi', U)/\partial(\phi, U')|$ is the Jacobian of the diffeomorphism from (ϕ, U') to (ϕ', U). Defining $\phi' = h(\phi, U') = U'$ and $\phi = h^*(\phi', U) = U$, the Jacobian is unity [Godsill 2001]. For further detail on RJMCMC, refer to [Green 2003; Andrieu et al. 2003].

Based on the RJMCMC formulation, we follow the natural strategy to define the jump moves as adding/removing a clothing item to/from the outfit, which induce a dimension change, and diffusion moves as swapping items or modifying an item's color, which involve no dimension change, as follows:[4]

---

[4]The acceptance probabilities of the proposed RJMCMC moves are:
Adding an Item (m_a):

$$\alpha(\phi'|\phi) = \min(1, \frac{p_r}{p_a} \frac{\frac{1}{|\phi'|}}{\frac{1}{|W \setminus \phi|}} \frac{f(\phi')}{f(\phi)}) \quad (9)$$

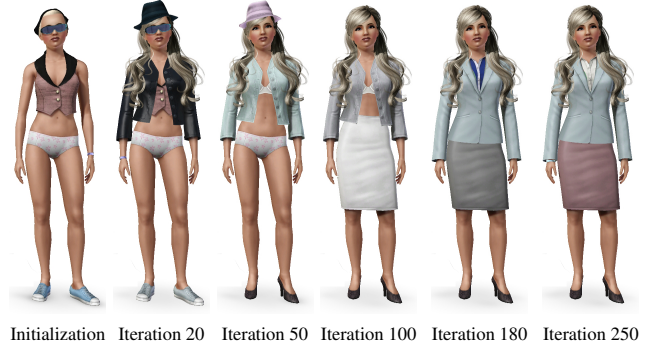Initialization Iteration 20 Iteration 50 Iteration 100 Iteration 180 Iteration 250

Figure 11: *Outfit optimization from a random initial configuration (left) for dress code* Business. *As the optimization process proceeds, the clothing items are iteratively updated until the outfit converges to the desired clothing item combination with coordinated color.*

Adding an Item (m_a): Randomly pick an available clothing item θ_j from wardrobe W and add it to outfit ϕ, so that $\phi' = \phi \bigcup \{\theta_j\}$.

Removing an Item (m_r): Randomly remove a selected clothing item θ_i from outfit ϕ, so that $\phi' = \phi \setminus \{\theta_i\}$.

Swapping Items (m_s): Randomly pick a selected clothing item θ_i from outfit ϕ, and swap it with an available clothing item θ_j from wardrobe W, so that $\phi' = \phi \setminus \{\theta_i\} \bigcup \{\theta_j\}$.

Modifying an Item Color (m_m): Randomly pick a selected clothing item θ_i from outfit ϕ and change its color c_i. Hence, θ_i is updated as: $\theta_i' = (c_i + \delta c_i, n_i, s_i)$, where $\delta c_i \sim [\mathcal{N}(0, \sigma_c^2) \, \mathcal{N}(0, \sigma_c^2) \, \mathcal{N}(0, \sigma_c^2)]^T$ and, with $\mathcal{N}(\mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} e^{-(x-\mu)^2/2\sigma^2}$, a Gaussian distribution of mean μ and variance σ^2. The variance σ_c^2, which determines the average magnitude of the change, is proportional to the temperature.

$$= \min(1, \frac{p_r}{p_a} \frac{|W \setminus \phi|}{|\phi'|} \frac{f(\phi')}{f(\phi)}). \quad (10)$$

Removing an Item (m_r):

$$\alpha(\phi'|\phi) = \min(1, \frac{p_a}{p_r} \frac{\frac{1}{|W \setminus \phi'|}}{\frac{1}{|\phi|}} \frac{f(\phi')}{f(\phi)}) \quad (11)$$

$$= \min(1, \frac{p_a}{p_r} \frac{|\phi|}{|W \setminus \phi'|} \frac{f(\phi')}{f(\phi)}). \quad (12)$$

Swapping Items (m_s):

$$\alpha(\phi'|\phi) = \min(1, \frac{p_s}{p_s} \frac{\frac{1}{|\phi'|} \frac{1}{|W|}}{\frac{1}{|\phi|} \frac{1}{|W|}} \frac{f(\phi')}{f(\phi)}) \quad (13)$$

$$= \min(1, \frac{f(\phi')}{f(\phi)}). \quad (14)$$

Modifying an Item Color (m_m):

$$\alpha(\phi'|\phi) = \min(1, \frac{p_m}{p_m} \frac{p(\theta_i|\theta_i')}{p(\theta_i'|\theta_i)} \frac{f(\phi')}{f(\phi)}) \quad (15)$$

$$= \min(1, \frac{f(\phi')}{f(\phi)}). \quad (16)$$

In our implementation, we simply set the prior distribution uniformly over the moves as $p_a = p_r = p_s = p_m = 0.25$.

Figure 12: *Outfit synthesis results for the models, associated items, and the 5-color palette. From top to bottom: "Mag" (Female, Cool), "Eddie" (Male, Cool), "Ce" (Female, Warm), "Jacen" (Male, Warm). From left to right:* Casual, Sportswear, Business-Casual *and* Business.

Figure 13: *Close-up views of populated virtual scenes with and without outfit consideration. (a) Outfits synthesized randomly; (b)–(c) Outfits synthesized under dress code* Business*; (d) Outfits re-synthesized after changing dress code to* Sportswear. *An unnatural appearance clearly results in the absence of a proper dress code.*

5 Results and Discussion

To demonstrate the efficacy of our optimization approach, we tested it on six different virtual human models, three males and three females. Figure 12 depicts two males and two females. The remaining characters were used in our perceptual study and can be found in the supplementary material. For the males, "Thor" has white skin and dark brown hair, "Eddie" has yellow skin and black hair, and "Jacen" has black skin and black hair. For the females, "Fiona" has white skin and blonde hair, "Mag" has yellow skin and black hair, and "Ce" has dark brown skin and black hair.

We synthesized all four test dress codes *Sportswear*, *Casual*, *Business-Casual* and *Business* for all the models. We optimized the male and female model outfits using the Bayesian networks learned for males and females, respectively. The clothing items are also segregated into male and female wardrobes. Each wardrobe contains about 10 clothing items for each of the 40 states in the Bayesian network, so there are about 400 clothing items in total. We simply used a budget of 250 optimization iterations for each outfit synthesis, which takes about 1-2 second per synthesis on a 3.33GHz Intel Xeon PC.

The final optimized outfits with the corresponding selected items are shown in Figure 12. We also show the corresponding 5-color palette alongside with the items. Mag and Eddie are classified as 'cool' and a 'cool' color palette was assigned to them prior to the optimization. Meanwhile Ce and Jacen are classified as 'warm'. The 'cool' and 'warm' color palettes are shown in Figure 2. For all the generated results, the color ratings are greater than 3.3.

The *dress code* as the root node determines the style of synthesis; i.e. what clothing items should be chosen and how they should combine. For example, in the 3rd row showing the synthesis for Ce, the same sweater is chosen for *Casual* and *Business*. However, the sweater is worn alone in *Casual*, but with a suit jacket in *Business*.

When we designed our Bayesian networks, we defined more than one node for the chest to permit the coexistence of different items. Several generated results reflect this property, which is important for creating variation. It happens more often for the dress code *Business*; for example, Eddie in the 2nd row wears a dress shirt, a vest, and a suit jacket for his upper body outfit, a combination which is occasionally observed in the *Business* training data.

Our outfit optimization can lead to two potential applications:

Outfit Suggestion Engine: The outfit suggestions can readily assist shoppers in boutique websites or fitting rooms, in which case the clothing items are those available in the store; or it can be used as a personal outfit advisor, in which case the clothing items are those available in the user's wardrobe. The support of efficient,

Figure 14: *Populating virtual scenes. Our approach can automatically suggest appropriate outfits to a large number of virtual characters. Dress codes* Sportswear *and* Casual *were used in accordance with the virtual beach scene.*

arbitrary probabilistic queries can handle scenarios commonly encountered in the clothes matching process. For example, conditional queries allow one to fix one or multiple clothing items and ask for multiple matching suggestions. Refer to Figure 7 for two examples. One can also change the preferred color palette, after which the optimizer will update the suggestion accordingly, as shown in Figure 9. As a personal outfit advisor, given a dress code, it can automatically suggest many decent outfits out of the user's wardrobe, thereby making full use of it. Refer to Figure 10 for two examples.

Virtual Character Modeling: Our approach is also useful for dressing human-like characters in large-scale virtual worlds, in which case the artist can specify dress codes and allow the computer to synthesize coordinated clothing combinations for each character in a fully automated manner. This can be readily incorporated on top of character modeling engines in gaming applications, which commonly support automatic clothes meshing on virtual characters[5], but lack support for reasoning about the many possible outfits out of the massive amount of clothing items available.

Figure 13 shows virtual scenes with and without outfit consideration. One can easily see that the scene appears unnatural if the characters are not properly dressed; e.g., donning a suit jacket, or wearing a dress in a gym, or dressing in sportswear in the office. Figure 14 shows a beach scene populated by approximately 100 virtual characters automatically dressed up in *Sportswear* and *Casual* dress codes. With our optimization, the characters are appropriately

---

[5]Examples include Playstation Home, XBox 360 Avatars (http://marketplace.xbox.com/en-US/AvatarMarketplace), Second Life, etc.

Figure 15: *Example images in Experiment 2 of the perceptual study. Left: Outfits synthesized with the corresponding dress codes. Right: Outfits synthesized without dress code consideration. Note that all syntheses considered the color cost terms.*

dressed in multiple ways to create variety suitable to the scene.

While we demonstrated our approach based on the four dress codes that are common nowadays, our framework offers the flexibility to cope with specific clothing styles matching a theme. An interesting example is for a massively-multiplayer online game featuring the Medieval Fantasy, in which case the node states can be replaced by medieval clothes and specific nodes such as "weapon" may be added. In this case, training examples may be collected directly from the player-created game characters, and our trained framework can be used to provide outfit suggestions in the character modeling engine used by new players, or for the automatic, realistic synthesis and dressing of non-player characters.

6 Perceptual Study

We performed a perceptual study to evaluate the functional and visual appearance of our outfit synthesis framework. Since comparisons of outfits are inherently subjective, one possible way is to evaluate our synthesis results against comparable results produced by human fashion designers. However, assessing metrics and performing pairwise comparisons is very difficult when there are significant differences, and they may not lead to meaningful conclusions. For example, a particular subject may be fond of some particular skirt and be biased in favor of women wearing this skirt.

The goal of our system is to synthesize visually reasonable or pleasing outfits under certain dress codes. To evaluate the efficacy of our approach, we must demonstrate that the clothing items enforce the selected dress code and that their colors are nicely coordinated. Since color coordination was extensively evaluated in [O'Donovan et al. 2011] and crowd perception as such was studied comprehensively in [O'Sullivan 2009], our perceptual study was focused on whether the matched clothes are functionally sound individually. We attempted to verify the following two conditions, by two experiments: First, a classification experiment to testify the outfit recommendations that our system produces successfully reflect the dress code and, hence, validate our Bayesian network training. Second, a discrimination experiment to verify that the incorporated dress code yields a benefit over outfit synthesis results obtained in its absence.

Similar to those of other authors [Jagnow et al. 2008; Jimenez et al.

2009; Yeung et al. 2011; Yu et al. 2011], our experiments were conducted using a subjective, five-alternative/two-alternative, forced-choice preference approach. In Experiment 1, our null hypothesis H_0 was that users cannot recognize the dress code of the syntheses for each category; i.e., recognition rate is at chance level. In Experiment 2, our null hypothesis H_0 was that users show no preference among the syntheses with and without dress code consideration.

Participants: 32 volunteer participants were recruited who were unaware of the purpose of the perceptual study. This number of participants was comparable with similar studies in which 16 users were recruited [Jagnow et al. 2008; Jimenez et al. 2009]. The participants included 16 males and 16 females whose ages ranged from 20 to 60. All the subjects reported normal or corrected-to-normal vision with no color-blindness and reported that they are familiar with the dress codes to be tested in the study. 29 subjects reported that they did not have any expertise in fashion design.

Data: We picked 4 virtual models to cover both genders: Thor and Jacen are male, Fiona and Mag are female. For each virtual model, we synthesized 20 outfits (5 per dress code) with the complete objective function, and 20 outfits with an objective function lacking the style cost term. Figure 12 depicts example matching results with their associated items used in the user study. For the pairwise comparison, examples are shown in Figure 15. With multiple outfits per dress code we can create variety in the comparisons. The images used in perceptual study are listed for visual inspection in the supplementary materials.

Procedure: The study was conducted in two experiments. Participants were encouraged to ask any question prior to the study. After completing a consent form and questionnaire, they were given a sheet detailing the task descriptions.

Experiment 1 (Classification): The main goal was to test whether our generated results reflect the corresponding dress code faithfully and, hence, verify our Bayesian network encoding. To achieve this, we asked the subject whether the synthesized clothing combinations fall into any of our encoded dress codes:

> "This experiment involves selecting a dress code from an image of a dressed model. There are 80 images.
>
> Your task in each evaluation is to select one of the following dress codes which you feel best describes the outfit shown in the image: *Sportswear*, *Casual*, *Business-Casual*, *Business*, and *Other* if the image does not match any of the previous four. You can view the test image for an unlimited amount of time, but we suggest that you spend around 15 seconds on each image before making your selection."

Experiment 2 (Discrimination): The main goal was to evaluate if incorporating the style cost term really shows a significant preference on the functionality of the outfit compared to outfits synthesized without consideration of a dress code:

> "This experiment involves selecting a dressed model from a pair of images, and there are 160 pairs in total. You will be shown the images side-by-side with a grey image displayed between each evaluation.
>
> Your task in each evaluation is to select the model based on their outfit in which you would prefer to dress for a particular occasion which is depicted in the top of the image pairs: *Casual*, *Sportswear*, *Business-Casual*, or *Business*. You can view the test pair for an unlimited amount of time, but we suggest that you spend around 15 seconds on each pair before making your selection."

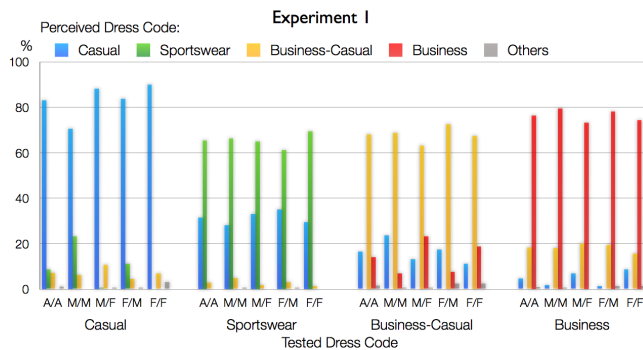Each participant viewed a total of 160 trials (4 models × 4 dress

Figure 16: *Recognition rates of Experiment 1: Perceived dress code versus tested dress code. "A/A": All participants perceiving all syntheses. "M/F": Male participants perceiving female syntheses. Similar for "M/F", "F/M" and "F/F". All recognition rates are significantly above chance level.*
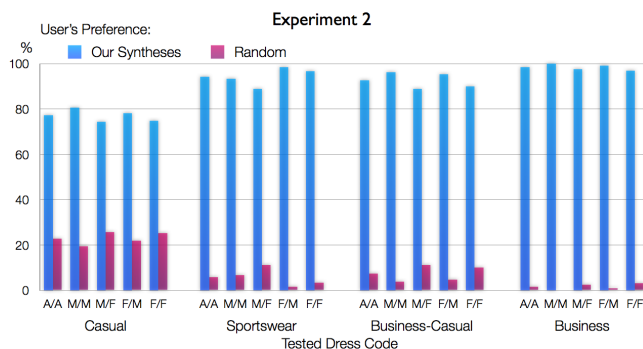

Figure 17: *User's preference of Experiment 2: Our syntheses versus random syntheses. The rates of picking our syntheses are significantly above chance level.*

codes × 5 pairs × 2 trials). Each pair comprises a full objective result and a result randomly chosen among those synthesized without considering the style cost term. The pairs were presented to each participant in a different random order. Counterbalancing was used to avoid any order bias—each paired comparison was assessed twice by each participant: In half of the trials the full objective result was displayed on the left side and in the other half on the right.

Results and Analysis: Figure 16 shows the correct recognition rates of Experiment 1. We display the results by gender of the participants versus the gender of the syntheses. Overall, the correct recognition rates are: *Casual* (83.125%), *Sportswear* (66.875%), *Business-Casual* (67.969%), *Business* (76.25%). The detailed recognition rates tabulated by gender can be found in the supplementary materials.

Figure 16 also shows some interesting observations. While all correct recognition rates were significantly above chance, *Sportswear* and *Business-Casual* have lower recognition rates. A certain portion of *Sportswear* was perceived as *Casual*, while a certain portion of *Business-Casual* was perceived as *Business* and *Casual*, respectively. This is probably because in reality, the perception of different dress codes can be ambiguous and may overlap; e.g., some people may regard a subset of *Sportswear* and a subset of *Business-Casual* also as belonging to *Casual*, which tends to be more frequently chosen as a result and received higher recognition rates. This also accounts for the recognition rates of *Business* that have minor portions perceived as *Business Casual*, which if added up together should give rates over 90% for both genders.
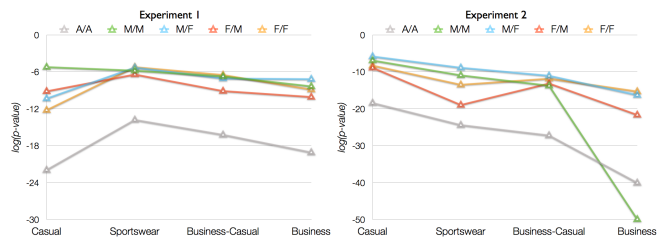

Figure 18: *Results of t-tests against chance for Experiment 1 (left) and Experiment 2 (right) shown as log(p-value). Notations are the same as in Figure 16. Test for A/A has $d.f. = 31$. Other tests have $d.f. = 15$. All tests have $log(p\text{-}value) < -5$ which is equivalent to p-value < 0.00001.*

With respect to gender difference, we note that men's *Business* tend to be more definitive than women's *Business*, with slightly higher correct recognition rates on men's *Business* syntheses and less men's *Business-Casual* syntheses being perceived as *Business*. On the contrary, men's *Casual* tend to have more overlap with *Sportswear* perceptually. The *Casual* plot shows a certain portion of men's *Casual* being perceived as *Sportswear*, while this is rarely the case for women's *Casual*. Finally, we note that male and female participants tend to give similar response trends in classification.

Figure 17 depicts the results of Experiment 2 by comparing the rates of choosing our synthesized outfit and random syntheses. In all the cases our syntheses are much more preferable than random syntheses. Notice that the relatively lower recognition rates on dress code *Casual*, which is not surprising due to its less restrictive nature. To ascertain that our results are significant, we performed t-tests against chance in both experiments. Figure 18 summarizes the p-values. In all cases, we have p-values less than 0.00001, which are very small. Therefore, we reject the null hypothesis H_0 in both experiments. For Experiment 1, this concludes that subjects can correctly recognize the dress code of the syntheses as one of the 4 encoded dress codes. For Experiment 2, this concludes that subjects also prefer the syntheses that include dress code consideration.

7 Conclusion and Future Work

We have introduced an automated framework for outfit synthesis, which is a highly practical topic both in daily life and computer graphics. Our approach optimizes outfits in a way similar to real-world situations. The body color tone classifier automates the classification pre-process in fashion practices, avoiding cumbersome, obscure, manual classification. From the user's perspective, our framework is highly intuitive in practical use. On the one hand, if one fixes item colors and permits only addition, removal, or swapping moves during optimization, one is mimicking the scenario of a fixed wardrobe, and the optimizer jointly considers style and color when synthesizing outfits out of the available clothing items. On the other hand, if one permits the changing of certain clothing item colors, this is similar to buying new clothes, and it is particularly useful for populating virtual worlds with characters that exhibit realistic sartorial variety.

Currently, we have incorporated four different dress codes into our outfit synthesis system, but our learning and synthesis framework is flexible enough to accommodate additional criteria such as season, texture pattern, clothing shape, age, body proportion, or even associating outfits with multiple dress codes during training. For simplicity, we assumed each clothing item is represented by its dominant color. More sophisticated representations, such as representing each clothing item with an arbitrary number of colors (e.g., one color for a plain shirt and two colors for a checkered shirt),

can be readily handled by our RJMCMC formulation, which flexibly allows varying the number of dimensions. On the other hand, the color palette suggestion is motivated from the fashion literature, and it is easy for users to change according to their own preference; e.g., using a more colorful palette for a festive occasion; or replacing the color palette with one tailor-made by a fashion professional for a specific client; or trained by large-scale commercial datasets. Our framework is novel in formulating the seemingly abstract fashion matching problem as a combinatorial optimization problem in which style and color are jointly considered. Speeding up the automated outfit synthesis process for crowds through parallel computation, and a comprehensive human perceptual study on different outfits are additional interesting avenues for future work.

Acknowledgements

References

ANDRIEU, C., DE FREITAS, N., DOUCET, A., AND JORDAN, M. I. 2003. An introduction to mcmc for machine learning. *Science 50*, 1, 5–43.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 43–54.

CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3D modeling. *ACM Transactions on Graphics 30*, 4.

COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 25*, 3, 624–630.

CRISTIANINI, N., AND SHAWE-TAYLOR, J. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. *ACM Trans. Graph. 29*, 4.

DOBBYN, S., MCDONNELL, R., KAVAN, L., COLLINS, S., AND O'SULLIVAN, C. 2006. Clothing the Masses: Real-Time Clothed Crowds With Variation. In *Eurographics Short Papers (EG'06)*, 103 – 106.

FENG, W.-W., YU, Y., AND KIM, B.-U. 2010. A deformation transformer for real-time cloth animation. *ACM Trans. Graph. 29*, 4.

FISCHER-MIRKIN, T. 1995. *Dress Code: Understanding the Hidden Meanings of Women's Clothes*. Clarkson Potter, New York, NY.

FLUSSER, A. J. 2002. *Dressing the Man: Mastering the Art of Permanent Fashion*. HarperCollins.

FRIEDMAN, N., GEIGER, D., AND GOLDSZMIDT, M. 1997. Bayesian network classifiers. *Mach. Learn. 29*, 2-3 (Nov.), 131–163.

GILCHRIST, R. A. 2011. *The Encyclopedia of Men's Clothes*.

GODSILL, S. J. 2001. On the relationship between markov chain monte carlo methods for model uncertainty. *Journal Of Computational And Graphical Statistics 10*, 2, 1–19.

GREEN, P. J. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika 82*, 711–732.

GREEN, P. J. 2003. *Highly structured stochastics systems*. Oxford University Press, Oxford, UK, ch. Trans-dimensional Markov Chain Monte Carlo, 179–196.

GUAN, P., REISS, L., HIRSHBERG, D. A., WEISS, A., AND BLACK, M. J. 2012. Drape: Dressing any person. *ACM Trans. Graph. 31*, 4, 35.

HASTINGS, W. 1970. Monte Carlo samping methods using Markov chains and their applications. *Biometrika*, 97–109.

HENDERSON, V., AND HENSHAW, P. 2008. *Color Me Confident: Change Your Look - Change Your Life!*. Hamlyn, London, UK.

JACKSON, C., AND LULOW, K. 1984. *Color for Men*. Ballantine Books, New York, NY.

JACKSON, C. 1987. *Color Me Beautiful*. Ballantine Books, New York, NY.

JAGNOW, R., DORSEY, J., AND RUSHMEIER, H. 2008. Evaluation of methods for approximating shapes used to synthesize 3D solid textures. *ACM Trans. Appl. Percept. 4*, 4, 1–27.

JIMENEZ, J., SUNDSTEDT, V., AND GUTIERREZ, D. 2009. Screen-space perceptual rendering of human skin. *ACM Trans. Appl. Percept. 6*, 4, 1–15.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2008. Simulating knitted cloth at the yarn level. *ACM Trans. Graph. 27*, 3.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2010. Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph. 29*, 4.

KAVAN, L., GERSZEWSKI, D., BARGTEIL, A. W., AND SLOAN, P.-P. 2011. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph. 30*, 4, 93.

KOLLER, D., AND FRIEDMAN, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

MCDONNELL, R., DOBBYN, S., AND O'SULLIVAN, C. 2006. Crowd Creation Pipeline for Games. In *International Conference on Computer Games (CGames'06)*, 183 – 190.

MCDONNELL, R., LARKIN, M., DOBBYN, S., COLLINS, S., AND O'SULLIVAN, C. 2008. Clone attack! perception of crowd variety. *ACM Trans. Graph. 27*, 3.

MCDONNELL, R., LARKIN, M., HERNÁNDEZ, B., RUDOMÍN, I., AND O'SULLIVAN, C. 2009. Eye-catching crowds: saliency based selective variation. *ACM Trans. Graph. 28*, 3.

MERRELL, P., SCHKUFZA, E., AND KOLTUN, V. 2010. Computer-generated residential building layouts. *ACM Trans. Graph. 29*, 181:1–181:12.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. 1953. Equation of state

calculations by fast computing machines. *Journal of Chemical Physics 21*, 1087–1092.

NICHOLSON, J. 2003. *Dressing smart for women: 101 mistakes you can't afford to make and how to avoid them.* Impact Publications, Manassas, VA.

O'DONOVAN, P., AGARWALA, A., AND HERTZMANN, A. 2011. Color compatibility from large datasets. *ACM Trans. Graph. 30*, 4, 63.

O'SULLIVAN, C. 2009. Variety is the spice of (virtual) life. In *MIG*, 84–93.

PEARL, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

PROVOT, X. 1995. Deformation constraints in a mass–spring model to describe rigid cloth behavior. In *Graphics Interface '95*, 147–154.

SCHNEIDER, J. J., AND KIRKPATRICK, S. 2006. *Stochastic Optimization (Scientific Computation).* Springer-Verlag, Berlin.

SCHOEFFLER, O. E., AND GALE, W. 1973. *Esquire's encyclopedia of 20th century men's fashions.* McGraw-Hill, New York, NY.

SONDAG, G. R. 2011. *Anything Other Than Naked - A guide for men on how to dress properly for every occassion.* Two Harbors Press.

TALTON, J. O., LOU, Y., LESSER, S., DUKE, J., MECH, R., AND KOLTUN, V. 2011. Metropolis procedural modeling. *ACM Trans. Graph. 30*, 2, 11.

TECCHIA, F., LOSCOS, C., AND CHRYSANTHOU, Y. 2002. Image-based crowd rendering. *IEEE Computer Graphics and Applications 22*, 36–43.

TERZOPOULOS, D., AND FLEISCHER, K. W. 1988. Deformable models. *The Visual Computer 4*, 6, 306–331.

TERZOPOULOS, D., PLATT, J. C., BARR, A. H., AND FLEISCHER, K. W. 1987. Elastically deformable models. In *SIGGRAPH*, 205–214.

THALMANN, D., O'SULLIVAN, C., YERSIN, B., MAM, J., AND MCDONNELL, R. 2007. EG 2007 Course on Populating Virtual Environments with Crowds . 23–123.

TSUJITA, H., TSUKADA, K., KAMBARA, K., AND SIIO, I. 2010. Complete fashion coordinator: a support system for capturing and selecting daily clothes with social networks. In *Proceedings of the International Conference on Advanced Visual Interfaces*, ACM, New York, NY, USA, AVI '10, 127–132.

TU, Z., AND ZHU, S.-C. 2002. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. Pattern Anal. Mach. Intell. 24* (May), 657–673.

ULICNY, B., CIECHOMSKI, P. D. H., AND THALMANN, D. 2004. Crowdbrush: Interactive authoring of real-time crowd scenes. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '04, 243–252.

UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph. 30* (Aug.), 90:1–90:12.

VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph. 28*, 4.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. F. 2010. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph. 29*, 4.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM Trans. Graph. 30*, 4, 71.

YEUNG, S. K., TANG, C.-K., BROWN, M. S., AND KANG, S. B. 2011. Matting and compositing of transparent and refractive objects. *ACM Transactions on Graphics 30*, 1, 2.

YU, L.-F., YEUNG, S. K., TANG, C.-K., TERZOPOULOS, D., CHAN, T. F., AND OSHER, S. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics 30*, 4, 86.

ZYLA, D. 2010. *The Color of Style.* Dutton Adult, New York, NY.