**INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)**

<table>
<thead>
<tr>
<th>(51) International Patent Classification</th>
<th>(11) International Publication Number: WO 89/08409</th>
</tr>
</thead>
<tbody>
<tr>
<td>A41H 3/00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(21) International Application Number:</th>
<th>PCT/US89/01063</th>
</tr>
</thead>
<tbody>
<tr>
<td>(22) International Filing Date:</td>
<td>14 March 1989 (14.03.89)</td>
</tr>
<tr>
<td>(31) Priority Application Numbers:</td>
<td>168,648, 169,046, 169,047</td>
</tr>
<tr>
<td>(32) Priority Dates:</td>
<td>16 March 1988 (16.03.88), 16 March 1988 (16.03.88)</td>
</tr>
<tr>
<td>(33) Priority Country:</td>
<td>US</td>
</tr>
</tbody>
</table>

| (71) Applicant: MINNESOTA MINING AND MANUFACTURING COMPANY [US/US]; 3M Center, P.O. Box 33427, Saint Paul, MN 55133-3427 (US); |
| (72) Inventors: COLLINS, John, E.; GRUNES, Mitchell, Benjamin; P.O. Box 33427, Saint Paul, MN 55133-3427 (US); KOZAK, Daryn, J.; VITTAL, AliceMae, H.; P.O. Box 33427, St. Paul, MN 55133-3427 (US); |

| (54) Title: SYSTEM FOR PREPARING, STORING AND PREALTERING GARMENT PATTERN DATA |

**Abstract**

Disclosed is a computerized constraint based system for preparing, storing and prealtering garment pattern data.

![Diagram](image)

**Agents:** HAMRE, Curtis, B.; Merchant, Gould, Smith, Edell, Welter & Schmidt, 3100 Norwest Center, 90 South Seventh Street, Minneapolis, MN 55402 (US) et al.

**Designated States:** AT (European patent), AU, BE (European patent), BR, CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent).

Published

*With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

**Diagram**

- **User Interface**
- **Data Conversion**
- **Pattern Preparation**
- **Classification**
- **Annotation**
- **Prepared Pattern Knowledge Base**
- **Fitting**
- **Grading**
- **Adjustment**
- **Redrawing**
- **Output Prep.**
- **Pre-Altered Pattern Data**
- **Post Processor**
- **Electronic Pattern Data**
- **Initial Pattern Knowledge Base**
- **Templates**
- **Body Measurements**
- **Measurement Completion**
- **Pre-Altered Printed Pattern**
- **Garment Manufacturing**
**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>Austria</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
</tr>
<tr>
<td>DE</td>
<td>Germany, Federal Republic of</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>SU</td>
<td>Soviet Union</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
</tr>
</tbody>
</table>
SYSTEM FOR PREPARING, STORING AND PREALTERING GARMENT PATTERN DATA

Technical Field

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

The present invention relates to a computerized system for producing pre-altered garments and the patterns (electronic or paper) used in the construction of garments, which relate to a set of standard or individual body measurements.

Background Art

The problem of getting clothing to fit well without requiring custom tailoring has been an elusive goal in the garment and clothing pattern industries for many years. Many strategies have been developed for dealing with this problem. Creating and selling loose-fitting styles makes the problem less evident. Many companies take pains to identify the body characteristics of a "target market", a subset of the general population with identifiable body proportions. Many sizing schemes have been developed, particularly for women's clothing, including Misses, Womens, Half-sizes, Juniors, Petites, Junior Misses, etc. The proliferation of sizes makes it uneconomic for a manufacturer to produce and a retailer to stock all styles in all sizes. Extensive statistical studies have been undertaken with the aim of developing sets of standard sizes that will fit the majority of the population.

Typically, a prior art pattern used in the construction of clothing is generated as follows:

1. A skilled designer sketches a new design. After approval these sketches are sent to a pattern making department.
2. A pattern maker drafts a standard size (typically size Misses 10 for women's garments) flat-paper realization of the design using a set of generic parts called slopers as a starting point. This realization is then implemented in muslin and other fabrics and tested on mannequins and models. Once the designer approves the final realization of the design, the pattern pieces are sent to the grading department.

3. A grader generates a set of garment part outlines for each of the standard sizes in which it is desired to produce the design (e.g. Misses 6, 8, 10, 12, 14, 16, 18, 20, 22). This may be done by hand or with the use of a computerized grading system. The hand method involves tracing the original, moving the important points in or out as needed, and then redrawing the lines. If grading is being done with a computerized grading system, the grader assigns grading rules to each important point of each pattern piece, and the computer generates the outlines and draws the results. A grading rule is typically a set of \((x, y)\) offsets, one per size, that specify how a given point is to be moved to represent a particular grade. Steps 2 and 3 are typically repeated for each size range (e.g. Misses, Juniors, Petites, Womens, etc.) because of the difficulty in expressing the more complex changes required to move between ranges with current systems.

4. If the pattern is intended for the home sewing market, the pattern pieces are labeled and laid out for printing on tissues; if the pattern is for a manufacturing operation, a manual or computerized marker-making system lays out the pattern pieces on the fabric, and either the markers are printed, for a manual cutting operation, or the computer drives an automated fabric cutter.
There are a number of inefficiencies with these processes. They affect manufacturers, retailers, and home sewers. These include but are not limited to high labor costs, printing costs, reorder and handling costs, inventory costs, space costs, stock shortages on popular items, and poor fit.

Many tailors and seamstresses spend their time hand-altering commercially manufactured garments. Many people are simply unable to find a ready-to-wear garment that fits in a desired style either because the garment is not manufactured in their size or because the standard sizes produced by a particular manufacturer do not match well with a particular customer's body measurements.

Fit problems for the pattern industry have been addressed by a number of approaches. The pattern houses often include alteration lines for lengthening and shortening sleeves and legs. Even so, a home sewer typically modifies a pattern before sewing the garment, and then disassembles, alters, and re-sews the garment repeatedly in an attempt to achieve satisfactory fit. Several companies host traveling seminars designed to teach pattern modification techniques, and there have been previous attempts to produce computer-altered garments and garment patterns (a description of pattern modification techniques is in Jan Minott, *Pants and Skirts Fit for Your Shape*, Burgess Publishing, Minneapolis, Minnesota, 1974). None of these approaches adequately addresses the range of alterations necessary on the wide selection of garments and garment patterns that are available today.

Computerized fitting systems have been developed in the past; examples include systems used commercially by Clothing Design Concepts, Box 1188, Manhattan, Kansas, and by Richman Brothers Company, Cleveland, Ohio, also described in U. S. Patent 4,598,376. An analytic approach to producing custom-fitted patterns may be
found in Francesann Heisey, *A Quantitative Methodology for Generating Specifically Fitted Garment Patterns*, PhD. Thesis, University of Minnesota, 1984. These systems have been limited to specific garment patterns, because of the difficulty involved in encoding a garment pattern with detailed fitting information.

Computerized grading systems, which have been in use in the industry for several years, do not have the ability to relate generic garment knowledge to garment pattern data; therefore, a grader is responsible for providing this knowledge through a process of manually assigning grading rules to patterns.

The present invention addresses the major problem that has existed to date in the use of a computerized fitting or grading system, which is the time consuming and error prone process of preparing new patterns as input to the system. This is accomplished by applying knowledge about generic garment types to each new pattern, and by extensive automatic analysis of new patterns based on that knowledge. Once this knowledge has been applied to a new pattern, it may be used to eliminate manual grading operations through a process of the present invention called "knowledge-based grading".

The present invention also offers prealteration of garment patterns yielding a high quality fit for a wide range of standard or individual body measurements. As used here, the term "prealteration" refers to automatically applying alterations to a garment pattern, as needed to fit a set of body measurements, before the garment is constructed or the pattern printed. This effectively eliminates the current labor-intensive grading and alteration processes.

The system of the present invention contains knowledge about generic garment styles, including landmarks and other garment features that are expected to be found in garments representative of a garment style and generic constraints that describe detailed
relationships between garments and body measurements in terms of the landmarks and other garment features. This knowledge may be applied to raw pattern data by identifying a generic garment style of a garment depicted by the raw pattern data, identifying the expected landmarks and other garment features in the raw pattern data, and then applying the generic constraints to the new pattern. The result is prepared pattern data that includes the knowledge necessary to fit the garment to a set of body measurements. A fitting system is also described that operates by satisfying constraints in a prepared pattern. Such a fitting system need not contain specific knowledge about particular garments.

A prior art system disclosed in patent number 4,149,246 describes storing information on limitations for a garment (ref. column 7, line 39-40). These limitations refer only to limited choices such as whether a particular type of pocket is appropriate for a particular body type. It is assumed that these limitations have been entered into the prior art system by a system user, such as a designer, for each particular garment or pattern to be processed by the prior art system. Although in column 2, lines 52-54, there is a reference to using nineteen special body measurements for "highly accurate fitting (optional)", there is no disclosure for accomplishing such fitting in the patent, there being reference only to prior art grading systems, such as U. S. Patent 3,391,392 (see column 5 line 49). Further, there is no disclosure of any type of generic knowledge of garment styles, or of the application of generic knowledge to specific garment patterns, as in the present system.

Another prior art system, described in patent number 4,598,376, operates by modifying reference or pattern points that are stored in a pattern file (ref. column 7, lines 37-39). However, this system includes no storage of generic knowledge of garment styles. Nor is there
any disclosure to teach or suggest a data storage
structure for the pattern file or a pattern preparation
process for identifying the reference or pattern points
stored in the pattern file of the system. In contrast
to the present system, in which a fitting system need
not contain specific knowledge about particular
garments, there is no disclosure in U. S. patent
4,598,376 to suggest that this is possible.

 Disclosure of Invention

The present invention relates to a computerized
system for preparing garment pattern data to enable
subsequent computerized prealteration based on standard
or individual body measurements. The system comprises
pattern storage means for storing garment pattern data
comprising points and lines depicting one or more
garments. The system further comprises garment
knowledge means for storing generic garment knowledge
for each of a plurality of garment styles. The garment
knowledge means comprises feature description means for
storing descriptions of landmarks and other garment
features that are expected to be found in garments
representative of the garment style. The garment
knowledge means further comprises generic constraint
means for storing generic constraints comprising at
least one of algorithms and declarative structures for
relating body measurements to changes in one or more of
the landmarks and other garment features found in
garments representative of the garment style. The
system also includes processor means for preparing
garment pattern data for modification by relating the
generic garment knowledge to the garment pattern data.

The present invention also relates to a computerized
data storage structure for storing garment pattern data
in machine-readable form for use in connection with a
computerized structure for manipulating the garment
pattern data. The structure comprises coordinate data
means for storing points and lines depicting parts of a
garment. The structure further comprises garment description means for storing a description of the garment. The garment description means comprises geometric constraint means for storing constraint descriptions that specify limits on relationships among the points and lines that depict the garment. The garment description means further comprises measurement means for storing one or more measurement constraints that map the physical dimensions of the garment onto the points and lines and specify relationships between the physical dimensions of the garment and standard or individual body measurements.

The present invention further relates to a computerized system for prealtering garment pattern data based on standard or individual body measurements. The system includes body measurement acceptance means for accepting standard or individual body measurements and pattern acceptance means for accepting garment pattern data. The pattern acceptance means comprises coordinate data means for accepting coordinate data comprising points and lines depicting a garment and garment description means for accepting a description of a garment. The garment description means comprises geometric constraint means for accepting geometric constraints that specify limits on relationships among the points and lines that depict the garment. The garment description means further comprises garment measurement means for accepting one or more measurement constraints that map the physical dimensions of the garment onto the points and lines and specify relationships between the physical dimensions of the garment and the standard or individual body measurements. The system also comprises constraint satisfaction means for producing prealtered pattern data by satisfying the geometric and measurement constraints through altering points and lines in the garment pattern data.
Brief Description of Drawings

Fig. 1A represents a preferred flow of data and user interaction through pattern preparation and fitting.

Fig. 1B schematically represents a hardware embodiment of the preferred system.

Fig. 2 describes an object oriented model for representation of a system.

Fig. 3 depicts a preferred hierarchical relationship among the instances in a project KB and the relationships of these instances to templates.

Fig. 4 details the elements of a preferred project instance.

Fig. 5 details the elements of a preferred garment instance for a garment that belongs to a project.

Fig. 6 details the elements of a preferred part instance for a part that belongs to a garment.

Fig. 7 details the elements of a preferred feature instance for a feature that belongs to a part or to a garment.

Fig. 8 details the elements of a preferred subclass of feature, called a line.

Fig. 9 details the elements of a preferred subclass of feature, called a point.

Fig. 10 details the elements of a preferred garment template.

Fig. 11 details the elements of a preferred garment measurement.

Fig. 12 details the elements of a preferred part template.

Fig. 13 illustrates a pair of angle dependencies.

Fig. 14 details the elements of a preferred subclass of a feature, called a dart.

Fig. 15 details the elements of a preferred set of standard or individual body measurements.

Fig. 16 represents a preferred flow of data through a knowledge-based grading system.

Fig. 17 illustrates a midpoint-endpoint dependency.
Fig. 18 schematically shows the steps of a preferred classification process.

Fig. 19 details the elements of a preferred style template.

Fig. 20 details the elements of a preferred grade set.

Fig. 21 illustrates conceptual relationships among standard sizes.

Fig. 22 illustrates an example of a knot scaling problem.

Fig. 23 illustrates an example of isomorphic curve reshaping.

Fig. 24 illustrates an example of sinusoidal curve reshaping.

Fig. 25 illustrates an example of cubic curve reshaping.

Fig. 26 illustrates an example of linear curve reshaping.

Fig. 27 illustrates a preferred dart lengthening process.

Fig. 28 illustrates a preferred dart widening process.

Fig. 29 illustrates an angle dependency.

Fig. 30 illustrates a preferred process of adjustment and redrawing steps on a single line.

Fig. 31 schematically shows the steps of a preferred data conversion process.

Fig. 32A illustrates a preferred process of computing to-close information for darts and pleats.

Fig. 32B illustrates a preferred process of computing to-close information for a symmetrical dart.

Fig. 32C illustrates a preferred process of computing to-close information for an asymmetrical dart.

Fig. 32D illustrates a preferred process of computing to-close information for a converging pleat.

Fig. 32E illustrates a preferred process of computing to-close information for a parallel pleat.
Fig. 33 details the elements of a preferred line template.
Fig. 34 schematically shows the steps of a preferred pattern preparation process.
Fig. 35 illustrates a preferred process of closing darts.
Fig. 36 schematically shows the steps of a preferred annotation process.
Fig. 37 details the elements of a preferred measurement template.
Fig. 38 schematically shows the steps of a preferred fitting preparation process.
Fig. 39 illustrates darts and pleats.
Fig. 40 illustrates a preferred process of angle restoration.
Fig. 41 shows a preferred coupling data structure.

Detailed Description

GENERAL INTRODUCTION

Referring to Fig. 1A, in the preferred embodiment of the present invention data may enter the system in digital electronic form 100. After patterns 100 have been converted to an internal format by a data conversion process 104, a pattern preparation process 110 analyzes, identifies, measures, and prepares them for fitting, and produces a prepared pattern knowledge base 128. This process makes use of a set of templates 122, which contain generic knowledge about typical garments in a variety of styles, and knowledge about components of those garments. Although a major portion of pattern preparation 110 is automatic, certain steps preferably involve human interaction 123; therefore, a user interface 124 is provided. A fitting process 134 may then make use of a prepared pattern knowledge base 128 together with a set of measurements 132 corresponding to an individual person or a standard size to produce pre-altered pattern data 136, a version of the original pattern data which has been altered to fit
the body measurements 132 that were used to drive the fitting process 134. This data may then be plotted to produce a printed pattern 518 or used as input to a fabric cutter or a marker making system or some other portion of a garment manufacturing process 139.

Referring to Fig. 1B, the preferred embodiment operates on a computer 85 comprising a processor 80 with associated memory 81, the latter being subdivided for explanatory convenience into pattern storage 84, garment knowledge 82, and measurement storage 83. Those skilled in the art will recognize that computer memory 81 may be embodied as a combination of internal memory along with fixed and removable mass storage. Those skilled in the art will also recognize that appropriate user interface, input, and output means (not illustrated) are normally provided with such equipment.

One embodiment of the present invention involves a separate pattern preparation system (computer 85 implementing processes 104, 110, and 124 and storing templates 122) that produces a portable data storage structure (removable mass storage 81 or electronically transmitted data structured as prepared pattern knowledge base 128). Such an embodiment could also involve a separate fitting system (computer 85 implementing process 134) that accepts data from the portable data storage structure and produces prealtered garment pattern data based on standard or individual body measurements 132.

As is further described below (see also Figs. 1A, 1B, and 3), such a separate pattern preparation system, or the pattern preparation portion of an integrated system, could comprise pattern storage means 84 for storing garment pattern data 108 comprising points and lines depicting one or more garments 148, garment knowledge means 82 for storing generic garment knowledge 122 for each of a plurality of garment styles, and processor means (processor 80 implementing processes 104
and 110) for preparing the garment pattern data 100 for modification by relating the generic garment knowledge 122 to the garment pattern data. In such a system, the garment knowledge means could comprise feature description means 82 for storing feature descriptions in templates 676, 685, and 634, and generic constraint means 82 for storing generic constraints in templates 676, 685, 600, 634 and 612. A feature description stored by the feature description means could comprise descriptions of landmarks and other garment features that are expected to be found in garments representative of a garment style. A generic constraint stored by the generic constraint means could comprise algorithms or declarative structures for relating body measurements to changes in one or more of the landmarks and other garment features 154 found in garments representative of the garment style.

Also as described below, a portable data storage structure (memory 81 structured as prepared pattern data 128) could be used in connection with a variety of computerized systems for manipulation of garment pattern data, in addition to the pattern preparation and fitting systems described herein. Referring to Fig. 3, the system could comprise coordinate data means for storing points 164 and lines 162 depicting a garment, along with garment description means 148 for storing a description of a garment. The garment description means could comprise geometric constraint means (e.g., angle dependencies 384, reshape methods 390, Fig. 8; attached points 403, Fig. 9; to-close 558, Fig. 14) for storing one or more geometric constraints that specify limits on relationships among the points and lines that depict the garment. The garment description means could further comprise measurement means for storing one or more measurement constraints 150 that map the physical dimensions of the garment onto the points and lines and specify relationships between the physical dimensions of
the garment and standard or individual body measurements 132.

A separate or integrated fitting system (processor 80 executing process 134) for prealtering garment pattern data 128 based on standard or individual body measurements 132, as described below, could comprise body measurement acceptance means 80 for accepting standard or individual body measurements 132, and pattern acceptance means 80 for accepting prepared garment pattern data 128. The pattern acceptance means could be designed to accept portable data storage structure as described above. A fitting system could further comprise constraint satisfaction means (processor 80 implementing process 134) for producing prealtered pattern data 136 by satisfying the geometric and measurement constraints contained in the prepared pattern data through altering points and lines in the data.

The remainder of this document contains descriptions of the processes of the preferred embodiment and the data or knowledge used by these processes. The section entitled "Process Description" describes in detail the various processes that make up the preferred embodiment, including data conversion 104, pattern preparation 110, and fitting 134.

The following introductory sections describe the conventions used in the subsequent detailed descriptions.

INTRODUCTION TO KNOWLEDGE REPRESENTATION

Information and processes in the preferred embodiment of the present system are represented as a collection of objects. The objects of the system, as illustrated in Fig. 2, are organized into classes or object frames 138. Each class object includes a collection of slots. There are three types of slots that may be contained in an object: object pointers 140 (illustrated by rectangles), methods 142 (illustrated by
ovals), and data 144 (illustrated by round cornered rectangles).

An object pointer 140 typically points to an instance of a particular class, and these pointers may be used to navigate from one instance to another. For example, to find all the parts 152 in a particular garment 148, an access may be made to the parts' slot 314 (see Fig. 5) of the particular garment 148. In the preferred embodiment, parts slot 314 contains pointers to all the part instances 152 belonging to a particular garment.

Methods 142 are illustrated by ovals. As described below, a method may be unique to the class of objects in which it is defined. Thus, a method defined in one object may have the same name, but comprise different code or commands, than a method having the same name which is defined in another object. A good example of this is the method named create which contains unique code for each object class.

Data 144 may be attributes, graphic data, products of methods, or any information which is necessary to the system.

New objects may be created as instances of particular classes. Space for slots, within an instance of a class, is typically automatically reserved when the instance is created. Exceptions are "class slots", also called class method, class data, or class pointer, which exist only in a class, and are not passed on to its instances; for example, a "create" class method may be used to create an instance of a class, but would not be passed on to instances of the class, since there is typically no need to create an instance of an instance. This distinction may be less important in an object-oriented programming system, such as KEE™ from IntelliCorp, that does not distinguish between classes and instances.

Referring to Fig. 2, method slots 142 may be
automatically filled in when an instance is created from a definition stored in the class level object, although they may be overridden by different definitions in specific instances. Typically, object pointers 140 and data slots 144 are initially filled with any default values that may be present in the class level objects. When no defaults are available, the slots may be empty when the instance is created and may later be filled during the process of producing prealtered patterns.

As previously suggested, each class object typically contains one method in a class slot, called create, which may be invoked to create an instance of that class.

A special kind of method is called a "demon". Two kinds of demons are illustrated in Fig. 2, a "fetch" demon 143, and a "put" demon 147. The typical function of a fetch demon 143 is to filter the data coming out of a slot to which it is attached; an access or "fetch" of the slot does not directly access the slot, but instead sees the value produced by the demon method. This is typically transparent in the sense that accessing the data slot invokes the demon automatically. A put demon 147 normally works in a similar fashion, except that in this case a value placed in a slot may be filtered by a demon method. In other words, writing to a slot that has an attached put demon method normally does not directly write to the slot, but instead invokes the demon method, whose output value is ultimately placed into the slot.

In the preferred embodiment, one or more garments are treated as a project, and a separate set of instances is created for each project and is collected together in a "knowledge base", sometimes called a "pattern knowledge base", "project knowledge base", or "project KB" 128. Such a project KB encapsulates information about a garment or set of related garments, including graphic data representing the individual
parts, knowledge about how the garment or garments are intended to fit the body, and knowledge about how various aspects of fitting process 134 apply to the pattern.

Fig. 3 depicts a preferred hierarchical relationship among instances in a project KB and relationships of these instances to templates. This may be thought of as a "structural" hierarchy as opposed to an inheritance hierarchy composed of classes and instances. For example, a project 146 contains a collection of garments 148; a garment 148 contains a collection of parts 152; and so forth. In the preferred embodiment, various instances in a project KB contain information that is derived from templates. For example, slots in a part instance 152 that is created to represent a particular pants back part may be filled with data that is derived from information contained in a part template 634 that generally describes pant back parts.

PROCESS DESCRIPTIONS

As illustrated in Fig. 1A, the main processes in the preferred embodiment of the present system are data conversion 104, pattern preparation 110 (comprising classification 112, annotation 114 and fitting preparation 116), fitting 134 (comprising grading 210, adjustment 212, redrawing 216 and output preparation 218), and post processing 137. Each of these preferred processes is described in detail in the following sections.

DATA CONVERSION 104

INTRODUCTION

The first preferred process in creating a prealtered pattern, as illustrated in Fig. 1A, is data conversion 104. The purpose of data conversion in the preferred embodiment is to accept and store electronic pattern data 100 and convert them to an initial pattern knowledge base 108 (also called a pattern KB or a project KB) which can be processed by preferred pattern
preparation process 110. The electronic pattern data 100 as described herein is typically substantially devoid of information relating the data to body measurements, and may be nothing more than raw coordinate data describing points and lines.

Preferred data conversion process 104, as illustrated in Fig. 31, creates instances of project 146, garment 148, part 152, line 162, and point 164 classes and fills appropriate slots with information obtained from the electronic pattern data; most of the slots residing in the created instances are not filled during data conversion, but are filled later during pattern preparation.

PROJECT CREATION

The main function of the preferred data conversion process is to create a project instance for a particular project and to invoke other methods to create garment and part instances.

GARMENT CREATION

A create method 319 of the garment class 148 (see Fig. 5) is preferably invoked each time data for a garment is encountered. When no garment data blocks are included in electronic pattern data 100, garment instances are typically created later during classification process 112.

PART CREATION

Preferred electronic pattern data 100 includes information about each part of the project. Typically, this includes coordinate positions of the points, lines and alignment guides in each part. Additionally, the electronic pattern data may include part names, which garment a particular part is a part of, and line names. When these data items are not available in the electronic pattern data, they are preferably filled in during classification process 112 and annotation process 114.

Coordinate positions in electronic pattern data 100
may be provided for a number of different sizes. In this case, a known sizes slot 308 in project 146 preferably contains a list of the sizes for which data are provided. Alternatively, data may only be provided for one size in which case known sizes slot 308 contains only a single size. During preferred fitting process 134, a grade process 210 either uses coordinate data for one of the known sizes or derives new coordinate positions by grading from a single known size.

As implemented in the present system, a create method 343 (see Fig. 6) for a part accepts an input argument that includes information from the electronic pattern data pertaining to a particular part (e.g., part data block). LINE CREATION

In the preferred embodiment, a line instance 162 (see Fig. 8) is created for each line in a part 152, and a pointer to each line instance is stored in a features slot 334 in the part (see Fig. 6). As implemented, a line is represented by a series of (x,y) positions for each of the known sizes 308 of a project (see Fig. 4); separate point instances 164 are created for each point of a line; and pointers to these point instances are stored in features slot 334 of the part.

POINT CREATION

In the preferred embodiment, the method to create points accepts three input arguments: the (x,y) position of the point for each known size (position data), a pointer to the part instance that the point is a part of (Part 152, Fig. 6), and an optional pointer to the line instance that the point is a part of (Line 162, Fig. 8).

PATTERN PREPARATION 110

INTRODUCTION

In the preferred embodiment, the purpose of pattern preparation (see Fig. 34) is to transform raw pattern data contained in an initial pattern knowledge base 108 into a prepared pattern knowledge base 128 that contains
the knowledge needed for fitting process 134 to produce prealtered patterns. The process of transforming raw pattern data into a prepared pattern knowledge base operates in the current system by selecting an appropriate set of generic garment knowledge (templates 122) for a project and then applying the selected generic garment knowledge to the raw pattern data by identifying important landmarks and other garment features and then applying various prealteration constraints.

In the current implementation of the system, generic garment knowledge is stored in templates (garment templates 676, style templates 685, part templates 634, line templates 612, and measurement templates 600) and is structured to contain feature descriptions that generically describe landmarks and other garment features that are expected to be found in garments representative of a garment style, and to include a description of generic constraints comprising algorithms or declarative structures for relating body measurements to changes in the landmarks and other garment features in garments representative of a garment style.

PROCESS ORGANIZATION

In the current implementation, pattern preparation is divided into three processes: classification 112, annotation 114, and fitting preparation 115. Each of the three processes embodied in the preferred pattern preparation process performs specific steps to modify the initial knowledge base in some way and then passes the resulting modified knowledge base (KB) to the next process.

Preferred classification process 112 is concerned with identifying a garment type and corresponding styles for each of the garments in a project; the identification of garment type and styles provides means for selecting a set of templates from generic garment knowledge that pertains to each specific garment.
Annotation process 114 preferably includes feature identification means for identifying the points and lines in the raw pattern data that correspond to the features described in the selected templates. Preferred fitting preparation process 116 is primarily concerned with deriving physical measurements for each garment in a project and then applying generic constraints described in the selected templates to the specific features identified during annotation process 114, resulting in prealteration constraints. Both the generic constraints and the prealteration constraints preferably describe how to prealter garments by altering various garment features based on a comparison between physical garment dimensions and body measurements; generic constraints typically refer to landmarks and other garment features in an abstract way, while prealteration constraints typically refer directly to specific points and lines in a particular garment pattern.

CLASSIFICATION 112

INTRODUCTION

In the preferred embodiment, the purpose of classification process 112 (see Fig. 18) is to begin with an initial pattern knowledge base, as provided by data conversion process 108, and to produce a classified pattern knowledge base 500 in which one or more garments and parts within a project are classified by type (e.g., a garment type of pants or skirt, a part type of front or back), parts are collected together to form one or more garments, and garments are identified with measurements and features useful in prealtering each garment within the project to standard or individual body measurements. Subsequent preferred processing by annotation process 114 and fitting preparation process 116 completes the description of the pattern knowledge base, which is eventually used in a fitting process to prealter each garment.
Although each garment or part within an initial pattern knowledge base 108 may or may not have a name associated with it, the generic type of each garment or part may not have been classified in a manner general enough to be useful to the preferred system.

In the current implementation of the system, the way in which classification is able to classify garment and part types is through knowledge that is resident in garment templates, part templates, and style templates. The following section describes the role that these templates play in the current system during the process of classification.

TEMPLATES

In the preferred embodiment, information about each basic garment type is stored in garment templates 676 (see Fig. 10), with each garment template being identified by a name contained in a type slot 677. For example, basic garment types of pants, skirt, and bodice could be used as the name in type slot 677 for templates corresponding to such garment types. As implemented in the present system, a garment template corresponds to a basic garment, referred to as a “sloper” by the garment industry, that is typically a plain garment without pleats, collars, cuffs and so forth.

A pattern designer typically adds various style elements and design ease to a basic garment (sloper) to create a new design. Design ease, a term used to describe the extra fabric added to a basic sloper to create a desired style, does not affect the classification process of the present system; the way in which design ease is processed in the current system is described in fitting preparation 116.

In addition to design ease, a pattern designer may style a pattern by adding style elements. A style element, as implemented in the current system, is either a part added to a garment (such as a pocket) or a characteristic of a part (such as a mid-calf-length
skirt). In the preferred embodiment, information about each style element is contained in a style template 685 (see Fig. 19).

A collection of part templates 634 is also resident in the current system. Each part template 634 (see Fig. 12) contains generic part knowledge and may be identified by its type based on the contents of a type slot 660. A part's type may consist of a part name (e.g., back) and an optional garment type (e.g., pants).

When a garment type is included in a part's type slot 660, the part template typically applies only to parts of a particular garment. When there is not a garment type included, a part template typically applies to all parts of a specific type for all garment types. For example, a part template for a pants back may include both "back" (part name) and "pants" (garment name), since the information associated with a pants back part is different than the information for other garment back parts, e.g., a bodice back part. By way of a contrasting example, a part template for a straight waistband in the preferred embodiment does not include a garment name, since all straight waistbands in the current system typically contain the same information.

In the preferred embodiment, the important measurements needed by fitting process 134 are listed in templates. Each garment template (Fig. 10) includes a list of measurements that are important for the fitting of a particular basic garment (e.g., a garment template whose type is "pants" may include a "waist circumference" measurement). Each style template 685 (Fig. 19) may also include a list of measurements that are important for fitting. For example, a style element of type "full length" may include a "waist to floor" measurement. Further, part templates 634 (Fig. 12) may include a list of measurements that are important for the fitting of that part. For example, for a garment that has a waistband, a part template 634 having a type
"waistband" would normally include a "waistband" measurement in slot 692.

As will be described in the remainder of this section, a primary function of the preferred classification process is to match garments and parts to their corresponding templates and to identify which styles are included in each garment; after the appropriate templates are identified, the information contained in the templates may be used to provide the basis from which information is derived for the fitting process.

In the current embodiment of the system, classification process 112 is divided into four steps: garment classification, part classification, style classification, and template copying.

GARMENT CLASSIFICATION

In the preferred embodiment, the purpose of the garment classification step is to identify the types of garments in a project by associating each garment with a particular garment template. Garment classification attempts to complete the classification automatically based on available descriptions and then elicits missing information and verification from the system user.

In the present system, garment classification begins by analyzing the description of a project. When electronic project data 100 includes a project description, it is typically contained in a description slot 306 of project instance 146. The project description contained in description slot 306 of each project instance is normally a textual description of the garments in the project. For example, a project description is customarily printed on the back of a prior art pattern envelope or in pattern or garment catalogs. An example of a project description is:

"Straight skirt, below mid-knee or tapered pants have waistband and back zipper. Skirt: back slit. Pant: front pleats and side
pockets. Purchased top and belt."

When a project description is not available, one may be elicited from the system user.

Each garment and style element template preferably contains a list of identifiers used to match words and phrases of a project description. For example, a garment template 676 (Fig. 10) whose type slot 677 contains the name "pants" may include "pants", "shorts", "slacks", and "culottes" in identifiers slot 678. If, in an implementation of the current system, there are two garment templates, one containing the word "skirt" and another containing the word "pants" in identifier slot 678, the system would identify two garments from the above project description (a "pants" garment and a "skirt" garment) for this project.

After the system matches garment template identifiers to the project description, the system user may be asked to verify the system derivation. Verification may include adding new garments, removing system derived garments, and correcting system mismatches between garments and garment templates. In the current implementation of the system, a garment instance 148 is created for each garment type identified, and each created garment instance is associated with one of the garment templates 676.

In the preferred embodiment, classification occurs at the project level, and the method used to classify garments is contained in a slot 695 (see Fig. 4). PART CLASSIFICATION

When all garments of a project are classified and garment instances created, the next step in the preferred embodiment is to classify each part in a project in order to associate each part instance with a particular part template and to associate each garment instance with its constituent parts.

In the preferred embodiment, each garment template 676 represents a basic garment which is typically
comprised of a minimum set of parts expected to be found in garments of a particular type. For example, a basic pants garment normally includes a front part and back part. As implemented in the present system, each garment template 676 includes an expected parts slot 679 which contains a list of pointers to related part templates 634, and this list is used to represent a minimum set of parts that comprise a particular garment.

The first step of the preferred part classification process is to identify the project parts that correspond to the expected part templates of each garment and to associate those parts to their corresponding part templates and garments. Each part template in the system as implemented contains in an identifiers slot 690 a list of identifiers which are typically words and phrases that are associated with parts of a particular type. When part names are available, as provided during data conversion, the system preferably attempts to automatically match the minimum set of parts listed for each garment to their corresponding part template by matching part names to the words and phrases contained in identifiers slot 690. When part names are not available, they are preferably elicited from the system user.

At this point, there may still be project parts, outside of the minimum list of expected parts, that are not matched to garments and part templates. A next step of part classification is preferably to involve the system user in completing the matching of parts to garments and to part templates. A list of parts that have not been matched is preferably presented to the system user. For each unmatched part, the system user may then match the part to one of the project garments.

To complete part classification, each part may be presented to the system user along with its current part template association. The system user may then verify that each part is properly matched to its corresponding
part template; additions and changes may also be accepted from the system user at this time.

In the preferred embodiment, a method to classify parts of a project is contained in a part classification slot 697.

STYLE CLASSIFICATION

After garments and parts are classified, a next step in the preferred embodiment is to identify the style elements that pertain to each of the garments. To find styles, the current system re-analyzes the phrases of the project description. Using the same project description example above, the phrases: "mid-knee" "waistband" "back zipper" "front pleats" and "side pockets" match identifiers that are stored in various style templates.

When a phrase of a project description matches an identifier of a garment's possible style template, that style may then be associated with the particular garment. When the present system has completed associating styles with garments based on the project description, the system user is typically presented with the derived information; from this point, the system user normally verifies the derived associations and modifies the information as needed.

In the preferred embodiment, a style classification slot 698 (Fig. 4) contains a method to classify styles for a project.

TEMPLATE COPYING

At this point, each garment within a project in the preferred embodiment has a pointer to its corresponding garment template and pointers to the styles and parts that are contained in the garment, and each part has a pointer to its corresponding part template and to the garments in which it is a part (part-of slot 336, see Fig. 6). In the current system, the next step of classification is to copy information from the template objects to the garment and part instances in the pattern
knowledge base. It will be recognized by those skilled in the art that the process of copying template information is a preferred approach to facilitate greater performance by a subsequent process. An alternate approach is not to copy template information, whereby a subsequent process (e.g., a fitting process) would retrieve template information as needed.

There are two types of information that are typically copied at this time: measurements and expected features. Each garment, style and part template preferably contains a slot called measurements, with each measurements slot containing a list of measurement templates corresponding to physical dimensions of garments that are important to the prealteration of garments of a given type and to the prealteration of garments that contain particular parts and styles. The list of measurement templates important to the prealteration of a particular garment, as implemented in the current system, is the collection of the important measurement templates found in a garment’s parts, styles and corresponding garment template.

A second type of information typically copied at this time from template information is expected features. An expected feature may include a name and a feature type (e.g., line or point; see also Fig. 3). Each garment template typically includes expected features that may be found in any of a particular garment’s parts. Each style template may also contain a list of expected features. Expected features, which may be declared in a style template, sometimes correspond to a part name. In the absence of a part name, an expected feature is typically found in any part of a particular garment with that style. Accordingly, each part template may contain a list of expected features that are expected to be in a particular part.

In the preferred embodiment, expected features that may be found anywhere on a garment are copied to an
expected features slot 521 of the corresponding garment instance, and expected features associated with a particular part are copied to an expected features slots 519 of the particular part instance.

A typical method to copy template information may be contained in a template copying slot 699.

ANNOTATION 114

INTRODUCTION

As illustrated in Fig. 1A, annotation process 114 is part of preferred pattern preparation process 110. The purpose of annotation is to produce an annotated pattern knowledge base 502. Fig. 36 illustrates a processing flow of annotation.

In the preferred embodiment, the first set of functions performed during annotation is related to transforming various graphic elements of a project into their final form. These graphic transformations typically include attaching alignment guides, intersecting lines, and combining line segments.

In the current implementation of the system, the primary function performed during annotation process 114 is feature identification. The purpose of feature identification is to identify and name the actual lines and points corresponding to the expected landmarks and other garment features of a particular project. The features that are expected to be contained in a project are preferably determined during classification 112 and are included in classified pattern knowledge base 500. It is the function of feature identification to match those expected features to their corresponding lines and points on the pattern parts in a particular project.

MAKE INTERSECTIONS

When a pattern part is designed, it is comprised of a set of lines. Some of these lines may intersect.

With pants, for example, a dart typically intersects a waist line in two points. In order to preserve the correct graphic relationships among lines, it is
important in the current implementation of the system that intersecting lines share points. In this way, when one line is moved (such as during a fitting process), the intersecting line will be adjusted accordingly.

5 When data is received in electronic project data 100, the lines that form the outside of a part are typically connected. That is, the end point of one outside line is the same as the starting point of another outside line. However, internal lines (not on the outside) may not share any points with perimeter lines in original electronic project data 100. Because of this, a make intersections method 510 is preferably used.

The purpose of make intersections method 510 is to force the sharing of points between internal lines and outside lines when intersections occur. Intersections are preferably made separately for each of a project's known sizes because two lines may intersect at different places at different sizes.

20 FEATURE IDENTIFICATION

During preferred classification process 112, features that are expected to be found in a project are determined and stored in expected features slot 519 of each part (see Fig. 6) and expected features slot 521 of each garment (see Fig. 5). An entry for an expected feature typically contains a feature description including a feature name and a feature type (e.g., line or point). In the preferred embodiment of the system, the next step in the annotation process is to identify the expected features corresponding to the entries in expected features slots 519 and 521.

A preferred processing strategy employed during feature identification is to automatically identify as many features as possible. After the system has completed automatic identification, the system user is preferably asked to identify any unnamed features and to verify the accuracy of the derived information. As
currently implemented, the system user interacts with the system through an agenda-driven user interface that is further described in the section entitled User Interface.

The preferred process to identify the expected features of a project is divided into three subprocesses, line identification, dart and pleat identification and point identification.

LINE IDENTIFICATION

The strategy used in the current implementation of the system is to begin by identifying the lines that correspond to a part’s perimeter. In the preferred embodiment, each part 152 has a corresponding part template 634 (see Fig. 12) that contains an ordered list of entries corresponding to the expected perimeter lines of a particular part (perimeter slot 693), and each entry in a perimeter slot 693 is a pointer to a line template 612. The perimeter entries are typically ordered so that the end of one line in the list is expected to be connected to the beginning of the next line in the list. For example, perimeter slot 693 of a part template to describe a pants back part typically includes an ordered list of pointers to the line templates for waist, side, hem, inseam, and crotch.

Using the ordered list of perimeter lines, the preferred system tries to find one line instance in each part that is already identified and matches that line to one of the perimeter line names. When the preferred system is unsuccessful at matching any lines to perimeter line names, the system user is asked to locate the first perimeter line of a part, and after a single line is matched, the system attempts to automatically derive the remaining perimeter lines by associating connected lines in each part with corresponding line names contained in each line template.

After the system automatically derives as many perimeter lines as possible, the system user is
preferably asked to verify the system derivations and to identify any missing names that are contained in expected features slots. Whenever the system user identifies a line, the system typically continues to try automatically deriving more feature names, given the additional input provided by the user. Those skilled in the art will recognize that there are alternate methods for identification. For example, a system user could be asked to identify key points instead of lines, and the system could derive line information from the identified points.

A special condition may arise during feature identification when neither the system or the user is able to locate one of the expected lines. For example, it is not uncommon for a pants pattern to be digitized without a hem line even though the current implementation of fitting process 134 may need a hem line in order to fit pants. When this case arises, the preferred system requests that the system user declare to the system that the line is not present, at which time the endpoints of that line (as preferably defined in endpoints slot 674 in the corresponding line template 612) are included in the list of expected features. In this way, the system ensures that the two endpoints for the absent line will be named and that a subsequent preferred process (called add invisible lines described in fitting preparation process 116) will create the necessary line from the endpoint data.

DART AND PLEAT IDENTIFICATION

The next step in the current implementation of feature identification is to analyze internal lines for darts and pleats. A dart may be detected by finding a set of connected line segments that, when combined, either have one or three corners (see Fig. 32B and Fig. 32C). A pleat may be detected by finding a set of connected line segments that, when combined, have two corners (see Fig. 32D and Fig. 32E). Darts and pleats
typically intersect one of the perimeter lines (a waist, for example).

When a dart or a pleat is identified in the current implementation of the system, it is assigned a name that contains "dart" or "pleat" as a first name and the name of the line that it intersects, if any, as a second name. Examples of a dart and pleat names are (dart side), (pleat waist) and (dart). The current subsystem names darts and pleats in this way so that they may later be identified during fitting preparation process 116.

After the current system has identified darts and pleats based on geometric considerations as described above, the system user is preferably asked to verify the system derived information.

POINT IDENTIFICATION

When all expected lines are named or declared not to be present, as in the special case described above, the preferred system automatically names points of a project based on the available line names. As currently implemented, the two endpoints of each expected line are named with the names contained in endpoints slot 674 contained in the corresponding line template 612 of each line. At this point in the preferred embodiment, the system user is asked to supply any remaining point names from the list of expected features that were not named by the system.

FITTING PREPARATION 116

INTRODUCTION

In the preferred embodiment, a fitting preparation process performs a series of processes in order to produce a prepared pattern knowledge base containing complete garment descriptions accepted by fitting process 134. Accordingly, the purpose of fitting preparation process 116, as illustrated in Fig. 38, is to create a prepared pattern knowledge base 128 beginning with an annotated pattern knowledge base 502.
Fitting preparation typically performs without system user interaction by making use of the information in the annotated pattern knowledge base that was provided by classification process 112 and annotation process 114.

In the preferred embodiment, much of the information that ultimately resides in prepared pattern knowledge base 128 is derived from templates. In particular, the templates of the present system contain generic constraints that describe how to prealter garments of a particular style. Preferred fitting preparation process 116 generates prealteration constraints by substituting the lines and points identified during feature identification 513 for general feature descriptions contained in the generic constraints.

ROTATE PARTS

The fitting process of the preferred embodiment expects parts to be oriented in a specific way. For example, the waist on pants is expected to be at the top of the part. Within each preferred part template 634 (see Fig. 12), a slot named top 656 contains the name of a line that needs to be positioned at the top of the part. In the current system, the line instance corresponding to the top line is established, and then the part is rotated about the center point of the part so that the top line is at the top of the part.

ADD INVISIBLE LINES

Some of the lines that are needed by preferred fitting process 134 may not exist in a particular part. For example, it is common for a pants part not to include a hem line, although a hem line is needed by a fitting process to prealter inseam lengths. In the preferred embodiment, the names of the lines that are required by fitting are stored in an expected features slot 519 in each part instance 152. As described above in annotation process 114, unnamed lines may still reside in this slot when the system operator declares that these lines do not exist on a particular pattern
part. Because these lines are crucial to a normal fitting process, they are typically added at this time. In the current implementation of the system, a special attribute, called invisible, is assigned to these lines so that a preferred post processing process 137 is aware not to plot them.

An endpoints slot 674, residing in each preferred line template 612 (see Fig. 33), lists the names of the two endpoints of each line. In the preferred embodiment, the process to add invisible lines uses the endpoints slot information to create a line for each unnamed line in expected features slot 519 of a part.

**DERIVE MEASUREMENTS**

During preferred classification process 112, measurements important to the alteration of each garment are identified and associated with the garment. Each garment may carry a different set of important measurements. For example, when fitting pants it is important to know about crotch related measurements, and when fitting bodices it is important to know about neck line measurements. As implemented in the present system, a list of the measurements that are important to each particular garment are found in a measurements slot 316 of each garment 148.

In the preferred embodiment, each measurement in measurements slot 316 corresponds to a measurement template 600. For example, the name "crotch length" listed in measurements slot 316 corresponds to a measurement template named "crotch length". Each measurement template preferably includes a description about how to derive a particular physical dimension of a garment. In the preferred embodiment, this description is found in a measurement declaration slot 602 in each measurement template.

All garment styles and variations can normally be measured with the general description contained in each measurement template. For example, the measurement
declaration in the preferred "crotch length" template describes how to measure the crotch length for any garment style. For example, a crotch length measurement declaration could include instructions that convey the following: "compute the crotch length on a particular garment by summing the length of the crotch line (from the waist to the crotch point) on the back and the front parts". These instructions are typically not stored as English sentences, however, but rather as declarative structures.

In the preferred embodiment, a derive measurements method 542 is invoked to create a measurement instance 150 for each important measurement of a particular garment and to perform the declared measurement calculation. The measurement calculation uses the general declarative information found in the measurement template and calculates a specific value for a particular garment. As implemented, the result of a measurement calculation is stored in a graded-data slot 520 in each newly created measurement instance.

Graded data slot 520 typically contains entries of the form: (garment measurement, fit ease, design ease) for each of a project's known sizes. Derive measurement method 542 typically calculates only a garment measurement. Fit ease and design ease are preferably calculated later by a derive ease values method 652 described in adjustment 212.

In addition to deriving a physical dimension, the current implementation of derive measurements method 542 copies declarations including a name 606, a determine adjustment 603, a preference values 673, and a grading factors 671 from preferred measurement template 600 to a newly created measurement instance 150. The preferred way in which adjustments are calculated and used is described in adjustments process 212.

DERIVE COUPLINGS

As implemented in the present system, the next step
in preparing garments is to derive couplings. A coupling is preferably implemented as a declaration that couples an adjustment to a set of garment features for a specific measurement. The amount of adjustment applied to specific features of a garment may be a function of the difference between specific garment measurements (as computed, for example, by derive measurements method 542 shown above) and specific personal measurements. The way in which adjustments are calculated and couplings are applied to alter garments in the preferred embodiment is described in fitting process 134.

The aspect of couplings described here (as part of the fitting preparation process of the present system) is determining which features of a particular garment are affected by a coupling. For example, a coupling for a waist circumference typically needs to adjust all lines of the garment that are affected by lengthening or shortening the waist. On one garment, this may only involve a front and back part; on another garment there may be front, back, waistband and pocket parts that are affected by waist circumference changes. In order to accommodate differences between garment styles, the present system declares in a general way the features which are included in a coupling; then, a derive couplings method 544 described below may be used to determine specific affected features of a particular garment from the general declaration of affected features.

In the preferred embodiment, a generic coupling is stored in a coupling declaration slot 604 of each measurement template 600. The preferred process to derive couplings, as contained in method slot 544, copies the coupling declaration information from the measurement template to a couplings slot 534 contained in each measurement instance 150 corresponding to each particular measurement. When copying a coupling declaration, preferred derive couplings method 544
searches for general feature descriptions and translates them to specific lines and points identified during feature identification 513.

General feature descriptions may be of the form:

\[ \text{FEATURE} \quad \langle \text{type} \rangle \quad \langle \text{part} \rangle \quad \langle \text{name} \rangle \quad \langle \text{match} \rangle, \]

where \text{FEATURE} is a keyword to denote the beginning of a general feature description.

\langle \text{type} \rangle contains the type of feature, for example, LINE, POINT, or DART/PLEAT.

\langle \text{part} \rangle contains a reference to the parts on which features of the specified type are desired. The \langle \text{part} \rangle specification may be a single part name, a list of part names or ALL.

\langle \text{name} \rangle contains a reference to the feature's name or a part of the feature name. A feature name may contain a number of words (e.g., (WAIST UPPER) and (WAIST LOWER) are two lines of a waistband).

\langle \text{match} \rangle tells how to match the \langle \text{name} \rangle to garment features. Three types of matching typically take place: synonym, primary or secondary. Synonym matching may be used to match features with exactly the same name as specified in \langle \text{name} \rangle or a synonym of \langle \text{name} \rangle as established in a list of synonym names residing in the system. Primary matching may be used to match features that have the same "first name" as the name specified in \langle \text{name} \rangle. For example, (WAIST UPPER) is a primary match with WAIST. Secondary matching may be used to match features that have the same "first name" and "second name" as the name specified in \langle \text{name} \rangle.

An example of a general feature description is:

\[ ((\text{FEATURE} \ \text{LINE} \ \text{ALL} \ \text{WAIST} \ \text{PRIMARY})) \].
This feature description may be used to prompt a search through all part instances of a particular garment and find all the lines whose primary name (first name) is WAIST. The result of processing this description is typically a list of pointers to all line instances 162 of a particular garment that correspond to waist lines.

SET LINE DIRECTIONS

In the preferred embodiment, the next step in fitting preparation is to insure that the order of x,y positions in each line is correct for the fitting process, and this is preferably accomplished through a set line directions method 640. The order of x,y positions that comprise a line can affect the way in which a line is altered. When scaling a line, for example, the first point of the line may be fixed, and each successive point may be moved according to some scale factor (the various operations that alter x,y positions of lines is described in fitting). On the other hand, for some lines, order of x,y positions is not important.

When order for a particular line is important, information about the required order may be stored in an anchor slot 614 of a corresponding line template 612.

As implemented in the present system, the anchor slot contains either a point name or a line name, a point name indicating which point needs to be the starting point for the line, and a line name indicating that the line needs to begin at the point of intersection with a line whose name is contained in anchor slot 614.

SET ANGLE DEPENDENCIES

Angle dependencies may be established to specify that limits on an angle between two connected lines are to be maintained during fitting. An angle dependency involves two lines: an independent line and a dependent one. At a specific time in fitting, as further described in the fitting section, the dependent line may
be adjusted so that the angle between it and the independent line conforms to a specified amount. For example, when fitting pants, an angle dependency of 90 degrees typically exists between an independent crotch line and a dependent waist line.

Angle dependency relationships are stored in the preferred embodiment in an angle dependencies slot 384 in each line instance corresponding to an independent line of an angle relationship. The information contained in the angle dependencies slot typically includes a pointer to the dependent line instance and an angle.

As implemented in the present system, the contents of angle dependencies slot 384 are filled based on information contained in each part template 634; each part template contains an angle dependencies slot 654 that contains a list of angle dependency entries; and each entry contains the name of an independent line, the name of a dependent line, and an angle. The angle may either be a fixed value (e.g., 90 degrees) or a calculated value, which may be indicated by the word "measure". When a calculation is specified, the angle between the lines is preferably calculated for the base size of the project (as preferably contained in a base size slot 312 of project 146), with the calculated value used as the angle. Measuring is typically used in cases when a fixed angle does not apply to all garment styles where the angle in the original pattern design needs to be preserved.

An example of an angle dependencies declaration for a pants back part is:

```scheme
  ( ((CROTCH) (WAIST) 90)
    ((INSEAM) (CROTCH) MEASURE)
    ((SIDE UPPER) (WAIST) 270) )
```

This example shows that there are three angle dependencies for parts that correspond to pants back, that two of the angle dependencies are fixed, and that
one is measured from the base size.

In the preferred embodiment, set angle dependencies method 642 analyzes the declaration in each part template corresponding to the parts of a particular project and fills angle dependencies slot 384 in each independent line as needed.

SET ATTACHED POINTS

There are certain aspects of a garment that are not changed when garments are prealtered to match individual body measurements. For example, when pants are sewn to a finished length, the length of a pants hem that is folded up does not typically change as the inseam is lengthened or shortened. Even when a portion of a part is not directly altered by the fitting process, it may still have to be moved. In the example of a pants hem, when the hem point is lowered to lengthen an inseam, the bottom of the pants need to be lowered by the same amount.

To accomplish the necessary movement of points that are not directly altered, those points may be attached to other points that are altered by the fitting process. Then, when a point is altered, the attached points may be automatically moved by an identical amount.

In order to implement the concept of attached points, the preferred embodiment employs an attached points slot 403 residing in each point instance 164 and containing a list of attached points (if any); the contents of the attached points slot is preferably filled by a set attached points method 666. Based on this process, when a line falls outside of the perimeter lines of a part (the perimeter lines being preferably stored in perimeter slot 342 of each part instance 152), the endpoint of the outside line is attached to the endpoint of the perimeter line.

PREPARE DARTS AND PLEATS

Darts and pleats require special handling by the preferred fitting process beyond that which is required
of normal lines. In the preferred embodiment, the next step of fitting preparation is to prepare darts and pleats so that fitting process 134 can alter darts, pleats and lines that have embedded darts and pleats.

Darts and pleats often intersect other lines. For example, a waist often includes one or more darts and pleats. When a waist line containing a dart is sewn, the area between the dart ends is closed which renders a smooth waist line. Preferred fitting process 134 analyzes the effect of closing darts and pleats so that lines to which darts and pleats are connected can be treated as single continuous entities.

Fig. 39A illustrates the components of a dart. In the preferred embodiment, the line segments that form a dart are combined into a single line, referred to as a dart line (818 in Fig. 39A); the two endpoints of a dart line (819 and 820) are called dart ends; and when dart ends intersect a perimeter line, the intersecting perimeter line is referred to as a parent line (labelled 821). In the current system, the point at which the line segments of a dart line form a corner is referred to as a dart point (822 in Fig. 39A).

Pleats, as illustrated in Fig. 39B, may have pleat ends (823 and 824) and a parent line 825. The line segments that form the pleat are also combined, in the preferred embodiment, to form a pleat line (labelled 826). A pleat’s line segments normally form two corners which, in the current system, are identified as pleat points (827 and 828).

In the preferred embodiment, darts and pleats are prepared by a prepare darts and pleats method 644 in order to facilitate prealteration of garments containing darts and pleats. Method 644 may be implemented by creating a dart instance 160 for each dart and pleat in a project and then filling in the following slots of each dart/pleat instance:

attributes: type either "dart" or "pleat".
ends two endpoints of the dart or pleat line.

features line instance of the dart or pleat line.

line line instance of the parent line.

point point instance(s) of the dart or pleat points (see Figs. 39A and 39B).

graded data information about how to close the dart or pleat.

A graded data slot 558 may contain information, referred to here as "to-close" information, about how to close a particular dart or pleat for each of a project's known sizes. To-close information may be calculated identically for darts and pleats and, therefore, the term dart will be used in this discussion to refer to both darts and pleats.

The purpose of to-close information in the preferred embodiment is to describe the graphic transformations needed so that the two dart ends and the two dart lines projecting from the dart ends coincide or align with each other. Aligning the two dart ends and dart lines is one way to approximate the way that a dart is sewn together. In the preferred fitting system, when a dart is closed, the near dart end (that is, the dart end closest to the beginning of the parent line) is fixed, and the far dart end is moved so that the final positions of the two dart ends coincide.

When closing a dart, the current system either applies a translation or it applies a rotation. A translation is typically applied when the two line segments of a dart are parallel. Fig. 32E illustrates an example of parallel lines for which a translation may be calculated as the displacement needed to move far dart point 816 onto near dart point 815.

When the two line segments of a dart are not
parallel, the current system establishes a pivot point and a rotation angle. A way in which a pivot point and rotation angle may be computed is illustrated in Fig. 32A. Point 803 is the near dart end and point 804 is the far dart end. Lines 805 and 807 are the two line segments of a dart line.

A first preferred step in closing such a dart is to compute two directions (806 and 808) formed by two line segments (805 and 807) at the dart ends (803 and 804). The rotation angle, call it r, is defined by (direction 806 - direction 808). The next step in the current process is to compute a pivot point. The pivot point typically falls on a line 809 that begins midway between the two dart ends and projects in a direction 810 which is perpendicular to a line 812 connecting the two dart ends. The position of pivot point 811 may be computed by moving a distance along line 89 equal to:

\[
\left[ \frac{d}{2 \cdot \tan(r/2)} \right]
\]

where d is the distance between the two dart ends and r is the rotation angle.

Various examples of computing to-close information is illustrated in the figures. Fig. 32B illustrates an example of a symmetrical dart where pivot point 813 coincides with the dart point. Fig. 32C illustrates an asymmetrical situation where pivot point 814, obtained by applying the algorithm described above, is outside of the dart. Fig. 32D illustrates a pleat in which the pleat lines are not parallel. Fig. 32E illustrates an example of parallel pleat lines. In this case, a translation is calculated.

A prepare darts and pleats method 644 typically creates dart and pleat instances for each line that is known to represent a dart or pleat. A line instance may be known to represent a dart or pleat when the first
name, as contained in name slot 360 of the line instance 162, is either "DART" or "PLEAT".

SET UPDATE SEQUENCES

As is described in the section entitled Fitting 134, the lines in each part instance 152 are typically updated in a particular order during operations of fitting 134. At least two situations arise that affect the need to order the way in which lines are updated. First, when there is an angle dependency, the independent line of an angle dependency relationship is normally updated before the dependent one. Second, when there is an endpoint-midpoint relationship, as illustrated in Fig. 17, the line containing the midpoint is preferably updated before the one containing the same point as an endpoint. The reasons for updating lines in this particular order is described in fitting.

During fitting process 134, an update sequence slot 352 in each part instance 152 may be accessed to determine the correct order. Furthermore, as part of fitting preparation 116, a set update sequences method 646 may be invoked for each part instance of a project to fill update sequence slots 352.

SET RESHAPE METHODS

Reshaping is a process, performed in the preferred embodiment during fitting, to reshape a line after points on a line have been moved. In the present system, the way in which a line is reshaped depends, among other things, on the contents of a line's reshape method slot 390; this slot preferably contains a list of reshape points and a reshape method for each of the reshape points. The way in which reshape points may be used and in which lines may be reshaped is fully described in the section entitled Fitting 134.

In the preferred embodiment, each line template 612 (see Fig. 33) includes a reshape declaration slot 616 containing a declarative structure that is used to fill a reshape method slot 390 in each line instance 162.
The declarative structure may include a list of entries, where each entry is of the form ((point name) (reshape method) (parameter)). The typical process to set reshape methods as contained in method dot 648 in each line instance is to find the line template corresponding to a particular line and to copy the entries contained in reshape declaration slot 616 to reshape method slot 390 in each line instance. In the process of copying entries, each point name may be converted to a pointer to its corresponding point instance 164 in the current project.

DERIVE EASE VALUES

Ease refers to extra fabric that is needed to achieve a comfortable fit or that is intentionally added to achieve a desired style by a pattern designer. In the current system, a distinction is made between these two types of ease.

Fit ease, as defined in the current system, is the ease required to achieve a comfortable fit. For example, extra fabric may be needed around the hips to be able to comfortably walk and to sit down.

Design ease, as defined in the current system, is additional fabric added for stylistic reasons. For example, extra fabric is added under the arm to create a basic dolman sleeve; accordingly, in the current system, the extra fabric under the arm of a dolman sleeve is treated as design ease.

In the current implementation of the system, information about standard measurements and standard fit ease values are stored in measurement templates 600, each of which typically contain the information needed to establish ease for a particular measurement. As implemented in the current embodiment, a standard measurement, stored in a standard values slot 668, contains a standard value for a particular measurement for each standard size; for example, a measurement template for waist circumference contains a standard
waist circumference for each standard size.

In addition, a standard amount of fit ease may be stored in a fit ease slot 670 of each measurement template for each standard size. In the case of fit ease, slot 670 typically contains either a value (the actual amount of fit ease) or a method to calculate the value. A method, when it applies, may be stored as a fetch demon.

Design ease, for a particular measurement, may be defined as the amount by which the measured value exceeds the total of the standard measurement plus the standard fit ease. Accordingly, design ease may be calculated by taking the garment measurement, stored in graded-data slot 520 of each measurement instance 150, and subtracting the sum of the standard measurement and the standard fit ease.

In the preferred embodiment, each measurement instance 150 contains an ease slot 528 that contains a fetch demon for deriving the total ease for a particular measurement. The fetch demon contents of ease slot 528 are typically copied from a fetch demon stored in an ease slot 672 in each corresponding measurement template 600. The fetch demon contents then reside in ease slot 528 where they are typically accessed during preferred fitting process 134 when adjustments are being calculated.

In the system as disclosed, a derive ease values slot 652 contains a method to establish ease information in each measurement instance 150 of a particular garment. Prior to deriving ease values, each measurement's graded-data slot 520 typically contains a garment measurement, zero fit ease and zero design ease. A derive ease values method 652 may then be used to compute fit ease and design ease and to replace the zero ease amounts in the graded-data slot with newly calculated ease values.
INTRODUCTION

In the preferred embodiment, the purpose of fitting process 134 is to alter the geometry of pattern data in a prepared pattern KB 128 that has been prepared by pattern preparation process 110 so that a pattern or garment represented by the data fits a specific set of standard or individual body measurements 132 (Fig. 15), thus producing prealtered pattern data 136 customized to fit the given standard or individual body measurements. In an embodiment where pattern preparation process 110 and fitting process 134 are integrated on the same processor 80, a fitting process may operate from data stored by pattern preparation process 110. Alternately, a fitting process 134 may accept a prepared pattern KB 128 from portable mass storage media 81 or through other data communication. Body measurements 132 may be accepted from a variety of media or from other means.

In the preferred embodiment, fitting process 134 is accomplished by satisfying geometric and measurement constraints in prepared pattern KB 128 through a constraint satisfaction mechanism that alters points and lines that depict a garment 148. As shown in Fig. 1A, fitting process 134 may be implemented by a sequence of four steps; grading 210, adjustment 212, redrawing 216, and output preparation 218. The detailed operation of the fitting process as implemented in the present embodiment may vary somewhat from one project to the next. This variation may arise because the particular set of constraints contained in one prepared pattern 128 may differ from the set contained in another prepared pattern, and may be controlled by the structure of the prepared pattern itself, particularly with regard to preferred constraints represented by adjustments 530 and couplings 534 in measurement objects 150, as explained below. Another source of variation is that the preferred grading step 210 may be considered optional in
the sense that the pattern data may not contain information necessary to drive a grading process, in which case grading does nothing in the preferred embodiment.

5 In the fitting process discussed below, the processes as implemented in the preferred embodiment are described; those skilled in the art will recognize that many alternate implementations are possible.

FITTING GARMENTS

10 A project 146, which may contain data describing one or more garments 300, may be altered to fit a set of standard or individual body measurements 132 by altering the data which depicts each of its garments and preparing the data for output.

15 In the preferred process 148 for altering the garment, a garment is first graded to one of the known sizes 308 (Fig. 4) of the current project, the size being selected by sending a Determine-Closest-Grade message 325 (Fig. 5) to the garment. The approach used in the preferred embodiment of grading to a known size before altering to fit has two advantages. First, the magnitude of the adjustments that have to be made is generally kept small, so that unintended distortion of the pattern may be minimized. Second, freedom is maintained for a garment designer to alter the overall proportions of the garment from one standard size to the next. For example, the designer may want to make the smaller sizes of a skirt pattern flare more than the larger sizes to achieve a desired visual effect.

20 However, it is not necessary that there be more than one known size for the fitting process to work. In particular, if the body measurements correspond to a standard size rather than a particular person, the fitting process described herein may be used to grade patterns in accordance with specifications of standard sizes, which may be in the form of body measurements corresponding to the standard sizes, or may be in the
form of specifications about how garment measurements themselves vary from one standard size to the next.

In the present embodiment, actual alteration of garment and part features begins by sending a process-couplings message 536 (Fig. 11) to each of the measurements 150 listed in the garment’s measurements slot 316 (Fig. 5). Referring to Fig. 11, couplings slot 534 in each measurement instance may contain a set of couplings 220 (Fig. 41) which describe how an adjustment 530 determined for that measurement is to be applied to the features of the garment and its parts. The function of preferred process-couplings method 536 is to interpret these descriptions, and will be described below.

In the preferred embodiment, after process-couplings method 536 (Fig. 11) has completed the desired moving, scaling, rotating, etc. of features, many lines will have become discontinuous, as shown in Fig. 30, and will require redrawing. Accordingly, the preferred embodiment employs a redrawing process 216 in order to restore continuity of lines and restore important relationships between lines. This preferred redrawing process is further described in a subsequent section.

Although in the preferred embodiment couplings 220 in slot 534 may refer to values stored in adjustments slot 530, the adjustments need not be calculated ahead of time. As implemented in the current system, the adjustments are filtered by a fetch demon method, determine-adjustment 531; accordingly, the adjustments are preferably calculated when they are requested for the first time. This approach has the advantage that the various adjustments can depend on one another, and they can be made to automatically "chain" together to accommodate interdependencies among measurement adjustment calculations without requiring that a dependency graph be explicitly analyzed beforehand, as long as there are no cycles in the dependency graph. If
the demon mechanism were not available, an approach consisting of generating and analyzing a dependency graph would also be effective. DETERMINATION OF CLOSEST KNOWN SIZE

As described above, a typical process for altering garments 322 (Fig. 5) will start by grading the garment in question to a known size 308 (Fig. 4) that is closest to corresponding body measurements 132. A sorted list of entries may be created, each entry containing one of the project known sizes 308 and a number representing the difference between a value 516 and an ease 528 for a garment’s grading-measurement 326. The resulting list may then be searched, comparing each entry with the value in a slot in body measurements 132 having a name that matches the garment’s grading-measurement 326. The known size contained in the first entry that satisfies the search criterion may then be taken as the size for grading. The search criterion is that the difference between value 516 and ease 528 be less that the chosen body measurement. If the body measurement is less than the number in the first entry in the list, then the first entry will be the size chosen, thus guaranteeing, in the preferred embodiment, that a size will be chosen. If there is only one known size for a garment, a system may bypass this processing.

Garment measurements are fetched from graded-data slot 520 of the measurement instance 150 in which the contents of name slot 512 matches the contents of the garment’s grading-measurement slot 326. The graded-data slot is assumed to contain an entry of the form <value fit-ease design-ease> corresponding to each of the known-sizes 308.

GRADING 210

INTRODUCTION

Grading 210 is a preferred process of configuring a project, garment, part, measurement, or feature in a particular pre-defined size. In the preferred
embodiment, grading process 210 may be used as an initial step in a fitting process 134 in order to configure a garment to approximately fit a set of body measurements 132, in preparation for the "fine-tuning" of an adjustment step 212. When used in this way, the inclusion of a grading step in an overall fitting process can simplify a subsequent adjustment step by eliminating many details from consideration during adjustment. For example, such details as placement of pockets, width of collars and cuffs, number of buttons, etc. may be adequately specified by the overall size of a garment and need not be further adjusted during the remainder of prealteration.

Grading 210 is accomplished in the preferred embodiment in one of two ways. If all the relevant data (positions of points, values of measurements, dart and pleat geometry, etc.) for each of the known sizes 308 of a current project 146 is available before the fitting process 134 starts, grading is a simple retrieval process, called "retrieval grading". If graphical and other data is available for only the base size, then either grading may not be done at all, or a "knowledge-based" grading process may be used. In the case where the grading step is skipped, the preferred subsequent adjustment 212 and redrawing 216 processes may be implemented by simply working from a base size.

RETRIEVAL GRADING

Retrieval grading may be used for a particular project when all of the graphical data for each of the desired grades is made available to the system during pattern preparation as described earlier. In the preferred embodiment, a list of available grades is contained in a known-sizes slot 308 of the current project 146 (Fig. 4). Retrieval grading as described here is incapable of configuring a garment in a size other than one of these known sizes.

Referring to Fig. 5, a garment 148 may be graded by
sending a grade message 320 to all its constituent parts 152 and measurements 150, listed in corresponding slots 314 and 316, to grade to the requested size, and finally by recording the new size in size slot 328, as shown in the following method description. This approach assumes that each of the components of a garment, parts 152 and measurements 150, contains a grade method slot that may be used to retrieve the appropriate known size data.

Parts may be graded by grading each of their features and then generating snapshots for each of their lines. In the present embodiment these snapshots are later used, during redrawing process 216, to determine the extent to which the lines have been distorted by the fitting process; this is preferably used to guide a reshape operation, discussed as part of redrawing process 216.

Measurements 150 (Fig. 11) may be graded to a given size by retrieving a record of the form <value fit-ease design-ease> from a graded-data slot 520 corresponding to the given size, splitting it up, and filling in a value slot 516, a fit-ease slot 524, and a design ease slot 526 with the results. In the system as described, ease 528 and adjustment 530 slots are emptied at this step in the process to allow the demon mechanism to operate during adjustment step 212.

Points 164 (Fig. 9) may be graded by fetching the stored position for the requested size from a graded-positions slot 408 and saving it in a position slot 406. In the preferred embodiment, a position is simply an (x, y) pair.

A line 162 (Fig. 8) in the system as described here may vary from size to size for two reasons. First, the positions of points comprising the line (which may be contained in a knots slot 382) may vary from size to size, and that data typically resides in the constituent points. Therefore, the method for grading points 164 as described above will preferably automatically cause a
line to be drawn in proper position for a given size. Second, as discussed above in the description of line object 162 and in the description of annotation 114, the set of points 164 comprising a particular line 162 may vary from size to size due to the placement of alignment guides and the occurrence of line intersections. A process for grading lines fetches a size-specific list of points from a line's graded-data slot 383 and places the list in the line's knots slot 382.

In the present embodiment, adjustment process 212 operates on individual points, lines, darts, etc., and redrawing process 216 resolves the discontinuities that may result, as shown in Fig. 30. Redrawing process 216 discovers these discontinuities by comparing the positions of points with their positions in the unaltered lines. The positions of points in an unaltered line may be stored in a snapshot slot 388 (see Fig. 8) in each line instance 162. In the system as described, snapshots are generated for each line in a garment after the garment is graded by invoking a snap method 389 for each line. The resulting snapshot, stored in the line's snapshot slot 388, may be a collection of records of the form <point position> where point is a pointer to a point instance 164 in the list of the line's knots 382, and position is the \((x, y)\) position of that point as fetched from the point's position slot 406 at the time of the snap operation.

Darts 160 (Fig. 14) may be graded by filling in a to-close slot 558 with data retrieved from a graded-data slot 560. In the present system, the to-close slot of each dart contains a record of the form <pivot rotation translation> specifying what must be done to close the dart so that the line broken by the dart will be continuous. The semantics of the to-close record are provided in the description of dart objects in the section "Description of Objects".

KNOWLEDGE-BASED GRADING
Knowledge-based grading (Fig. 16) is a preferred process of configuring a project or garment in one of a set of pre-determined sizes, given graphical data for another size. In the preferred embodiment, it is not normally used as a grading step 210 in fitting process 134, but rather is typically used to produce graded pattern data 131 for the pre-determined sizes themselves.

Two approaches to knowledge-based grading are described herein; those skilled in the art will recognize that others are also possible. The first approach, as illustrated in Fig. 16, simply uses a set of standard body measurements 120 corresponding to each of the desired graded sizes to drive a normal fitting process 135 from a single base size, thereby producing graded pattern data 131.

If the fitting process being used requires body measurements that are not known for the various standard sizes, they may be derived from known measurements by establishing proportionality factors among the measurements based either on statistical studies of anthropometric data or on measurements of existing garments in various sizes that are known to fit satisfactorily. These proportionality factors may then be expressed in terms of derivation descriptions 735 (Fig. 15), which may be simply algebraic formulas of the form \( m = f(m_1, m_2, \ldots) \), where \( m \) is the unknown measurement, and \( m_1, m_2, \ldots \) are either known or can in turn be derived from known measurements. A measurement completion process 133 may then use the derivation descriptions to automatically derive estimates for body measurements that are expected by measurement constraints 150 but are missing from a particular set of standard or individual body measurements.

A second approach to knowledge-based grading uses knowledge about particular garments as derived during pattern preparation 110 along with generic knowledge.
about grading as contained in measurement template objects 600 (Fig. 37) and knowledge about the desired pre-determined sizes as contained in grade-sets 775 (Fig. 20). The first approach described here has the advantage of being very easy to use; the second has the advantage of enhanced flexibility for the garment designer in terms of changing garment proportions across the size range. The second approach will be described in more detail below.

Generic knowledge about grading could comprise grade descriptions (e.g., a grading factors slot 517 in a measurement instance 150 (Fig. 11)) that describe how physical dimensions of garments should vary from one size to the next. As shown in Fig. 21, a set of pre-determined sizes may be thought of as a graph. A knowledge-based grading process can operate by finding a path through this graph from the current base size to a desired new-size, then adjusting the various measurements according to this path in a manner analogous to preferred adjustment process 212 as used in preferred fitting process 134 and as described in a subsequent section, then updating part geometry as in preferred redrawing process 216, and finally preparing the data for output as in preferred output preparation process 218. Line snapshots are generated twice, once before applying the couplings that account for grading adjustments, and once before applying couplings as part of a subsequent adjustment process 212.

As shown in Fig. 21, a set of pre-determined sizes may be thought of as having two types of connections. Connections among individual sizes within a grade set, as symbolized by arrows 771, 772, and 773, may be thought of as grade-range connections. Connections among the base sizes of the various grade sets 775 (such as Misses 10 774), as symbolized by arrow 770, may be thought of as grade-set connections. A grade path may comprise a series of steps through this graph along
defined connections. For example, a path from Misses 12 to Juniors 5 according to the connections defined in Fig. 21 would be (Misses 12, Misses 10, Juniors 9, Juniors 7, Juniors 5).

In the preferred embodiment of the present system, both types of connections are implemented by grade set objects 775 (see Fig. 20). Grade range connections such as 772 (Fig. 21) may simply be successive elements in a sizes slot 778 of a particular grade set instance 775.

In a system containing more that one grade set, a special grade set may be allocated whose sizes are a list of base sizes 777 of various grade sets 775.

Adjustment step 212 of fitting process 134 in the preferred embodiment consists primarily of calculating adjustments and applying couplings. These operations are described below under adjustment process 212. A grade-adjust process is an alternative to a fitting adjustment process. In this case, the calculated adjustment is not typically based on body measurements, but instead on grading factors 517 stored in individual measurement objects 150 and on grading indices stored in sizes slots 778 in grade-set objects 775. The grade-path calculation may be broken down into a sequence of grade steps, and each transition may be matched with a range entry in a grading-factors slot 517 in a measurement object 150 and with entries in sizes slot 778 in grade-set objects 775. The resulting factors may be multiplied together and summed to yield an adjustment value.

The remainder of the knowledge-based grading process in the preferred embodiment is identical with adjustment 212 and redrawing 216 steps of fitting process 134.

PROJECT GRADING

The capability to grade an entire project is provided in the present embodiment for completeness and convenience. At the project level (Fig. 4), grading process 302 consists of first checking to ensure that
the requested size is available, and then sending grade messages 320 to all the garments listed in garments 300 specifying the new size. If retrieval grading is to be used as defined above, it is preferably an error to request grading to a size that has not been supplied in the original input data; available sizes may be those that comprise a list in known-sizes slot 308 of a particular project 146.

ADJUSTMENT 212

INTRODUCTION

In the preferred embodiment, an adjustment step 212 of the fitting process is responsible for computing and applying the changes that are required to alter standard size garments to fit an individual. Prior to this step, the garments in a pattern KB have been graded to a known size 308 that most closely matches an individual's body measurements 132. After preferred adjustment process 212 is completed, the various graphical features will have been moved, scaled, rotated, etc. as needed, but these changes will not have been coordinated between features. This situation is illustrated in Fig. 30, which shows a line 234 after grading 210 has positioned its points according to one of the known-sizes 308, as previously described; the same line 235 after adjustment 212, the process currently being described, has moved a point; and again the same line 236 after redrawing 216, a process yet to be described, has redrawn the line to account for the movement of the point.

In the preferred embodiment, four major operations comprise the adjustment step, all controlled by methods in measurement instances 150 (Fig. 11). These preferred methods are determine-fit-ease 525, which adjusts the fit ease of a particular measurement to account for body proportions as determined from body-measurements 132 and to account for personal preferences as listed in ease preference slot 733 (Fig. 15), determine-ease 529, which combines fit-ease and design-ease and takes into account
dependencies among the various measurements of a garment with respect to total ease, determine-adjustment 531, which computes the amount by which a particular physical dimension of a garment needs to be changed to fit the given body-measurements 132, and process-couplings 536, which applies the computed adjustments to the garment pattern data in order to accomplish the desired change in physical dimension by altering the points and lines depicting a garment as determined by couplings 220 (Fig. 41), residing in slot 534, that were generated during pattern preparation.

COUPLINGS

In the preferred embodiment, adjustment process 212 may be initiated by invoking a process-couplings method 536 in each of the measurement instances 150 listed in a measurements slot 316 of a garment 148. The remainder of an adjustment process 212 may be implemented to occur as fetch demon methods determine-fit-ease 525, determine-ease 529, determine-adjustment 531, and get-adjusted-value 535 are invoked; the details of the control flow preferably depend on the contents of the couplings and the various demon methods. Alternative approaches would include fetching the ease and adjustment values before invoking the process-couplings method, thereby causing the demon methods to perform the necessary calculations, or calculating the various ease and adjustment values explicitly in such a way as to ensure that a calculation is not attempted until the results of other calculations it depends on are available.

In the preferred embodiment, a coupling 220 (Fig. 41) is a description of what has to be done to a garment to account for an adjustment to a particular measurement; it "couples" an adjustment to the features of the garment and its parts. Couplings 220 may be records of the form <comment features operation parameters predicate> as shown in Fig. 41. As
implemented, comment 222 contains a human-readable description of the coupling, for the convenience of a programmer. The list of features that are the subject of the coupling may be contained in a features slot 224. Preferred operation 226 may specify the operation to be performed on the features, such as move, scale, rotate, etc. Parameters 228 may be a list of forms to be evaluated that yield a list of parameters appropriate for the specified operation. For example, if the operation is move, evaluation of the parameters may yield a list of the form (x-offset y-offset). A predicate 230, when present, may be evaluated to yield a boolean value that will block application of the coupling if it is false.

The following coupling, from an example measurement instance called "Waist-Circumference", shows how the waist seams on a garment could be changed in response to a calculated adjustment. In this example, features 224 is a list of affected garment features, listed by name, operation 226 is "scale", which will cause the lines to be expanded uniformly according to the adjustment, and parameters 228 is a list of one item, a form to be evaluated by a Lisp interpreter. In standard notation, it would read:

\[ 1 + \text{Adjustment \ self:value} \]

In other words, if the value of the waist-circumference were 50 cm and the adjustment turned out to be 10 cm, the argument to the scale operation would be 1.2. In this example, predicate 230 is empty, meaning that the coupling will always be applied when a measurement instance containing this coupling receives a process-couplings message.
WAIST-CIRCUMFERENCE:

Comment 222: "Scale waist seams on back and front"

Features 224: (waist-front waist-back)

Operation 226: Scale

Parameters 228: (+ 1 (/ Adjustment (self:Value)))

Predicate 240: -

Couplings 220 as described above may be interpreted or "applied" by a process-couplings method 536. There may be several couplings in couplings slot 534, each of which is typically processed individually. When predicate 230 is blank, or if evaluation of the predicate produces a true value, then processing may proceed; otherwise no further action is typically necessary. Parameters 228 may then be evaluated, and the resulting list may be used as the set of arguments for a message, as specified by operation 226, to be sent to each of the listed features 224.

A critical step in the preferred method is setting up an evaluation context for parameters 228; recall that evaluation, in the preferred embodiment, refers to invoking an interpreter. The evaluation context in the preferred method defines four symbols, which may be referred to by the text of the couplings. They are self, the measurement instance itself; garment, fetched from the measurement's garment slot 514; person, the body measurements 132 being adjusted for; and adjustment, fetched from the measurement's adjustment slot 530. Since fetching from adjustment slot 530 may be implemented to automatically invoke determine-adjustment demon method 531, the adjustment may automatically be calculated at this time. An adjustment calculation in turn typically refers to ease slot 528, automatically invoking a determine ease demon method 529, which in turn preferably refers to fit-ease slot 524, automatically invoking a determine-fit-ease demon
method 525. An adjustment calculation may also refer to the adjustments of other measurement instances, in turn triggering the necessary calculations in those instances.

ADJUSTMENTS

In the preferred embodiment, adjustment slot 530 of a measurement instance 150 may contain either a default process, given here, or a specialized process, an example of which is provided below. The default process in the preferred embodiment subtracts the contents of value slot 516 of the measurement instance from the sum of the corresponding body measurement, as determined from body-measurements 132, plus the contents of ease slot 528 of the measurement. As is typical in a fetch demon method, the current value of the slot, when present, may be used; otherwise a new value may be computed and placed in the slot. In this way, each slot value need be computed only once per fitting, regardless of the number of times it is accessed. Note that this approach typically requires that the slot be emptied before the commencement of adjustment process 212. In the current system, this is accomplished in measurement:grade method 522.

Calculation of an adjustment 530 normally refers to the value of ease slot 528 of the same measurement instance 150. The value contained in an ease slot 528 may automatically be derived on demand by demon method determine-ease 529. A default process for a determine-ease method 529 may simply add the fit-ease 524 to the design-ease 526.

Ease calculations may be used to determine relationships between body measurements 132 and physical dimensions of a garment. These calculations normally refer to contents of a fit-ease slot 524 of the same measurement instance, which may automatically invoke a determine-fit-ease demon method 525. In the preferred embodiment, a user's personal preference related to
desired fit may be accepted as an ease preference slot 733 of body measurements 132 (Fig. 15). As implemented, a default process for a determine-fit-ease demon method 525 returns the value of fit-ease slot 524 modified in accordance with a user's ease preference value. Preference-values slot 533 in measurements 150 (Fig. 11), in the system as described, may contain constraints on ease preference in the form of two ratios, called tight-ratio and loose-ratio, and the ease-preference slot 733 of body measurements 132 may have one of the values tight, normal, or loose.

ADJUSTMENT OPERATIONS

In the preferred embodiment, once parameters 228 of a coupling 220 (Fig. 41) have been evaluated, a final step in applying couplings is to send the message specified by operation 226 to each of the features listed in features 224 of the coupling. The repertoire of coupling operations available in the present system includes point:move 410, line:move 393, line:scale 395, line:change-angle 396, dart:move 561, dart:lengthen 562, and dart:widen 564.

A point 164 may be translated or moved by invoking its move method 410, which in the preferred embodiment changes the x and y components of the contents of its position slot 406, as shown below.

A line 162 may be translated or moved by invoking its move method 393, which in the preferred embodiment moves all points 164 listed in knots slot 382, and by then modifying each of the entries in snapshot slot 388 by the same delta-x, delta-y amounts, as shown below. In the preferred embodiment, modification of the snapshot entries 388 is necessary because moving a line does not distort it, although it may distort other lines, and because comparison between the snapshot data and the final actual positions of the same points is used during redrawing process 216 to determine the amount by which the line has been distorted during
adjustment process 212.

In the preferred embodiment, each line 162 includes a scale method slot 395, which may be used to change the length of the line without changing its shape by isomorphic scaling, as illustrated in Fig. 22. Because the shape is not changed, the snapshot data 388 will also preferably be modified to avoid having redrawing process 216 attempt to reshape the line. The scale factor may be a ratio specifying the amount by which the line is to be shrunken or expanded, with a value of 1.0 causing no change to the line. The scale method may also take an optional to specify that one of the points on the line is to remain fixed during the scaling operation. If this option is not provided, the first point in the list of knots will typically remain fixed. Fig. 22 illustrates the operation of this method; 287 is a line before scaling, and 288 is the same line after scaling by 1.5 with a point 280 as the origin. Note that darts may be handled in such a way as to be unchanged by this operation.

In the preferred embodiment, most of the work of scaling lines is performed by a scale-knots process. Referring to Fig. 22, point 280 is the first point in the list "knots". As implemented, the positions of points 281 and 282 are scaled about point 280; then, the dart and points beyond the dart, that is points 283, 284, 285, 286, and all points on line 287 beyond point 286, are moved the amount by which point 282 was moved; the origin is then moved to point 284, and the remainder of the knots are then scaled about point 284 in the same fashion.

A change-angle method 396 may be invoked by a coupling 220 to change the angle specified by an angle dependency in slot 384 of a line 162 (Fig. 8). As previously noted in the description of line object 162, angle-dependencies slot 384 may contain records of the form <point dependent-line angle> which specify that the
angle between a line 162 containing the record in its angle-dependencies slot 384 and a dependent-line specified in the record about the specified point must be the specified angle. Since many lines are curved, the angle specified in an angle dependency record is typically the angle between the tangents of the lines 162 at the point 164 where they join.

Both darts and pleats in a garment may be represented by instances of dart 160 (Fig 14), and they may be distinguished by the value of the type attribute, listed in attributes slot 556. For the purpose of the present discussion, both will be referred to as "darts".

A move method 561 for dart instances 160 may be invoked during line:scale 395, as described above.

Features 154 listed in the dart's features slot 555 may then be moved by sending move messages to them, and the position of the pivot listed in the dart's to-close slot 558 may be translated by the given delta-x, delta-y.

In the preferred embodiment, adjustment methods for darts 160 may include lengthen 562 and widen 564. The lengthen operation may change the length of a dart without changing its angle or the length of the line intersected by the dart, as shown in Fig. 27, and is preferably meaningless for dart instances 160 that are identified as pleats, the identification being accomplished in the preferred embodiment by the value of a type attribute in an attributes slot 556. Darts may be lengthened by scaling them by the ratio new-length/old-length. Referring to Fig. 27, the dart itself, including point 754 and the to-close data for the dart, and the portion of the line 757 between the two dart ends 752 and 755 may be scaled by this process; the portion of line 757 beyond point 755, including 756, may then moved by the amount by which point 755 is moved. Therefore this process preserves the length of the original line 757 as it appears on a garment; i.e. with the dart closed.
In the preferred embodiment of the system, widening a dart or pleat causes it to occupy more space on the line 162 that is broken by the dart, the line preferably being listed in the dart's line slot 554. This may be done without changing the length that the line in slot 554 will have when the dart is closed or sewn together, as in a finished garment. The rotation angle in the dart's to-close slot 558 is also typically changed unless the dart closes with a translation rather than a rotation, in which case the translation in the to-close slot 558 is preferably changed instead. As shown in Fig. 28, this may be achieved with a one-dimensional scaling operation anchored along a line 767 through a point 762 parallel to the centerline of the dart or pleat. The length of the line on the finished garment may be preserved by shifting all points on a line 768 that lie to the right of point 765 by the same amount that point 765 was moved by this process.

INTRODUCTION

In the preferred embodiment, a redrawing process 216 is used during fitting process 134 to restore continuity to lines on which individual points have been moved as a result of applying the couplings during adjustment process 212 and to satisfy other geometric constraints that are represented in the prepared pattern KB 128. Parts 152 may be redrawn by sending update messages 380 to the individual lines 162 (Fig. 8) as listed in features slot 334 of each part 152 (Fig. 6). In the present system, the order in which this is done is controlled by an ordered list of lines 162 contained in a part's update-sequence slot 352, which may be used to ensure that, in each dependency relationship between two lines 162, the independent line is updated before the dependent line. In the system as described, dependency relationships may be either endpoint-midpoint dependencies or angle dependencies. In an endpoint-
midpoint dependency, shown in Fig. 17, the independent member may be shown as a line 238 on which a shared point 240 is a midpoint, and the dependent member may be shown as a line 239 on which a shared point 240 is an endpoint. An angle dependency, shown in Fig. 29, is typically a record of the form <point dependent-line angle> contained in an angle-dependencies slot 384 of a line 162. The independent member of this relationship 242 may be a line containing the angle-dependency record, and the dependent member 243 may be a line referred to as dependent-line in the angle dependency record.

REDRAWING PARTS

Parts 152 may be redrawn by sending update messages 346 to them. In the preferred embodiment, each line included in a part's features, as listed in slot 334 (Fig. 6), includes a requested-changes slot 394 (Fig. 8) that is used to communicate dependencies among lines during redrawing 216. These slots are typically emptied at the beginning of the process. Each line 162 may then be sent an update message, the order typically being controlled by the part's update-sequence slot 352 as described above.

REDRAWING LINES

Lines 162 (Fig. 8) are redrawn, in the preferred embodiment, by a process residing in update slot 358. Update slot 358 may be inherited from features 154, a superclass of line 162, but the process described here is preferably unique to instances of the line class 162. Since darts or pleats 160 may break a line 162 up into segments, any darts that lie along a line are typically closed before redrawing occurs and opened again afterward. A preferred process of closing and opening darts is described subsequently and is illustrated in Fig. 35.

In the preferred embodiment, an update process 380 is used to restore continuity to a line, otherwise
called reshaping the line, as illustrated in Fig. 30. For each point 164 in the line's knots slot 382 for which the current actual position of the point 164 differs from the position recorded for that point in the line's snapshot slot 388, the line may be reshaped according to either a reshape-method recorded for that point in a reshape-methods slot 390 corresponding to the line, or according to a default reshape method.

An additional function of update process 380 as described here is to satisfy angle dependency constraints by propagating angle adjustment requests from independent lines to dependent lines, and by processing angle adjustment requests. As was explained earlier, in the system as described herein, an angle-dependencies slot 384 of an independent line may contain angle-dependency records that specify a dependent line, a point 164 that the two lines share, and an angle that the two lines must form around the given point. A graphic example of an angle dependency is illustrated in Fig. 29, where line 242 is an independent line, 243 is a dependent line, 244 is a point where they meet, and 245 is an angle between them that is to be preserved. As a final step in the process of updating an independent line as described herein, each angle-dependency record residing in the line's angle-dependencies slot 384 is processed, calculating the proper direction for the corresponding dependent line at the common point. A request may then be posted in a requested-changes slot 394 of the dependent line, specifying the type of request, "restore-angle", the common point, and the direction that the dependent line must take at the point. A dependent line, after being reshaped as described below, may pick up requests from its requested-changes slot 394 and invoke its restore-angle method 398 for each one in turn.

Darts 160 (and pleats, which are distinguished, in the preferred embodiment, from darts only by graphic
representation and the contents of a type entry in attributes slot 362) may be closed or opened by invoking a close-darts process, which preferably combines segments of a line broken by darts or pleats using coordinate transformation. As illustrated in Fig. 35, the line containing the dart comprises two segments 255 and 257 (the points between the two dart ends have been omitted for clarity). The close-darts process may be implemented by searching the given list of points for a point having an entry for a dart-end in an attributes slot 362. One such point is identified as 261 in Fig. 35. If one is found, the dart 160 may be closed using the information in its to-close slot 558. If the to-close data specifies a rotation, the remainder of the points in knots (points beyond 261) may be rotated through an angle 260 specified by rotation about a pivot point 256. Note that this is not necessarily the same point as the dart point 258. If, on the other hand, the to-close data specifies a translation, the remainder of the points in knots may be moved by the specified translation. At this point, the two points 261 and 262 lie on top of one another. Finally, the routine may be invoked recursively, using the list of points beyond the second end 262 of the dart that was closed, to close any further darts that might lie on the line.

Lines 162 may be reshaped by invoking a method in a reshape slot 397 (Fig. 8). Reshaping may be necessary when an individual point 164 is moved independently of a line that includes the point in its knots slot 382.

This situation is illustrated in Fig. 30, where 234 is an unaltered line, 235 is the same line after a point 237 has been moved independently of the line, and 236 is the same line after completion of a reshape process such as the one described below. The fact that a point has been moved independently of a line whose knots list 382 contains the point may be detected by comparing the contents of a position slot 406 of the point 164 with
the position of the point as recorded in the line's snapshot slot 388. If a line 162 is moved or scaled, the snapshot data for the line is typically updated, as described in adjustment process 212, to reflect the fact that these operations do not change the shape of the line. In general, however, since moving a line means moving all of the points listed in its knots slot 382, other lines that connect to the line being moved will also have those points moved independently of the connecting lines.

In the preferred process, a list of points 164 that have been moved independently of the line 162 is collected, along with, for each such point, the position as recorded in the line's snapshot slot 388, the current position as stored in the point's position 406, and two items fetched from the line's reshape-methods slot 390, a reshape-method and a parameter. These two items may be used to specify a desired shape characteristic of a line. This collection of points and associated reshape data may then be passed off to a process called reshape-curve, to be described subsequently.

A reshape-curve process takes apart the reshape-points list and activates individual reshape processes. For each of the available reshape methods, which in the present embodiment include linear, isomorphic, sinusoidal, and cubic, there is a process that accepts a list of points, an offset, and an optional parameter. Each of them, as described herein, operates by assuming that the last point in the list of points has been moved by the given offset, and moves the remainder of the points to shape the line. For this reason, the process described here manipulates the list of points to ensure that last point is the point being moved. Therefore, in the preferred embodiment, if the first point in a parameter "knots" is the point being moved, the knots are reversed; if a middle point is being moved, the knots are split, and the second portion reversed; and if
more than one middle point appears in the reshape-points parameter, the line is split multiple times. This is accomplished in the process described herein by making a recursive call to a reshape-curve process.

In the preferred embodiment, there are four different methods for reshaping a line in response to independent movement of one of the points in the line's knots slot 382. They are isomorphic, which distributes the change evenly along the length of the line, thereby preserving the overall curvature profile of a line; sinusoidal, which has the effect of preserving the tangents of the original line at the end points and at the moved point; cubic, which concentrates the change closer to the point that was moved, and linear, which applies an anisomorphic coordinate scaling operation to the line. This particular set of reshaping operations has been arrived at by attempting to emulate the manual operations performed by persons skilled in the art of garment pattern alteration. The more typical cubic spline approach was rejected because of the difficulty encountered by inexperienced operators in placement of guide points. Each of the reshaping methods as enumerated above is described in detail below. Each of them is implemented in the preferred embodiment as a process accepting three arguments: "knots", a list of points 164 describing the segment being reshaped; "offset", an amount by which the far end of the segment has been moved; and "parameter", an optional argument whose meaning varies according to the particular reshape method as described below.

The operation of the isomorphic reshape method is illustrated in Fig. 23. A graph 250 shows the preferred relationship between point motion and distance along a segment. The illustration shows an original segment 251 and the same segment 252 after a point 253 is moved and the segment reshaped to accommodate the motion of point 253 according to this method. As can be seen from the
illustration in Fig. 23, the effect of this reshaping method is to distribute the change evenly along the length of the segment, thereby causing a minimal amount of overall distortion to the shape of the line.

Sinusoidal reshaping, as illustrated in Fig. 24, may be accomplished in a similar fashion to the isomorphic method explained above, except that a sinusoidal transfer function 265 is used instead of a linear one to relate point motion to distance along the line. As can be seen from Fig. 24, the effect of this method of reshaping is to preserve the direction of the segment at both of its end points. For this reason, it is typically used when a middle point along a line has been moved, to avoid the formation of "corners". A cubic reshaping method, illustrated in Fig. 25, also operates on a "transfer function" principle, as did the isomorphic and sinusoidal methods described above. A cubic transfer function 267 has the property that if the slope is \( p \) at \((0, 0)\), the slope will be \( 1/p \) at \((1, 1)\).

Practical limits for the value of \( p \) with this approach are from 0.33 to 3.0, since with values much outside this range, the second derivative of the transfer function changes sign within the interval \((0, 0)\) to \((1, 1)\), leading to serious distortion of the line being reshaped. As may be seen in Fig. 25, the effect of this method of reshaping is to concentrate the effect of the point motion closer to the end point that was moved, thereby minimizing disturbance at the opposite end of the segment.

A linear reshape method operates on a different principle than the "transfer function" principle described for the three methods described previously. This reshape method has the advantage that a curve can be "bounded" as illustrated in Fig. 26. In this example, no point on curve 292 will cross line 296 as long as offset 295 does not cause point 294 to cross line 296. This property holds when fixed point 290 lies
on a vector 296 that determines the orientation of a local coordinate system as described below, and when all points on the original line 292 lie on the same side of vector 296 as a point 294 that is to be moved.

According to this method, a rectangle 293 is inscribed about a curve 292, and a local coordinate system within the rectangle is distorted by anisomorphic scaling to accomplish the reshaping. Referring to Fig. 26, the linear reshape method as may accept an argument, point 290 in this example, which is a point in the segment described by the "knots" argument that is used to determine the orientation of the inscribed rectangle. In this example, fixed point 290 and orientation-point 290 are the same point in order to illustrate the boundedness property of this reshape method; they need not be.

As implemented, a rectangle 293 is created to enclose all points of line 292, with a vector 296 through fixed point 290 parallel to the tangent of curve 292 at orientation point 290. In order to simplify processing, the coordinate data given in position slot 406 of each point 164 in knots may be rotated about point 290 through an angle that makes the orientation vector 296 be horizontal (parallel to the x-axis), and then translated so that point 290 lies at the origin (0, 0). Once this is accomplished, horizontal and vertical scale factors may be derived from the offset given, and applied to each point in knots. Finally, the translation and rotation as described previously may be undone to complete the process.

A process for restoring angles specified in angle-dependencies, illustrated in Fig. 40, is a constraint satisfaction means for altering the coordinate data of a line such that the tangent at one end of the line is changed while the tangent at the other end remains fixed. In Fig. 40, line 270 depicts a segment before angle restoration and line 271 shows the same segment
after angle restoration. A situation in which angle-
dependencies might be used to advantage is illustrated
in Fig. 13. In this example, lines 275 and 278 are to
be sewn together in a finished garment, after which
lines 274 and 277 are intended to form a continuous
seam. The use of angle dependencies 273 and 267 ensures
that a "corner" is not formed at the point where lines
274 and 277 meet.

Referring to Fig. 40, angle restoration may be
accomplished by using a cubic formula 272 of the form \( y = ax^3 + bx^2 + cx + d \), whose slope at the origin is equal
to the desired angle change and whose slope at a point a
distance from the origin along the x-axis equal to the
length of the line is 0, to control the motion of points
along the line. The cubic function \( y = x^3 - 2x^2 + x \) has
a slope of 1 at \( (0, 0) \) and 0 at \( (1, 0) \), and may be
scaled to accomplish the same effect; this is more
efficient than calculating coefficients. This function
is scaled in the x-dimension by the length of the line,
and in the y-dimension by the tangent of the angle
between the current direction and desired direction of
the line at the specified point. The y-values derived
from this formula are then used to shift points along
the line in a direction perpendicular to the slope of
the line at the point being shifted.

OUTPUT PREPARATION 218

In the preferred embodiment, the purpose of output
preparation 218 is to prepare pattern parts for eventual
output to an output device, such as a plotter, marker
making system, or other equipment related to producing
garments or garment patterns.
CLAIMS:

1. A computerized system related to preparing garment pattern data to enable subsequent computerized prealteration based on standard or individual body measurements, the garment pattern data comprising points and lines, the system comprising:
   pattern storage means for storing garment pattern data comprising points and lines depicting one or more garments;
   garment knowledge means for storing generic garment knowledge for each of a plurality of garment styles, comprising:
      feature description means for storing feature descriptions comprising descriptions of landmarks and other garment features that are expected to be found in garments representative of a garment style; and
      generic constraint means for storing generic constraints comprising at least one of algorithms and declarative structures for relating body measurements to changes in one or more of the landmarks and other garment features found in garments representative of the garment style; and
   processor means for preparing garment pattern data for modification by relating the generic garment knowledge to the garment pattern data.

2. The system of claim 1 wherein the processor means comprises constraint satisfaction means for producing prealtered pattern data by satisfying the generic constraints through altering points and lines in the garment pattern data.
3. The system of claim 1 wherein the processor means comprises:
   classification means for selecting at least one garment style within the generic garment knowledge that applies to a garment in the garment pattern data; and
   feature identification means for identifying one or more points or lines in the garment pattern data that correspond to a feature description within the selected generic garment knowledge.

4. The system of claim 3 wherein the processor means further comprises constraint generation means for generating prealteration constraints that relate directly to the lines and points in the garment pattern data by substituting the lines and points identified by the feature identification means for the descriptions of landmarks and other garment features referred to in the generic constraints of the selected garment knowledge.

5. The system of claim 3 wherein the feature identification means comprises:
   means for determining expected landmarks and other garment features for a garment in the garment pattern data from the one or more selected garment styles for the garment; and
   means for identifying one or more points or lines in the garment pattern data that correspond to the expected landmarks and other garment features associated with the garment.

6. The system of claim 3 wherein the generic constraint means comprises means for storing one or more measurement descriptions for a garment style, a measurement description comprising a description of a physical dimension in terms of the expected
landmarks and other garment features of the garment style.

7. The system of claim 6 wherein the processor means comprises:
   means for associating a measurement description with a garment in the garment pattern data; and
   measurement derivation means for deriving a value for a physical dimension of the garment based on the measurement description.

8. The system of claim 6 wherein:
   the generic constraint means comprises means for storing generic couplings which describe how changes to the physical dimensions of a garment within a particular garment style are applied to the expected landmarks and other garment features of that garment style; and
   the processor means comprises means for selecting a generic coupling for a garment in the garment pattern data from the one or more selected garment styles for that garment.

9. The system of claim 6 wherein:
   the generic constraint means comprises means for storing generic adjustment descriptions which describe how the body measurements relate to changes in the physical dimensions of a garment within a particular garment style; and
   the processor means comprises means for determining a generic adjustment description for the garment from the one or more selected garment styles for the garment.

10. The system of claim 1 wherein the processor means comprises means for producing prepared pattern data, the prepared pattern data comprising one or more
garment descriptions, a garment description comprising one or more geometric constraints that specify limits on relationships among the points and lines that depict the described garment.

11. A computerized data storage structure for storing garment pattern data in machine-readable form for use in connection with a computerized system for manipulating the garment pattern data, comprising:

coordinate data means for storing points and lines depicting parts of a garment; and

garment description means for storing a description of the garment, the garment description means comprising:

geometric constraint means for storing constraint descriptions that specify limits on relationships among the points and lines that depict the garment; and

pattern measurement means for storing one or more measurement constraints that map the physical dimensions of the garment onto the points and lines and specify relationships between the physical dimensions of the garment and standard or individual body measurements.

12. The data storage structure of claim 1 wherein the pattern measurement means comprises:

means for storing total ease values that relate to the amount by which a physical garment dimension should differ from the body measurements;

means for storing fit ease values related to the amount by which a physical garment dimension should differ from the body measurements in order to allow for comfort and wearability; and

means for storing design ease values related to the amount by which the total ease differs from the fit ease for a particular physical garment.
dimension.

13. The data storage structure of claim 11 wherein the pattern measurement means comprises means for storing adjustment descriptions which describe how changes in the physical dimensions of the garment are derived from the body measurements.

14. The data storage structure of claim 11 wherein the pattern measurement means comprises means for storing adjustment couplings which describe how changes in the physical dimensions of a garment are translated into changes to the points and lines depicting the parts of the garment.

15. The data storage structure of claim 11 wherein the geometric constraint means comprises means for storing constraints that specify the manner in which a line may be reshaped.

16. A computerized system for prealtering garment pattern data based on standard or individual body measurements, comprising:

   body measurement acceptance means for accepting standard or individual body measurements;

   pattern acceptance means for accepting garment pattern data, the pattern acceptance means comprising:

   coordinate data means for accepting coordinate data comprising points and lines depicting a garment; and

   garment description means for accepting a description of the garment, the garment description means comprising:

   geometric constraint means for accepting geometric constraints that specify limits on relationships among the
points and lines that depict the garment; and
garment measurement means for accepting one or more measurement
constraints that map the physical
dimensions of the garment onto the points
and lines and specify relationships
between the physical dimensions of the
garment and the standard or individual
body measurements; and
constraint satisfaction means for producing
prealtered pattern data by satisfying the geometric
and measurement constraints through altering points
and lines in the garment pattern data;

17. The system of claim 16 wherein the constraint
satisfaction means further comprises measurement
completion means for automatically deriving
estimates for body measurements that are expected by
the measurement constraints but are missing from a
particular set of standard or individual body
measurements.

18. The system of claim 16 wherein:
the garment description means comprises means
for accepting grade descriptions that describe how
physical dimensions of garments vary from one size
to the next; and
the constraint satisfaction means comprises
measurement modification means for producing graded
pattern data by modifying the physical dimensions of
a garment through altering points and lines in the
garment pattern data.

19. The system of claim 16 wherein the constraint
satisfaction means comprises:
adjustment means for determining an adjustment
that specifies how a physical dimension of a garment is to be changed based on the body measurements;

coupling means for applying the adjustment to the garment pattern data in order to accomplish the change in physical dimension by altering the points and lines depicting the garment; and

redrawing means for restoring continuity to lines on which individual points have been moved as a result of applying the coupling means.

20. The system of claim 19 wherein:

the garment description means comprises reshape description means for accepting a reshape description that specifies a desired shape characteristic of a line;

the redrawing means comprises means for restoring continuity to a line, after a point on the line has been moved, by altering the coordinate data of the line in accordance with a reshape description.
**Fig. 1B**

- COMPUTER
- PROCESSOR
- PATTERN STORAGE
- GARMENT KNOWLEDGE
- MEASUREMENT STORAGE

**Fig. 2**

- OBJECT POINTERS
- DATA
- METHOD
- DATA
- ACTIVE VALUE METHOD
- DATA
- ACTIVE VALUE METHOD
Fig. 4
Fig. 5
Fig. 6
Fig. 7
Fig. 8
Fig. 9
Fig. 10
Fig. 14
Fig. 15

© 1988 3M COMPANY
Fig. 16

Fig. 17
ELECTRONIC PATTERN DATA

CREATE KNOWLEDGE BASE

CREATE PROJECT INSTANCE

READ NEXT BLOCK OF DATA

GARMENT TYPE OF DATA BLOCK

CREATE GARMENT INSTANCE

CREATE PART INSTANCE

MORE LINES?

CREATE LINE INSTANCE

CREATE POINT INSTANCES

MORE ALIGNMENT GUIDES?

CREATE ALIGNMENT GUIDE INSTANCE

CREATE POINT INSTANCE

MORE DATA?

INITIAL PATTERN KNOWLEDGE BASE

© 1988 3M COMPANY

Fig. 31
Fig. 33
Fig. 37
# INTERNATIONAL SEARCH REPORT

**International Application No:** PCT/US 89/01063

## I. CLASSIFICATION OF SUBJECT MATTER

According to International Patent Classification (IPC) or to both National Classification and IPC:

<table>
<thead>
<tr>
<th>Int.Cl.</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A41H3/00</td>
<td></td>
</tr>
</tbody>
</table>

## II. FIELDS SEARCHED

<table>
<thead>
<tr>
<th>Classification System</th>
<th>Classification Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int.Cl. 4</td>
<td>A41H</td>
</tr>
</tbody>
</table>

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched

## III. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of Document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to Claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US, A 4149246 (R. N. GOLDMAN) 10 April 1979 see column 2, line 25 - column 14, line 17; figures 1-4; abstract <em>(cited in the application)</em></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>EP, A 0137612 (THE RICHMAN BROTHERS COMPANY) 17 April 1985 see page 26, line 8 - page 30, line 17; claims 1-10; figures 1-8 &amp; US, A 4598376 <em>(cited in the application)</em></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>US, A 3391392 (H. W. DOYLE) 02 July 1968 <em>(cited in the application)</em></td>
<td></td>
</tr>
</tbody>
</table>

## IV. CERTIFICATION

<table>
<thead>
<tr>
<th>Date of the Actual Completion of the International Search</th>
<th>Date of Mailing of this International Search Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>06 JULY 1989</td>
<td>02 AUG 1989</td>
</tr>
</tbody>
</table>

International Searching Authority: EUROPEAN PATENT OFFICE

Signature of Authorized Officer: GARNIER F. M. A. C.

Form: PCT/ISA/15 (second sheet) (January 1985)
ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO. PCT/US 89/01063
SA 27418

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 06/07/89

<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EP-A- 0227642</td>
<td>01-07-87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US-A- 4598376</td>
<td>01-07-86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA-A- 1224535</td>
<td>21-07-87</td>
</tr>
<tr>
<td>US-A-3391392</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NL-A- 6614635</td>
<td>19-04-67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB-A- 1165223</td>
<td>24-09-69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE-A- 1524345</td>
<td>20-08-70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FR-A- 1490284</td>
<td></td>
</tr>
</tbody>
</table>

For more details about this annex: see Official Journal of the European Patent Office, No. 12/82.