This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Inverse Procedural Modeling of Knitwear

Elena Trunz, Sebastian Merzbach, Jonathan Klein, Thomas Schulze
Michael Weinmann, Reinhard Klein
Institute of Computer Science II, University of Bonn, Germany
{trunz,merzbach,kleinj,mw,rk}@cs.uni-bonn.de, s6tsschu@uni-bonn.de

Abstract

The analysis and modeling of cloth has received a lot of attention in recent years. While recent approaches are focused on woven cloth, we present a novel practical approach for the inference of more complex knitwear structures as well as the respective knitting instructions from only a single image without attached annotations. Knitwear is produced by repeating instances of the same pattern, consisting of grid-like arrangements of a small set of basic stitch types. Our framework addresses the identification and localization of the occurring stitch types, which is challenging due to huge appearance variations. The resulting coarsely localized stitch types are used to infer the underlying grid structure as well as for the extraction of the knitting instruction of pattern repeats, taking into account principles of Gestalt theory. Finally, the derived instructions allow the reproduction of the knitting structures, either as renderings or by actual knitting, as demonstrated in several examples.

1. Introduction

Fabrics are an essential matter in our daily life. In contrast to their woven counterparts, knitted clothing visually sticks out due to complicated underlying stitch structures formed by various knitting operations, each inducing a characteristic appearance. Furthermore, knitting clothing of various types still belongs to a handcraft mastered by a rather large group of our society covering all ages. One reason for this may be the interest in manufacturing one's own clothing with knitting patterns following the individual subjective preferences. While there are books and websites that provide a wide range of patterns together with their respective construction instructions, it would be desirable to be able to reproduce patterns from images provided e.g. by standard search engines such as *Google*, which lack the corresponding knitting instruction.

Unfortunately, inferring the underlying knitting patterns by only "reading" a single image is particularly challenging, even for experts. The visual appearance of stitches exhibits a large variety as neighboring stitches may occlude them or cast shadows. The variability in the appearance of the basic stitch types is further increased by the properties of the used yarns, such as their thickness, type of yarns, etc., or the individual knitting styles of different people leading to various deformations, such as stretchings, holes and tight or loose stitches. Therefore, even experts often prefer analyzing the respective physical clothing pieces by stretching it and performing the inspection from both sides, in order to reliably infer the knitting instructions, including otherwise covered stitches. These manipulations are not possible when analyzing knitted fabrics in single photos.

Inverse procedural modeling of objects from only a few or even single examples has received a lot of attention in the last decade. Corresponding applications encompass the derivation of production rules for plants, woven fabrics, buildings and facades. However, the developed approaches are custom tailored for the corresponding applications and cannot be easily transformed to infer knitting patterns.

In this paper, we direct our attention to the inference of the complicated structures of knitwear and the derivation of the respective knitting instructions, to the best of our knowledge, for the first time from only a single image without annotations. This implies solving the labeling problem, i.e. the identification of the occurring stitch types as well as their proper localization from the visually complex appearance depicted in the input photographs. For this purpose, we introduce a novel pipeline that involves four major components represented by (1) the search for the individual stitch types across the image, (2) the inference of the underlying grid structure from the coarsely localized stitches from the previous step, (3) an error correction and pattern size detection step that determines the size of the desired pattern and corrects the labeling errors from the first step, and, finally, (4) the derivation of the final knitting instruction (in analogy to instructions in knitting books), based on the found pattern size and corrected underlying grid structure, taking into account the intuition of human perception by applying the Law of Symmetry and the Law of Prägnanz [5]. These

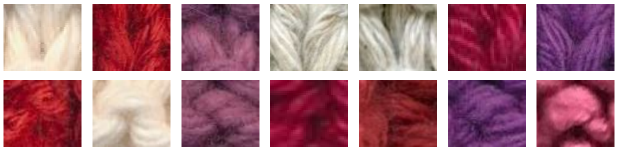

Figure 1: Exemplary appearance variations of knits (top) and purls (bottom).

derived instructions allow the reproduction of the knitting patterns with possibly different yarn types as demonstrated in several examples.

In summary, the key contributions of this paper are:

- A novel method for inverse procedural modeling of knitwear from a single image.
- The derivation of the underlying optimal regular grid structure from initially determined hypotheses regarding the coarse localization of stitch types.
- An error correction technique that determines the correct size of the knitting pattern and corrects possible recognition errors.
- A final induction of the knitting instruction from the derived grid structure following human intuition.

2. Background

Knitting relies only on a relatively small set of basic actions, and their combinations allow to generate the underlying knit structures for various patterns [34]. Therefore, even children can learn to produce caps, scarfs or potholders. These basic actions and their combinations result in a number of stitch types, which are used to generate a wide range of various knitting designs by repetitions and different orderings. In this paper, we focus on the two fundamental stitch types: *knit* and *purl* (see Figure 1).

The usual shape of a knit resembles the structure of a ”v”. If there is a purl stitch above or below (or both), a knit stitch becomes partially covered by the purl stitch(es). Additionally, the width of the stitch gets smaller than it would be if it had other knits as an upper and/or lower neighbor. The purl stitch usually resembles a wave structure. This wave becomes wider if there are knit stitches underneath and above the purl. However, in case there is a knit stitch on the right or the left side (or both) of a purl, then the purl stitch gets partially covered. Figure 1 illustrates some of the appearance/shape variations of purls and knits. Note that the shown stitch variations are solely induced by arranging the basic stitch types in different orderings. Additionally, these appearance variations heavily depend on the properties of the used yarn, as well as stitch deformations resulting from subjective knitting styles, making almost every stitch of a hand-made piece of knitted fabric individual.

3. Related Work

The major components of our framework for inverse modeling of knitwear include the search for occurrences of basic stitch types as marked by the user within the image, the inference of a grid structure based on the stitch candidates and underlying repeating patterns. As a consequence, we briefly review the developments in the areas of template matching and inverse procedural modeling. We refrain from a detailed discussion regarding the visualization of knitted fabrics, but only refer to the work by Yuksel et al. [34].

Template matching: Traditional techniques for efficiently searching a query patch within an image are usually based on using the Sum-of-Squared-Distances (SSD), the Sum-of-Absolute-Distances (SAD) or Normalized Cross-Correlation (NCC). Subsequent works addressed their lacking robustness towards handling noise [10] and illumination changes [13]. Further improvements came with the use of robust error functions [6, 22, 21, 18]. Later, Barnes et al. [2, 3] introduced the PatchMatch algorithm for nearest neighbor matching across translations, rotations and scales. However, all of these techniques only allow a one-to-one mapping between a template and the query region and rely on a strict rigid geometric deformation between the template patch and the target patch. As a consequence, they are not capable of dealing with the geometric deformations we expect for patches containing knitting primitives (knits and purls). Towards handling appearance variations for material recognition, other approaches rely on the matching of histograms extracted for different images by considering various descriptors (e.g. [20]) within classification frameworks. Furthermore, set-based matching has been explored to allow a more robust matching of textures based on the consideration of the appearance space of textures [14, 31].

Other approaches have been designed to explicitly handle parametric deformations such as 2D affine transformations [16] or more general non-rigid distortions [29]. However, despite the requirement of a parametric distortion model for the underlying geometry, these techniques also rely on the assumption of a one-to-one mapping between the query and target patch, which is susceptible to errors in the presence of occlusions or background clutter.

Further work explores the bi-directional similarity between target and query patch. Simakov et al. [23] represent images in terms of a set of patches and the considered bi-directional similarity (BDS) measure considers the sum of distances between a patch in the first image and its nearest neighbor in the second image and vice versa. To also distinguish between inliers and outliers arising from foreground/background parts of the considered patches, the Best Buddies Similarity (BBS) has been proposed [7], based on counting the Best Buddy Pairs and, hence, using the actual distances only implicitly. Therefore, an increased robustness in comparison to BDS has been achieved. Talmi et al. [26] extended this work by enforcing diversity in the mutual nearest-neighbor matching and explicitly considering the deformation of the nearest-neighbor field. To achieve a speed up of the matching process, they use an approximate nearest neighbor search.

While any of the template matching techniques could be applied to derive probability maps for the localization of certain basic primitives required in our approach, we use template matching based on BBS due to its proven robustness to deformations that are expected to occur for the primitives of knitting. We improved the BBS technique by the use of additional gradient information. In the evaluation, we compare several template matching techniques and show that the extended BBS approach outperforms the other techniques in the context of our particular problem.

Image-based detection of weave patterns: Cloth modeling has received a lot of attention so far. Especially approaches for detecting weave patterns from images are closely related to our work. In particular, the complete reverse-engineering of woven cloth at the yarn level as approached by Schröder et al. [19] and Guarnera et al. [11] has been demonstrated to be the current state-of-the-art technique. While these approaches are powerful for the analysis of woven cloth, they are not designed to handle knitted textiles. Knitted clothing is inherently 3D and the final shapes of stitches, especially hand-made stitches, do not possess the similarity and regularity of warp and weft of woven cloth, where occlusions and non-rigid deformations of the yarn have to be taken into account in order to be able to find the actual position and the type of the stitches in the image. To the best of our knowledge, we are the first to tackle the problem of detecting knitting patterns and the respective knitting instructions from a single image.

Inverse procedural modeling: Inverse procedural modeling (IPM) is the problem of inferring a set of parameters [28, 30] or even a whole procedural description for a given model. Early investigations on applying inverse procedural modeling for graphics applications include the works on 3D meshes [4] and 2D vector designs [24], but there has been a lot of progress in this area of research. Meanwhile, inverse procedural modeling is widely used and has been applied for varying purposes ranging from the inference of 3D design patterns [27] over the modeling of plants [25, 17] to editing of building point clouds [8] as well as inferring procedural descriptions of building facades [32, 33, 9, 17] and reverse engineering of woven cloth [19]. For a detailed and extensive survey on inverse procedural modeling, we refer to the report by Aliaga et al. [1].

4. Stitch Pattern Inference Approach

In this section, we introduce our approach to infer knitting patterns and the respective instructions for their generation from single input images. An initial pre-processing step compensates for non-axis-alignment of the depicted knit patterns and, hence, makes our framework capable of handling tilted images of knitwear. In the next step, the user provides exemplars of particular basic stitch types, such as knits or purls within the image via an intuitive interface. Subsequently, image patches containing these stitch types are searched within the whole image and the resulting coarse localization of the individual stitch types is used to infer the underlying grid structure. Furthermore, an error correction procedure allows to compensate for possible misclassifications of the stitch types in the grid and detects the size of the repeating pattern. The found size and the optimized grid structure are then used to find the starting position of the pattern, thus finalizing the process of stitch pattern inference. Finally, we derive the underlying production rules and convert them into corresponding knitting instructions that allow the reproduction of the knitting pattern depicted in the input image. Details regarding the involved components are described in the following sections.

4.1. Pre-Processing

Before allowing the user to specify templates for the relevant stitch types in the image, we perform a pre-processing step to facilitate the annotation process. To compensate for deviations from axis-alignment, we use Histograms of Oriented Gradients (HOG) to determine the most dominant directions in the photo, which is justified due to the inherent grid structure resulting from the production process. This allows the reversal of rotations to align the grid structure with the axes and, hence, makes our algorithm capable of also handling non-axis-aligned input patterns. Respective examples are shown in the supplemental material.

4.2. Interactive Selection of Relevant Stitch Types

The detection of stitch types could possibly be approached with a completely automatic pipeline. However, this would require huge annotated databases depicting the possibly occurring stitch types with various stitch neighborhoods and distortions with yarns of different properties (yarn thickness, reflectance behavior, etc.) under different illumination conditions. As such databases, to the best of our knowledge, are not yet publicly available, we refrain from relying on a completely automatic approach to detect stitch types across the image based on machine learning techniques. Instead we let the user guide the search for stitch types by providing a single template for the individual stitch types occurring in the input image, in order to keep user interaction as minimal as possible. For this purpose, we implemented an easy-to-use interface that allows the user to choose a sample for each stitch type by simply drawing a rectangle over a stitch. In turn, considering the possibly strong variations of the occurring stitches requires a subsequently applied robust template matching technique.

4.3. Derivation of Stitch Localization Hypotheses

Finding certain stitch types in an image is complicated because their appearance may significantly vary due to partial occlusions by neighboring stitches, variances in the used yarn types including their reflectance behavior, thickness and hairiness, as well as variations induced by the individual knitting style during manufacturing, manifested in deformations like tight or loose stitches. To be able to find stitch types across the image based on a given template, handling distortions and partial matches becomes an essential prerequisite for the derivation of hypotheses of where the respective stitch types are found.

Best Buddies Similarity (BBS) based template matching [7] has been designed towards these goals of matching distorted and partially occluded patterns and proven to outperform most previous techniques in this regard. We therefore apply this technique for the detection of hypotheses for the individual stitch types such as knits or purls. Following Dekel et al. [7], the BBS between two point sets $P = \{p_i\}_{i=1}^N$ and $Q = \{q_i\}_{i=1}^M$ extracted from a local image region and a template is defined according to

$$BBS(P,Q) = \frac{1}{\min(M,N)} \sum_{i=1}^{N} \sum_{j=1}^{M} bb(p_i, q_j, P, Q). \quad (1)$$

Here,

$$bb(p_i, q_j, P, Q) = \begin{cases} 1, & \text{if } \mathrm{NN}(p_i, Q) = q_j \\ & \quad \text{and } \mathrm{NN}(q_j, P) = p_i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

acts as an indicator function influenced by the nearest neighbor definition

$$\mathrm{NN}(p_i, Q) = \mathrm{argmin}_{q \in Q} d(p_i, q) \quad (3)$$

and the distance measure

$$\begin{aligned} d(p,q) = \quad & ||p_i^{(A)} - q_j^{(A)}||_2^2 + \lambda_G ||p_i^{(G)} - q_j^{(G)}||_2^2 \\ & + \lambda_L \, ||p_i^{(L)} - q_j^{(L)}||_2^2. \end{aligned} \quad (4)$$

In comparison to the original implementation [7], we extend the RGB-based appearance (A) and spatial distance (L) within the patch with an additional gradient constraint (G), that enforces similar gradients within the patches. Based on several examples, we determined $\lambda_G = 100$ to be suitable for our purpose, and otherwise follow the original implementation in using $\lambda_L = 2$ and a decomposition of image and template into $k \times k$ patches with $k = 3$.

As a result, we obtain BBS likelihood maps that indicate where the respective stitch types are (coarsely) localized. Finally, we merge the likelihood maps obtained for the different stitch types to a resulting likelihood map that contains the maximum likelihood of the individual stitch types obtained per pixel as well as the corresponding most likely stitch type. An example of these maps is shown in Figure 2.

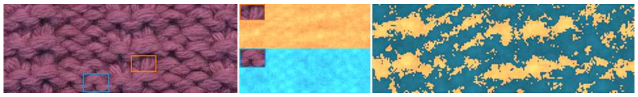

Figure 2: User-specified stitch templates (left) and corresponding likelihood maps (middle). The likelihood value is indicated with the colorization, i.e. the lighter the spot the higher the probability. The image on the right depicts the maximum likelihood map including the assignments to the stitch types (knit = orange, purl = blue).

4.4. Inference of Grid Structure of Stitches

The maximum likelihood map retrieved in the last step contains the coarse per-pixel likelihood regarding the local presence of respective stitch types. From this coarse localization we need to determine the fine-grained arrangement and the corresponding classes of the individual stitches. For this purpose, we exploit the presence of an underlying grid-like structure induced by the knitting process to account for the fact that the spatial extension of the individual stitches constrains their locations. In general, the latter will not be equidistant and the grid may exhibit significant distortions due to the non-ideal man-made manufacturing process or the respective treatment of the fabric. To model this behavior, we associate the centers of the stitches with a set of labeled points arranged in a 2D grid-like structure. These points have to fulfill the following properties:

- Each point is assigned a high likelihood of representing a certain stitch type such as knits or purls (P1).
- Neighboring points must preserve a minimal distance (on the order of magnitude of a stitch prototype) to each other (P2).
- Adjacent points cannot be further apart than the maximal extension of a stitch type (P3).
- The set of points has the structure of a regular approximately rectangular grid (P4).

Finding the optimal set of points fulfilling the above stated properties can then be formulated in terms of a point selection problem.

4.4.1 Stitch Localization as Point Selection Problem

To infer the positions of the centers of the individual stitches, we solve a point selection problem that can be formulated in terms of an integer linear program (ILP). Let P denote the set of all possible points (pixels) of the input image. Furthermore, let P_{opt} be the point set corresponding to the solution of our optimization problem. Denoting the likelihood value of a pixel $p_i \in P$ to be assigned to a certain stitch type according to the likelihood map from the previous step with $score(i)$ and using the binary variables

$$o_i = \begin{cases} 0 & \text{if } p_i \notin P_{opt} \\ 1 & \text{if } p_i \in P_{opt} \end{cases} \tag{5}$$

that determine whether a point p_i is assigned to the optimal solution, we maximize the functional

$$\sum_{p_i \in P} score(i)\ o_i \tag{6}$$

subject to the constraints following from the aforementioned properties P2, P3 and P4. To ensure the properties P2 and P3, we determine for each pixel $p_j \in P$ two corresponding rectangular regions $P_j^{min} \subset P$ and $P_j^{max} \subset P$ that represent the uncertainty in the location of neighboring points in the grid structure . P_j^{min} has the width w_{min} and the height h_{min} of the estimated minimal extension of the stitch and P_j^{max} has the width w_{max} and the height h_{max} of the estimated maximal extension. The computation of the corresponding extension size values and the uncertainties is discussed in Section 4.4.3. To account for the property P2, we constrain each region P_j^{min} to contain *at most* one optimal point $p \in P_{opt}$ and, to account for the property P3, each region P_j^{max} is constrained to have *at least* one point $p \in P_{opt}$, i.e.:

$$\sum_{\forall i: p_i \in P_j^{min}} o_i \leq 1 \quad \text{and} \quad \sum_{\forall i: p_i \in P_j^{max}} o_i \geq 1 \quad \forall p_j \in P. \tag{7}$$

In order to force the points of the optimal solution to have a grid-like structure (P4), we subdivide the input image into $r \cdot c$ grid cells $G_k \subset P$ with the width $\frac{w_i}{c}(1 - u_w)$ and the height $\frac{h_i}{r}(1 - u_h)$, where w_i and h_i denote the width and the height of the image respectively and c and r denote the number of rows and columns of the grid. These are precomputed, as described in Section 4.4.2. With u_w and u_h, we denote the uncertainties in the spatial extension of the stitches in x and y direction respectively, which are used here to allow the overlap of the cells. The values of u_w and u_h are computed as described in Section 4.4.3. We constrain the optimal solution to contain at least one point in each grid cell. Furthermore, the number of points in the solution is constrained to be equal to $r \cdot c$. Both constraints ensure (P4) while allowing for overlapping cells:

$$\sum_{\forall i: p_i \in G_j} o_i \geq 1 \qquad \forall G_j \in G, \tag{8}$$

$$\sum_{\forall i: p_i \in P} o_i = r \cdot c. \tag{9}$$

G denotes the set of all grid cells. To solve this ILP, we use the Gurobi solver [12]. From the points contained in the resulting optimal solution we construct the grid in the following manner: We sort all points according to the x coordinates of the pixels and assign to each row of the grid c points. The stitch types assigned to the individual points are stored in a matrix $M^{r \times c}$.

4.4.2 Computing the Number of Rows and Columns

To determine the number of columns in the grid, we use the position of one of the stitch samples selected by the user during the initial step of the inference approach and select the region around the stitch position within the likelihood map. The height of the selected region corresponds to the height of the selected template with some additional tolerance and the respective width is given by the image width. To account for possible distortions, the height is allowed to deviate up to u_y in each direction from the center of the chosen stitch sample (in our experiments, we use $u_y = 25\%$). Using the data of this truncated map, we apply a similar ILP formulation as before with slight changes. In contrast to the optimization described before, the point variables consist of the pixels from the chosen strip. The objective functional and all constraints except for the row and the column number constraints remain unmodified. The number r of rows is set to 1. Now we compute the possible minimal value of the number of columns c as $c_{min} = \frac{w_i}{W_{max}}$, where W_{max} denotes the maximal width of both templates, since we do not yet know the correct stitch labelings of this strip. To account for possible stitch occlusions, we compute the maximal value of c as $c_{max} = 2\frac{w_i}{W_{min}}$, where W_{min} denotes the minimal width of both templates. We iterate through the possible numbers of columns from c_{min} to c_{max} and divide the resulting objective function of each optimal solution by the current number of columns. Finally, we obtain the number c of columns corresponding to the largest value of the normalized objective function as the optimal solution. The number of rows is determined accordingly.

4.4.3 Uncertainty in the Locations of Adjacent Stitches

Because of occlusions and different deformations the stitches vary from each other in size. Additional variations are induced by the use of different yarn types and inconsistencies of the knitter. We implicitly take these aspects into account by analyzing the strips extracted from the previous step to estimate uncertainties in the spatial extensions of the stitches. First, we compute the average width w_a and height h_a of the stitches taken from the four strips (two for each sample). By computing the maximum absolute deviation for the width (d_w) and height (d_h) separately, we get the uncertainties $u_w = \frac{d_w}{w_a}$ and $u_h = \frac{d_h}{h_a}$, yielding the values $w_{min} = w_a - u_w$ and $w_{max} = w_a + u_w$ for the width and analogous values h_{min} and h_{max} for the height, which are then used for the optimization.

For the computation of the number of rows and columns we use the average of the actual sizes of the templates selected by the user and set the values of uncertainties u_x and u_y to be equal to 25% of the average template size each. Note that large uncertainty values result in an increasing number of variables during the optimization, hence significantly increasing computational time.

4.5. Error Correction and Repeat Size Detection

After the inference of the underlying grid structure M from the previous step, we aim at finding an intuitive repeating pattern of minimal size. For this purpose, we assume that the knitting pattern of interest is at least twice contained completely within the image, but unfinished repeats may occur as well. In order to find the pattern, we first find the correct size and subsequently determine the starting position of the pattern. As the extracted matrix M of stitch types resulting from the grid optimization step might still contain some wrongly recognized stitch types, the identification of the pattern size in the matrix M as well as the potentially required error correction have to be conducted simultaneously. While the size of the repeating structure may be derived from the matrix M without labeling errors of the stitch types using region growing procedures as proposed by Wu et al. [33], the occurrence of errors in M instead forces us to perform an exhaustive search over all possible repeat sizes and to compute an error score for each possible repeat size. Finally, the size with the least error score is assumed to be the correct one.

Let r and c denote the number of the rows and columns of a possible repeat. Assuming the presence of at least two occurrences of the pattern in the image, we only consider repeat sizes $s = (r, c)$ that satisfy at least one of conditions $r \leq \frac{R}{2}$ and $c \leq \frac{C}{2}$. In more detail, we fully partition M into a set of non-overlapping submatrices M^s for each possible repeat size $s = (r, c)$, where each $M_i^s \in M^s$ has the size s (or smaller if depicting an unfinished repeat on a boundary). The partitions are evaluated at different positions (m, n) of M with $m = r\,k$ and $n = c\,h$ with $k = 1, \ldots, \lfloor \frac{R}{r} \rfloor$ and $h = 1, \ldots, \lfloor \frac{C}{c} \rfloor$, respectively. Subsequently, we align all $M_i^s \in M^s$ according to their indices and compute the matrix M_{max}^s, which contains the stitch type with the maximal occurrence for each equal index of the submatrices. Then, we compute the Hamming distance D_i between each M_i^s and M_{max}^s. The sum of all Hamming distances yields the overall distance D_s of the current repeat size s.

We consider the size and the underlying stitch type matrix M_{max}^s with the minimal distance as the resulting pattern size. If there are several sizes with the same edit distance, we take the one with the smallest value $r + c$, since otherwise there is evidence for having another repeat withing the repeat. In the case that we cannot determine the type with maximal occurrence for an index position due to an equal number of the stitch types at this position, we compare the corresponding likelihood values of the pixels from which these types were derived to determine the final type.

If the distance of the resulting optimal repeat deviates from zero, errors have been detected in the matrix M. In this case, the corresponding M_{max}^s is determined to have the correct labelings of the stitches and all the submatrices are corrected according to M_{max}^s.

4.6. Repeat Position Determination

Finally, the localization of an as intuitive as possible repeating pattern from the underlying grid structure and the size of the repeating pattern has to be computed. In order to select an intuitive pattern repeat, we take inspiration from human perception and make use of two of the basic laws in Gestalt theory [5]. The Law of Symmetry states that symmetrical elements tend to be perceived as a unified group. Taking this into consideration, we search for symmetry along the x-direction of the pattern. If there is a symmetry, we take it into account when selecting the starting position of the repeat. If there is no symmetry in the structure of the repeat, we apply the Law of Prägnanz. According to this law, humans prefer simpler and ordered states that require less cognitive effort and, hence, can be faster processed than complex structures that, in turn, might have to be reorganized or even further decomposed. In our case, this corresponds to selecting the starting position of the pattern from all the possible positions that leads to the least amount of changes from one type of stitch to another when computing the sum of the type changes from each two adjacent rows and columns of the pattern in question. This ensures that individual structures such as squares or circles appearing in the pattern will not be broken.

5. Results and Discussion

Sample selection: In order to test our approach, we have chosen 25 photos and scans that depict knitting samples with different patterns and were produced with yarns of various types and colors. Eight of the photos were taken from the internet, one photo depicts a machine knitted piece and sixteen photos depict hand-made knitting fabrics. The focus on hand-made samples results from the fact that these exhibit a higher degree of variation and, hence, are more challenging than machine-knitted samples.

Performance analysis: Table 1 provides an overview over the computation times as well as the problem sizes for the four examples selected for this paper. More examples with corresponding running times are shown in the supplemental material. Since the most time-consuming operations were computations of the similarity maps with BBS and solving for the optimum with the Gurobi solver [12], we report the computation times only for these steps. The other steps required only a negligible amount of time. All computations were performed with an unoptimized implementation on an Intel(R) Core(TM) i7-5820K CPU with 3.30 GHz.

Table 1: The columns contain the image size (IS), the runtimes (in seconds) of BBS and ILP, as well as the size of both the grid (GS) and the pattern (PS). e_G and e_P denote the fraction of misclassified stitch types for the overall grid and the pattern after error correction.

ID	IS	BBS	ILP	GS	PS	e_G	e_P
1	673× 257	102.53	11.02	9×5	4×2	$1/45$	0
2	690×370	31.93	8.13	15×11	7×6	$1/165$	0
3	803×844	219.79	62.02	11×17	8×8	$4/187$	0
4	516×347	137.74	4.78	7×6	3×4	0	0

Visual quality: Figure 3 demonstrates the results of the individual steps of the pipeline for four of the example textiles. For the example in the second row one stitch in the input image (the sixth stitch from the left in the bottom row) has actually been wrongly knitted (knit instead of purl). This error was recognized and corrected by our method. To obtain the shown realistic renderings (last column), we synthesize yarns using the procedural model of Zhao et al. [35]. We then deform the yarn according to the discovered knitting instructions and discretize the resulting fiber geometry into a voxel grid, storing averaged densities and fiber orientations per voxel [15]. This voxel-based representation is then rendered using a volumetric path tracer [15]. Furthermore, Figure 4 shows results for the inferred grid structure obtained when applying our method on worn clothing.

Susceptibility to template selection: To evaluate the robustness regarding the selection of templates for the individual stitch types, we performed a study where 10 people aged from 10 to 67 years were asked to provide respective annotations. The results do not exhibit significant differences, as long as the testers follow the simple instruction of selecting two templates, which look similar to other stitches of the same type (Figure 1) (see supplemental material).

Table 2: Performance comparison of several template matching techniques: The rows contain the fractions of misclassified pixels e_{Px}, stitch types for the overall grid e_G and the pattern after error correction e_P. For the computation of the first measure we excluded pixels within a small band at the transitions between different stitch types. In 32% of the tests, the optimization based on the SAD likelihood maps did not succeed in detecting the correct number of rows and/or columns of the grid. These cases are excluded from the reported values e_G and e_P for SAD.

	SAD	NCC	DDIS	BBS	BBSg
e_{Px}	0.422	0.453	0.216	0.342	**0.194**
e_G	0.633	0.087	0.062	0.071	**0.040**
e_P	0.268	0.056	0.051	0.046	**0.029**

Suitability of different template matching schemes: We evaluated the suitability of different template matching schemes for the generation of likelihood maps for the individual stitch types. For this purpose, we compare our extended version of the BBS technique with additional gradient information (BBSg) to the normalized cross-correlation (NCC), the sum of absolute differences (SAD), the original BBS approach [7] without the proposed extension and the deformable diverse similarity (DDIS) approach [26]. Table 2 summarizes the respective results. In order to achieve meaningful results, the resolution of the input image is required to be sufficient so that the minimal template size is not smaller that 30×30 pixels.

Computational efficiency: In order to find the final center positions of stitches, we apply a global optimization that is formulated as an ILP problem. As ILP problems are known to be NP-hard, the computational times may be impractical. In order to speed-up the inference of optimal grids, which is particularly required for larger images, we downsize the corresponding likelihood maps by the factor of 0.5. The downsizing significantly decreases the computational time of the optimization, while still yielding similar results as without downsizing (for the evaluation of scaling we refer to the supplemental material).

Another possibility to decrease the computational time is to choose some iterative locally optimal approach instead of global optimization. For comparison, we use the likelihood and the stitch type assignments from the template matching step as the starting point for a greedy strategy to select neighboring stitch centers, where we also exploit uncertainty of the template sizes. In a first step, we take the maximum of the likelihood to find the the most likely location of a stitch and define a minimum distance within which no other stitch is allowed to occur depending on the template uncertainty. After discarding the respective area in the likelihood, we continue to search for the next highest likelihood, place a stitch center and again remove the region from the likelihood. This process is iterated until no further stitch center can be placed or the remaining likelihoods are lower than a certain threshold t (we used $t = 0.2$). As shown in Figure 3, this approach does not result in acceptable stitch center hypotheses, due to the iterative local optimization. Furthermore, this method does not compute the uncertainties automatically but requires their manual specification for each fabric sample individually. In contrast, our global optimization technique yields stitch center hypotheses at a higher quality.

Pattern search: In principle, once the size and correct labeling of the repeated pattern is found, one could reproduce the initial knitted example, since the knitting is done periodically. However, when knitting whole clothing pieces, the borders of the piece should be appealing. Hence, we need to identify the starting position of the correct or at

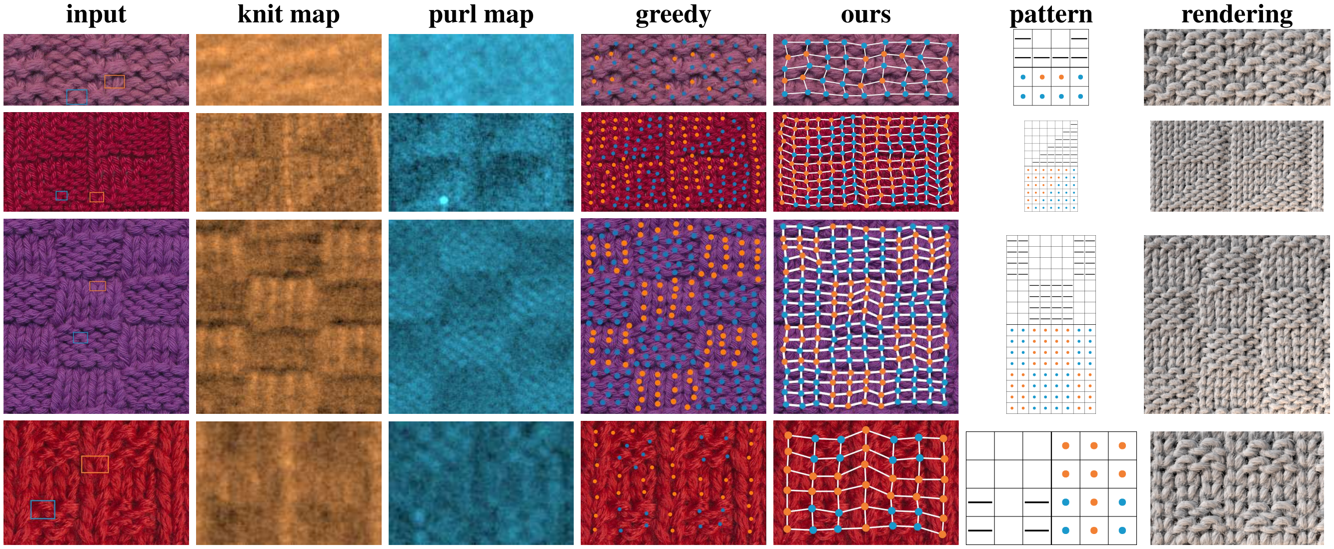

Figure 3: From left to right: input image, likelihoods for both stitch types, stitch center hypotheses derived via a greedy approach and grid structure inferred via our approach, corresponding knitting instruction (counting the rows from bottom to top, empty cells correspond to knits in odd rows and purls in even rows while cells containing bars correspond to purls in odd rows and knits in even rows) and rendering.

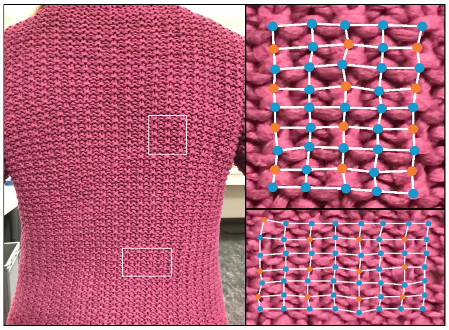

Figure 4: General, unrestricted setup of knitwear worn by a person (left) with detected grid structures for some regions of interest (right).

least of a nice pattern. In the supplementary material, we illustrate the problem of choosing an intuitive pattern. With our pattern search procedure we try to avoid breaking existing structures of the pattern, such as triangles or checkerboards, thereby following the Gestalt principles.

Limitations: In this paper, we limit our approach to the two fundamental stitch types: knit and purl. However, the number of stitch types is not strictly limited to two. In the supplemental material we also provide an example with three stitch types. However, including stitch types (e.g. holes) that deform the grid-like structure of the patten, requires including additional constraints, which we want to pursue in future work. Furthermore, if the input image is of low quality or contains almost completely occluded stitches, so that already the coarse localization does not yield meaningful results, the optimization technique will not produce the correct labeling.

6. Conclusion and Future Work

We have presented a novel practical framework for the inference of the complicated structures of knitwear as well as the corresponding knitting instructions from a single image. Templates for individual stitch types, as provided by the user, are roughly localized across the complete image and the resulting stitch positions are subsequently refined by optimizing the underlying grid structure within an integer linear program. The size of the repeating pattern is computed from the derived stitch labeling at the vertices of the resulting grid. Subsequently, we apply the Law of Symmetry and the Law of Prägnanz from Gestalt theory to find an intuitive pattern repeat and derive the corresponding knitting instruction. While our approach was demonstrated to allow the derivation of the knitting instructions for several different knitwears, there are still some open challenges to be addressed by future research. Including further stitch types into the framework as well as further reducing the degree of user interaction based on the combination of a large database of stitch types with their respective appearance variations and machine learning techniques is a promising avenue of research that we plan to pursue in future work.

References

[1] D. G. Aliaga, İ. Demir, B. Benes, and M. Wand. Inverse procedural modeling of 3D models for virtual worlds. In *ACM*

SIGGRAPH 2016 Courses, SIGGRAPH '16, pages 16:1–16:316, New York, NY, USA, 2016. ACM.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009.

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, pages 29–43, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] M. Bokeloh, M. Wand, and H.-P. Seidel. A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph.*, 29(4):104:1–104:10, July 2010.

[5] S. Bradley. Design principles: Visual perception and the principles of Gestalt. https://www.smashingmagazine.com/2014/03/design-principles-visual-perception-and-the-principles-of-gestalt/, 2014.

[6] J.-H. Chen, C.-S. Chen, and Y.-S. Chen. Fast algorithm for robust template matching with M-estimators. *IEEE Transactions on Signal Processing*, 51(1):230–243, Jan 2003.

[7] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2021–2029, June 2015.

[8] I. Demir, D. G. Aliaga, and B. Benes. Procedural editing of 3D building point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2147–2155, 2015.

[9] I. Demir, D. G. Aliaga, and B. Benes. Proceduralization for editing 3D architectural models. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 194–202, Oct 2016.

[10] E. Elboher and M. Werman. Asymmetric correlation: A noise robust similarity measure for template matching. *IEEE Transactions on Image Processing*, 22(8):3062–3073, Aug 2013.

[11] G. C. Guarnera, P. Hall, A. Chesnais, and M. Glencross. Woven fabric model creation from a single image. *ACM Trans. Graph.*, 36(5):165:1–165:13, Oct. 2017.

[12] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016.

[13] Y. Hel-Or, H. Hel-Or, and E. David. Matching by tone mapping: Photometric invariant template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):317–330, Feb 2014.

[14] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, Sep 1993.

[15] W. Jakob, A. Arbree, J. T. Moon, K. Bala, and S. Marschner. A radiative transfer framework for rendering materials with anisotropic structure. In *ACM Transactions on Graphics (TOG)*, volume 29, page 53. ACM, 2010.

[16] S. Korman, D. Reichman, G. Tsur, and S. Avidan. FasT-Match: Fast affine template matching. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, June 2013.

[17] S. Lienhard, C. Lau, P. Müller, P. Wonka, and M. Pauly. Design transformations for rule-based procedural modeling. *Comput. Graph. Forum*, 36(2):39–48, May 2017.

[18] O. Pele and M. Werman. Robust real-time pattern matching using bayesian sequential hypothesis testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1427–1443, Aug 2008.

[19] K. Schröder, A. Zinke, and R. Klein. Image-based reverse engineering and visual prototyping of woven cloth. *IEEE Transactions on Visualization and Computer Graphics*, 21:188–200, 2015.

[20] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision*, 103(3):348–371, Jul 2013.

[21] B. G. Shin, S.-Y. Park, and J. J. Lee. Fast and robust template matching algorithm in noisy image. In *2007 International Conference on Control, Automation and Systems*, pages 6–9, Oct 2007.

[22] A. Sibiryakov. Fast and high-performance template matching method. In *CVPR 2011*, pages 1417–1424, June 2011.

[23] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[24] O. Stava, B. Benes, R. Mech, D. G. Aliaga, and P. Kristof. Inverse procedural modeling by automatic generation of L-systems. *Comput. Graph. Forum*, 29(2):665–674, 2010.

[25] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mźch, O. Deussen, and B. Benes. Inverse procedural modelling of trees. *Comput. Graph. Forum*, 33(6):118–131, Sept. 2014.

[26] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1311–1319, 2017.

[27] J. Talton, L. Yang, R. Kumar, M. Lim, N. Goodman, and R. Měch. Learning design patterns with bayesian grammar induction. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 63–74, New York, NY, USA, 2012. ACM.

[28] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *ACM Trans. Graph.*, 30(2):11:1–11:14, Apr. 2011.

[29] Y. Tian and Srinivasa G. Narasimhan. Globally optimal estimation of nonrigid image distortion. *International Journal of Computer Vision*, 98(3):279–302, Jul 2012.

[30] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and P. Waddell. Inverse design of urban procedural models. *ACM Trans. Graph.*, 31(6):168:1–168:11, Nov. 2012.

[31] M. Weinmann and R. Klein. Material recognition for efficient acquisition of geometry and reflectance. In *Computer Vision - ECCV 2014 Workshops*, pages 321–333. Springer International Publishing, 2015.

[32] J. Weissenberg, H. Riemenschneider, M. Prasad, and L. Van Gool. Is there a procedural logic to architecture? In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 185–192. IEEE, 2013.

[33] F. Wu, D.-M. Yan, W. Dong, X. Zhang, and P. Wonka. Inverse procedural modeling of facade layouts. *ACM Trans. Graph.*, 33(4):121:1–121:10, July 2014.

[34] C. Yuksel, J. M. Kaldor, D. L. James, and S. Marschner. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Trans. Graph.*, 31(4):37:1–37:12, July 2012.

[35] S. Zhao, F. Luan, and K. Bala. Fitting procedural yarn models for realistic cloth rendering. *ACM Transactions on Graphics (TOG)*, 35(4):51, 2016.