Multimodal sensing-based camera applications
Miguel Bordallo López\textsuperscript{a}, Jari Hannuksela\textsuperscript{a}, Olli Silvén\textsuperscript{a} and Markku Vehviläinen\textsuperscript{b}
\textsuperscript{a}Machine Vision Group, University of Oulu, Oulu, Finland
\textsuperscript{b}Nokia Research Center, Tampere, Finland

ABSTRACT
The increased sensing and computing capabilities of mobile devices can provide for enhanced mobile user experience. Integrating the data from different sensors offers a way to improve application performance in camera-based applications. A key advantage of using cameras as an input modality is that it enables recognizing the context. Therefore, computer vision has been traditionally utilized in user interfaces to observe and automatically detect the user actions. The imaging applications can also make use of various sensors for improving the interactivity and the robustness of the system. In this context, two applications fusing the sensor data with the results obtained from video analysis have been implemented on a Nokia Nseries mobile device. The first solution is a real-time user interface that can be used for browsing large images. The solution enables the display to be controlled by the motion of the user’s hand using the built-in sensors as complementary information. The second application is a real-time panorama builder that uses the device’s accelerometers to improve the overall quality, providing also instructions during the capture. The experiments show that fusing the sensor data improves camera-based applications especially when the conditions are not optimal for approaches using camera data alone.

Keywords: Computer vision, data fusion, motion sensor, user interface, mobile device

1. INTRODUCTION
The increased sensing and computing capabilities of mobile devices can provide for enhanced mobile user experience. Integrating the data from different sensors such as cameras, ambiance-light sensors, gyroscopes, accelerometers and magnetometers offers a way to improve application performance or even create new user interaction concepts. In camera-based applications such as panorama building and high dynamic range imaging, the motion measured by accelerometers and gyroscopes enables a better interactivity and overall quality of the resulting images. Many of the current devices have also two built-in cameras, one for capturing high resolution photography, and the other for lower resolution video telephony. Even the most recent devices have not yet utilized these unique input capabilities enabled by cameras for purposes other than just image capture for viewing by humans.

Figure 1 shows an example of a modern mobile device: a Nokia N8 cellular phone. The phone includes a 3.5-inch, 640 by 360 pixel capacitive touch sensitive screen with tactile feedback. It also has a GPS sensor, a 12 Megapixel autofocus camera under with a dual LED flash, and a supplemental video call camera next to the display. Furthermore, it has multiple integrated sensors including an accelerometer, a magnetometer, a light sensor and a proximity sensor. The proximity sensor is clearly visible on the left of the earpiece while the video call camera and the light sensor that controls the screen brightness are on the right. The built-in accelerometer is primarily employed to turn the screen orientation.

In this paper, we demonstrate a set of novel application concepts that rely on built-in multiple sensors of handhelds for improving camera-based applications. Two interactive applications fusing the sensor data with the results obtained from video analysis have been implemented on a Nokia Nseries mobile device.

The first solution is a real-time user interface that can be used for browsing large images or documents such as maps on small screens with single hand operation. Based on video analysis, the solution enables the display to be controlled by the motion of the user’s hand. The motion sensor obtains the device orientation and it is used as complementary information for motion-based browsing. The system decides the motion input by fusing the measurements from the camera and accelerometers depending on the light conditions and image quality.

Further author information: Send correspondence to Miguel Bordallo López
E-mail: miguelbl@ee.oulu.fi, Telephone: +358 449170541
The second application is a real-time panorama builder that uses the mobile device’s accelerometers to increase the quality of source images by detecting motion blur and enhancing the user experience by providing instructions during the capture. In addition, the motion and device-orientation data guides the image registration process. The better robustness and interactivity of the system result in a significant quality improvement in the final panorama images.

The experiments described here, fuse the data obtained from accelerometers and image analysis to enhance the user interactivity and robustness of camera-based applications. The framework allows balancing several motion input sensors, improving the device movement estimates especially when the environmental conditions are not optimal for approaches using camera data alone.

2. MULTIMODAL USER INTERFACES

Cameras have traditionally been utilized in user interface research to build systems that observe and automatically sense and perceive the human users, including their identity, focus of attention, facial expression, gestures and movements. The most notable practical implementations such as Sony’s EyeToy and Microsoft’s Kinect, are related to gaming. For example, the EyeToy, a peripheral camera for Sony’s PlayStation game consoles has shown that computer vision technology is becoming feasible for consumer-grade applications. This device allows players to interact with games using simple motion estimation, color detection and also sound, through its built-in microphone. Respectively, the Kinect device includes a depth-sensing camera and a microphone array providing full-body 3-D motion capture, facial recognition, and voice recognition capabilities.

Despite the significant progress made, vision-based interfaces often require customized hardware and they work only in more or less restricted environments such as in these examples in which the sensors are stationary. However, to some extent the interaction needs in hand-held communication devices in mobile usage are similar.

It has been shown that different sensors provide viable alternatives to conventional interaction in portable devices. For example, tilting interfaces can be implemented with gyroscopes and accelerometers. Using both tilt and buttons, the device itself is used as input for navigating menus and maps. During the operation, only one hand is required for manipulation.

Nintendo Wii is an example of fitting together application interactivity and sensor functionality: the limitations of the three-axis accelerometer are cleverly hidden from the user by the characteristics of each game. Apple’s products make use of the multimodal user interaction technology in different ways. In the iPhone, users
are allowed to zoom in and out by performing multiple fingers gestures on the touch screen. In addition, a proximity sensor shuts off the display in certain situations to save battery power, and an accelerometer senses the orientation of the phone and changes the screen accordingly. In a category of their own are devices that employ a detachable stylus by which interaction is done by tapping the touch screen to activate buttons or menu choices.

The first mobile phone application utilizing the camera as a sensor was introduced by Siemens in 2003. An augmented reality game called Mozzies, developed for their SX1 cell phone, featured a real-time background image where the projection of several synthetic flying mosquitoes should be shot down by moving the phone around and clicking at the right moment. When the user is executing some action, the motion of the phone is recorded using a simple optical flow technique. Today, there is an increasing interest among researchers towards mobile user interfaces and a number of solutions have been proposed in the literature.

Much of the previous work on vision-based user interfaces with mobile phones has utilized measured motion information directly for controlling purposes. For instance, Möhring et al. presented a tracking system for augmented reality on a mobile phone to estimate 3-D camera pose using special color-coded markers. An alternative to markers is to estimate motion between successive image frames with similar methods to those commonly used in video coding. For example, Rohs divided incoming frames into the fixed number of blocks and then determined the relative x, y, and rotational motion using a simple block matching technique.

A recent and generally interesting direction for mobile interaction is to combine information from several different sensors. In their feasibility study, Hwang et al. combined forward and backward movement and rotation around the Y axis data from camera-based motion tracking, and tilts about the X and Z axis from the 3-axis accelerometer. In addition, a technique to couple wide area, absolute, and low resolution global data from a GPS receiver with local tracking using feature-based motion estimation was presented by DiVerdi.

Recently, the motion input has also been applied to more advanced indirect interaction such as sign recognition. This increases the flexibility of the control system as the abstract signs can be used to represent any command, such as controls for a music player. A number of authors have examined the possibility of using phone motion to draw alpha-numeric characters. Liu et al. showed examples of Latin and Chinese characters drawn using the ego-motion of a mobile device, although these characters are not recognized or used for control. Kratz and Ballagas propose using a simple set of motions to interact with the external environment through the mobile device. In their case, there are four symbols, consisting of a three-sided square in four different orientations. Due to the small size of the symbol set, they report good performance with no user training.

Mobile phones equipped with a camera can also be used for interactive computational photography. Adams et al. presented an online system for building 2D panoramas. They used viewfinder images for triggering the camera whenever it is pointed at previously uncaptured part of the scene. Ha et al. also introduced the auto shot interface to guide mosaic creation using device motion estimation.

The approaches described above have utilized camera motion estimation to improve user interaction in mobile devices. In our contribution, we describe an intuitive user interaction framework which fuses camera and sensor data to improve the motion estimation results. Two applications are described as examples which can benefit from the enriched robustness and interactivity.

3. SENSOR DATA FUSION SYSTEM

The proposed system determines motion of the device using both image and accelerometer data. Camera-based motion estimation has a number of apparent performance limitations, caused by lighting conditions and fast movements. Similarly, pure accelerometer measurements lead to errors that increasingly grow with time. These two sensing modalities can be combined to compensate for each other’s limitations and therefore provide for a more robust device motion estimates.

In vision-based user interfaces, special attention needs to be paid to the design of a proper lighting system. One possibility is to use special infrared LEDs for illuminating the user’s face. Energy consumption is, of course, the main limitation of such designs. Therefore, if cameras become standard user interface components in mobile devices, energy efficiency requires that the bulk of computing is carried out using hardware acceleration.
Another possibility to solve this issue is to use an adaptation method to switch to another input source. Several motion sensors are included in most of the newest mobile devices. Linear accelerometers can capture data at a very high rate, increasing the system’s robustness when the user input consists on fast movements or the camera image does not present a sufficient amount of valid features.

Combining inertial sensors with camera motion estimation has been active research area in the past decade. In many cases, the fusion is done using an Kalman filtering (KF) framework. For example, Klein et al. used a KF-based method for fusing rate gyroscope information with model-based tracking. Jiang et al. proposed a real-time system for outdoor augmented reality integrating gyroscope data and natural line features from images. In the proposed system, KF is used to fuse measurements from accelerometers and camera motion estimation.

The main features of the Kalman filter are modeling the random process under consideration using a system model and recursive processing of the noisy measurement data. A filter is optimal if the dynamic model is linear, the measurement model is linear, and the noise processes involved are Gaussian distributed. Furthermore, the recursive nature of the algorithm makes it convenient to use in real-time systems.

3.1 System model

In sensor fusion, the objective is to recursively estimate the state in the dynamic model. We model the device motion using following model

\[ x_{k+1} = \Phi_k x_k + \Gamma_k \varepsilon_k, \]  

where the parameters to be estimated are presented by the state vector \( x_k \) at time instant \( k \), and \( \Phi_k \) is the state transition matrix. The state transition matrix relates the state at time instant \( k \) to the state at time instant \( k+1 \). \( \Gamma_k \varepsilon_k \) models the uncertainty of the motion model. The process noise \( \varepsilon_k \) is assumed to be Gaussian distributed with an expected value \( E\{\varepsilon_k\} = 0 \) and the covariance matrix \( Q_k = E\{\varepsilon_k \varepsilon_k^T\} \). \( \Gamma_k \) is the process noise transition matrix.

The state vector \( x_k \) consists of the position \( (x_k, y_k, z_k) \), velocities \( (\dot{x}_k, \dot{y}_k, \dot{z}_k) \) and accelerations \( (\ddot{x}_k, \ddot{y}_k, \ddot{z}_k) \) of the device at time instant \( k \). It is defined as follows

\[ x_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, \ddot{x}_k, \ddot{y}_k, \ddot{z}_k]^T. \]

In the beginning, the elements of the state vector are set to zero. The time step between two successive images is normalized to 1. We approximate the variances of the process noise from the maximum accelerations allowed.

3.2 Measurements

The measurement model is needed to relate the state to the 2-D image motion and accelerometer observations. In our case, the model is defined as

\[ w_k = H x_k + \eta_k, \]  

where \( H \) is the observation matrix. The measurement noise \( \eta_k \) models uncertainty in the motion measurements and it is assumed to be Gaussian distributed with an expected value \( E\{\eta_k\} = 0 \) and the covariance matrix \( R_k = E\{\eta_k \eta_k^T\} \). The noise covariance can be adjusted based on lighting conditions. For example, in dark lighting condition the uncertainty of image motion estimation is larger.

In actual measurements, the ego-motion of the device is estimated from 2-D image motion measured between two successive frames. We propose an approach where a sparse set of feature blocks is first selected from one image and then the displacements are determined. We pay attention especially to the confidence analysis of block matching since the results of this process can be utilized in further analysis.

First, the previous frame is split to 16 subregions, and one 8 by 8 pixel block is selected from each region based on analysis of spatial gradients. Displacement of selected blocks is estimated using zero mean sum of squared differences (ZSSD) criterion which is applied over some range of candidate displacements (e.g. 16 pixels). Refinement to subpixel precision is done in the neighborhood of the displacement minimizing the ZSSD measure using fitting of the second order polynomials. The ZSSD values are also used to analyze uncertainty
information related to the local displacement estimate. Figure 2 shows an example result for an image sequence. This information is used in RANSAC style outlier analysis which provides reliable motion features for 2-D motion estimation.

For our purpose, a four-parameter affine model is sufficient for approximating motion between frames as it can represent 2-D motion consisting of x-y translation ($\theta_1, \theta_2$), rotation around the z-axis ($\phi$), and scaling $s$. Scaling $s$ is related to the translation in the z-direction, $\Delta Z$, as $(s - 1) = \Delta Z/Z$, where $Z$ is the scene depth.

In our system, images are captured at a rate of 15 fps and the motion is estimated at the same rate. Therefore, we acquire accelerometer data also at a rate of 15 fps. This leads to an easier implementation of the measurement fusion. Figure 3 shows example data for x-, y-, and z-acceleration for the same sequence presented in Figure 2.

3.3 Hybrid tracker

In order to obtain smoother output as a result of motion estimation, Kalman filtering is applied for implementing sensor fusion. The Kalman filter algorithm estimates the motion recursively, repeating two stages: prediction and correction. At the first stage, the state at the next time instant is predicted based on the previous state estimate and the dynamical model. In the correction stage, the predicted state is adjusted by using the measurements of the image motion and device acceleration.

4. APPLICATION CASE IMPLEMENTATIONS

The previous section dealt with the method to fuse measurements from camera and accelerometers in order to obtain more robust motion estimates. This chapter describes the novel possibilities that this multimodal method provides: input to mobile user interfaces and applications. Overall, the applications presented show that mobile user interfaces benefit from the enriched interaction modalities.
The first solution is a real-time motion-based user interface that can be used for browsing large images or documents such as maps on small screens. In this interface, the motion information of the device itself, the face, or the hand of the user is extracted from the image sequence. The second solution presented is a real-time panorama builder that composes a mosaic image from individual video frames using the motion information obtained by both the camera and the device accelerometers.

We have implemented our method using only fixed-point arithmetic due to the lack of a fast and energy efficient floating-point unit in most current devices. The use of integer operations in the inner loops guarantees high performance. The solution can also take advantage of the hardware acceleration used with other video processing applications. Acceleration hardware is designed to support the block-based and pixel-level processing tasks that are not efficiently handled by the CPU architecture. Typically, such hardware contains highly optimized motion estimation instructions on blocks from 16 by 16 to 4 by 4 pixels, which are also the usual sizes for the blocks in our method. The solutions can also take advantage of the acceleration hardware by using the graphic processing unit to perform several processing tasks.

4.1 Application case 1: Motion-based image browser

Navigating large information spaces can be disorienting even on a large screen. In mobile devices with a small screen, the user often encounters situations where the content that is needed for display exceeds what can be shown on the screen. For example, large digital images are becoming commonplace due to the increasing availability of high resolution imaging and map navigation applications.

A viable alternative for improving interaction capabilities consists on spatially aware displays. The solution is to provide a window to a larger virtual workspace where the user can access more information by moving the device around. In a touch screen-based approach, the user’s hand or finger can partially obstruct the view and compromise the eventual perception of the displayed augmented information. On the other hand, with motion input, the user can operate the phone through a series of movements whilst holding the device with a single hand. During these movements, the motion is extracted from the image sequence captured by the frontal camera, with the user being in its field of view.

As an application example, the solution has been implemented on Nokia N-series mobile phones, allowing the user to browse large image documents on small screens as shown in Figure 4. In this solution, the camera is pointing towards the user’s face, which guarantees that there are always good features available in the view.

Figure 4. Motion-based user interface estimates the motion of the device relative to the user or the scene enabling browsing and zooming functionalities.
In the solution, only a small part of the high resolution image is visible at a time and the estimated motion is used as input. For instance, the lateral movement upwards scrolls the focus towards the upper part of the display, and back and forth motion is interpreted as zooming in and out. The rotation component is utilized to change the orientation of the display. In practice, the user can also tilt the device in order to navigate over the display, which is a more natural and convenient way of controlling the device. Compared to the use of HW accelerometers or computer vision alone, the multi-sensing approach allows a more robust way of controlling zooming effects and adapts better to cases where the user is moving or walking.

The multimodal motion estimation approach used in this application was already introduced in the previous section. To improve the robustness of the system, the ambient-light sensor present on the device determines the lighting conditions of the environment. In case the illumination is insufficient, the night-mode is turned on and the motion estimation model assigns increases the value of trust of the accelerometer measurements, decreasing the value of the features extracted from the camera frames.

In the experiments, the user made some typical controlling movements. In these tests, there was no ground truth available, but the trajectories followed the movements that the user made with reasonable accuracy. A frame rate of 15 fps was used on a Nokia N900 device.

4.2 Application case 2: Motion sensor assisted panorama imaging

Our panorama building solution analyzes each individual VGA video frame for motion and moving objects, quantifies the quality of each frame, and stitches up to 360 degree panoramas from the best available images. We have developed a method where the devices are able to stitch images in real time, obtaining a result image that is growing with the frame acquisition. Fig. 5 shows the four basic blocks of a panorama building application.

---

**REGISTRATION**

![Registration blocks](image)

**CORRECTION**

![Correction blocks](image)

**SELECTION**

![Selection blocks](image)

**BLENDING**

![Blending blocks](image)

---

Figure 5. The four steps of a panorama building application.

In the panorama capturing procedure, in order to get a final panorama image, the user focuses the camera to the desired starting point of the mosaic. The camera starts turning around up to 360 degrees and a sequence of images starts to be captured. Each image is then individually processed to estimate the camera motion. The user is guided with instructions to adjust the speed and direction of the capture. The blurriness of each picture
is measured and moving objects are identified. Based on the quality of each individual frame, a selection process takes place. Image quality feedback is also displayed providing for user interactivity.

The idea of selection is to consider only good quality frames for creating the best possible output. In the selection process, each frame is either accepted or discarded based on the results obtained from video analysis and the accelerometer measurements. For every selected frame, if a moving object is present and it fits the sub image, the image is blended drawing a seam that is outside the boundaries of the object. If only a partial object is present, the part of the frame without the object is the one blended.

Image registration relies on the sensor fusion method described in the previous chapter. The sensor fusing method offers an accurate description of the camera movement with increased robustness against poor light conditions. Only a fixed square template on the central part of each frame, where the image quality is better, is used as an input image for the subsystem registration subsystem.

To perform motion detection, the difference between the current frame and the previous frame is computed. The result is a two-dimensional matrix that covers the overlapping area of the two frames. Then, this matrix is low-pass filtered to remove noise and is thresholded against a fixed value to produce a binary motion map. If the binary image contains a sufficient amount of pixels that are classified as motion, the dimensions of the assumed moving object are determined statistically. First, the center point of the object is approximated by computing the average coordinates of all moving pixels. Second, the standard deviation of coordinates is used to approximate the dimensions of the object.

In order to determine the quality of each frame, the amount of motion blur in the frame is computed with summed derivatives. The method estimates the image’s sharpness by summing together the derivates of each row and each column of the overlapping part. Blur calculation produces one single number that expresses the amount of high-frequency detail in the image. The value is sensible if it is used to compare images: if a certain image Ia acquires a higher result than image Ib, it means that Ia has more high-frequency detail than image Ib (implying that both images depict approximately the same scene). Usually this means that Ia is sharper than image Ib, but in some occasions the difference in the image content distorts the result. To add robustness to the selection system, the accelerometer measurements are integrated with the blur in a scoring system. A simple measure of the involuntary tilting or shaking is done by calculating the average of the last motion vectors provided by the accelerometers and subtracting the result from the current vector. The result of this operation is then thresholded to determine if a frame is too blurry.

Frame selection is done based on the score of the frame quality, calculated with the values of rotation, change of scale, involuntary tilting and the motion detection process. High values of tilting or motion between frames mean high probability of registration errors or bad image quality. Among the set of images that present an overlap with the previous blended frame, only the best frame is selected, while the others are discarded. The frame blending is done with feathering method, where a linear or Gaussian function gradually merges one frame to the next by changing the frames weight.

5. CONCLUSIONS

The objective of the presented framework and application scenarios has been to experimentally evaluate the value of the integration of camera and sensor data on a mobile platform for interactivity purposes. Our target has been in increasing the robustness of vision-based user interfaces. The motivation is a practical one: bad illumination conditions, direct light or reflections make the camera unreliable as an input method since in low-light conditions, the image can be too noisy to perform any image analysis. The presented framework fuses image analysis with data from motion sensors, improving the user interactivity and reliability of camera-based applications, especially when the environmental conditions are not optimal for approaches using camera data alone. The fusion of the data also decreases the number of operations that are needed for an image-based motion estimation, improving the energy efficiency of the system.

Other than the accelerometer data, the framework proposed allows the integration of data from other types of sensors that describe the device’s motion, such as gyroscopes or magnetometers. Other sensors present in the device such as ambient-light or proximity sensors, can be used to adapt the fusion process, evaluating the conditions and adjusting a proper balance for the contribution of each sensor to the final results.
We believe the current demonstrations represent a more general class applications that employ multimodal sensory information, and may provide for ingredients for novel device designs. Motion estimation can be identified as a potential platform level service to be offered via the multimedia API. It plays key roles in the demonstrated applications, and it is most likely to be employed in many others.

REFERENCES