EXHIBIT 3.05

gesture interface for your laptop. Bored at your board meetings? The iGesture Pad is a perfectly silent mouse for stealthy web surfing.

Plugs and plays with Macs, Windows, and Linux.

Home

Products | Forums | Site Map | Resellers | Contact

Sarah A. Douglas and Anant Kartik Mithal

The Ergonomics of Computer Pointing Devices

Springer

Sarah A. Douglas, AB, MS, PhD
Anant Kartik Mithal, BTech, MS, PhD
Department of Computer and Information Science
College of Arts and Sciences
1202 University of Oregon, Eugene, OR 97403-1202, USA

Series Editors
Professor Peter J. Thomas, BA (Hons), PhD, AIMgt, FRSA, FVRS
Centre for Personal Information Management, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, UK
Professor Ray J. Paul, BSc, MSc, PhD
Department of Computer Science and Information Systems at St. John's, Brunel University, Kingston Lane, Uxbridge, Middlesex UB8 3PH, UK

ISBN 3-540-19986-1 Springer-Verlag Berlin Heidelberg New York

British Library Cataloguing in Publication Data
Douglas, Sarah Ann
The ergonomics of computer pointing devices. - (Advanced perspectives in applied computing)
1. Computer input-output equipment - Design and construction
2. Human-computer interaction
I.Title II.Mithal, Anant Kartik
621.3'986
ISBN 3540199861

Library of Congress Cataloging-in-Publication Data
Douglas, Sarah A., 1949-
The ergonomics of computer pointing devices / Sarah A. Douglas and Anant Kartik Mithal.
p. cm. -- (Advanced perspectives in applied computing)
Includes bibliographical references and index.
ISBN 3-540-19986-1 (pbk. : alk. paper)
1. Computer input-output equipment--Design. 2. Human engineering. 3. Mice (Computers)--Design. I. Mithal, Anant Kartik, 1960- . II. Title. III. Series.
TK7887.5.D665 1997
621.39'86--dc21
96-49162

The use of registered names, trademarks etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Contents

Preface

We first began looking at pointing devices and human performanc
1990 when the senior author, Sarah Douglas, was asked to evaluate
human performance of a rather novel device: a finger-controlled isom
joystick placed under a key on the keyboard. Since 1990 we have l
involved in the development and evaluation of other isometric joystic
foot-controlled mouse, a trackball, and a wearable computer with h
mounted display. We unabashedly believe that design and evaluatio
pointing devices should evolve from a broad spectrum of values w
place the human being at the center. These values include performance
ues such as pointing-time and errors, physical issues such as comfort
health, and contextual issues such as task usability and user acceptance

This book chronicles this six-year history of our relationship as tea
(Douglas) and student (Mithal), as we moved from more traditional ev
ation using Fitts' law as the paradigm, to understanding the basic rese
literature on psychomotor behavior. During that process we became
foundly aware that many designers of pointing devices fail to unders
the constraints of human performance, and often do not even cons
experimental evaluation critical to usability decisions before marketi
device. We also became aware of the fact that, contrary to popular beli
the human–computer interaction community, the problem of predic
pointing device performance has not been solved by Fitts' law. Simil
our expectations were biased by the cognitive revolution of the pas
years with the belief pointing device research was 'low-level' and unir
esting.

In due consideration of our evolving process, we set the following g
for this book. Our first goal is to provide the reader with a basic b
ground on human performance research with pointing devices. We c
the psychomotor research on pointing, the ergonomics literature eval
ing specific pointing devices, and the work on developing perform
models, such as GOMS, which predict pointing time. Where it is neces
to provide an explanation of concepts or terms, we have attempted t
so, for work in this area is cross-disciplinary, and incorporates both c
puter science and psychology. Our second goal is to demonstrate the v
of empirical and experimental methods with typical users in the evalua
and design of pointing devices. While we are aware of the difficultie
empirical study, requiring the investment of time, money, and exper

5.1 Overview

A keyboard with an integrated isometric velocity-controlled joystick located under the 'J' key, called a key joystick, was compared in learning and skilled human performance with a standard mouse. To support a broad study of human performance using these two devices, data was collected via: videotape, computer-generated data collection and questionnaire on performance time, errors, and subjective experience during learning and practicing with the experimental devices. This presents a fairly comprehensive picture of the evolution of a learner into a competent experienced user and allows us to compare human performance on the key joystick with a standard mouse.

Participants performed three tasks: pointing, dragging and device/mode switching between typing and pointing. The experimental procedure tested Fitts tasks in a two-dimensional (i.e. motion in a plane) environment. Skilled key-joystick performance was comparable to that reported in the literature for other isometric velocity-control joysticks. The key joystick was significantly slower for pointing and dragging in both learning and skilled performance, and had more errors. For practiced performance the pointing task time for the key joystick was approximately 58% slower than the mouse; the dragging task time was approximately 46% slower. For the mode-switching task, the key joystick had a significantly faster homing time, but it was not enough to compensate in overall task time for the significantly longer pointing time and non-significant difference in typing time. Both devices were shown to obey Fitts' law for pointing tasks. This result suggests that Fitts' law does accurately describe motion of finger-operated, isometric velocity-controlled devices. Finally, a comparison of index-of-performance (IP) ratios within the experimental literature places the finger-controlled isometric joystick at the same performance level as a hand-operated isometric joystick. This leads to our theoretical conclusion that continuing to reduce mode-switching or homing time will have little effect on the overall performance time of this device, and that *IP* is still a better indicator of pointing device performance than mode-switching time. In other words, in order to make this device faster, it is more fruitful to concentrate on improving the pointing speed than on the mode-switching speed.

5.2 Introduction

While numerous studies have shown the mouse to be a superior pointing device when compared to other devices such as trackballs and joysticks (Card *et al.*, 1978; Epps, 1986; MacKenzie *et al.*, 1991), the research literature has described draw-

backs in the use of mice from a very
described a homing time, which was t
from one input device to another. In t
mouse and was 360ms. (See Table 4.9
significant that Card, Moran and Ne
keystroke level model (Card *et al.*, 198
pointing task incorporates a pointing
each). Note that the original empirical
second. Moving from the keyboard to
imately 40% of the total time it takes to

In general, we can think of a homin
time that encompasses the physical ac
user to change from using one input d
combination, therefore, there are two
from the keyboard to the mouse, and
the keyboard.

The key joystick was designed to red
rates an isometric, rate-controlled joy
keyboard. An isometric device is one
applied to it, so there is no discernib
joystick. Holding the 'J' key down activ
cursor around the computer screen. R
typing mode. A second drawback of the
it requires. It requires a surface to oper
approaches have addressed these issu
trackballs integrated into the keyboard.
that it eliminates the need for additic
situations where space is at a premium

Although the key joystick appears to
time through the reduction of device-sw
that is a hypothesis which can, and s
should be given to accepting it since re
ciency of the mouse as a pointing devi
1978; MacKenzie *et al.*, 1991) and may
these reasons our experiment uses the
In addition, replication of the findings
experiments develop confidence in resu
ings concerning the key joystick.

5.3 Previous Research

While there is a broad range of studies e
Fitts' law to pointing devices (see MacK

Toolglass and Magic Lenses: The See-Through Interface

Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton(1), Tony D. DeRose(2)
Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304
(1) of Toronto, (2)University of Washington

Abstract

Toolglass(TM) *widgets* are new user interface tools that can appear, as though on a transparent sheet of glass, between an application and a traditional cursor. They can be positioned with one hand while the other positions the cursor. The widgets provide a rich and concise vocabulary for operating on application objects. These widgets may incorporate visual filters, called *Magic Lens*(TM) *filters*, that modify the presentation of application objects to reveal hidden information, to enhance data of interest, or to suppress distracting information. Together, these tools form a *see-through interface* that offers many advantages over traditional controls. They provide a new style of interaction that better exploits the user's everyday skills. They can reduce steps, cursor motion, and errors. Many widgets can be provided in a user interface, by designers and by users, without requiring dedicated screen space. In addition, lenses provide rich context-dependent feedback and the ability to view details and context simultaneously. Our widgets and lenses can be combined to form operation and viewing macros, and can be used over multiple applications.

CR Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques-interaction techniques; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces-interaction styles; I.3.3 [**Computer Graphics**]: Picture/Image Generation-viewing algorithms; I.3.4 [**Computer Graphics**]: Graphics Utilities-graphics editors
Key Words: multi-hand, button, lens, viewing filter, control panel, menu, transparent, macro

1. Introduction

We introduce a new style of graphical user interface, called the *see-through interface*. The see-through interface includes semi-transparent interactive tools, called *Toolglass*(TM) *widgets*, that are used in an application work area. They appear on a virtual sheet of transparent glass, called a *Toolglass sheet*, between the application and a traditional cursor. These widgets may provide a customized view of the application underneath them, using viewing filters called *Magic Lens*(TM) filters. Each lens is a screen region together with an operator, such as "magnification" or "render in wireframe," performed on objects viewed in the region. The user positions a Toolglass sheet over desired objects and then points through the widgets and lenses. These tools create *spatial modes* that can replace temporal modes in user interface systems.

Two hands can be used to operate the see-through interface. The user can position the sheet with the non-dominant hand, using a device such as a trackball or touchpad, at the same time as the dominant hand positions a cursor (e.g., with a mouse or stylus). Thus, the user can line up a widget, a cursor, and an application object in a single two-handed gesture.

Published as: Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and Magic Lenses: The See Through Interface. Proceedings of Siggraph 93 (Anaheim, August), *Computer Graphics Annual Conference Series*, ACM, 1993, pages 73 80.

A set of simple widgets called *click-through buttons* is shown in figure 1. These buttons can be used to change the color of objects below them. The user positions the widget in the vicinity and indicates precisely which object to color by clicking through the button with the cursor over that object, as shown in figure 1(b). The buttons in figure 1(c) change the outline colors of objects. In addition, these buttons include a filter that shows only outlines, suppressing filled areas. This filter both reminds the user that these buttons do not affect filled areas and allows the user to change the color of outlines that were obscured.

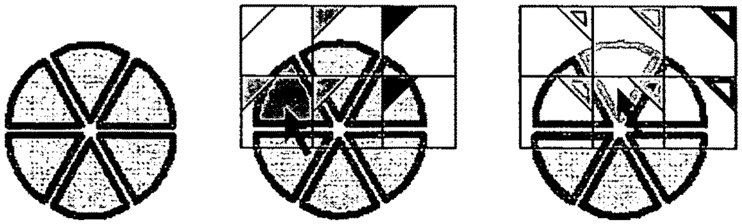


Figure 1. Click-through buttons. (a) Six wedge objects. (b) Clicking through a green fill-color button. (c) Clicking through a cyan outline-color button.

Many widgets can be placed on a single sheet, as shown in figure 2. The user can switch from one command or viewing mode to another simply by repositioning the sheet.

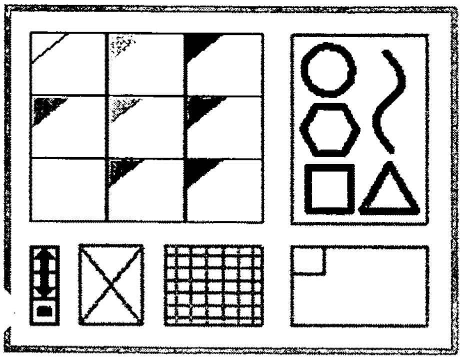


Figure 2. A sheet of widgets. Clockwise from upper left: color palette, shape palette, clipboard, grid, delete button, and buttons that navigate to additional widgets.

Widgets and lenses can be composed by overlapping them, allowing a large number of specialized tools to be created from a small basic set. Figure 3 shows an outline color palette over a magnifying lens, which makes it easy to point to individual edges.

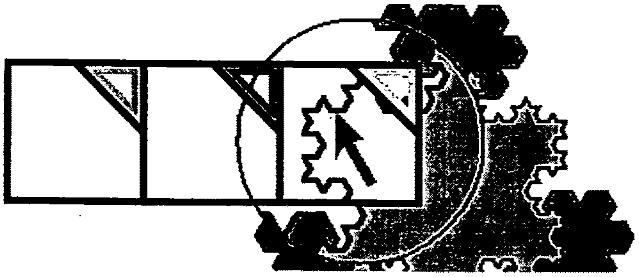


Figure 3. An outline color palette over a magnifying lens.

The see-through interface has been implemented in the Multi-Device Multi-User Multi-Editor (MMM) framework5 in ᐩhe Cedar programming language and environment,24 running on the SunOS UNIX(TM)-compatible operating system n Sun Microsystems SPARCstations and other computers. The Gargoyle graphics editor,20 as integrated into MMM, serves as a complex application on which to test our interface. We use a standard mouse for the dominant hand and a MicroSpeed FastTRAP(TM) trackball for the non-dominant hand. The trackball includes three buttons and a thumbwheel, which can be used to supply additional parameters to the interface.

The remainder of this paper is organized as follows. The next section describes related work. Section 3 describes some examples of the tools we have developed. Section 4 discusses general techniques for using the see-through interface. Section 5 discusses some advantages of this approach. Section 6 describes our implementation. Sections 7 and 8 present our conclusions and plans for future work.

Except for figures 12 and 16, all of the figures in this paper reflect current capabilities of our software.

2. Related Work

The components of the see-through interface combine work in four areas: simultaneous use of two hands, movable tools, transparent tools, and viewing filters. In this section, we describe related work in these four areas.

Multi-Handed Interfaces

Several authors have studied interfaces that interpret continuous gestures of both hands. In Krueger's VIDEOPLACEs system,15 the position and motion of both of a participant's hands, as seen by a video camera, determine the behavior of a variety of on-screen objects, including animated creatures and B-spline curves. Buxton and Myers discovered that users naturally overlap the use of both hands, when this is possible, and that, even when the two hands are used sequentially, there is still a performance advantage over single-hand use.7,8

Other work characterizes the situations under which people successfully perform two-handed tasks. Guiard presents evidence that people are well-adapted to tasks where the non-dominant hand coarsely positions a context and the dominant hand performs detailed work in that context.4 Similarly, Kabbash presents evidence that a user's non-dominant hand performs as well or better than the dominant hand on coarse positioning tasks.13

Our system takes full advantage of a user's two-handed skills; the non-dominant hand sets up a context by coarsely ositioning the sheet, and the dominant hand acts in that context, pointing precisely at objects through the sheet.

Movable Tools

Menus that pop up at the cursor position are movable tools in the work area. However, such a menu's position is determined by the cursor position before it appears, making it difficult to position it relative to application objects.

Several existing systems provide menus that can be positioned in the same work area as application objects. For example, MacDraw "tear-off menus" allow a pull-down menu to be positioned in the work area and repositioned by clicking and dragging its header.17 Unfortunately, moving these menus takes the cursor hand away from its task, and they must be moved whenever the user needs to see or manipulate objects under them.

Toolglass sheets can be positioned relative to application objects and moved without tying up the cursor.

Transparent Tools

Some existing systems that allow menus to be positioned over the work area make these menus transparent. For example, the Alto Markup system18 displays a menu of modes when a mouse button goes down. Each menu item is drawn as an icon, with the space between icons transparent. Bartlett's transparent controls for interactive graphics use stipple patterns to get the effect of transparency in X Windows.2

While these systems allow the user to continue to see the underlying application while a menu is in place, they don't llow the user to interact with the application through the menu and they don't use filters to modify the view of the .pplication, as does our interface.

Viewing Filters

Many existing window systems provide a pixel magnifier. Our Magic Lens filters generalize the lens metaphor to many representations other than pixels and to many operations other than magnification. Because they can access application-specific data structures, our lenses are able to perform qualitatively different viewing operations, including showing hidden information and showing information in a completely different format. Even when the operation is ıagnification, our lenses can produce results of superior quality, since they are not limited to processing data at screen resolution.

The concept of using a filter to change the way information is visualized in a complex system has been introduced before.25,10,14 Recent image processing systems support compostition of overlapping filters.23 However, none of these systems combine the filtered views with the metaphor of a movable viewing lens.

Other systems provide special-purpose lenses that provide more detailed views of state in complex diagrams. For example, a fisheye lens can enhance the presentation of complicated graphs.21 The bifocal display22 provides similar functionallity for viewing a large space of documents. The MasPar Profiler3 uses a tool based on the magnifying lens metaphor to generate more detail (including numerical data) from a graphical display of a program.

Magic Lens filters combine viewing filters with interaction and composition in a much broader way than do previous systems. They are useful both as a component of the see-through interface and as a general-purpose visualization paradigm, in which the lenses become an integral part of the model being viewed.

3. Examples

This section shows several tools that demonstrate features of the see-through interface. Because we have implemented primarily in the graphical editing domain, most of these tools are tailored to that application. However, the see-through interface can be used in a wide variety of other application domains.

Shape and Property Palettes

Palettes are collections of objects or properties that can be added to a scene. Figure 1 showed two widgets that apply color to shapes. Similar tools can be designed to apply other graphical properties, such as type and line styles to an illustration, shading parameters to a 3D model, or initial values to a simulation. Figure 4 illustrates a widget containing graphical shapes that can be "pushed through" from the tool into the illustration below. In figure 4(a), the user has positioned a shape palette widget (shown in cyan) over an illustration (shown in magenta). When the user clicks on a shape on the tool, a copy of that shape is added to the illustration. The widget attaches the copied shape to the cursor for interactive dragging until the final shape position is achieved (figure 4(b)).

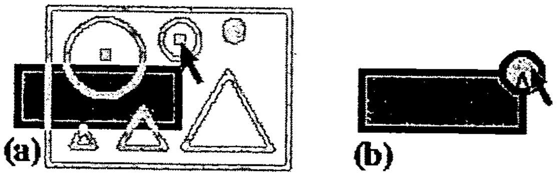
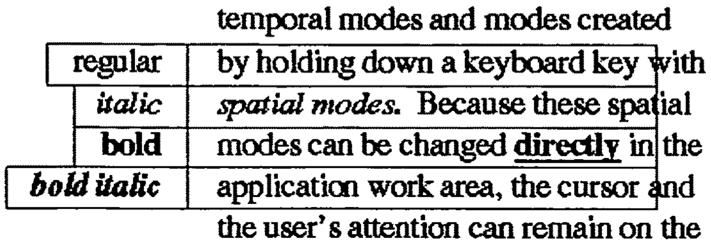


Figure 4. Shape palette. (a) Choosing a shape. (b) Placing the shape.

Figure 5 shows a design for a property palette for setting the face of text in a document. Each face (regular, bold, etc.) has an active region on the right side of the tool. Selecting the text displayed in this region changes its face.



temporal modes and modes created by holding down a keyboard key with *spatial modes.* Because these spatial modes can be changed **directly** in the application work area, the cursor and the user's attention can remain on the

Figure 5. Font face palette. The word "directly" is being selected and changed to bold face.

Clipboards

Clipboard widgets pick up shapes and properties from underlying objects, acting as visible instantiations of the copy and paste keys common in many applications. Clipboards can pick up entire objects or specific properties such as color, dash pattern or font. They can hold single or multiple copies of an object. The objects or properties captured on the clipboard can be copied from the clipboard by clicking on them, as in the palette tools.

Figure 6 shows a symmetry clipboard that picks up the shape that the user clicks on (figure 6(a)) and produces all of the rotations of that shape by multiples of 90 degrees (figure 6(b)). Moving the clipboard and clicking on it again, the user drops a translated copy of the resulting symmetrical shape (figure 6(c)). Clicking the small square in the upper left corner of the widget clears the widget so that new shapes can be clipped.

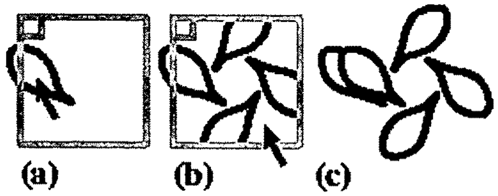


Figure 6. Symmetry clipboard. (a) Picking up an object. (b) Rotated copies appear. (c) The copies are moved and pasted.

Figure 7 shows an example of a type of clipboard that we call a *rubbing*. It picks up the fill color of an object when the user clicks on that object through the widget (figure 7(a)). The widget also picks up the shape of the object as a reminder of where the color came from (figure 7(b)). Many fill-color rubbings can be placed on a single sheet, allowing the user to store several colors and remember where they came from. The stored color is applied to new shapes when the user clicks on the applicator nib of the rubbing (figure 7(c)).

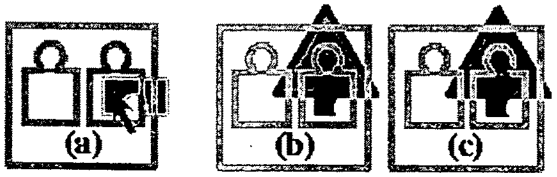


Figure 7. Fill-color rubbings. (a) Lifting a color. (b) Moving the clipboard. (c) Applying the color.

Besides implementing graphical cut and paste, clipboards provide a general mechanism for building customized libraries of shapes and properties.

Previewing Lenses

In graphical editing, a lens can be used to modify the visual properties of any graphical object, to provide a preview of what changing the property would look like. Properties include color, line thickness, dash patterns, typeface, arrowheads and drop shadows. A previewing lens can also be used to see what an illustration would look like under different circumstances; for example, showing a color illustration as it would be rendered on a black/white display or on a particular printer. Figure 8 shows a Celtic knotwork viewed through two lenses, one that adds drop shadows and ne that shows the picture in black and white. The achromatic lens reveals that the drop shadows may be difficult to distinguish from the figure on a black/white display.

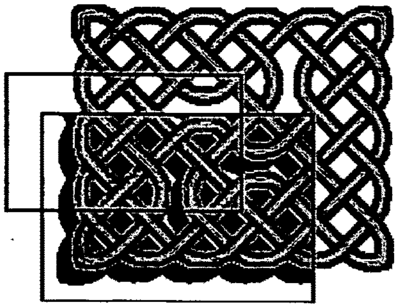


Figure 8. An achromatic lens over a drop shadow lens over a knotwork. (Knotwork by Andrew Glassner)

Previewing lenses can be parameterized. For example, the drop shadow lens has parameters to control the color and displacement of the shadow. These parameters can be included as graphical controls on the sheet near the lens, attached to input devices such as the thumbwheel, or set using other widgets.

Selection Tools

Selection is difficult in graphical editing when objects overlap or share a common edge. Our selection widgets address .iis problem by modifying the view and the interpretation of input actions. For example, figure 9 shows a widget that makes it easy to select a shape vertex even when it is obscured by other shapes. This tool contains a wire-frame lens that reveals all vertices by making shape interiors transparent. Mouse events are modified to snap to the nearest vertex.

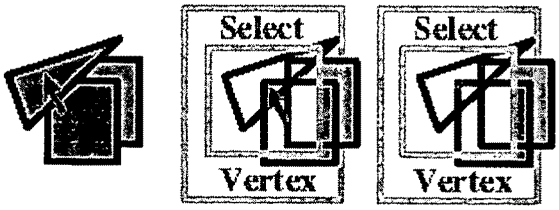


Figure 9. Vertex selection widget. (a) Shapes. (b) The widget is placed. (c) A selected vertex.

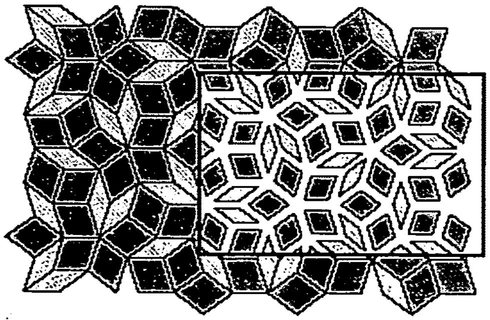


Figure 10. The local scaling lens. (Tiling by Doug Wyatt)

Figure 10 shows a lens that shrinks each object around its own centroid. This lens makes it easy to select an edge that is coincident with one or more other edges.

Grids

Figure 11 shows three widgets, each of which displays a different kind of grid. The leftmost two grids are rectangular with different spacings. The rightmost grid is hexagonal. Although each grid only appears when the lens is in place, the coordinates of the grid are bound to the scene, so that grid points do not move when the sheet moves. By clicking on the grid points and moving the widget, the user can draw precise shapes larger than the widget. If the sheet is moved by the non-dominant hand, the user can quickly switch between the grids during an editing motion.

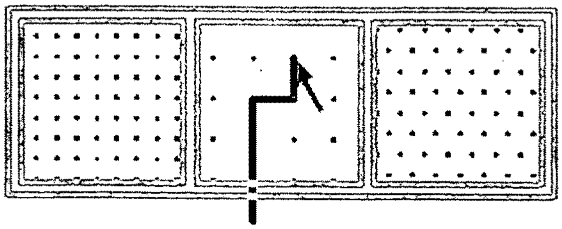


Figure 11. Three grid tools.

Visualization

Figure 12 illustrates the use of tools and lenses to measure Gaussian curvature in the context of a shaded rendering of a 3D model. The pseudo-color view indicates the sign and relative magnitude of the curvature,9 and the evaluation tool displays the value at the point indicated.

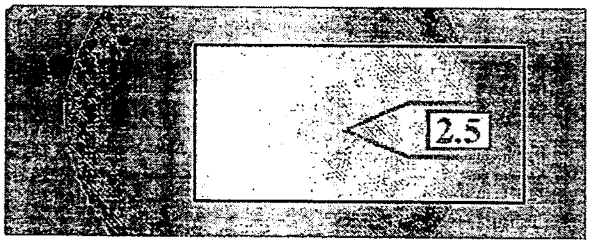


Figure 12. Gaussian curvature pseudo-color lens with overlaid tool to read the numeric value of the curvature. (Original images courtesy of Steve Mann)

4. Using the See-Through Interface

Widgets and lenses are most effective when supported by appropriate conventions specifying how to position, size, organize, and customize them. This section discusses a few of these issues.

Moving and Sizing the Sheet or the Application

A Toolglass sheet can be moved by clicking and dragging on its border with a mouse or by rolling the trackball. The sheet and all its widgets can stretch and shrink as a unit when the user works a a second controller such as a thumbwheel. With these moving and sizing controls, the user can center a widget on any application object and size the widget to cover any screen region. Large widgets can be used to minimize sheet motion when applying a widget to several objects. A widget that has been stretched to cover the entire work area effectively creates a command mode over the entire application.

By clicking a button on the trackball, the user can disconnect the trackball from the sheet and enable its use for scrolling and zooming a selected application area. If a sheet is over this application, the user can now move an application object to a widget instead of moving a widget to an object. This is a convenient way to use the see-through interface on illustrations that are too large to fit on the screen.

Managing Sheets

A typical application will have a large number of widgets in its interface. To avoid clutter, we need a way to organize these widgets and sheets. One approach is to put all of the widgets on a single sheet that can be navigated by scrolling and zooming. Perlin and Fox's paper in these proceedings19 describes techniques for creating and navigating unlimited structures on a single sheet. A second approach is to have a master sheet that generates other sheets. Each of these sheets could generate more sheets, like hierarchical menus. A third technique, used in our prototype, is to allow a single sheet to show different sets of widgets at different times. The set to display can be selected in several ways: the user can click a special widget in the set, like the arrows in HyperCard,(TM)11 that jumps to another set. In addition, a master view provides a table of contents of the available sets allowing the user to jump to any one. To use different sets simultaneously, the user creates additional sheets.

Customizing Sheets

Because sheets can contain an unlimited number of widgets, they provide a valuable new substrate on which users can create their own customized widgets and widget sets. In effect, the sheets can provide a user interface *editor*, allowing users to move and copy existing widgets, compose macros by overlapping widgets, and snap widgets together in new configurations. Indeed, with the techniques described in this paper, one Toolglass sheet could even be used to edit another.

5. Advantages of See-Through Tools

In this section, we describe some advantages we see for using the see-through interface. Most of these advantages result from placing tools on overlapping layers and from the graphical nature of the interface.

In most applications, a control panel competes for screen space with the work area of the application. Toolglass sheets exist on a layer above the work area. With proper management of the sheets, they can provide an unlimited space for tools. The widgets in use can take up the entire work area. Then, they can be scrolled entirely off the screen to provide an unobstructed view of the application or space for a different set of widgets.

The see-through user interface can be used on tiny displays, such as notebook computers or personal digital assistants, that have little screen real estate for fixed-position control panels. It can also be used on wall-sized displays, where a fixed control panel might be physically out of reach from some screen positions. These tools can move with the user to stay close at hand.

A user interface layer over the desktop provides a natural place to locate application-independent tools, such as a clipboard that can copy material from one window to another.

These widgets can combine multiple task steps into a single step. For example, the vertex selection widget of figure 9 allows the user to turn on a viewing mode (wire-frame), turn on a command mode (selection), and point to an object in a single two-handed gesture.

Most user interfaces have temporal modes that can cause the same action to have different effects at different times. With our interface, modes are defined spatially by placing a widget and the cursor over the object to be operated on. hus, the user can easily see what the current mode is (e.g., by the label on the widget) and how to get out of it (e.g., move the cursor out of the widget). In addition, each widget can provide customized feedback for its operation. For example, a widget that edits text in an illustration can include a lens that filters out all the objects except text. When several widgets are visible at once, the feedback in each one serves a dual role. It helps the user make proper use of the widget and it helps the user choose the correct widget.

The visual nature of the see-through interface also allows users to construct personalized collections of widgets as described above.

6. Implementation

This section provides an overview of our implementation of the see-through interface.

Toolglass Sheets

We describe three Toolglass subsystems: one that handles simultaneous input from two pointing devices and updates the screen after multiple simultaneous changes, one that modifies pointing events as they pass through widgets, and one that modifies graphical output as it passes up through each widget.

Multi-Device Input and Screen Refresh

Our Toolglass software uses the MMM framework.5 The see-through interface relies on the following features of MMM.

MMM takes events from multiple input devices, such as the mouse and trackball, keeps track of which device produced which event, and places all events on a single queue. It dequeues each event in order and determines to which application that event should be delivered. MMM applications are arranged in a hierarchy that indicates how they are nested on the screen. Each event is passed to the root application, which may pass the event on to one of its child applications, which may in turn pass the event on down the tree. Mouse events are generally delivered to the most deeply nested application whose screen region contains the mouse coordinates. However, when the user is dragging or ɹubberbanding an object in a particular application, all mouse coordinates go to that application until the dragging or rubberbanding is completed. Keyboard events go to the currently selected application.

To support Toolglass sheets, MMM's rules for handling trackball input were modified. When a sheet is movable, trackball and thumbwheel events go to the top-level application, which interprets them as commands to move or resize the sheet, respectively. When the sheet is not movable, the trackball and thumbwheel events are delivered to the selected application, which interprets them as commands to scroll or zoom that application.

Filtering Input Through Lenses and Widgets

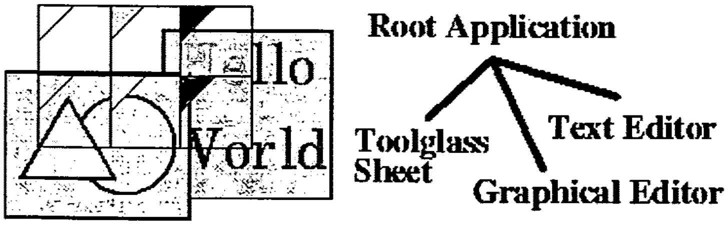


Figure 13. A simple hierarchy of applications

Ordinarily, MMM input events move strictly from the root application towards the leaf applications. However, to support the see-through interface, input events must be passed back up this tree. For example, figure 13(b) shows an application hierarchy. The left-to-right order at the lower level of this tree indicates the top-to-bottom order of ɪplications on the screen. Input events are first delivered to the Toolglass sheet to determine if the user is interacting with a widget or lens. If so, the event is modified by the sheet. In any case, the event is returned to the root application, which either accepts the event itself or passes it on to the child applications that appear farther to the right in the tree.

The data structure that represents an MMM event is modified in three ways to support Toolglass sheets. First, an event is annotated with a representation of the parts of the application tree it has already visited. In figure 13, this prevents the root application from delivering the event to the sheet more than once. Second, an event is tagged with a command string to be interpreted when it reaches its final application. For example, a color palette click-through button annotates ach mouse-click event with the command name "FillColor" followed by a color. Finally, if the widget contains a lens, the mouse coordinates of an event may be modified so the event will be correctly directed to the object that appears under the cursor through that lens.

Figure 14. Composing color-changing widgets.

Widgets can be composed by overlapping them. When a stack of overlapped widgets receives input (e.g., a mouse click), the input event is passed top-to-bottom through the widgets. Each widget in turn modifies the command string that has been assembled so far. For example, a widget might concatenate an additional command onto the current command string. In figure 14, a widget that changes fill colors (figure 14(a)) is composed with a widget that changes line colors (figure 14(b)) to form a widget that changes both fill and line colors (figure 14(c)). If the line color widget is on top, then the command string would be "LineColor blue" after passing through this widget, and "LineColor blue; FillColor cyan" after both widgets.

Filtering Output Through Lenses and Widgets

Ordinarily, MMM output is composed from the leaf applications up. To support lenses, the normal screen refresh composition has been extended to allow information to flow down and across the tree as well as up. For example, if the widgets in figure 13 contain one or more lenses, and if any of those lenses is situated over the graphical editor, each lens must examine the contents of the graphical editor (which is the lens's sibling in the hierarchy) in order to draw itself.

In addition, to improve performance, MMM applications compute the rectangular bounding box of the regions that have recently changed, and propagate this box to the root application, which determines which screen pixels will need to be updated. Generally, this bounding box is passed up the tree, transformed along the way by the coordinate transformation between each application and the next one up the tree. However, lenses can modify the set of pixels that an operation affects. A magnifying lens, for example, generally increases the number of pixels affected. As a result, the bounding box must be passed to all lenses that affect it to determine the final bounding box.

Magic Lens Filters

A Magic Lens filter modifies the image displayed on a region of the screen, called the *viewing region*, by applying a *viewing filter* to objects in a model. The *input region* for the lens is defined by the viewing region and the viewing filter. It may be the same size as the viewing region, or different, as in the magnification lens. For a 3D model, the input region is a cone-shaped volume defined by the eye point and the viewing region. Input regions can be used to cull away all model objects except those needed to produce the lens image. Our current implementations do not perform this culling; as described below, there are advantages to lenses that operate on the entire model.

When several lenses are composed, the effect is as though the model were passed sequentially through the stack of lenses from bottom to top, with each lens operating on the model in turn. In addition, when one lens has other lenses below it, it may modify how the boundaries of these other lenses are mapped onto the screen within its own boundary. The input region of a group of lenses taken as a whole can be computed by applying the inverses of the viewing filters to the lens boundaries themselves.

Our lenses depend on the implementation of Toolglass sheets to manage the size, shape and motion of their viewing regions. This section describes two strategies we have tried for implementing viewing filters: a procedural method that we call *recursive ambush*, and a declarative method that we call *model-in model-out*. We also describe a third method that promises to be convenient when applicable, called *reparameterize-and-clip*. Finally, we discuss issues that arise in the presence of multiple model types.

Recursive Ambush

In the recursive ambush method, the original model is described procedurally as a set of calls in a graphics language such as Interpress(TM)12 or PostScript.(R)1 The lens is a new interpreter for the graphics language, with a different implementation for each graphics primitive. In most cases, the implementation of a given graphics primitive first performs some actions that carry out the modifying effect of the lens and then calls the previous implementation of the primitive. For example, a lens that modifies a picture such that all of its lines are drawn in red would modify the "DrawLine" primitive to set the color to red and then call the original "DrawLine" primitive.

When lenses are composed, the previous implementation may not be the original graphics language primitive, but another lens primitive that performs yet another modification, making composition recursive.

Recursive ambush lenses appear to have important advantages. Because they work at the graphics language level, they work across many applications. Because they work procedurally, they need not allocate storage. However, the other methods can also work at the graphics language level. In addition, recursive ambush lenses have three major disadvantages. First, making a new lens usually requires modifying many graphics language primitives. Second, debugging several composed lenses is difficult because the effects of several cooperating interpreters are hard to understand. Finally, performance deteriorates rapidly as lenses are composed because the result of each lens is computed many times; the number of computations doubles with the addition of each lens that overlaps all of the others.

Model-In Model-Out

In the model-in model-out (MIMO) method, we make a copy of the original model as the first step. This model might e the data structure of an editor, a representation of graphics language calls, an array of pixels or some other picture representation. The implementation walks through this data structure and modifies it in accordance with the desired behavior of the lens. When composed with other lenses, a MIMO lens takes each model that is produced by each lens under it, produces a modified version of that model, and associates it with the clipping region formed by intersecting its clipping region with that of the lens underneath. The resulting models are passed on to lenses above.

Although MIMO lenses must allocate storage, this investment pays off in several ways. First, during the rendering of a single image, each lens computes its output models only once, and then saves them for use by any lenses that are over it. In addition, if the computed model is based on the entire original model, then redrawing the picture after a lens moves is just a matter of changing clipping regions; no new model filtering is needed. In this case, each lens maintains a table of the models it has produced. The table is indexed by the models it has received as input and when they were last modified. The action of such a lens often consists of a single table lookup.

MIMO lenses have many other advantages. Given routines to copy and visit parts of the model, the incremental effort to write a MIMO lens is small. Many of our lenses for graphical editor data structures were written in under 20 minutes and consist of under 20 lines of code. Debugging composed lenses is easy because the intermediate steps can easily be viewed. Finally, MIMO lenses can perform a large class of filtering functions because they can access the input model in any order. In particular, they can compute their output using graphical search and replace,16 as shown in figure 15 where each line segment is replaced by multiple line segments to create a "snowflake" pattern.

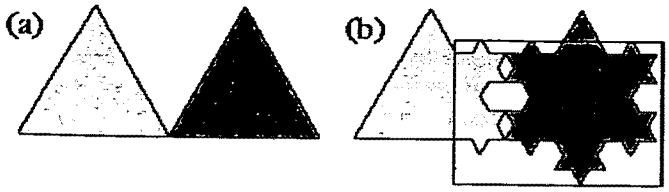


Figure 15. The snowflake lens. (a) Two triangles. (b) Snowflake lens over part of the scene.

An important variation of MIMO is to allow the output model to differ in type from the input model. For example, a lens might take a graphics language as input and produce pixels as output. In this case, the lens walks the original model, rather than copying it, and allocates data structures of the new model type.

.eparameterize and Clip

If the original image is being produced on the screen by a renderer with variable parameters, it is easy to implement lenses that show the effects of varying those parameters. To function, the lens modifies a renderer parameter and asks the renderer to redraw the model clipped to the boundary shape of the lens. For example, a lens showing the wireframe version of a 3D shaded model can be implemented this way.

Several reparameterize-and-clip lenses can be composed if the parameter changes made by these lenses are compatible. In the region of overlap, the renderer re-renders the original model after each of the overlapping lenses has made its changes to the renderer parameters. The flow of control and performance of a stack of these lenses is like that of MIMO lenses; a new output is computed for each input region received from lenses underneath. These lenses differ from MIMO in that each output is computed from the original model, and each output is always a rendering.

Multiple Model Types

In our discussion above, lenses are used to view a single type of model, such as a graphical editor data structure or a graphical language. In practice, multiple model types are often present, for two reasons. First, a lens can overlap multiple applications at the same time, where the applications have different model types, as shown above in figure 13. Second, a lens may overlap both an application and a lens, where the lens output and application model are of different types. For example, in figure 16, the wireframe lens converts from a 3D model to a 2D line drawing. The magnifier lens, which operates on 2D drawings, overlaps both the original image and the output of the wireframe lens. Rich illustrations can be produced by permitting lenses to overlap multiple model types in this way.

upporting multiple model types requires *type conversion* and *type tolerance*. When a lens that expects one type of model as input is moved over a model of a different type, the system may automatically convert the model to be of the type required; this is type conversion. For example, all of our applications produce Interpress graphics language calls as part of drawing themselves on the screen. When a lens that takes Interpress as input is positioned over one of these applications, that application converts its model to Interpress on demand for that lens.

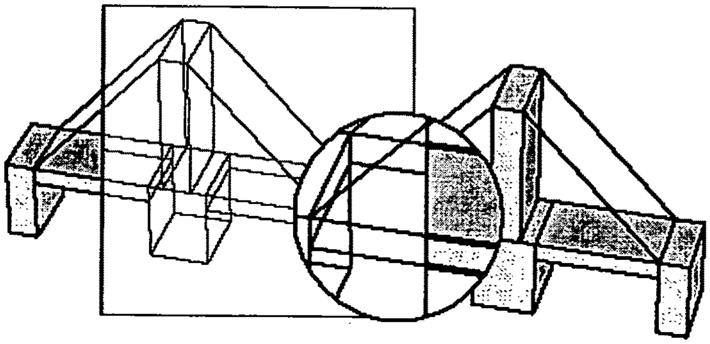


Figure 16. A bridge made of shaded, 3D blocks showing a 3D wireframe lens and a 2D magnifier.

Alternatively, when presented with a model it does not understand, a lens can simply pass that model through unchanged; this is type tolerance. For example, a lens that operates only on a graphics editor's data structures will only modify the image in the part of that lens's boundary that overlaps the graphics editor; other regions are unchanged.

.omposing Widgets and Lenses

When a widget and a lens are composed, their functions combine. For example, consider a click-through button on top of a magnifying lens. Mouse events pass through the button, are annotated with a command, and then pass through the lens, which applies the inverse of its transformation to the mouse coordinates. During screen refresh, the widget adds its appearance to the output of the lens. If the lens is on top of the widget, input events are first transformed by the lens and then tested to see if they fall within the button or not; during refresh, the widget adds its appearance to the model, which is then acted on by the lens. A widget and lens can be very tightly coupled. For example, an editing tool could include a lens that displayed control points or editing handles implemented as widgets.

Performance

Our sheets and lenses are already fast enough to be useful on current hardware, but need to be faster for smooth motion. For example, using our prototype on a SPARCstation 10, we measured the time it takes to redraw the screen after moving a wireframe lens of size 70 by 70 pixels over the Penrose tiling of figure 10, containing 117 filled and outlined shapes. For the MIMO implementation of the lens, once it has cached its output scene, it takes an average of 300 milliseconds to repaint the scene, of which 120 milliseconds are spent drawing the lens interior. The same lens implemented using recursive ambush takes %15 longer to redraw the lens interior, which we attribute to the procedure call overhead of the recursive approach. Computing the filtered scene for the MIMO lens takes an average of 480 milliseconds for this example. This computation is performed whenever the illustration under the lens is changed or lens parameters are modified.

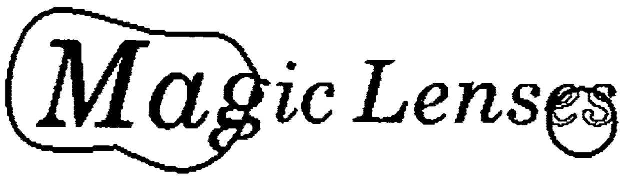


Figure 17. The Magic Lenses logo.

7. Conclusions

We have described a new style of user interface, the see-through interface, based on Toolglass widgets and Magic Lens filters. The see-through interface offers a new design space for user interfaces based on spatial rather than temporal modes and provides a natural medium for two-handed interaction. Because the interface is movable and overlays the application area, it takes no permanent screen space and can be conveniently adapted to a wide range of display sizes. Because the overlaid tools are selected and brought to the work area simply by moving the Toolglass sheet, the user's attention can remain focused on the work area. Because the operations and views are spatially defined, the user can work without changing the global context.

The see-through interface provides a new paradigm to support open software architecture. Because Toolglass sheets can be moved from one application to another, rather than being tied to a single application window, they provide an interface to the common functionality of several applications and may encourage more applications to provide common functionality. Similarly, Magic Lens filters that take standard graphics languages as input work over many applications.

In addition to their role in user interfaces, Magic Lens filters provide a new medium for computer graphics artists and a new tool for scientific visualization. When integrated into drawing tools, these filters will enable a new set of effects and will speed the production of traditional effects. Figure 17 shows a magnifying lens and a wireframe lens used to produce our Magic Lenses logo.

Integrated into scientific visualization tools, these filters can enhance understanding by providing filtered views of local regions of the data while leaving the rest of the view unchanged to provide context, as was shown in the visualization example in figure 12.

We hope the see-through interface will prove to be valuable in a wide variety of applications. While the examples in this paper stress applications in graphical editing, these tools can potentially be used in any screen-based application, including spreadsheets, text editors, multi-media editors, paint programs, solid modelers, circuit editors, scientific visualizers, or meeting support tools. Consider that most applications have some hidden state, such as the equations in a spreadsheet, the grouping of objects in a graphical editor, or the position of water pipes in an architectural model. A collection of widgets and lenses can be provided to view and edit this hidden state in a way that takes up no permanent .reen space and requires no memorization of commands.

We believe that the see-through interface will increase productivity by reducing task steps and learning time, providing good graphical feedback, and allowing users to construct their own control panels and spatial modes.

8. Plans for Future Work

The see-through interface is a framework that can be used to create many new tools in many application domains. Exploring the current space of possibilities will take many people many years. Furthermore, this design space will be enlarged by future software and hardware. We will carry out some of this exploration ourselves, creating new widgets in different application domains, working out taxonomies for the tools we discover, designing new conventions for composing, editing, navigating, organizing and triggering these tools, combining them with existing user interface techniques, and testing them on users performing real work.

We are building two Toolglass widget toolkits. The first is a traditional toolkit in which widgets are created through object-oriented programming. The second toolkit is based on our EmbeddedButtons project;6 here, users draw new widgets and collections of widgets using a graphical editor and then apply behavior to these graphical forms, where the behavior is expressed in a user customization language.

We are designing new algorithms to increase the speed of these tools. It is clear that Magic Lens filters and, to a lesser extent, Toolglass widgets provide a new way to consume the graphics power of modern computers.

inally, we are working to better understand how to model and implement general composition of widgets and lenses, specially those that work with multiple model and applications types.

Acknowledgments

We thank Blair MacIntyre for implementing our first lenses for 2D graphics and Ken Fishkin for his demonstration of lenses for text editing. We thank many of our colleagues at PARC for fruitful discussions and enthusiasm, including Stu Card, Ken Fishkin, Andrew Glassner, David Goldberg, Christian Jacobi, Jock Mackinlay, David Marimont, George Robertson, Marvin Theimer, Annie Zaenen, and Polle Zellweger, plus our consultants Randy Pausch and John Tukey. Finally, we thank Xerox Corporation for supporting this work.

Trademarks and Patents: Toolglass, Magic Lens and Interpress are trademarks of Xerox Corporation. Postscript is a trademark of Adobe Systems, Inc. UNIX is a trademark of AT&T. FastTRAP is a trademark of MicroSpeed Inc. Patents related to the concepts discussed in this paper have been applied for by Xerox Corporation.

References

1. Adobe Systems Incorporated. *PostScript*(R) *Language Reference Manual, second edition.* Addison-Wesley, 1990.

2. Bartlett, Joel F. Transparent Controls for Interactive Graphics. WRL Technical Note TN-30, Digital Equipment Corp., Palo Alto, CA. July 1992.

Beck, Kent, Becher, Jon, and Zaide, Liu. Integrating Profiling into Debugging. *Proceedings of the 1991 International Conference on Parallel Processing, Vol. II, Software,* August 1991, pp. II-284-II-285.

4. Guiard, Yves. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model.

The Journal of Motor Behavior, 19, 4, (1987), pp. 486-517.

5. Bier, Eric A. and Freeman, Steve. MMM: A User Interface Architecture for Shared Editors on a Single Screen. *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology* (Hilton Head, SC, ovember 11-13), ACM, New York, (1991), pp. 79-86.

6. Bier, Eric A., EmbeddedButtons: Supporting Buttons in Documents. *ACM Transactions on Information Systems, 10,* 4, (1992), pp. 381-407.

7. Buxton, William and Myers, Brad A.. A Study in Two-Handed Input. *Proceedings of CHI '86* (Boston, MA, April 13-17), ACM, New York, (1986), pp. 321-326.

8. Buxton, William. There's More to Interaction Than Meets the Eye: Some Issues in Manual Input. *Readings in Human-Computer Interaction: A Multidisciplinary Approach.* (Ronald M. Baecker, William A.S. Buxton, editors). Morgan Kaufmann Publishers, Inc., San Mateo, CA. 1987.

9. Dill, John. An Application of Color Graphics to the Display of Surface Curvature. Proceedings of SIGGRAPH '81 (Dallas, Texas, August 3-7). *Computer Graphics, 15,* 3, (1981), pp. 153-161.

10. Goldberg, Adele and Robson, Dave, A Metaphor for User Interface Design, *Proceedings of the University of Hawaii Twelfth Annual Symposium on System Sciences,* Honolulu, January 4-6, (1979), pp.148-157.

11. Goodman, Danny. *The Complete HyperCard Handbook.* Bantam Books, 1987.

12. Harrington, Steven J. and Buckley, Robert R.. *Interpress, The Source Book.* Simon & Schuster, Inc. New York, NY. 1988.

3. Kabbash, Paul, MacKenzie, I. Scott, and Buxton, William. Human Performance Using Computer Input Devices in ne Preferred and Non-preferred Hands. *Proceedings of InterCHI '93,* (Amsterdam, April 24-29), pp. 474-481.

14. Krasner, Glenn and Hope, Stephen, A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80, *Journal of Object-Oriented Programming, 1,* 3, (1988), pp. 26-49.

15. Krueger, Myron W., Gionfriddo, Thomas, and Hinrichsen, Katrin. VIDEOPLACE - An Artificial Reality. *Proceedings of CHI '85* (San Francisco, April 14-18). ACM, New York, (1985), pp. 35-40.

16. Kurlander, David and Bier, Eric A.. Graphical Search and Replace. Proceedings of SIGGRAPH '88 (Atlanta, Georgia, August 1-5) *Computer Graphics, 22,* 4, (1988), pp. 113-120.

17. *MacDraw Manual.* Apple Computer Inc. Cupertino, CA 95014, 1984.

18. Newman, William. *Markup User's Manual.* Alto User's Handbook, Xerox PARC technical report, (1979), pp. 85-96.

19. Perlin, Ken and Fox, David. Pad: An Alternative Approach to the Computer Interface. this proceedings.

20. Pier, Ken, Bier, Eric, and Stone, Maureen. An Introduction to Gargoyle: An Interactive Illustration Tool. *Proceedings of the Intl. Conf. on Electronic Publishing, Document Manipulation and Typography* (Nice, France, April). Cambridge Univ. Press, (1988), pp. 223-238.

?1. Sarkar, Manojit and Brown, Marc H.. Graphical Fisheye Views of Graphs. *Proceedings of CHI '92,* (Monterey, CA, lay 3-5, 1992) ACM, New York, (1992), pp. 83-91.

22. Spence, Robert and Apperley, Mark. Data Base Navigation: An Office Environment of the Professional. *Behaviour and Invormation Technology, 1,* 1, (1982), 43-54.

23. *ImageVision*, Silicon Graphics Inc., Mountain View, CA.

24. Swinehart, Daniel C., Zellweger, Polle T., Beach, Richard J., Hagmann, Robert B.. A Structural View of the Cedar Programming Environment. *ACM Transactions on Programming Languages and Systems*, *8,* 4, (1986), pp. 419-490.

25. Weyer, Stephen A. and Borning, Alan H., A Prototype Electronic Encyclopedia, *ACM Transactions on Office Systems*, *3,* 1, (1985), pp. 63-88.

EBVElektronik
I An Avnet Company I

1000-up price
TSOP6238TR
€ 0.64

TSOP6238
IR Receiver Modules for Infrared Remote Control Systems

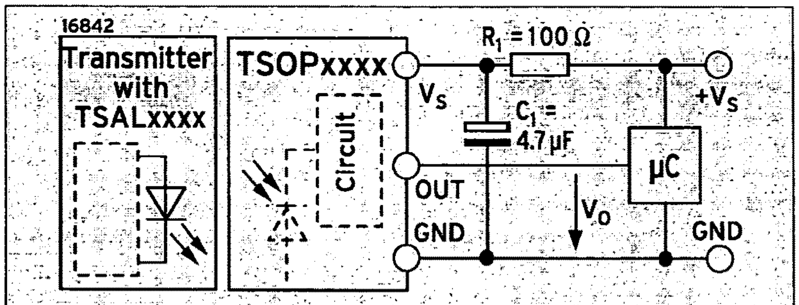
from Vishay

The TSOP62 series are miniaturized SMD-IR receiver modules for infrared remote control systems. A PIN diode and preamplifier are assembled on one lead frame, the epoxy package is designed as an IR filter. The demodulated output signal can be directly decoded by a microcontroller.

TSOP62xx is a standard IR remote control SMD receiver series, supporting all major transmission codes. The circuit of the TSOP62xx is designed in a way that unexpected output pulses due to noise or disturbance signals are avoided. A band-pass filter, an integrator stage and automatic gain control are used to suppress such disturbances. Data signals are distinguished from disturbance signals by the carrier frequency, the burst length and the duty cycle.

Features

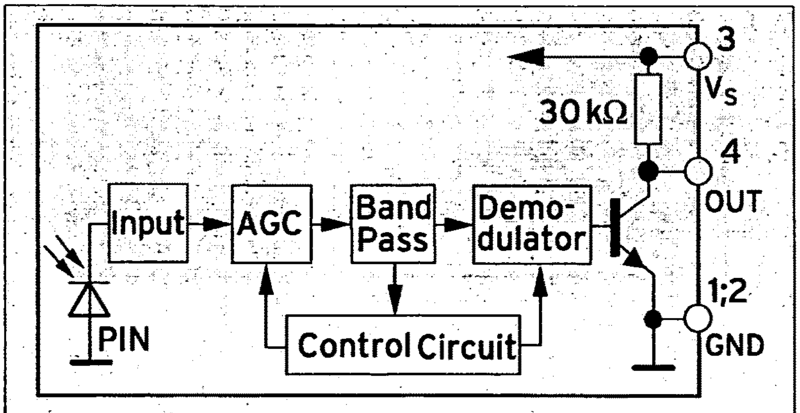
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Continuous data transmission possible
- TTL and CMOS compatibility
- Output active low
- Low power consumption
- High immunity to ambient light



$R_1 + C_1$ recommended to suppress power supply disturbances.

The output voltage should not be hold continuously at a voltage below $V_O = 3.3$ Vb y the external circuit.

Application Circuit



Block Diagram

Part	Carrier Frequency
TSOP6230	30 kHz
TSOP6233	33 KHz
TSOP6236	36 KHz
TSOP6237	36.7 KHz
TSOP6238	38 KHz
TSOP6240	40 KHz
TSOP6256	56 KHz

Parts List

Cordula Carlin, EBV Poing

HUMAN FACTORS, 1993, 35(2), 283–304

Repetitive Motion Disorders: The Design of Optimal Rate-Rest Profiles

DONALD L. FISHER,[1] ROBERT O. ANDRES, DAVID AIRTH, *and* STEPHEN S. SMITH, *University of Massachusetts, Amherst, Massachusetts*

Repetitive motion disorders of the upper extremities are the primary cause of lost time in hand-intensive industries. Typical remedies include the introduction of job aids and the redesign of the job. An alternative approach is considered here. Procedures are developed that can be used to determine the parameters of a job (e.g., the rate of work and the number and duration of the rest breaks) that minimize repetitive motion disorders and maximize productivity. It is shown that in theory the job parameters can be set so that one can achieve not only a decrease in repetitive motion disorders but also an increase in productivity. The application of the procedures to an actual manufacturing job requiring high-repetitive, high-force hand motions is discussed in detail.

INTRODUCTION

Repetitive motion disorders of the upper extremities such as carpal tunnel syndrome, DeQuervain's disease, tenosynovitis of finger flexor and extensor tendons, epicondylitis, ganglionic cysts, and neuritis are a major cause of lost days in industries requiring extensive hand and wrist movements (Armstrong, 1986). Estimates of the percentage of workers afflicted with such disorders have been as high as 25% (Armstrong, Radwin, Hansen, and Kennedy, 1986). Incidence rates in industries requiring repetitive motions have ranged from as low as 4.6 cases per 200 000 work hours in one area of a telecommunication's manufacturing facility (McKenzie, Storment, Van Hook, and Armstrong, 1985) to as high as 129.6 cases per 200 000 work hours in one division of a poultry processing plant (Armstrong, Foulke, Joseph, and Goldstein, 1982).

Causes

A number of causes of repetitive motion disorders have been identified. These include posture (e.g., position of the wrist), concentrations of mechanical stress in the region of the palm, vibration, cold, and gloves (e.g., see Armstrong et al., 1986; Silverstein, Fine, and Armstrong, 1986). More recent epidemiological studies have documented the importance of two additional factors: rate of repetition and force of tendon contraction. For example, Silverstein, Fine, and Armstrong (1987) reported that carpal tunnel syndrome is 15 times more likely to appear among workers in jobs requiring high-repetitive, high-force hand motions than it is to appear among

[1] Requests for reprints should be sent to Donald L. Fisher, 114 Marston Hall, College of Engineering, University of Massachusetts, Amherst, MA 01003.

workers in jobs requiring low-force, low-repetitive hand motions. More troubling still, Armstrong, Fine, Goldstein, Lifshitz, and Silverstein (1987) reported that hand and wrist tendinitis is 29 times more likely to appear in high-repetitive, high-force jobs than in low-force, low-repetitive jobs. Attempts to separate the effects of repetitiveness and force indicate that high repetitiveness appears to be a greater risk factor than is high force (Silverstein et al., 1987).

Physiological explanations of how repeated stress leads to the development of various repetitive motion disorders depend greatly on the nature of the stress and the identity of the affected tissue. Perhaps most is known about the relation between repetitive loading of the tendons and tenosynovitis of the finger flexor and extensor tendons. For example, recently Goldstein (1981; also see Goldstein, Armstrong, Chaffin, and Matthews, 1987) recorded both the elastic strain and the viscous strain and recovery of flexor digitorum profundus tendons subjected to uniaxial step stress and cyclic loads in intact human cadaver hands. Most important in the current context, the measurements indicated that above a certain rate, viscous strain accumulated over time. Furthermore, the accumulation increased more rapidly as the rate of repetition increased. The documentation of a sustained or increasing level of viscous strain is of consequence because such strain is thought to produce the microdamage that eventually leads to tenosynovitis of the finger flexor and extensor tendons (Armstrong, Castelli, Evans, and Diaz-Perez, 1984; Chu and Blatz, 1972; Goldstein, 1981; Rowe, 1987). Some have postulated that this microdamage involves a failure of the molecular links between the tissue matrix and filler material (e.g., Chu and Blatz, 1972).

Much the same explanation may apply to other repetitive motion disorders in which tendons are repeatedly loaded (for exan carpal tunnel syndrome, DeQuervain's ease, epicondylitis, and ganglionic cy Clearly a different explanation will be nee for repetitive motion disorders (e.g., neur that involve the application of other type stress (e.g., repeated pressure to the side the fingers). However, in general this va tion in underlying physiological models not change the procedure for identifying optimal rate-rest profile because, for the m part, this procedure does not depend on exact form of the equation governing the lation between stress and trauma.

Remedies

The remedies available depend greatly which particular risk factor or factors a present in a given job. Most remedial a tempts have focused on introducing job ai or redesigning the physical layout of the j (e.g., Armstrong et al., 1982; Damon, 196 Emanuel, Mills, and Bennet, 1980; McKenz et al., 1985; Tichauer, 1976). These attemp have proved to be extremely useful, fr quently leading to large reductions in repet tive motion disorders.

To date, however, no remedial efforts hav been reported that attempt to manipulate th rate-rest profile—that is, to change the rate c repetition or to break up the repetition by in troducing rest breaks. Modifications to th rate of work and the rest break schedule ca be used in addition to or in place of the intro duction of job aids and the redesign of th job. Modifications to the rate-rest profil alone are reasonable to consider when the lead to relatively large reductions in repetitive motion disorders and no change (or actual increases) in productivity because such modifications are relatively easy and quick to implement.

Attempts to modify the rate-rest profile may have been ignored primarily because it was believed that if repetitive motion disorders are decreased, then productivity must also decrease (Silverstein et al., 1986). Clearly, repetitive motion disorders can be decreased by decreasing productivity. However, it is by no means clear that a decrease in repetitive motion disorders *must* be accompanied by a decrease in productivity. For example, suppose that the work rate is increased and several relatively long rest breaks are replaced by many relatively short rest breaks. In this case it might be possible to decrease repetitive motion disorders while increasing productivity. Although the suggestion has recently been made that viscoelastic tendon models may be useful when studying the influence of the rate-rest profile on both repetitive motion disorders and productivity mstrong et al., 1987), no formal attempts o characterize this influence have been undertaken.

The notion that the rate-rest profile can influence productivity—and, in particular, the notion that the scheduling of work periods and rest breaks can affect the level of output—has been pursued by a number of investigators (Barnes, 1968; Gilbert 1968; Gilbreth and Gilbreth, 1953; Johnson and Ogilvie, 1972; Lee, 1974; McFarland, 1971; Streimer, 1971; Taylor, 1967). However, until recently there was no formal characterization of the problem which could be used to maximize productivity (Bechtold, Janaro, and Sumners, 1984; Janaro and Bechtold, 1985). The formal characterization assumes that the work rate decreases linearly with the length of the work period and increases linearly with ncreases in the duration of the rest period. However, the formal model does not apply vhen trauma increases over time and one vants to minimize any attendant disorders. urthermore, the model does not apply when he work rate must remain constant for some period of time, a typical constraint in many manufacturing jobs.

The primary objective of this article is the development of a procedure that can be used to identify the rate-rest profile that minimizes repetitive motion disorders and maximizes productivity. The procedure can be applied to any job in which it is possible to modify the work rate and to introduce one or more rest breaks in addition to those already in place. For example, suppose that currently a job is 3 h long and is followed by a 15-min rest break. The procedure might indicate that repetitive motion disorders could be *decreased* and productivity *increased* by introducing 5-min rest breaks every hour and increasing slightly the work rate in the interval between rest breaks.

The following discussion is divided into four major sections. First, the characteristics of the repetitive jobs considered in this article are discussed. It is shown that some of these characteristics are related to the level of trauma experienced on the job. Second, a simple method is described that can be used to compute, for any given rate-rest profile, both the productivity (number of goods produced on a given job) and the likelihood of repetitive motion disorders. Third, an actual manufacturing job with recorded instances of repetitive motion disorders is discussed. It is shown that for this existing job, it is relatively easy to identify a rate-rest profile that increases productivity and (in principle) decreases the incidence of repetitive motion disorders. Finally, an analytic procedure for identifying the optimal rate-rest profile is described.

JOBS AND TRAUMA

In order to identify the optimal rate-rest profile, it is necessary first to describe in more detail the characteristics of the specific set of repetitive jobs to be considered. In addition, it is necessary to make explicit the relation between the characteristics of a given job and the level of trauma.

The Job

To begin, consider the overall characteristics of the job. First, assume that a job of length τ consists of $p \geq 1$ identical *work periods* and *rest breaks* (Figure 1a), where p is an integer. A profile associated with such a job will be called a *regular profile* (to distinguish it from a profile associated with a job that does not consist of an identical, integer number of work periods and rest breaks). Second, assume that each work period consists of $c \geq 1$ identical *work cycles*, where c is an integer.

Next, consider the characteristics of a work cycle. First, assume that each work cycle is composed of one *stress interval* of length $\tau_s \geq$

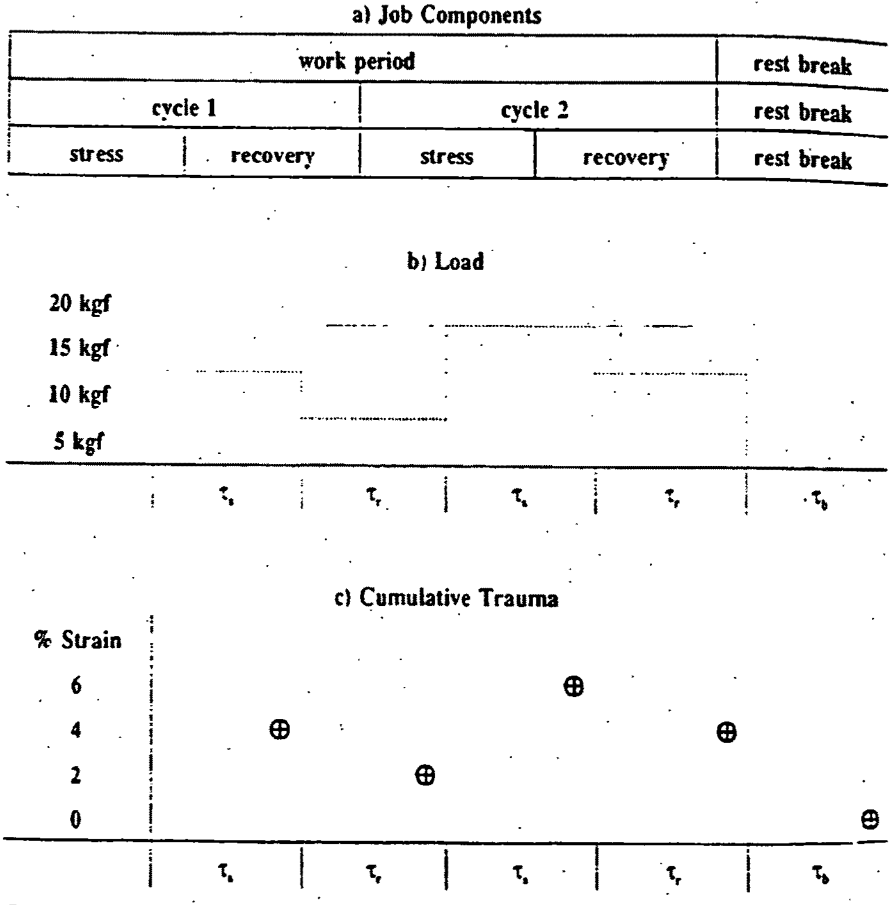


Figure 1. *(a) The components of a job where the number c of work cycles is equal to two. (b) The step load applied on the job during the work cycle stress and recovery intervals and rest break. (c) The level of trauma experienced at the end of the work cycle stress and recovery intervals and rest break.*

$\tau_s > 0$, where τ_s is the minimum duration of a stress interval, and one *recovery interval* of length $\tau_r \geq \tau_r > 0$, where τ_r is the minimal time that must intervene between each work cycle stress interval (Figure 1a). Second, assume that in the stress interval the stress is increased by amount δ_s to some new level and held constant at that level (this is referred to as a *step stress*). Third, assume that in the recovery interval the stress is decreased by amount δ_r to some new level and held constant at that level. Finally, assume that the stress interval output λ_s (i.e., the number of goods produced in one work cycle stress interval) remains constant throughout the work period. Because stress can take any of a number of forms, the actual units of stress will vary with the application. As noted, in many jobs tissues are stressed because they are subjected either to repeated stretching (in which case the relevant unit is load or force) or to repeated and direct mechanical insult (in which case the relevant unit is pressure).

Finally, consider the characteristics of the rest break. First, assume that each rest break consists of exactly one cycle, a break of length $\tau_b \geq \tau_6 > 0$, where τ_6 is the minimal time that must be set aside for a single rest break. Second, assume that in the rest break the load is decreased by amount δ_b. Set β equal to the total time that is spent in rest breaks—that is, $\beta = p\tau_b$—and set β equal to the minimal total time that must be spent in rest breaks (note that β need not necessarily equal $p\tau_6$).

A typical stress cycle is displayed in Figure 1b, in which we equate stress with load. The increase δ_s in load at each work cycle stress interval is set equal to 10 kg force (kgf), and the decrease δ_r in load at each work cycle recovery interval is set equal to 5 kgf. The decrease δ_b in load in the rest break is equal to 10 kgf, exactly that needed at the end of the work period to reduce cumulative trauma to zero.

Trauma

When tissue is stressed, it leads to trauma, the exact nature of which depends on the specific type of stress and the particular tissue involved. We want to introduce some generic trauma functions. At the same time we will show how these functions are realized for specific stresses and specific tissues.

To begin, set $f_s(\delta_s, \tau_s)$ equal to the increase in trauma during the work cycle stress interval. We assume that the stress interval trauma is a monotonically increasing function of the step stress δ_s applied throughout the interval and the duration τ_s of the interval. For example, if one equates trauma with viscous strain produced by repeated loading on tendons, then Goldstein (1981) shows that the work cycle stress interval trauma function has the form,

$$f_s(\delta_s, \tau_s) = 0.059\,\tau_s^{0.409}\,e^{-4.71+0.0176\delta_s}. \quad (1)$$

The units of stress (i.e., of δ_s) are now kilograms of force. Note that the function behaves as desired—that is, as either the duration of the stress interval or the load increases, so does the trauma. (It is assumed for the sake of convenience that the strain is produced by a male worker extending the second digit in the neutral position; other positions, digits, or sex would require a slight modification to the constants in Equation 1 [see Goldstein, 1981].)

Next, set $f_r(\delta_r, \tau_r)$ equal to the decrease in trauma during the work cycle recovery interval. We assume that the recovery interval trauma is a monotonically increasing function of δ_r and τ_r. Again, for the specific case in which one equates trauma with viscous strain, Goldstein has indicated the form that the trauma function should take:

$$f_r(\delta_r, \tau_r) = 0.001\,e^{-1.26+0.0806\tau_r+0.0108\delta_r}. \quad (2)$$

Note that, as required, the decrease in

trauma (in this case, strain) grows larger with both increases in the duration of the recovery interval and increases in the reduction in load.

Finally, set $f_b(\delta_r, \delta_b, \tau_r, \tau_b)$ equal to the decrease in trauma in the rest break. As with the decrease in the recovery interval, we assume that the decrease in the rest break trauma is a monotonically increasing function of $\delta_r, \delta_b, \tau_r$, and τ_b. In the special case in which the tendons are repeatedly stretched *and* the load is decreased to zero in the recovery interval (i.e., $\delta_s = \delta_r$), we can use Goldstein (1981) to obtain the rest break trauma function:

$$f_b(\delta_r, \delta_b, \tau_r, \tau_b) = 0.001\, e^{-1.26 + 0.0806(\tau_r + \tau_b) + 0.0108\delta_r} - f_r(\delta_r, \tau_r). \quad (3)$$

The decrease δ_b in load in the rest break does not appear on the right-hand side of Equation 3 because we assume that the load has already been reduced to zero during the last recovery interval.

The most important index in the current context, *cumulative trauma*, is derived from changes in trauma. Specifically, define $\Gamma_b(j)$ as the cumulative trauma at the end of rest break j and set it equal to j times the increase in cumulative trauma over each of the preceding work periods. This increase will equal the maximum of two quantities, 0, or the net increment in the cumulative trauma over one work period and rest break. More formally, for $j \geq 0$,

$$\Gamma_b(j) = (j)\max\{0, c[f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r)] - f_b(\delta_r, \delta_b, \tau_r, \tau_b)\}. \quad (4)$$

Define $\Gamma_r(i, j)$ as the cumulative trauma at the end of the recovery interval of cycle i in work period j, and set it equal to the sum of the cumulative trauma $\Gamma_b\ (j - 1)$ at the end of rest break $j - 1$ plus the increase in trauma over i cycles of work period j. This increase will be equal to one of two quantities. Specifically, if there is no net change in trauma from one work cycle to the next—that is, $f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r) \leq 0$—then the increase will equal zero. Otherwise the increase will equal i times the difference between the increase in trauma in the stress interval and the decrease in trauma in the recovery interval. More formally, for $i \geq 0$ and $j \geq 1$,

$$\Gamma_r(i, j) = \Gamma_b(j - 1) + \max\{0, i[f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r)]\}. \quad (5$$

Note that $\Gamma_r(0, j)$, the cumulative trauma a the beginning of work cycle j, is simply equa to $\Gamma_b(j - 1)$, the cumulative trauma at the en of rest break $j - 1$.

An example of the relation between loadin cycle, changes in trauma, and cumulativ trauma is displayed in Figure 1c, where fo purposes of this example cumulative traum is expressed as percentage strain. Note tha the increase $f_s(\delta_s, \tau_s)$ in trauma in the first an second work cycle stress intervals is equal t 4%. The decrease $f_r(\delta_r, \tau_r)$ in trauma in th first and second work cycle recovery interva is equal to 2%, and the decrease $f_b(\delta_r, \delta_b, \tau_r, \tau_b)$ in trauma in the rest break is equal to 4% Cumulative trauma at the end of the fir work cycle recovery interval is 2%—that i $\Gamma_r(1,1) = 2$—and increases to 4% at the end the second work cycle recovery interval $\Gamma_r(2,1) = 4$. Cumulative trauma at the end the rest break has returned to $0: \Gamma_b(1) = 0$.

Cumulative trauma is an important quantity to know because there is a level of th trauma, called *critical cumulative traum* above which it is hypothesized that dama to tissue occurs (Goldstein, 1981). We will s α equal to the critical level of the cumulati trauma. Note that α will presumably be function of the type of stress applied and t tissue to which it is applied. However, we w not subscript α to indicate which of the particular applications we have in mind. The a plication will be made clear by the surrounding material.

ʔate-Rest Profile

In the foregoing paragraphs we have de-ined a total of 15 parameters; 4 are associ-ited with each stress interval (τ_s, τ_t, δ_s, λ_s), 3 ire associated with each recovery interval (τ_r, τ_t, δ_r), and 3 are associated with each rest break (τ_b, τ_β, δ_b). In addition, there are a total of 5 general parameters (τ, p, c, α, β). We will define a rate-rest profile as an ordered list ω consisting first of the 5 general parameters and next of the three lists of stress interval, rest interval, and rest break parameters.

PRODUCTIVITY AND REPETITIVE MOTION DISORDERS

We have defined formally a rate-rest profile and made clear the relation between a given rate-rest profile and the level of trauma (Equations 4 and 5). It is now a straightforward matter to compute the productivity of any given profile. It is an equally straightforward matter to determine for any given pro- whether cumulative trauma rises above tne critical cumulative trauma at any point in the job.

Productivity

To begin, consider the productivity or work. Assume that production continues through to the end of the job. Then the productivity $W(\omega)$ for a regular rate-rest profile ω can easily be computed:

$$W(\omega) = pc\lambda_s. \qquad (6)$$

For example, suppose that a job consists of 100 work periods (p = 100) and 50 work cycles in each work period (c = 50). Suppose also that 2 goods are produced in each work cycle (λ_s = 2). Then the productivity for the job is 100 × 50 × 2 = 10 000 units.

Repetitive Motion Disorders

We are interested in identifying those rate-rest profiles that improve productivity and at the same time minimize repetitive motion disorders. No constraints were placed on the rate-rest profiles in Equation 6. Thus, although Equation 6 could be used to compute the productivity of any regular profile, the profile might well be one that greatly increased repetitive motion disorders. At this point, therefore, it needs to be determined whether a given profile will in theory minimize repetitive motion disorders. This determination will depend on whether or not the level of the critical cumulative trauma is known. In the following paragraphs it is assumed that this level is known or can be estimated. Later it will be shown what can be done when the level of the critical cumulative trauma is not known.

Assume that the level of the critical cumulative trauma is known. Given the characteristics of the repetitive jobs described earlier, it can be shown that for such jobs, cumulative trauma at all points in work cycle i, i + 1, ... $c(i \geq 1)$ of work period j will remain above the critical cumulative trauma α if cumulative trauma $\Gamma_r(i - 1, j)$ at the end of the $i - 1^{st}$ work cycle recovery interval is greater than α. Thus it is the cumulative trauma at the end of each work cycle recovery interval that will be of particular importance in designing a model that minimizes repetitive motion disorders while maximizing work.

Specifically, a profile will minimize repetitive motion disorders (i.e., it will theoretically reduce to zero such disorders) if cumulative trauma at the end of each work cycle recovery interval is no greater than the critical cumulative trauma. This will be true if the maximum level to which cumulative trauma rises at the end of each work cycle recovery interval is no greater than the critical cumulative trauma. Given the types of jobs considered and the monotonicity of the trauma functions, the maximum cumulative trauma $g(\omega)$ for a given profile ω can be computed as follows:

$$g(\omega) = \Gamma_b(p - 1) + \max\{0, c[f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r)]\}. \quad (7)$$

For example, suppose that $p = 100$ and $c = 50$. Suppose also that $f_s(\delta_s, \tau_s) = 2/50$, $f_r(\delta_r, \tau_r) = 1.9/50$, and $f_b(\delta_r, \delta_b, \tau_r, \tau_b) = 0.1$. Then, using Equation 4, we obtain $\Gamma_b(p - 1) = \Gamma_b(99) = 0$. Using Equation 7, we obtain $g(\omega) = 0 + \max\{0, 50(2 - 1.9)/50\} = 0.1$. Thus if $g(\omega) = 0.1 \leq \alpha$, there should in theory be no incidents of repetitive motion disorders.

APPLICATION

We have now identified how to compute productivity and how to determine whether a given profile will lead to repetitive motion disorders. At this point we could go on to identify the optimal profile. However, we have chosen not to do so. Rather, in order to ground our discussion before introducing the more theoretical work, we believe it is important to work through an actual application.

Two questions become central from the standpoint of applications. First, one can ask whether the current methods can ever be applied in practice, given that no estimates exist of the level of the critical cumulative trauma associated with any one of the several types of stress. The underlying concern motivating this question is that the methods may have to await the development of physiological models of cumulative trauma in order to be applied—that is, to identify the exact level of the critical cumulative trauma. Second, assuming that the methods can be applied, one can ask whether jobs exist in which the current rate-rest profile ω could in principle be replaced by an alternative profile ω' that both reduces repetitive motion disorders and increases productivity. The underlying concern motivating this question is that no jobs may exist in which the rate-rest profile could profitably be changed (perhaps because most such jobs already have profiles that are optimal or near optimal). Note that a positive answer to the foregoing two questions would mean only that the underlying rate-rest model could be applied now and that jobs exist to which, in theory, it could usefully be applied. There still remains the larger question of whether the predicted change in the level of trauma experienced by workers would indeed materialize if the theoretically optimal profile were to be put in place. We will speak to this larger question later in the article.

The answer to the two questions raised in this section will be developed within the context of an actual manufacturing job for which relevant ergonomic data are available.

Workers in the finishing area of a plant were required to execute a large number of high-force hand and wrist motions. This number could on occasion exceed 600 motions per hour. The high repetition rate and high force appeared to be taking a toll on worker productivity, considering that workers in the finishing area were increasingly reporting symptoms consistent with carpal tunnel syndrome, DeQuervain's disease, and tenosynovitis. Some of the incidents required surgical intervention; others resulted in lost time or employee attrition. The adverse impact on productivity provided the stimulus for management to seek solutions.

The Job

Four conveyor belts were in continuous operation. Two workers were stationed on either side of a conveyor belt. Each worker had to grab a handful of folded boxes off the line and put these boxes into a shipping container (the *grab-put stress interval*). The worker then could take a very brief pause (the *grab-put recovery interval*), followed by another grab and put. Measurement of the various durations and loads of the grab-put stress interval and the grab-put recovery interval occurred in two stages.

The details of these measurements do not bear directly on the current concerns, so we will note only that in the first stage, eight different observations of the static grip strength were recorded concurrently with the registration of muscle activity patterns in the form of root mean square electromyographs (RMS EMGs) for finger and wrist flexors and extensors from both hands of four subjects. The recordings were made before starting work and after at least 6 h of work. In the second stage, workers' hand movements were videotaped while on the job. Concurrently, muscle activity patterns (RMS EMGs) were recorded. Approximately 10 min of dynamic data were collected from each worker. The measurements are fully documented in Andres and Fisher (1989).

Productivity

The line is typically run at 20 000 boxes per hour. There are two workers per line, so each worker handles 10 000 boxes in 1 h. The job runs for 3 h, followed by an effective rest break of 20 min ($\beta = 20$). Thus a worker handles, on average, about 30 000 boxes—$W(\omega)$—in the first 3 h 20 min. Each grab consists on average of 50 boxes ($\lambda_x = 50$). Thus the worker makes approximately 600 grabs and puts ($p = 1$, $c = 600$) before the rest break (Equation 6).

Discussion with workers and inspection of production data indicated that τ_s, the duration of the minimum stress interval, was about 5 s; that τ_r, the duration of the minimum recovery interval, was about 5 s; and that τ_6, the duration of the minimum single rest break, was about 30 s (though this was relatively flexible). Thus in theory production could be increased to 54 000 boxes in one morning shift. However, this increase in production could well lead to a similar increase in the incidence of repetitive motion disorders, effectively offsetting the productivity gains.

Repetitive Motion Disorders

Without more information it is not possible either to determine how long the current job places the workers at risk for repetitive motion disorders or to identify the optimal rate-rest profile, because little is known about the level that the critical cumulative trauma actually assumes. However, in the current situation something can be learned about this level. In previous years workers in the finishing area had been given a 5-min break every ½ h (instead of the current schedule, one 20-min rest break every 3 h). There was no loss in worker productivity attributable to repetitive motion disorders. Assuming that workers would have reported the occurrence of symptoms had they indeed existed, the foregoing description implies that, as a conservative estimate, the critical cumulative trauma was reached in 25 min. The estimate is a conservative one because we do not know that the incidence of repetitive motion disorders would have increased had the line been run faster. (Note that the assumption that workers will report symptoms if they occur may not always be a good one, given that a worker might not assign a work-related cause to the appearance of symptoms associated with a repetitive motion disorder. In this case one could use an estimate of the critical cumulative trauma obtained from some other plant where similar work was performed and where it was reasonably safe to make the aforementioned assumption.)

Given this information, one can easily solve for α, the level of critical cumulative trauma. The job is one that repeatedly stresses the tendons, and thus we assume that the trauma is best described by the viscoelastic model of tendon strain developed by Goldstein (1981). Only the computations for the worker who used only her right hand will be illustrated. For this worker τ_s, the average duration of the grab-put stress interval, was 5 s and τ_r, the average duration of the grab-put recovery interval, was 13 s. Thus τ, the length of one work cycle, is equal to 18 s. In addition, for this worker the average load during the grab-put stress interval was 15.366 kgf (33.875 pounds), and there was no load during the recovery interval.

We needed to transform the force measured by the EMG recordings into the actual load on the tendons of the second digit because it is this load that appears in Goldstein's equations. The transformation required multiplying first by a factor of 4.32 (to obtain the force on the tendon in a pinch grasp; see Chao, Opgrande, and Axmear, 1976) and then by a factor of 0.25 (to obtain the load on digit two; see Hazelton, Smidt, Flatt, and Stephens, 1975). Doing such, δ_s = 16.59 kgf. The load was reduced to zero in the work cycle recovery interval, so that $\delta_r = \delta_s$ = 16.59 kgf. To obtain an estimate of an upper bound for the critical cumulative trauma, one has simply to multiply the number of work cycles in 25 min (converted to seconds) times the net work cycle cumulative trauma—$f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r)$, where the first and second functions are obtained, respectively, from Equations 1 and 2—and set this equal to α:

$$\alpha = \left\{\frac{(25)(60)}{18}\right\}\{[(0.059)(5)^{0.409}e^{-4.71+(0.017)(16.59)}] - [0.001e^{-1.26-(0.0806)(13)-(0.0108)(16.59)}]\}. \quad (8)$$

Doing these computations, one obtains α = 0.032746. The computations for the other workers are similar; they yield a somewhat higher α. The conservative estimate obtained earlier (i.e., the smallest estimate of α) is the one used in the following computations.

Now that an estimate of α is available, one can determine whether cumulative trauma in the current profile rises above critical cumulative trauma. As noted, $\delta_s = 16.59$, $\tau_s = 5$, $\delta_r = 16.59$, $\tau_r = 13$, $p = 1$, and $c = 600$. Using Equation 5, cumulative trauma $\Gamma_r(600,1)$ at the end of the recovery interval of the last work cycle can be computed as follows:

$$\Gamma_r(600.1) = 600\{[(0.059)(5)^{0.409}e^{-4.71+(0.017)(16.59)}] - [0.001e^{-1.26-(0.0806)(13)-(0.0108)(16.59)}]\}. \quad (9)$$

Solving for $\Gamma_r(600,1)$, one obtains 0.235774, which is more than seven times higher than the critical cumulative trauma.

Next, consider an alternative profile that satisfies the constraints on the various job parameters and in which the level of cumulative trauma never rises above the critical cumulative trauma. The details for producing such a profile will be described in the next section, but here it suffices to list the parameters: p = 26, c = 39, τ = 3 h 20 min (12 000 s), δ_s = 16.59 (remains unchanged), τ_s = 5 (remains unchanged), τ_r = 5.315, τ_b = 59.25, $\delta_r = \delta_s$, and by implication δ_b = 0. Note that all the requirements are satisfied.

First, p and c are integers and $p[c(\tau_r + \tau_s) + \tau_b]$ = 12 000 = τ. Second, $\tau_s \geq \tau_\sigma$ (because the duration of the stress interval is not changed), $\tau_r \geq \tau_\rho = 5$, and $\tau_b \geq \tau_\beta = 30$. We leave it to the reader to show that cumulative trauma rises to exactly the critical level in one work period and falls to zero during the rest break (in fact, it falls to zero in the first 51.58 s of the rest break, a quantity that is computed from knowledge of the estimated α and the recovery interval and rest break trauma functions).

We have succeeded in producing a profile that satisfies the relevant job constraints while keeping trauma below the critical level, so we can now ask about the productivity associated with this profile. Significantly, the productivity of this profile is 50 700 boxes, an increase of 69% in the number of boxes handled using the current profile. In short, we ıave produced for an actual manufacturing ob a profile that, in theory, reduces repetiive motion disorders while increasing proluctivity. Furthermore, we have produced :his profile in the absence of estimates of crit.cal cumulative trauma. Finally, note that :ven though the profile we produced is sensitive to trauma, it leads to a level of productivity equal to fully 94% of the level achieved by a profile that is not sensitive to trauma (i.e., 50 700 is approximately 94% of 54 000).

OPTIMAL RATE-REST PROFILE

We have shown that when the critical cumulative trauma is known (or can be estimated), there exists for at least one real job a rate-rest profile that increases productivity and decreases repetitive motion disorders. It can be shown that for any given job there are many rate-rest profiles that have these two 'red characteristics. At this point the ıader would probably like to know whether there is one best such profile. That is, can it be determined analytically which profile (or set of profiles) is optimal?

Feasible Rate-Rest Profiles

The set of profiles that could potentially serve as the optimal profile must satisfy the various job constraints defined earlier. In addition, we will need three more constraints. First, assume that during the rest break the cumulative trauma is reduced to zero:

$$\max\{0, c[f_s(\tau_s, \tau_r) - f_r(\delta_r, \tau_r)] - f_b(\delta_r, \delta_b, \tau_r, \tau_b)\} = 0. \qquad (10)$$

Note that this means that $\Gamma_b(j) = 0.0 \leqslant j \leqslant p$. Another way of stating this constraint is to require that the duration τ_b of a single rest break be greater than τ_b, the time to reduce cumulative trauma from α to zero at the end of the last work cycle recovery interval.

Second, assume that cumulative trauma $\Gamma_r(c, j)$ at the end of the last work cycle recovery interval of each work period j is equal to critical cumulative trauma α. Because $\Gamma_b(j - 1) = 0$, Equation 5 can be written more simply as:

$$\Gamma_r(c,j) = c[f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r)] = \alpha. \qquad (11)$$

This last assumption is an attempt to balance judiciously two needs: the need to continue work as long as possible (given that in most cases starting up a job will involve some costs, which presumably are reduced as the length of the work period is increased) and the need to keep cumulative trauma at or below the critical cumulative trauma (to keep repetitive motion disorders to a minimum).

Finally, assume that 3 of the general job parameters (τ, α, β), all of the stress interval parameters (τ_s, τ_s, δ_s, λ_s), 2 of the 3 recovery interval parameters (τ_r, δ_r), and 2 of the 3 rest break parameters (τ_b, δ_b) cannot be modified. Perhaps the only parameter that it may seem we could modify, but have decided not to, is τ_s, the duration of the stress interval. We decided not to modify this because we had no way of knowing a priori whether employees in a particular job were working at their fastest during the stress interval. Thus our results remain most general if we assume that the duration of this interval remains unchanged. In order for the rate-rest profile to be consistent with the above assumptions, the remaining parameters that we assume can be modified—c, p, τ_r and τ_b—must satisfy the following constraints:

$$\begin{aligned} &\tau_r \geqslant \tau_r, \\ &\tau_b \geqslant \max\{\tau_b, \tau_b, \beta/p\}, \\ &p[c(\tau_s + \tau_r) + \tau_b] = \tau, \end{aligned} \qquad (12)$$

where p and c are integers. Note that only the second inequality modifies in any way the job constraints set out earlier in the article.

Any profile that satisfies the constraints in Equations 10–12 will be defined as a *feasible profile*. Note again that the set of feasible rate-rest profiles is confined to profiles that never lead to an increase in cumulative trauma above the critical cumulative trauma during the work period (Equation 11) and which always lead to a decrease in cumulative trauma to zero during the rest break (Equation 10). Thus, in theory, no profile in the set of feasible rate-rest profiles should give rise to incidents of repetitive motion disorders.

Regular Profiles

We can now identify the optimal rate-rest profile. This will depend in part on the characteristics of the particular job under consideration. To begin, we want to focus on the identification of the optimal, feasible profile in the set of regular, feasible profiles. (Recall that a regular profile is defined as one in which the number of work periods and rest breaks are identical and equal to the same integer.) Define $E(\omega)$ as the *work cycle efficiency* for profile ω and set it equal to the number of goods produced in one time unit of a single work cycle:

$$E(\omega) = \lambda_s/[\tau_s + \tau_r]. \quad (13)$$

Intuition suggests that if the work cycle efficiency $E(\omega')$ of one profile—say, ω'—is greater than the work cycle efficiency $E(\omega)$ of some other profile—say, ω—then the productivity $W(\omega')$ of profile ω' ought to be greater than the productivity $W(\omega)$ of profile ω. In fact this intuition is true, but only for regular profiles with no slack.

Before proving the above-mentioned intuition, we first need to define slack and to show how one identifies a rate-rest profile with the most efficient work cycle. Let τ_d represent the *slack*, and set it equal to the difference between the total time actually spent resting in the profile under consideration and the total time β that must be spent in rest breaks:

$$\tau_d = \tau - pc(\tau_s + \tau_r) - \beta. \quad (14)$$

For example, consider a job of length 9 (τ = 9). Suppose that the work cycle stress interval is 2 units long (τ_s = 2) and that the work cycle recovery interval is 1 unit long (τ_r = 1). Suppose that there are two work periods (p = 2) and one work cycle (c = 1) in each work period. Finally, suppose that the total duration of the rest breaks must be at least 2 units long (β = 2). Then, using Equation 14, the slack for this profile is positive and equal to 9 − (2)(1)(2 + 1) − 2 = 1. If the job were of length 8 or the minimum rest break were of length 3, then the slack would be zero, as can be confirmed using Equation 14.

Next, consider identification of ω^*, the regular, feasible profile with the most efficient work cycle. Because λ_s and τ_s remain constant across changes in profiles, it is clear from Equation 13 that a regular, feasible profile with the most efficient work cycle is one that minimizes τ_r—that is, one that rises most quickly to the critical cumulative trauma. This requires identification of the four free parameters in the rate-rest profile: p^*, the number of work periods and rest breaks in the regular, feasible profile with the most efficient work cycle; c^*, the number of work cycles in this profile; τ_r^*, the duration of the corresponding recovery interval; and τ_b^*, the duration of the corresponding rest break.

In order to identify c^*, one needs simply to divide α, the level of critical cumulative trauma, by the difference $f_s(\tau_s, \delta_s) - f_r(\tau_r, \delta_r)$, the net increment in cumulative trauma in one work cycle. If the resulting number is an integer, set c^* equal to it. If not, set c^* equal to the next highest integer (e.g., if the division produces the quantity 3.2, set c^* equal to 4). In order to identify τ_r^*, find the smallest value of τ_r ($\geq \tau_r$) such that cumulative trauma rises to α in exactly c^* cycles. In order to identify p^* and τ_b^*, find the largest positive integer k such that $k[c^*(\tau_s + \tau_r^*) + t_b^*] = \tau$ and $t_b^* = \{\tau - k[c^*(\tau_s + \tau_r^*)]\}/k$, subject to $t_b^* = [\tau - kc^*(\tau_s + \tau_r^*)]/k \geq \max\{\tau_b, \tau_b, \beta/p^*\}$. Set $p^* = k$ and $\tau_b^* = t_b^*$. All other parameters in ω^* are set equa to the corresponding parameters in ω. In Lemma A1 (Appendix A) we show that if ω^* is a regular, feasible profile, then there exists no other regular, feasible profile with a more efficient work cycle than ω^*. In Theorem A1 (Appendix A), we show that if there is no slack in a job, then the regular, feasible profile with the maximally efficient work cycle is the most productive profile. (If ω^* is not a regular, feasible profile, alternative measures need to be considered. These are discussed in Appendix A.)

Examples. Two examples follow: one in which a regular, feasible profile with the maximally efficient work cycle does not contain slack and one in which it does. To begin, consider a regular, feasible profile ω that does not contain any slack. Specifically, assume that $\alpha = 100$ and $\beta = 200$. Assume also that $p = 2$, $c = 100$, $\lambda_s = 1$, $\tau = 800$, $\tau_s = 2$, $\tau_{\tilde{s}} = 2$, $\tau_r = 1$, $\tau_f = 1$, $\tau_b = 100$, $\tau_{\hat{b}} = 40$, $\tau_{\tilde{b}} = 100$, $f_s(\delta_s, \tau_s) = 2$, $f_r(\delta_r, \tau_r) = 1$, and $f_b(\delta_r, \delta_b, \tau_r, \tau_b)$ 100. Now, consider the identification of the most efficient regular, feasible profile ω^*. Because $\alpha'[f_s(\tau_s, \delta_s) - f_r(\tau_r, \delta_r)] = 100$, it follows that $c^* = c$ and $\tau_r^* = \tau_r$. Furthermore, solving for p^* and τ_b^*, we find $p^* = 2$ and $\tau_b^* = 100$. It is easy to show that profile $\omega^* = \omega$ is both a regular and a feasible one (Equations 10–12). Thus, by Lemma A1, $\omega^* = \omega$ has a maximally efficient work cycle. Furthermore, using Equation 14, it can be shown that the slack, $\tau_d = 800 - (2)(100)(2 + 1) - 200$, is zero. Thus, by Theorem A1, $\omega^* = \omega$ is the most productive regular, feasible profile.

Next, consider a regular profile ω with a maximally efficient work cycle that does contain slack. As in the previous example, assume that $\alpha = 100$, $\beta = 200$, $p = 2$, $c = 100$, $\lambda_s = 1$, $\tau = 890$, $\tau_s = 2$, $\tau_{\tilde{s}} = 2$, $\tau_r = 1$, $\tau_f = 1$, $\tau_b = 145$, $\tau_{\hat{b}} = 40$, $\tau_{\tilde{b}} = 100$, $f_s(\delta_s, \tau_s) = 2$, $f_r(\delta_r, \tau_r) = 1$, and $f_b(\delta_r, \delta_b, \tau_r, \tau_b) = 145$. Then, again, $c^* = c$, $\tau_r^* = \tau_r$, $p^* = p$, and $\tau_b^* = \tau_b$. Now, however, $\tau_b^* = \tau_b = 145$ (as opposed to 100 in the previous example). The profile $\omega^* = \omega$ is both a regular and a feasible one. Thus, by Lemma A1, this profile has a maximally efficient work cycle. However, using Equation 14, it follows that the slack, $\tau_d = 890 - 2(100)(2 + 1) - 200$, is positive and equal to 90. Thus the conditions of Theorem A1 are not satisfied. In fact, one can produce an alternative regular, feasible profile with a less than maximally efficient work cycle that is actually more productive. Construction of such a profile is potentially instructive.

To begin, note that the efficiency, $E(\omega^*)$, of the aforementioned maximally efficient, regular, feasible profile ω^* is 1 3 and the productivity, $W(\omega^*)$, is 200. Now consider a second regular profile—say, ω'—in which $\alpha = 100$ and $\beta = 200$. Assume also that $p' = 2$, $c' = 110$, $\tau_r' = 1.0909$, $\tau_b' = 105$, $f_r(\delta_r, \tau_r') = 1.0909$, and $f_b(\delta_r, \delta_b, \tau_r', \tau_b') = 105$ (all other parameters remain unchanged). The slack in this case is 10. The efficiency of the alternative profile ω', $E(\omega') = 11/34$, is clearly smaller than that of the most efficient profile ω^*, $E(\omega^*) = 1/3$. Furthermore, because the less efficient profile is a feasible one, cumulative trauma never rises above the critical cumulative trauma. However, the productivity of the alternative, less efficient profile ω', $W(\omega')$ = 220, is clearly greater than the productivity of the most efficient profile ω^*, $W(\omega^*) = 200$. Thus even if ω^* is a regular, feasible profile with the most efficient work cycle, there can still exist a more productive alternative regular, feasible profile if the slack associated with ω^* is positive.

Irregular Profiles

Given the foregoing, one is left wondering how to identify the optimal profile when the most efficient regular profile contains slack. Toward this end, we now want to address jobs with *irregular profiles*. Specifically, we define an irregular profile as a profile with p identical work periods and rest breaks and one *partial work period* that begins at the end of the last rest break. A partial work period is defined as a work period that consists of d work cycles, $0 \leq d \leq c$, where d is an integer.

The constraints (Equations 10–12) defining the set of feasible profiles must be changed slightly when considering an irregular profile. In particular, because the number of work periods and rest breaks is no longer identical, we need to change the expression in Equation 12 for the duration of the job to include the addition of the partial work period. Thus Equation 12 now reads:

$$\tau_r \geq \tau_{\hat{r}}, \tag{15}$$

$$\tau_b \geq \max\{\tau_{\hat{b}}, \tau_{\check{b}}, \beta/p\},$$

$$p[c(\tau_s + \tau_r) + \tau_b] + d(\tau_s + \tau_r) = \tau,$$

where p, c, and d are integers.

The efficiency of an irregular profile is computed just as is the efficiency of a regular profile (Equation 13). However, the slack is computed slightly differently because the slack time is now filled with $d \geq 0$ work cycles. Specifically,

$$\tau_d = \tau - pc(\tau_s + \tau_r) - d(\tau_s + \tau_r) - \beta. \tag{16}$$

The reason for considering irregular profiles is straightforward. In particular, it can be shown that such profiles are at least as productive as regular profiles (Lemma B1, Appendix B). Suppose now that we define p^*, c^*, τ_r^*, and τ_b^* for irregular, feasible profiles as we did for regular, feasible profiles. Then it can be shown that if the irregular, feasible profile ω^* exists, it has the maximally efficient work cycle (Lemma B2, Appendix B). Finally, it can be shown that ω^* is the most productive irregular, feasible profile if there is no slack (Theorem B1, Appendix B).

Example. An example to illustrate the aforementioned lemmas and theorem may prove to be useful. Consider again a job in which the regular profile ω produces positive slack: $\alpha = 100$, $\beta = 200$, $p = 2$, $c = 100$, $\lambda_s = 1$, $\tau = 890$, $\tau_s = 2$, $\tau_{\hat{s}} = 2$, $\tau_r = 1$, $\tau_{\hat{r}} = 1$, $\tau_b = 145$, $\tau_{\hat{b}} = 40$, $\tau_{\check{b}} = 100$, $f_s(\delta_s, \tau_s) = 2$, $f_r(\delta_r, \tau_r) = 1$, and $f_b(\delta_b, \delta_r, \tau_b, \tau_r) = 145$. The slack in this case equals 90. Now suppose that we fill this slack period with as many work cycles as possible ($d^* = 30$) and set $p^* = 2$, $c^* = 100$, $\tau_r^* = 1$, and $\tau_b^* = 100$. Assume $f_b(\delta_r, \delta_s, \tau_r^*, \tau_s^*) = 100$. Profile ω^* now forms an irregular, feasible profile (Equations 10, 11, and 15). Thus, by Lemma B2, the new irregular profile ω^* has a maximally efficient work cycle. Then, using Equation 16 to compute the slack, this slack, $\tau_d^* = 890 - (2)(100)(2 + 1) - 30(2 + 1) - 200$, is zero. Thus by Theorem B1 the irregular profile ω^* is the most productive one. Finally, by Lemma B1 there is no alternative, regular profile that is more productive.

DISCUSSION

Two important results have emerged. First, profiles do exist that can both reduce repetitive motion disorders and increase productivity. Such profiles exist not only for simulated jobs but for actual ones as well. Second, among the set of profiles that minimize repetitive motion disorders and satisfy certain feasibility constraints, it was possible to identify the profile that maximized productivity. Thus the goal of developing a procedure for identifying the optimal rate-rest profile has been achieved.

The success of the procedure depended on three significant assumptions. First, it was assumed that the critical cumulative trauma was known or a conservative estimate could be obtained. Although it is difficult to imagine circumstances in which a conservative estimate could not be obtained, such circumstances could indeed arise. It is important to generalize the procedures for reducing trauma and increasing productivity to situations for which the critical cumulative trauma is not known. Second, it was assumed that the job resulted in a slack of zero length. This may be unrealistic in many situations.

ıus it is important to generalize the proceıres to situations in which there is a positive ıck. Third, it was assumed that the load ıring the work cycle stress interval reained constant across changes in the rate of ork. This assumption appears not to hold in ıme cases (Arndt, 1987), and thus it is imortant to generalize the procedures to jobs ı which the load changes with changes in the ork rate.

ritical Cumulative Trauma: Not Known

Sometimes the critical cumulative trauma ıill not be known (or a conservative estimate annot be obtained). In such cases, suppose it ; possible to identify an alternative profile—ay, ω'—with the following two important haracteristics. First, $g(\omega')$, the maximum cuıulative trauma of the alternative profile ω', s less than $g(\omega)$, the maximum cumulative rauma of the current profile ω. Second, $_\gamma(\omega')$—the length of time in one work period the following rest break that a tendon is ...etched above any given level γ of the cuıulative trauma ($\gamma > 0$) in the alternative rofile ω'—is less than $t_\gamma(\omega)$, the length of time in a work period and following rest break that the tendon is stretched above the same level γ of the cumulative trauma in the current profile ω. The γ *trauma time* for profile ω—that is, $t_\gamma(\omega)$—will be measured from the end of the first recovery interval in which cumulative trauma rises above γ until that point in time when cumulative trauma dips below γ in the immediately succeeding rest break. (Note that the γ trauma time does not change across work periods because we assume that each work period consists of an identical number of work cycles and that cumulative trauma is reduced to zero in the intervening rest breaks.)

Suppose that the above two conditions are satisfied—that is, that

$$g(\omega') < g(\omega), \tag{17}$$

$$t_\gamma(\omega') < t_\gamma(\omega). \tag{18}$$

Then, given what is known about the etiology of repetitive motion disorders, it is reasonable to assume that the number of such disorders will be no larger with the alternative profile than it is with the current profile. This number will be smaller if, in the current profile, the critical level of the cumulative trauma is exceeded. Otherwise, it will remain unchanged.

Decreases in repetitive motion disorders. The question at this point is whether, when the critical cumulative trauma is *not* known, a procedure can be developed that functions similarly to that developed when the critical cumulative trauma is known. Specifically, can a procedure be developed for identifying profiles when the critical cumulative trauma is not known which keeps constant or reduces repetitive motion disorders and at the same time either increases or does not change productivity? Consider first a procedure for identifying profiles that keep constant or reduce repetitive motion disorders and do not change productivity.

To keep things simple, we will constrain our search to quasi-feasible profiles. A *quasi-feasible profile* is defined as a (regular or irregular) profile that satisfies the constraints in Equations 10 and 12 or Equations 10 and 15. (Equation 11 and the quantity τ_6 in Equations 12 and 15 are excluded because α is not known.) In certain situations intuition suggests that quasi-feasible profiles can be constructed so that the maximum cumulative trauma $g(\omega)$ and the γ trauma time $t_\gamma(\omega)$ can be reduced for all $\gamma > 0$ without decreasing productivity. For example, consider a profile ω with one work period (a single period profile), one rest break, and $c > 1$ work cycles. Then, the single period profile ω can sometimes be changed into an alternative multiple-period, quasi-feasible profile ω' in which the only parameters that change are p' (> 1), c' ($< c$), and τ_b' ($= \tau_b/p'$).

For example, a single-period, regular,

quasi-feasible profile ω with $\dot{\beta} = 100$, $p = 1$, $c = 100$, $\tau = 400$, $\tau_s = 2$, $\tau_{\hat{s}} = 2$, $\tau_r = 1$, $\tau_{\hat{r}} = 1$, $\tau_b = 100$, $\tau_{\hat{b}} = 25$, $f_s(\delta_s, \tau_s) = 2$, $f_r(\delta_r, \tau_r) = 1$, and $f_b(\delta_r, \delta_b, \tau_r, \tau_b) = 100$ can be changed into a multiple-period, regular, quasi-feasible profile with $p' = 2$, $c' = 50$, $\tau_b' = 50$, and $f_b(\delta_r, \delta_b, \tau_r', \tau_b') = 50$ (all other parameters and values of the relevant functions remain unchanged). A cursory inspection indicates that the maximum cumulative trauma (50) and the γ trauma time of the multiple-period profile ω' are each less than, respectively, the maximum cumulative trauma (100) and the γ trauma time of the single-period profile ω for all $\gamma > 0$. For example, for $\gamma = 50$, $t_\gamma(\omega') = 0$ and $t_\gamma(\omega) = 200$. Thus the multiple-period profile should keep constant or reduce repetitive motion disorders.

More generally, by modifying p, c, and τ_b, it should be possible to derive multiple-period, quasi-feasible profiles from single-period, quasi-feasible profiles that keep constant or reduce repetitive motion disorders while leaving the productivity unchanged. In fact, it can be shown that such is the case (Theorem C1, Appendix C). In addition, one can ask whether there is a multiple-period, quasi-feasible profile that reduces the incidence of repetitive motion disorders at least as much as every other multiple-period, quasi-feasible profile. We show that there is such a profile (Corollary C1, Appendix C) and that it is equal to that profile which both minimizes c and satisfies Equations 10 and 12 (for regular profiles) or Equations 10 and 15 (for irregular profiles).

Increases in productivity. We argued in the previous paragraphs that if the current profile was a single-period, multiple-cycle, quasi-feasible profile (i.e., $p = 1$, $c > 1$), then a multiple-period profile ($p' > 1$, $c' < c$, $\tau_b' = \tau_b p'$) derived from it would yield the same productivity. We assume that both the duration of the work cycle recovery interval and the slack remained unchanged in the multiple-period profile. However, there is no a priori reason to keep them as is. In fact, there may be good reasons to change them because decreasing the duration of the work cycle recovery interval can sometimes increase productivity, as we demonstrated in the previous sections. Thus the question to ask at this point is whether a procedure can be developed that identifies a multiple-period profile ω' that increases productivity and keeps constant or decreases repetitive motion disorders when either the duration of the work cycle recovery interval or the slack is no longer the same as it was in the associated single-period profile. Unfortunately, we have not been able to generate an analytic procedure that permits us to proceed with the identification. Clearly more work is needed.

Positive Slack

We have shown in Theorems A1 and B1 how to minimize repetitive motion disorders and maximize productivity when the cumulative trauma is known and there is no slack. We have shown in Theorem C1 how to minimize repetitive motion disorders when the cumulative trauma is not known and there is (or is not) slack. The following shows what to do when the cumulative trauma is known and there is positive slack.

When the cumulative trauma is known, there must be a finite number of regular, feasible profiles because c (the number of work cycles) and p (the number of periods) are both finite, and because the two remaining parameters (τ_r and τ_b) that vary across profiles are determined by the value of c and p. Furthermore, because the number of regular, feasible profiles is finite when the cumulative trauma is known, it is possible to search exhaustively through such profiles for the optimal one. The search can easily be automated on the computer. A similar argument applies to irregular, feasible profiles.

Variable-Stress Interval Load

It has been assumed throughout that the effective load remained constant during the work cycle stress interval. However, Arndt (1987) has reported that forces can actually increase as the work rate increases. For example, the results from EMG recordings of one worker indicate that a 10% increase in speed produced a 38% increase in pinch force. Although important, this finding may not easily generalize to jobs other than the one Arndt analyzed. First, the workers were producing at very close to their maximum speed (one part every 1.1 s). The observed increase in forces with increases in work rate might not have been noted had the workers been working at a more relaxed pace. Second, the workers were paid on an incentive basis. This may inflate the importance to the worker of the relative speed of production, producing more tension than otherwise might be the case. Consistent with this hypothesis, Arndt found that increases in EMG were observed when workers were asked to speed up production even when *no* objective change in work rate was recorded. Also, decreases in EMG were observed when workers were asked to slow production, again even when no change in the objective work rate was recorded.

In situations in which one suspects that the effective load in the work cycle stress interval will increase with increases in work rate, a conservative course of action is recommended. Specifically, assume that the current job has one work period and one rest break. Then cumulative trauma will be reduced and the productivity left unchanged if a multiple-period profile is used in place of the current single-period profile (Theorem C1). A more aggressive approach (one that both decreases repetitive motion disorders and increases productivity) could be taken if one knew how the effective load changed with the rate of work. However, a discussion of this approach is beyond the scope of this article.

Verification

Finally, it is necessary to say something about the steps one can take to verify our model. Because something is known about the relation between the particular characteristics of a job, tendon strain, and tendinitis/tenosynovitis, we will confine our remarks to jobs in which this relation holds. For any given level α of the critical cumulative trauma, it is possible to compute the $\gamma\ (= \alpha)$ trauma time using Goldstein's (1981) equations. Given other work in epidemiology, it is not unreasonable to assume that a logistic-response relationship exists between the γ trauma time $\tau_\gamma(\omega)$ and the probability p that a worker develops symptoms of tendinitis and/or tenosynovitis, where $p = \exp[\beta_0 + \beta_1\tau_\gamma(\omega)]/\{1 + \exp[\beta_0 - \beta_1\tau_\gamma(\omega)]\}$. This critical additional step makes it possible to fit the model to data from current and future epidemiological studies. Any of a number of measures could then be used to determine the relative goodness of the resulting fit. Preliminary work is now under way.

CONCLUSION

Repetitive motion disorders create real hardships both for employee and employer. Remedies now exist that can be used on some jobs, but not all. These remedies require the redesign of the job or the introduction of job aids. An alternative remedy was discussed here. Specifically, it was shown that one could achieve a reduction in repetitive motion disorders by introducing a multiple-period profile ω' in place of a single-period profile ω. Most important, the construction of the multiple-period profile did not depend on knowledge of the critical cumulative trauma, on the absence of slack, or on the effective load remaining constant across changes in the work rate. The multiple-period profile required merely the introduction of p' rest breaks each of length β/p'' (Theorem C1).

If, in addition to reducing the cumulative trauma, one wanted to minimize such disorders and also increase production, additional constraints would need to be imposed. First, if the critical cumulative trauma is known (or a conservative estimate can be obtained), if there is no slack, and if the load remains unvarying across changes in the work rate, then it is possible to identify analytically the rate-rest profile that reduces repetitive motion disorders to a minimum and maximizes productivity (Theorems A1 and B1). Second, if the critical cumulative trauma is known (or a conservative estimate can be obtained), if there is positive slack, and if the load remains unvarying across changes in the work rate, then it is possible to identify by enumeration the rate-rest profile that reduces repetitive motion disorders to a minimum and maximizes productivity. Third, if the critical cumulative trauma is not known, then the rate-rest profile that minimizes repetitive motion disorders and keeps productivity constant can be identified (Corollary C1); however, it is not possible (currently) to identify those profiles that both reduce repetitive motion disorders and increase productivity.

ACKNOWLEDGMENTS

The authors wish to thank two anonymous reviewers for their extensive comments and suggested modifications.

APPENDIX A: THE OPTIMAL REGULAR PROFILE

In this appendix we prove that the regular, feasible profile with a maximally efficient work cycle is the most productive regular, feasible profile if it contains no slack. To begin, recall that c^* is set equal to the number of work cycles that causes cumulative trauma to rise most quickly to the critical cumulative trauma and τ_r^* is the duration of the corresponding recovery interval. Furthermore, p^* is set equal to the largest integer k such that $k[c^*(\tau_s + \tau_r^*) + t_b^*] = \tau$, $t_b^* = \{\tau - k[c^*(\tau_s + \tau_r^*)]\}/k$, and $t_b^* \geq \max\{\tau_{\hat{b}}, \bar{\tau}_{\hat{b}}, \hat{\beta}/k\}$. Finally, τ_b^* is set equal to t_b^*. Now, consider the profile ω^* consisting of p^*, c^*, τ_r^*, τ_b^*, and the corresponding parameters in ω. Then,

Lemma A1: If ω^* is a regular, feasible profile derived as above, then there exists no other regular, feasible profile that has a more efficient work cycle than does ω^*.

To begin, we need to make clear our notational conventions. Recall that it is assumed that when explicitly comparing two profiles, the parameters α, $\hat{\beta}$, τ, δ_s, τ_s, τ_s, λ_s, δ_r, τ_f, and δ_b, τ_b, $\tau_{\hat{b}}$, $\bar{\tau}_{\hat{b}}$, do not change. Thus, for example, if we were comparing two profiles—ω^* and ω'—α would appear everywhere we would have written α^* or α', respectively. Furthermore, note that unless stated otherwise, it is assumed that the trauma functions as given in Equations 1–3 remain constant across changes in profiles. Thus, for example, $f_s^*(\delta_s, \tau_s) = f_s(\delta_s, \tau_s)$, where $f_s(\delta_s, \tau_s)$ is given in Equation 1.

Proof. Suppose that there exists some other regular, feasible profile—say, ω''—that is more efficient than ω^*. Then it follows from Equation 13 that τ_r'' must satisfy the following inequality: $\tau_f \leq \tau_r'' < \tau_r^*$. By definition, c^* is the smallest integer that causes cumulative trauma to rise to α. Therefore, if ω'' is a feasible profile, $c'' > c^*$. By Equation 11, $c''[f_s(\delta_s, \tau_s) - f_r(\delta_r, \tau_r'')] = \alpha$. However, because we assume monotonicity of the recovery interval trauma function, this implies $\tau_r'' > \tau_r^*$, a contradiction.

For any given job, there are many different profiles that have a maximally efficient work cycle—that is, that have a number c of work cycles equal to c^* and a duration τ_r of the recovery interval equal to τ_r^*. Not all of these profiles will be equally productive. However, if there is no slack in the profile, then we have the profile for which we have been searching. Specifically,

Theorem A1: If ω^* is a regular, feasible profile, if it has the maximally efficient work cycle, and if there is no slack in the job, then ω^* is the most productive regular, feasible profile.

Proof. Rearranging Equation 14, since the slack $\tau_d = 0$, the duration of profile ω^* can be written as the sum:

$$\tau = p^*c^*(\tau_s + \tau_r^*) + \dot{\beta}. \qquad (A1)$$

Consider some other profile, ω', which has a less than maximally efficient work cycle. Suppose that the number of work periods is equal to $p^* - p'$ and the number of work cycles is equal to $c^* + c'$. Furthermore, suppose that the duration of the recovery interval is equal to $\tau_r' + \delta$, where δ must be positive be- 'se, by assumption, the work cycle effi- .ncy of profile ω' is less than that of profile ω^*.

In order for the profile ω' with the less efficient work cycle to be more productive, two conditions must be satisfied. First, profile ω' must be completed in a period equal to τ, or, rearranging Equation 14:

$$(p^* - p')(c^* + c')(\tau_s + \tau_r^* + \delta) + \tau_d' \div \dot{\beta} = \tau, \quad \tau_d', \dot{\beta} \geqslant 0. \qquad (A2)$$

Substituting for τ from Equation A1, we obtain:

$$(p^* - p')(c^* \div c')(\tau_s + \tau_r^* + \delta) + \tau_d' + \dot{\beta} = p^*c^*(\tau_s + \tau_r^*) \div \dot{\beta}.$$

This constraint can be rewritten as follows:

$$p^*c^*\delta - (p^*c' - p'c^* - p'c')(\tau_s + \tau_r^* + \delta) \div \tau_d' = 0.$$

Note that this constraint can be satisfied only if:

$$p^*c' - p'c^* - p'c' < 0.$$

or,

$$c'(p^* - p') < p'c^*.$$

Second, the number $W(\omega')$ of goods produced using profile ω' must be greater than the number $W(\omega^*)$ goods produced using profile ω^*. Substituting from Equation 6 and canceling, we obtain:

$$(p^* - p')(c^* \div c') > p^*c^*. \qquad (A3)$$

Note that this constraint can be rewritten as follows:

$$c'(p^* - p') > p'c^*.$$

Clearly Equations A2 and A3 impose incompatible constraints. Thus, if ω^* is a regular, feasible profile with the maximally efficient work cycle and if ω^* contains no slack, then there does not exist an alternative regular, feasible profile that is more productive.

Note that in some cases there may not exist a regular, feasible profile associated with a particular job. There are two possibilities. First, this would happen if the job were relatively short and the trauma increased very little from cycle to cycle, never approaching the critical level. Second, assuming that the critical level was reached, then too little time may be available during the succeeding rest break either to return cumulative trauma to zero or to spend the required time ($\dot{\beta}$) in the one rest break. In both cases the profile that minimizes repetitive motion disorders and maximizes productivity is easily identified. We leave this as an exercise for the reader.

APPENDIX B: THE OPTIMAL IRREGULAR PROFILE

In this appendix we prove that the irregular, feasible profile with a maximally efficient work cycle and no slack is more productive than are all other regular or irregular feasible profiles. To begin, we show that irregular profiles are at least as productive as regular profiles:

Lemma B1: The productivity of an irregular, *feasible profile will always be at least as great as the productivity of the regular, feasible profile from which it is derived.*

Proof. In order to substantiate this last claim, one needs only to consider the slack in regular and irregular profiles. Nothing is produced during the slack period of the regular profile. However, this slack can be filled by d, $0 \leq d \leq c$, work cycles in an irregular profile. Thus, as claimed, an irregular profile is at least as productive as a regular profile.

Given that it makes good sense to consider irregular profiles, the same question can be asked of irregular profiles that was asked of regular profiles. In particular, one can identify the irregular profile with the most efficient work cycle. This we do in the following lemma, where p^*, c^*, τ_r^*, and τ_b^* are defined here as they were for regular profiles:

Lemma B2: If ω^* is an irregular, feasible profile derived as above, then there exists no other irregular profile that has a more efficient work cycle than does ω^*.

Proof. The proof is similar to the one given for Lemma A1 and so need not be repeated here.

Again, as for regular profiles, there are a number of irregular profiles that have the maximally efficient work cycle. We are interested in only one of these profiles. Specifically,

Theorem B1: If ω^* is an irregular, feasible profile, if it has the maximally efficient work cycle, and if there is no slack in the job, then ω^* is the most productive irregular, feasible profile.

Proof. Rearranging Equation 16, given that the slack $\tau_d^* = 0$, the duration of profile ω^* can be written as the sum:

$$\tau = p^*c^*(\tau_s + \tau_r^*) - d^*(\tau_s + \tau_r^*) + \hat{\beta}. \quad \text{(B1)}$$

As with Theorem A1, two constraints must be satisfied by any irregular profile—say, ω'—that is less efficient and more productive than profile ω^*.

First, the time to complete profile ω' must be equal to the time to complete ω^*:

$$\begin{aligned}(p^* - p')(c^* + c')(\tau_s + \tau_r^* + \delta) &+ \\ d'(\tau_s + \tau_r^* + \delta) - \tau_d' + \hat{\beta} & \\ = p^*c^*(\tau_s + \tau_r^*) + d^*(\tau_s + \tau_r^*) + \hat{\beta}. & \end{aligned} \quad \text{(B2)}$$

Second, the productivity of profile ω' must be greater than that of profile ω^*:

$$(p^* - p')(c^* + c') + d' > p^*c^* + d^*. \quad \text{(B3)}$$

It is easy to show that these two constraints quickly lead to a contradiction. Thus there exists no irregular, feasible profile ω' that is more productive than ω^*.

APPENDIX C: QUASI-FEASIBLE PROFILES

In this appendix we identify the multiple-period, quasi-feasible profile that will reduce repetitive motion disorders at least as much as all other multiple-period, quasi-feasible profiles identical to some single-period, multiple-cycle quasi-feasible profile in all respects except for p', c', and τ_b'. To begin, we show that any multiple-period, quasi-feasible profile derived from a single-period, multiple-cycle, quasi-feasible profile will keep constant or reduce the incidence of repetitive motion disorders and leave unchanged the productivity:

Theorem C1: Let ω be a single-period, multiple-cycle, quasi-feasible profile ($p = 1, c > 1, d \geq 0$) and let ω' be a multiple-period profile ($p' > 1, c' \geq 1$) derived from the single-period profile by changing only p, c, and τ_b, where $\tau_b' = \tau_b/p'$. If ω' is quasi-feasible, then ω' will keep constant or decrease repetitive motion disorders and leave productivity unchanged if the time that it takes the trauma to increase to the critical level after the rest break using profile ω' is greater than the time it takes the trauma to decrease to zero after the work period ends.

Proof. To begin, consider the claim that repetitive motion disorders will remain constant or decrease. This claim will be true if Equations 17 and 18 are satisfied. By assumption, $p' > p$ and $c' < c$. By assumption $\Gamma_k(j) = 0$, $j \geq 0$, so this implies that the maximum cumulative trauma under the multiple-period profile, $g(\omega')$, is strictly less than the maximum cumulative trauma, $g(\omega)$, under the single-period profile (Equation 7). Thus the first condition is satisfied.

Now consider the second condition. Suppose some level γ of cumulative trauma is chosen which is greater than the maximum cumulative trauma $g(\omega')$ of the multiple-period profile and less than or equal to the maximum cumulative trauma $g(\omega)$ of the single-period profile. Then, because the trauma of the multiple-period profile never rises above γ but the trauma of the single-period file does, the second condition is satisfied. uppose instead that some level of trauma γ is chosen that is less than or equal to the maximum cumulative trauma $g(\omega')$ of the multiple-period profile. In the first work period of the multiple-period profile ω', the γ trauma time $t_\gamma(\omega')$ of this profile will be the same as the γ trauma time $t_\gamma(\omega)$ of the single-period profile ω. However, as soon as the first work period ends, cumulative trauma will drop below γ at some point in the rest break of the multiple-period profile ω', whereas the trauma will continue to rise above γ in the single-period profile ω. Thus for each succeeding work period of profile ω', the cumulative trauma in profile ω will be above γ for a time equal to $t(\gamma_1) + t(\gamma_2)$, where $t(\gamma_1)$ represents that time that it takes the cumulative trauma to reach the level γ (for both profiles ω and ω') and $t(\gamma_2)$ represents the time that elapses between the point at which γ is first reached and the end of the work period (i.e., the beginning of the rest break) in the multiple-period profile ω'. Set $t(\gamma_3)$ equal to the time that it takes the trauma using profile ω' after the work period ends to decrease to γ. Then, in profile ω the time $t_\gamma(\omega,1)$ spent above γ in one work period of ω' (c' cycles of ω and ω') is equal to the sum $t(\gamma_1) + t(\gamma_2)$. In profile ω', $t_\gamma(\omega',1) = t(\gamma_2) + t(\gamma_3)$. Therefore $t_\gamma(\omega,1) > t_\gamma(\omega',1)$ if $t(\gamma_1) > t(\gamma_3)$. Given trauma functions such as those found in the box factory, the last assumption would appear to be reasonable. For example, in the box factory $t(\gamma_1) = 410.7$ s and $t(\gamma_3) < 56.69$.

Next, consider the claim that productivity remains unchanged. By assumption, for regular profiles $p'[c'(\tau_s + \tau_r) + \tau_b'] = p[c(\tau_s + \tau_r) + \tau_b]$. Because $\tau_b' = \tau_b p'$ and $p = 1$, this implies $p'c' = pc$, which in turn implies that the productivity of the single- and multiple-period profiles are identical. The proof for irregular profiles is similar.

This completes the proof of Theorem C1. One can ask at this point whether or not there is a best or optimal profile when the current profile is a single-period, multiple-cycle, quasi-feasible profile, the cumulative trauma is not known, and the productivity is to remain unchanged. In this context, by *optimal profile* we mean the multiple-period, quasi-feasible profile that reduces the γ trauma time at least as much as any other profile. We can show how to identify this profile in certain limited situations:

Corollary C1: Let ω be a single-period, multiple-cycle, quasi-feasible profile and let ω' be a multiple-period profile derived from the single-period profile by changing p, c, and τ_b such that $\tau_b' = \tau_b/p'$ and c' is minimized. If ω' is quasi-feasible, then the multiple-period profile ω' will reduce repetitive motion disorders at least as much as will the single-period profile from which it is derived and any other profile ω'' derived in a like manner where $c'' \geq c'$.

Proof. A formal proof of this claim will be only sketched here; the details can easily be filled in by the reader. Briefly, by Theorem C1, the multiple-period profile ω' reduces repetitive motion disorders when compared with the single-period profile from which it is derived. Thus we need only to compare the multiple-period profile ω' with other multiple-period profiles derived in a like manner.

Two possibilities present themselves: either the maximum cumulative trauma of profile ω' is greater than α or it is less than or equal to α (we cannot tell which because, by assumption, α is unknown). Assume that it is greater than α. Then, because profile ω' minimizes the maximum cumulative trauma (i.e., because c' is the smallest integer that satisfies the relevant constraints), the maximum cumulative trauma of any other profile ω^{*} must be greater than that of profile ω'. Furthermore, by Theorem C1 the γ trauma time of profile ω' must be less than the γ trauma time of profile ω. Assume instead that the maximum cumulative trauma of profile ω' is less than or equal to α. Then profile ω' minimizes repetitive motion disorders. There may exist other profiles that also minimize such disorders, but they can produce no greater reduction than does profile ω'. This completes the sketch.

References

Andres, R. O., and Fisher, D. L. (1989, August). *Phase I study: Documentation of catching jobs in the finishing area of Federal Paper Products Co., Inc.* Amherst: University of Massachusetts, Department of Exercise Science.

Armstrong, T. J. (1986). Ergonomics and cumulative trauma disorders. *Hand Clinics, 2,* 553–565.

Armstrong, T. J., Castelli, W. A., Evans, F. G. and Diaz-Perez, R. (1984). Some histological changes in carpal tunnel contents and their biomechanical implications. *Journal of Occupational Medicine, 26,* 197–201.

Armstrong, T. J., Fine, L. J., Goldstein, S. A., Lifshitz, Y. R., and Silverstein, B. A. (1987). Ergonomics considerations in hand and wrist tendinitis. *Journal of Hand Surgery, 12A,* 830–837.

Armstrong, T. J., Foulke, J. A., Joseph, B. S., and Goldstein, S. A. (1982). Investigation of cumulative trauma disorders in a poultry processing plant. *American Industrial Hygiene Association Journal, 43,* 103–116.

Armstrong, T. J., Radwin, R. G., Hansen, D. J., and Kennedy, K. W. (1986). Repetitive trauma disorders: Job evaluation and design. *Human Factors, 28,* 325–336.

Arndt, R. (1987). Work pace, stress, and cumulative trauma disorders. *Journal of Hand Surgery, 12A,* 866–869.

Barnes, R. M. (1968). *Motion and time study.* New York: Wiley.

Bechtold, S. E., Janaro, R. E., and Sumners, D. L. (1984). Maximization of labor productivity through optimal rest-break schedules. *Management Science, 30,* 1442–1458.

Chao, E. Y., Opgrande, J. D., and Axmear, F. E. (1976). Three-dimensional force analysis of finger joints in selected isometric hand functions. *Journal of Biomechanics, 9,* 387–396.

Chu, B. M., and Blatz, P. J. (1972). Cumulative microdamage model to describe hysteresis of living tissue. *Annals of Biomedical Engineering, 1,* 204–211.

Damon, F. (1965). The use of biomechanics in manufacturing operations. *Western Electric Engineer, 9*(4).

Emanuel, J., Mills, S., and Bennet, J. (1980). In search of a better handle. In *Proceedings of the Symposium: Human Factors and Industrial Design in Consumer Products.* Medford, MA: Tufts University.

Gilbert, O. (1968). *A manager's guide to work study.* London: Wiley.

Gilbreth, F. B., and Gilbreth, L. M. (1953). Fatigue study. In Spriegel and Myers (Eds.), *The writings of Gilbreths* (pp. 303–340). Homewood, IL: Irwin.

Goldstein, S. A. (1981). *Biomechanical aspects of cumulative trauma to tendons and tendon sheaths.* Unpublished Ph.D. Thesis, University of Michigan, Ann Arbor.

Goldstein, S. A., Armstrong, T. J., Chaffin, D. B., and Matthews, L. S. (1987). Analysis of cumulative strain in tendons and tendon sheaths. *Journal of Biomechanics, 20,* 1–6.

Hazelton, F. T., Smidt, G. L., Flatt, A. E., and Stephens, R. I. (1975). The influence of wrist position on the force produced by the finger flexors. *Journal of Biomechanics, 8,* 301–306.

Janaro, R. E., and Bechtold, S. E. (1985). A study of the reduction of fatigue impact on productivity through optimal rest break scheduling. *Human Factors, 27,* 459–466.

Johnson, S., and Ogilvie, G. (1972). *Work analysis.* London: Butterworths.

Lee, F. S. (1974). *The human machine and industrial efficiency.* Easton, PA: Hive.

McFarland, R. A. (1971). Understanding fatigue in modern life. *Ergonomics, 4,* 1–10.

McKenzie, F., Storment, J., Van Hook, P., and Armstrong, T. J. (1985). *American Industrial Hygiene Association Journal, 46,* 674–678.

Rowe, L. M. (1987). The diagnosis of tendon and tendon sheath injuries. *Seminars in Occupational Medicine, 2,* 1–6.

Silverstein, B. A., Fine, L. J., and Armstrong, T. J. (1986). Carpal tunnel syndrome: Causes and a preventative strategy. *Seminars in Occupational Medicine, 1,* 213–220.

Silverstein, B. A., Fine, L. J., and Armstrong, T. J. (1987). Occupational factors and carpal tunnel syndrome. *American Journal of Industrial Medicine, 11,* 343–358.

Streimer, I. (1971). Considerations of energy investments as determinants of behavior. In Singleton, W. T., Fox J. G., and Whitfield, D. (Eds.), *Measurement of man at work* (pp. 35–59). London: Taylor & Francis.

Taylor, F. W. (1967). *The Principles of Scientific Management.* New York: W.W. Norton Company.

Tichauer, E. (1976, February). Biomechanics sustains occupational safety and health. *Industrial Engineering, 8,* 63–71.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☑ BLACK BORDERS

□ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☑ FADED TEXT OR DRAWING

□ BLURRED OR ILLEGIBLE TEXT OR DRAWING

□ SKEWED/SLANTED IMAGES

☑ COLOR OR BLACK AND WHITE PHOTOGRAPHS

□ GRAY SCALE DOCUMENTS

☑ LINES OR MARKS ON ORIGINAL DOCUMENT

□ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

□ OTHER: ______________________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Active Click: Tactile Feedback for Touch Panels

FUKUMOTO, Masaaki
fukumoto@mml.yrp.nttdocomo.co.jp

SUGIMURA, Toshiaki
sugi@mml.yrp.nttdocomo.co.jp

NTT DoCoMo Multimedia Labs.
3-5 Hikari-no-oka, Yokosuka-shi
Kanagawa-ken, 239-8536 JAPAN
Fax: +81 468 40 3788

ABSTRACT

"Active Click" is a new interface mechanism for addling tactile feedback to touch panels. A small actuator is attached to a body of PDA or the backside of a touch panel. The tactile feedback, created by driving the actuator with a short pulse, is perceived by the grasping hand or tapping finger-tip when the panel is tapped. Active Click is effective in improving the input speed of touch panel operation especially in noisy situations. Active click is also useful for large touch panel devices such as public information terminals or ATMs.

Keywords

touch panel, click, PDA, interface device

INTRODUCTION

Touch panel displays are commonly used for PDAs because they are suitable for graphical and direct operation, and display size can also be increased through the deletion of keyboard buttons. However, ordinary touch panel devices fail to achieve comfortable and accurate operation because they do not provide the *click feel* when tapping. In many cases, a short beep sound is used to simulating the tactile click. However, this mechanism has many problems; for example, input mode (tactile tapping) differs from output mode (audible beep), and the beep sound may not be heard in noisy situations such as outdoor use.

"UnMouse"[1] is one solution for adding tactile feedback to touch panels. It uses the whole touch panel as a large push switch, however, nimble operation is difficult because the inertia of the push switch is large. "CC Click"[2] places transparent switch mechanisms over the display panel. The movable parts are small enough to operate airily, but switch position and display design are restricted.

CONCEPT

"Active Click" is a new mechanism that allows any touch panel device to provide tactile feedback. An actuator (electric to vibration transducer) is attached to a body of the PDA or the backside of the touch panel (**Figure 1**). When tactile feedback is desired such as tapping action by finger-tip, a short pulse signal is supplied to the actuator for making the actuator vibrate. The vibration is conveyed to the grasping hand or tapping finger-tip.

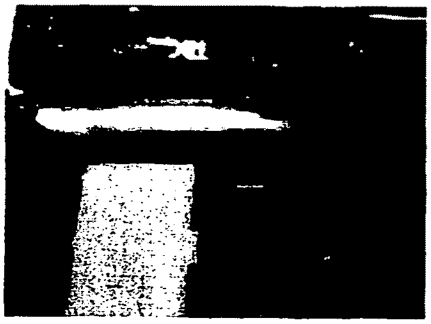


Figure 1: Example of actuator mounting
Putting actuator on the back panel of PDA.

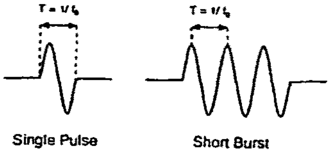
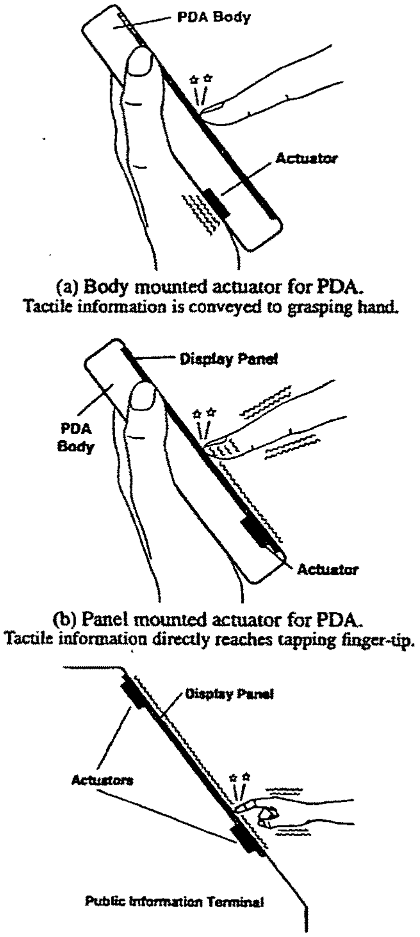


Figure 2: Example of driving signal for *click feel*.
f_0 = natural frequency of the actuator.

One commonly used vibrating device is a small motor that rotates an eccentrically weighted shaft. This approach cannot offer subtle control; only continuous vibration is well generated such as silent alarm of cellular phones. On the other hand, the structure of Active Click's actuator[1] is similar to a magnetic speaker, and delicate control is possible. For example, *click feel* is expressed by adding a single pulse or a short burst signal (**Figure 2**). Large vibration levels can be obtained using a small amplitude input signal if the pulse width or signal frequency equals the *natural frequency* of the actuator. In addition, the actuator can be used as *speaker* if the input signal contains audible frequency range. Therefore, both tactile feedback and audible output can be generated from just one device.

[1] TOKIN "MultiActor" [http://www.tokin.com]



(a) Body mounted actuator for PDA.
Tactile information is conveyed to grasping hand.

(b) Panel mounted actuator for PDA.
Tactile information directly reaches tapping finger-tip.

(c) Panel mounted actuator for large display.
Large or multiple actuators are used.

Figure 3: Mechanism of Active Click

VARIATION

Active Click has several variations according to the attaching position of the actuator. **Figure 3**-(a) shows the body mounted actuator for PDAs. Tactile feedback is conveyed to the grasping hand. Large feedback can be achieved with small amplitude input signal by attaching the actuator to the contact position of the grasping hand. This style is suitable for PDAs that requires low power operation.

Figure 3-(b) shows the panel mounted actuator for PDAs. The actuator is attached to the backside of the touch panel display. Tactile feedback is conveyed to the tapping fingertip directly, so this style can express most natural click feel. However, power consumption is higher since the whole touch panel must be vibrated.

Figure 3-(c) shows the panel mounted actuator for large displays such as public information terminals or ATMs. In this case, power consumption is insignificant therefore adequate tactile feedback can be realized by using large or multiple actuator(s).

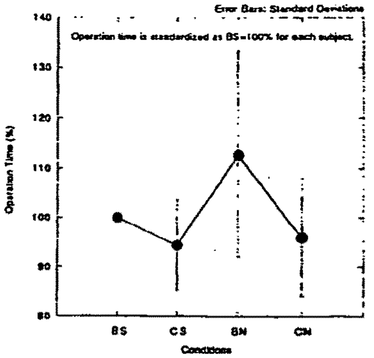


Figure 4: Time to perform calculating task.
Active Click is effective in reducing operation time.

EVALUATION

Simple calculation task[ii] was set to verify Active Click's performance. *Palm* compatible PDA[iii] and Palm-OS standard *Calculator* application were used in the evaluation. Both operation time and correct answer rate were measured using either Active Click feedback (body mounted actuator) or ordinary beep feedback. Two noise levels[iv] were set to check the influence of environmental noise. Four situations ([BS]: Beep&Silent, [CS]: Click&Silent, [BN]: Beep&Noisy and [CN]: Click&Noisy) were tested for each subject. Presentation order was randomized to eliminate any learning effect; the number of subjects was 10.

Difference in operation time is shown in **Figure 4**[v]. This graph shows that Active Click can reduce the operation time by about 5% (in silent situation) and 15% (in noisy situation)[vi].

CONCLUSION

Active Click can improve the usability of touch panels, especially in noisy situation, even though its mechanism is simple and easily implemented in ordinary PDAs. We are planning to create various tactile information by controlling the actuator's input signal, and to handle the feedback position by using plural actuators.

REFERENCES

[1] "UnMouse" by MicroTouch Systems Inc. [http://www.microtouch.com]
[2] "CC Click" by IDEC Corp. [http://www.idec.com]

[ii] One question: adding 5 pairs of 4-digit numbers. 20 questions in one task.

[iii] TRG Pro [http://www.trgnet.com]

[iv] 40dB(A) corresponds to a silent room, and 70dB(A) corresponds to an urban street. Pink noise was used as the signal source.

[v] Operation time was standardized as [BS] = 100% for each subject.

[vi] No significant difference was observed in the correct answer rate.

CHI 97 Electronic Publications: Papers

"Body Coupled FingeRing": Wireless Wearable Keyboard

FUKUMOTO, Masaaki
fukumoto@nttcvg.hil.ntt.co.jp
NTT Human Interface Laboratories
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa-ken, 239 JAPAN

TONOMURA, Yoshinobu
tonomura@nttvdt.hil.ntt.co.jp
NTT Human Interface Laboratories
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa-ken, 239 JAPAN

ABSTRACT

. really wearable input device "FingeRing" is developed for coming wearable PDAs. By attaching ring shaped sensors on each finger, many commands or characters can be input by finger-tip typing action. "FingeRing" can be used on any typing surface such as a knee or desk, so quick operation is realized in any situation while standing or walking. To improve wearability, a very small, ultra low power wireless transmitter is developed that uses the human body as part of an electric circuit. "Direct Coupling" method enables stable communication even when body contacts any grounded surface. A new symbol coding method that combines order and chord typing is also proposed, and useful typing patterns are chosen by typing speed evaluations. Expert users of musical keyboards can input 52 different symbols at speeds of over 200 symbols per minute by using the combination of FingeRing and the new coding method.

Keywords

wearable computer, PDA, interface device, input device, keyboard, PAN, BodyNet, FingeRing.

INTRODUCTION

The main reason for carrying a PDA is immediate access to information whenever desired. We want to carry information not a machine. Existing PDAs are much bigger and heavier than the information within them. However, PDAs become smaller and lighter given the progress in semiconductor technology. PDAs will be worn as accessories one of these days. The question is how to operate them.

Many concepts and prototypes of wearable computers have been proposed and partly developed. However, small interface devices suitable for wearing have not been well researched. Apple computer announced an image model of a wearable Macintosh[1]; a small wrist mounted trackball was used as its input device. The wearable computer project of MIT used a grip type chord keyboard[2], and some PDAs use miniaturized full or ten-digit keyboards. These interface devices depend on the physical size of the operative organ such as the human hand or finger. For instance, a keyboard whose key pitch is less than 14mm has lower input speed, higher fatigue levels and higher input error than the standard size keyboard[3]. Therefore, there is a trade-off between portability and usability. In other words, it is difficult to miniaturize ordinary interface devices without sacrificing their ease of operation. For the coming wearable PDAs, we think that specially designed interface devices that can be highly miniaturized are needed.

Glove or fingerstall style virtual keyboards which detect bending or typing action of fingers by sensors mounted at joint or tip of fingers, have been proposed[4] [5]. These systems seem suitable for wearable use because they do not require a key-top or key-pad. However, they cause trouble in daily life operation because they cover the finger-tip which has the highest tactile sensitivity or hand by sensor or glove. A virtual keyboard for a daily use wearable PDA should not cover the finger-tip or hand.

Considering these situations, we described an interface device which is suitable for wearable computers, and developed the FingeRing system[6][7], which is a "ring" shaped full-time wearable keyboard (Figure 1) . Users can input commands and characters by finger-tip typing actions on any support surface such as a knee or desk whenever desired. The small sensors do not cover finger-tips, so wearing such devices does not hinder our daily life.

Figure 1: Sensor part of FingeRing (wired version).
Detecting finger-tip typing actions by accelerometer.

FingeRing does not requires a particular space to be tapped by fingers such as a key-top or a key-pad, so usability does not worsen with miniaturization. However, the current FingeRing needs a direct electrical connection from the sensors on each finger to the symbol generator placed on a wrist. Even if the sensors and symbol generator are greatly miniaturized, the wire connection causes inconvenience in daily use. For example, the wires are frequently twisted and become tangled. Therefore, to realize truly wearable devices, we must establish wireless communication between the sensors and the symbol generator module. This paper compares several very short range, ultra low power wireless communication methods for their application to FingeRing. We choose the method called "Body Coupling" which uses the human body as an electric wire. We discuss the problems encountered in applying the body coupling method to FingeRing and propose "Direct Coupling" as a solution.

FingeRing is a kind of "chord" input keyboard, which makes symbols such as command or character through ɔmbinations of simultaneously typed fingers. Some chord keyboard systems have been proposed[10], but these systems tend to adopt useless (= hard to type) chord patterns to represent many symbols with one stroke typing actions. We propose a new coding method that combines of order and chord typing actions to increase the number of representable symbols without sacrificing input speed.

FINGERING

FingeRing is a prototype of a full-time wearable device for the input of commands and characters. A small accelerometer is worn on the base of each finger to detect the typing shocks generated by tapping the finger on any typing surface such as the thigh, knee or desk (called "finger-tip typing"). Commands and characters are generated from combinations of finger-tip typing actions. Each accelerometer is small and the finger-tip is not covered so they can be worn continuously in everyday life without trouble. In addition, no take-up action is needed for use, so immediate start of operation is possible.

Detection of finger-tip typing

Acceleration by finger-tip typing conveys from the finger tip to the sensor which is mounted on the base of typed finger. The acceleration is called "Self typing". However, the acceleration of the other fingers is also received by the same sensor; this is a type of cross-talk. Therefore, it is necessary to isolate the intended typing acceleration from the others. Figure 2 shows the frequency distribution of accelerometer output. Five subjects (154cm to 190cm in height) mounted accelerometers on the bases of their five fingers, and made finger-tip typing actions on a desk (reflects "Hard" typing surface) and on a thigh (reflects "Soft" surface). Figure 2 indicates that the self-typing and cross-talk signals have an amplitude difference of about 10 to 15 dB in the frequency area around 90Hz. Thus, a sharply carved ˡand Pass Filter (BPF), which passes only frequencies around 90Hz, can be used to eliminate the cross-talk regardless of typing surface stiffness. The example of BPF (24dB/Oct) setting is also shown in Figure 2. To be accurate, the filter property should different for each finger, but a simple resonance type BPF which has center frequency of 90Hz and Q (Quality factor: Sharpness of resonator) = 6 can be applied to all fingers.