July 2022

PHYSICALLY REALISTIC BOOK PAGE NAVIGATION

D. Shin

PHYSICALLY REALISTIC BOOK PAGE NAVIGATION

ABSTRACT

A computing device (e.g., a smartphone, a laptop computer, a tablet computer, a smartwatch, etc.) may use user gesture inputs to a graphical user interface (GUI) to determine a number of pages to move forward or back in paged digital content (e.g., electronic book, document file, digital article, word processing document, etc.). A machine learning model at the computing device may receive values such as contact location, pressure, and velocity of the user gesture to determine the number of pages to move forward or back.

Training data to construct the machine learning model may be obtained using physical books. One or more sensors, such as one or more cameras, may be focused on the physical books as test subjects turn the physical book's pages. Information such as page contact location, pressure, velocity, and the number of pages turned may be obtained from the sensor data. A machine learning algorithm can use this information as the ground truth to construct the machine learning model for the computing device.

DESCRIPTION

FIG. 1 below is a conceptual diagram illustrating a system 100 that includes a computing device 102 and a computing system 122. In accordance with various techniques described in this publication, computing device 102 may use a machine learning model 104 to determine the number of pages to move forward or back in paged digital content in a GUI of computing device 102. Machine learning model 104 may be trained on computing system 122 using physical books data, and later provided to and stored on computing device 102.

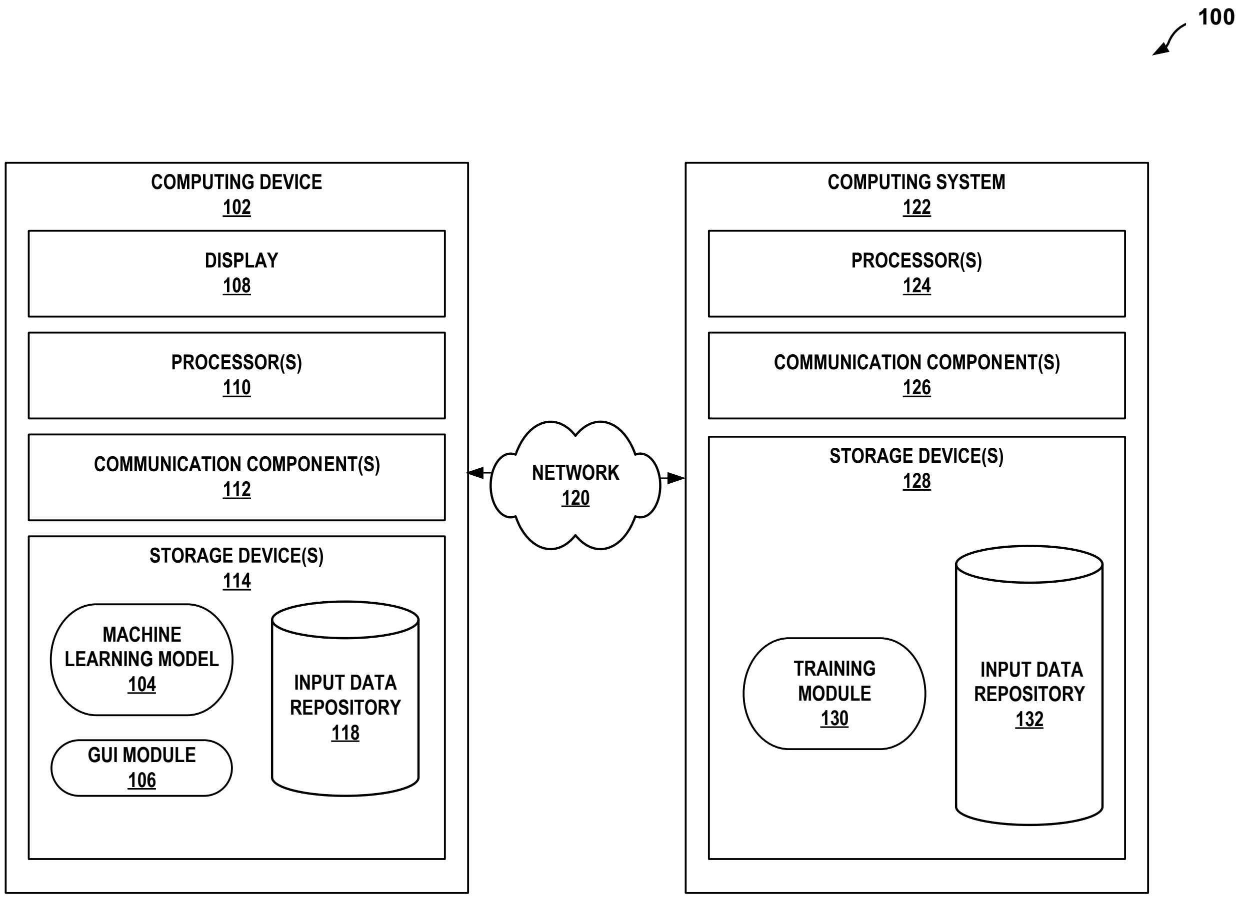

FIG. 1

As shown in FIG. 1, computing device 102 may include a display 108, one or more processors 110, one or more communication components 112 ("COMM components 112"), and one or more storage devices 114. Storage devices 114 may include machine learning model 104, a graphical user interface module 106 ("GUI module 106"), and an input data repository 118. Computing device 102 may be any mobile or non-mobile computing device, such as a cellular phone, a smartphone, a desktop computer, a laptop computer, a tablet computer, a portable gaming device, a portable media player, an e-book reader, a watch (including a so-called smartwatch), a gaming controller, and/or the like.

As shown in FIG. 1, computing system 122 may include one or more processors 124, one or more communication components 126, and one or more storage devices 128. Storage devices 128 may include training module 130 and an input data repository 132. Computing system 122 may be any suitable remote computing system, such as one or more desktop computers, laptop computers, mainframes, servers, cloud computing systems, virtual machines, etc., capable of sending and receiving information via a network 120. Computing system 122 may be a cloud computing system that provides one or more services via network 120. For example, computing system 122 may be a distributed computing system.

Display 108 of computing device 102 may be implemented using various display hardware. Display 108 may function as an input device using a presence-sensitive input component, such as a presence-sensitive screen or touch-sensitive screen, that receives tactile input from a user of computing device 102. The presence-sensitive input component may determine a contact location (e.g., an (x,y) coordinate) of the presence-sensitive input component at which the object was detected. The computing device 102 may also determine contact velocity (e.g., x and y velocity vectors) from changes in the contact location on display 108 over time. In one example, a pressure sensor in the display 108 may obtain the pressure information.

Computing device 102 may also determine the pressure information based on detected user contact with the display 108. For example, computing device 102 may use the contact area, contact shape and/or time of contact at the display 108 to determine the pressure information.

Processors 110 and processors 124 may implement functionality and/or execute instructions associated with computing device 102 and computing system 122, respectively. Examples of processors 110 and processors 124 may include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device. Machine learning model 104, and GUI module 106 may be operable by processors 110 to perform various actions, operations, or functions of computing device 102. Training module 130 may be operable by processor 124 to perform various actions, operations, or functions of computing system 122.

COMM components 112 and COMM components 126 may receive and transmit various types of information, such as input data stored in input data repository 118 and input data repository 132, over network 120. Network 120 may include a wide-area network such as the Internet, a local-area network (LAN), a personal area network (PAN) (e.g., Bluetooth®), an enterprise network, a wireless network, a cellular network, a telephony network, a Metropolitan area network (e.g., WIFI, WAN, WiMAX, etc.), one or more other types of networks, or a combination of two or more different types of networks (e.g., a combination of a cellular network and the Internet).

COMM components 112 and COMM components 126 may include wireless communication devices capable of transmitting and/or receiving communication signals using network 108, such as a cellular radio, a 3G radio, a 4G radio, a 5G radio, a Bluetooth® radio (or any other PAN radio), an NFC radio, or a WIFI radio (or any other WLAN radio). Additionally, or alternatively, COMM components 112 and COMM components 126 may include wired communication devices capable of transmitting and/or receiving communication signals via a direct link over a wired communication medium (e.g., a universal serial bus (“USB”) cable).

Storage devices 114 and storage devices 128 may include one or more computer-readable storage media. For example, storage devices 114 and storage devices 128 may be configured for long-term, as well as short-term storage of information, such as, e.g., instructions, data, or other information used by computing device 102 and computing system 122, respectively. Storage devices 114 and storage devices 128 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid-state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements, storage devices 114 and storage devices 128 may include one or more so-called "temporary" memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the storage devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), and/or the like. A user of computing device 102 may provide user input via display 108 to navigate with respect to GUIs of computing device 102.

Various aspects of the techniques described in this publication enable computing device 102 to use machine learning model 104 to determine the number of pages to move forward or back in the paged digital content., Computing device 102 may determine input data, such as location, velocity, and pressure data when a user gestures on display 108. Based on the input data, machine learning model 104 may determine a page delta. If the machine learning model 104 determines a non-zero page delta, computing device 102 may update the display of the paged digital content with the page delta so as to show a new page in the paged digital content. The display of the paged digital content may be updated using a page turning animation.

The computing device 102 may scale or modify the inputs such as the location, velocity, and pressure data before providing them to the machine learning model 104. For example, computing device 102 may scale locations from a page on a GUI of computing device 102 to location values expected by the machine learning model 104. The computing device 102 may associate one or more portions of the screen may with one or more sides of the paged digital content. One or more machine learning models constructed using physical books may be modified or analyzed to produce device-specific machine learning models or other control logic tailored to the specific computing device 102.

Computing system 122 may train the machine learning model 104. Training module 130 may train the machine learning model 104 on computing system 122 based on training data in input data repository 132. As described below, the data in the input data repository 118 may be obtained using physical books.

Training module 130 may train machine learning model 104 by optimizing an objective function. The objective function may represent a loss function that compares (e.g., determines a difference between) output data generated by the model from the training data and labels (e.g., ground-truth labels) associated with the training data. For example, the loss function may evaluate a sum or mean of squared differences between the output data and the labels. Training module 130 may train machine learning model 104 using supervised learning techniques. As described below, training module 130 may train machine learning model 104 on a training dataset obtained using physical books.

Computing device 102 may download machine learning model 104 (or, if computing device 102 already locally stores a version of machine learning model 104, update machine learning model 104 by downloading a more recent version of machine learning model 104) from computing system 122.

In addition to location information, the machine learning model 104 may use velocity and pressure to construct a more realistic page-turning experience. This more realistic page-turning experience may increase user engagement around the use and desirability of the digital paged content.

Machine learning model 104 may mimic page-turning behaviors that model how users read physical books. For example, when readers read physical books, they apply relatively intense pressure and relatively fast swiping movements to scan through the book quickly. For regular page-by-page reading, less force and slower swipes are typically used. Further, if a swipe is on the edge of the book, there is a higher probability of pages turning than if the swipe is near the middle of a page. Training the machine learning model 104 using location, velocity, and pressure information obtained from physical books may allow for such behaviors and more to be emulated on computing device 102.

In one example, the machine learning model 104 may model the relationship between Δ (a page delta or number of pages that get subtracted or added from the current page after swipe) and l (location), p (pressure), v (velocity) as described by a forward function:

$$\Delta = f\,(l, p, v)$$

This modeled function may allow a user navigating through paged digital content on computing device 102 to have a physically realistic experience.

To construct machine learning model 104, training module 130 may use a neural network such as a fully connected neural network. For the data generation, a trainer may use vision-based data collection methods to monitor test subjects turning pages of physical books to construct the ground truth. In this way, a series of mappings from the (l, p, v) triplets to Δ values may be generated from multiple test subjects. A moderately-sized deep neural network may fit and generalize pairs of (l, p, v) triplets to Δ values with as little as 1000 or fewer such pairs.

Fig. 2 is a diagram of an exemplary network architecture 200. Although Figure 2 shows a fully connected neural network, other Artificial Intelligence (AI) and/or Machine Learning (ML) architectures may also be used to create the machine learning model 104. Exemplary network architecture 200 is shown with an input layer size of five (two location coordinates, two velocity values, and one pressure value) and an output layer size of one (Δ value). In one example, the neural network may have a hidden layer size of around 32, and the depth of the neural network may be about 6. The training of the network may use an error function such as a continuous distance metric (e.g., mean squared error (MSE)).

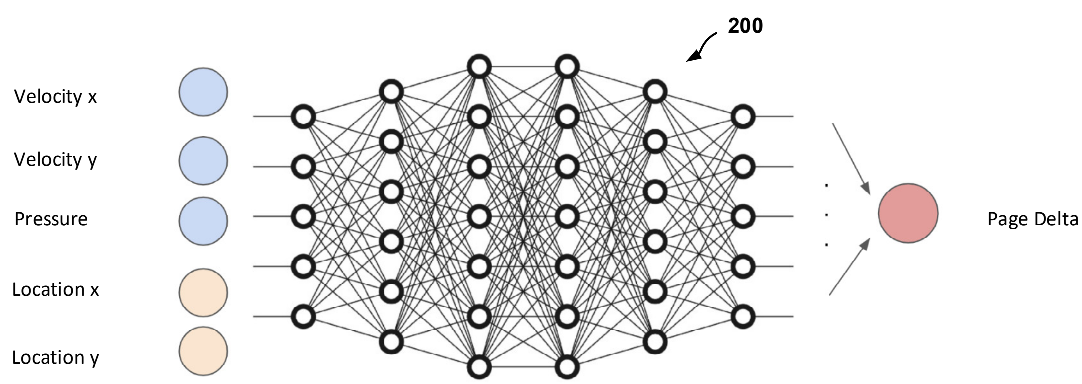

FIG. 2

In one example, the neural network may compute a non-integer output value for the page delta. This non-integer output may be converted to an integer for the final page delta (for example, using a simple staircase quantization function).

Fig. 3 is a conceptual diagram that shows the use of sensors, such as cameras, to collect the ground truth data to train and evaluate a neural network to construct the machine learning module 104.

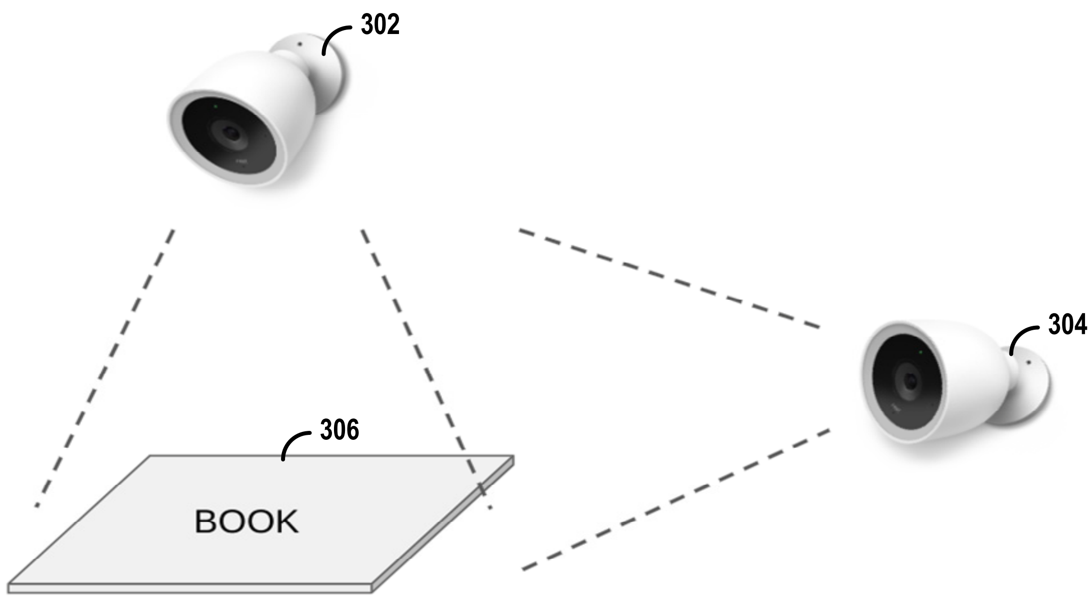

FIG. 3

A bird-eye view camera 302 may record the location and velocities of user gestures when the user is interacting with the physical book 306. Bird-eye view camera 302 may record the page number differences from the starting point to the endpoint (the Δ parameter or page delta).

Side-view camera 304 may determine pressures the user applies during a gesture. For example, different pressures may result in various distortions or bends in the pages that may be recorded with the side view camera 304. Thus, the computing system 122, or another computing system, may analyze images from side-view camera 304 to determine pressure values. Alternately, a force panel or other pressure sensor may produce an additional sensor input to help determine the pressure.

The computing system 122, or another computing system, may analyze images from side-view camera 304 to determine gesture types (e.g., book side pinching, palm swipes, single finger swipes, double finger swipes, surface page pinching, etc.). Training module 120 may use the determined gesture type as input to create the machine learning model 104. Alternately, training module 120 may construct separate machine learning models for each determined gesture type, and then computing device 102 may be loaded with these separate machine learning models. Computing device 102 may select a specific machine learning model at the computing device 102 using tactile input identifying the gesture type. Yet another machine learning model trained on the physical book data may determine the gesture type at the computing device 102.

In one example, the computing system 122 may produce more than one machine learning model. Machine learning models may be associated with different types of books (e.g., paperback, hardback, thick paper, thin paper, etc.). Computing system 122 may provide these multiple machine learning models to the computing device 102. Computing device 102 may then allow a user to select a specific machine learning model to provide different user experiences.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2011/0310005 A1. In another example, the techniques of this disclosure may be combined with the techniques described in PCT Publication WO 2018185585 A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2018/0307270 A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in “Force Gestures: Augmented

Touch Screen Gestures Using Normal and Tangential Force," published on ResearchGate, January 2011. DOI:10.1145/1979742.1979895.