STEREO CAPTURE UNIT FOR REAL-TIME COMPUTER VISION, FEATURING HARDWARE-ACCELERATED DIGITAL FILTER DESIGN

Iztok Kramberger, Zdravko Kačič

University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

Key words: computer vision, image processing, color filtering, spatial filtering, color space, skin color, user interface, interaction, hand tracking, head tracking, gesture recognition, skin formation, bit mask

Abstract: Real-time processing plays an important role in the achievement of more natural and physically intuitive ways for user-machine interaction. The main scope of this article is the development of a computer vision stereo capture unit that would fulfill more natural user-machine interaction within contemporary mobile telecommunication solutions. One of the main problems with current mobile devices is the restricted size of the showing area and the data input methods as a manner of device navigation. One of the most convenient ways to solve this problem is the use of gesture recognition within virtual or augmented reality applications.

In such cases the skin color can be a very comprehensive feature and so color tracking has been used to achieve the required goal. We have developed a method of color and spatial filtering for the purpose of indicating skin features and constituting skin formations within the level of the binary mask stream. A new stereo capture unit hardware structure with an accelerated digital nonlinear parametrical filter is shown. This features real-time searching for skin colored regions in a two-dimensional image stream where the parametrical design of the color filter also offers color detection in a comprehensive way. This parametrical scheme for the color filter enables automatic simultaneous adaptation of parameters due to changes in scene luminance and automatic adaptation possibilities of the color filter to the different types of skin.

Enota za zajemanje stereo slike za delovanje v realnem času s strojno izvedenim digitalnim filtrom

Ključne besede: računalniški vid, procesiranje slike, barvno filtriranje, prostorsko filtriranje, barvni prostor, barva kože, uporabniški vmesnik, interakcija, sledenje roke, sledenje glave, prepoznavanje kretenj, kožna formacija, bitna maska

Izvleček: Pomembno vlogo za čim bolj intuitiven način interakcije med uporabnikom in strojem ima procesiranje v realnem času. Članek prikazuje razvoj na računalniškem vidu zasnovanega vmesnika za zajemanje stereoskopskega video signala, ki omogoča poglobljeno interakcijo med uporabnikom in strojem znotraj aplikacij sodobnih telekomunikacijskih tehnologij. Glavni problem sodobnih mobilnih telekomunikacijskih tehnologij predstavlja omejena velikost prikazovalnega polja in načina vnosa podatkov oziroma navigacije same naprave. Eden izmed najprikladnejših načinov reševanja tega problema je uporaba razpoznavanja kretenj znotraj aplikacij navidezne in razširjene resničnosti.

Predstavljena je metoda barvnega sledenja, kjer smo razvili pristop barvnega in prostorskega filtriranja za označevanje kožnih značnic in tvorbe kožnih formacij na nivoju bitnih mask. Prikazana je struktura enote za zajemanje stereoskopskega video signala z uporabo strojnega pospeševanja v obliki digitalnega nelinearnega parametričnega filtra, ki omogoča iskanje kožno obarvanih regij v dvodimenzionalnem slikovnem zaporedju v realnem času. Parametrična zasnova predstavljenega pristopa omogoča samodejno sprotno prilagajanje na spremembe osvetlitve v sceni, kjer obstaja tudi možnost samodejne adaptacije na različne tipe kože.

1. Introduction

Most common intelligent environments suffer from the lack of natural and intuitive interactive devices. This can be especially felt within modern mobile telecommunication devices such as cellular phones, digital personal assistants, tablet computers and similar. The main problems for most of these mobile telecommunication devices that are becoming more and more intelligent and feature even more attractive services, is the lack of display size and interaction methods like data input or simply navigation and control. As a result conventional interaction devices such as the mouse and keyboard turn out to be unsuitable, which motivates the development of a vision-based tactile interface.

It can be seen on the basis of the human machine interaction that most communication is based on dynamic happening as the vision-based interface creates new content for each subject movement or gesture regardless of whether the movement or gesture was an expression of desired activity, selection, manipulation or just navigation. Dynamic happening within the captured scene is closely connected with any motion within the captured image sequence. When no subject movement is present at a given moment then the users have no requests to express themselves at the time.

It is reasonable to define user area for interaction within all three spatial dimensions because humans have a highly developed feeling of space. This enables the user more

natural and intuitive ways of communication. The user area for communication is defined as the area where the vision-based system is capable of tracking the movements and gesture of the navigational subject.

The basic idea of this work was to design an interface system that would enable tactile interaction within virtual or augmented reality user environments. As a control subject of tactile communication the user hand with stretched forefinger has been introduced, where the finger-tip represents the reference or navigation point of the interface.

From the beginning of the system design we had in mind that the interface should be able to gain real-time image processing, as this represents the basis for intuitive and more natural interaction with the machine. At this point the idea of interface usage was the ability to adapt the system into contemporary mobile telecommunication technology because the main problem lies within insufficient display size and user interaction methods. However, this is not the only problem concerning mobile technology because it suffers from available processing power and energy consumption. Real-time processing therefore was not the only concern to be considered although lower available processing power and data bandwidth was expected.

2. Stereo capture unit

In order to gain more natural and human-like interaction within the interface, a computer vision has been included that features the defining spatial position in all three dimensions. The use of stereoscopic vision requires two spatially and parametrically aligned image sensors shifted side by side, for which two analogue cameras have been used, as they offer an effective way of achieving the desired goal.

A stereo capture unit presents a special hardware design that can be used in fusion with digital signal processors from Texas Instruments. The testing hardware release was build for floating-point *DSP TMS320C6711* /19, 20/. This system design enables the capturing and preprocessing of two independent analogue video streams in real time. In addition stereoscopic vision requires the simultaneous capturing of two image sequences, thus the interface system should utilize two separate capturing units. As a matter of fact this approach would require more data bandwidth. Instead of utilizing two capturing units, a time-sequential video system is purposed where only one capturing unit is required. To achieve the stereoscopic cue, left and right images are spatially superimposed and temporally interleaved within one time-sequential image stream. It is reasonable to suppose that this approach lowers the quality of service, as the number of frames for a particular image sensor is halved but the interaction refresh rate still satisfies an adequate level of immersive communication. However, the drop in refresh rate is not the only concern of a time-sequential video system, as a time delay appears between the pair of captured stereo images. Consequently this time delay becomes a motion parallax problem when motion is present within the captured scene.

2.1. Skin color

Color is one of the most expressive visual features. Considerable work has been done on designing efficient descriptors for these features covering many applications. The color histogram is one of the most frequently used color descriptors that characterize the color distribution of an image. Common color description is based on dominant color, variable color and color structure within a color scheme.

Skin color has been chosen as the main feature in two-dimensional image, since this is the primary feature of the navigational subject. We have designed a hardware accelerated preprocessing stage on the level of programmable logic because of the pretension of color in spatial filtering for real-time. In several applications we encountered previous spatial filtering (low-pass filters) as it gives better results within the color segmentation stage. This requires an appropriate amount of free memory to store parts of the colored image. In our case the first stage of filtering is color feature extraction as a way of skin feature labeling, where in the next stage (spatial filtering) these features are spatially linked to skin formations, and noise singularities are removed. This approach for spatial filtering requires less memory requirements as the skin features exist within the binary mask, and gives adequate uniform color segmentation results. Color filter development has been accomplished through an appropriate mathematical model /34/ that issues the statistically acquired skin-color description in the *HSV* color space.

$$p(k)=\begin{cases}1, & h_{\min}\geq H(k)\leq h_{\max}, S(k)\geq s_{\max}, V(k)\geq v_{\max}\\ 0, & otherwise\end{cases} \quad (2.1)$$

The given distribution has been equipped with appropriate parameters that feature adaptation to different luminosity conditions within the captured scene, and adaptation to different skin types.

2.2. Motion detection

Motion detection within the motion analysis has been introduced using deterministic algorithms thus the processing time does not depend on the complexity of the captured scene, respectively the number of skin formations in the input image sequence. The motion detection core is based on a logical operator that performs its operation on three sequential binary masks. Continuous memory is needed for the storage of binary masks and so the motion detector algorithm is implemented within the digital signal processor.

$$LO=\left\{\left[m_i(T),m_i(T-1),m_i(T-2)\right]\mapsto mm_i(T)\right\} \quad (2.2)$$

The logical operator *LO* offers a fast and effective way of processing where its results are presented as a dynamic binary mask. The dynamic binary mask elements mm_i are those skin features m_i that are in motion. As in other similar methods for motion extraction using color filtering without any additional features like image edges, the given motion detector is liable to difficulties due to the mutual occlusion of individual skin formations that are projections of static or dynamic skin-colored subjects, and objects within the captured scene.

3. Hardware design for real-time processing

Appropriate hardware design has been suggested in order to achieve real-time processing of the given algorithms. The given hardware architecture features hardware acceleration using programmable logic that is implemented within the dynamically programmable *FPGA* circuits. The system's data path is divided into several processing stages.

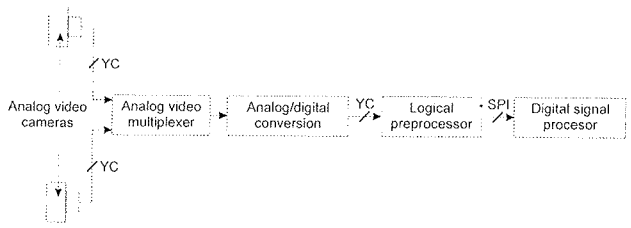

Figure 1: Hardware design for computer vision interface.

The first stage of the data path begins at the analog video multiplexer which combines two analog video streams from the employed analog cameras. Cameras are mounted in parallel configuration where the distance between them coincide with the average distance between human eyes. The parallel camera setup is advantageous over the toed-in approach since there is no vertical disparity introduced and the governing disparity equation is straightforward. However, the common field of view between the left and right acquired images becomes small and the correspondence problem becomes impossible for tokens at the outer extremes of the images. The analog video multiplexer creates a time-sequential video stream, where left and right video streams are spatially superimposed and temporary interleaved.

At the next processing stage the time-segmental video stream is captured and converted into a digital video stream which is described using a *YCbCr* color scheme. The analog video multiplexer and cameras are synchronized for capture period *T* depending on the used video standard. If the suggested *PAL* video standard is used then the capture period *T* is *12.5* frames per second, as the *PAL* video standard for the frame rate is defined at *25* frames per second. The processing requirements are not only defined by the frame rate but also by the employed video stream spatial resolution. The image resolution is defined at *625* lines for *PAL* video standard where each of them consists of *944* pixels. Of the given number of pixels, *768* are active within each line where *576* active lines are present. Thus the pixel frequency within the *PAL* video stream is *14.75 MHz*.

The high-quality single-chip digital video decoder *TVP5040* from Texas Instruments /17, 18/ was used as the video capture unit. This converts base-band analog *NTSC* and *PAL* video into digital component video. Both composite and S-video inputs are supported. The *TVP5040* includes two 10-bit A/D converters with 2x sampling. The output formats can be 8-bit, 10-bit, 16-bit, or 20-bit 4:2:2.

3.1. Logical preprocessor

The logical preprocessor is fully designed within the programmable logic devices and features hardware accelerated color and spatial filtering algorithms. The *FPSLIC* and *FPGA* circuits from Atmel were used as programmable logic devices. A logic device that interacts directly with the digital video decoder the *AT94K40 FPSLIC* device was used first /13/. This integrated circuit combines an *AT40K* dynamically configurable *SRAM FPGA* and high-performance 8-bit *AVR RISC* microcontroller with standard peripheral interfaces. The second logic device was a standard *SRAM* based *FPGA* integrated circuit AT40K40 /15/. Both devices can be efficiently used as coprocessors for high-speed (DSP/processor-based) designs by implementing a variety of computation intensive, arithmetic functions /9, 10/. These include adaptive finite impulse response (FIR) filters, fast Fourier transforms (FFT), convolvers, interpolators and discrete-cosine transforms (DCT) that are required for video compression and decompression, encryption, convolution and other multimedia applications. Both devices are capable of implementing Cache Logic (dynamic full/partial logic reconfiguration, without loss of data, on-the-fly) for building adaptive logic and systems. As new logic functions are required, they can be loaded into the logic cache without losing the data already there or disrupting

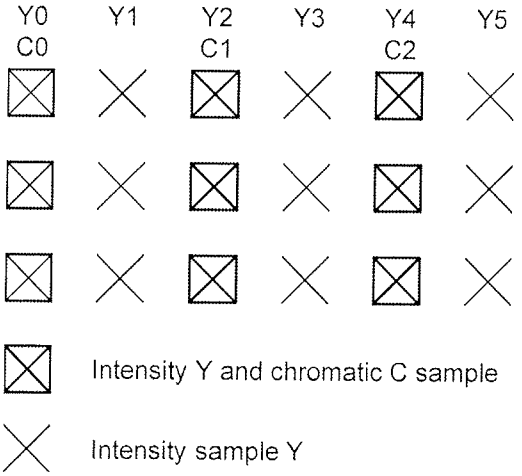

Figure 2: YCbCr 4:2:2 digital video format.

the operation remaining of the chip; replacing or complementing the active logic. The AT40KAL can act as a reconfigurable coprocessor /7/.

The captured digital video stream at the output of the conversion stage is given within the 4:2:2 *YCbCr* format. This means that the digital video stream includes twice the intensity information *Y* than chromatic information *Cb* and *Cr*. Prior to color filtering the employed sample format has to be converted into a 4:4:4 sampling scheme /26/. In such a way each of sampled image pixels has full color information. The given conversion is included within the preprocessing stage. The preprocessing stage can be gathered as a logical preprocessor (figure 4).

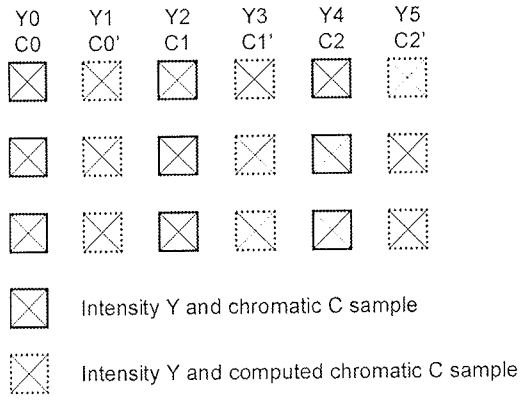

Figure 3: YCbCr *4:4:4 digital video format.*

The logical preprocessor input signal is the captured digital video stream within the *YCbCr* 4:2:2 sample format. The logical preprocessor output signal is represented as a binary mask stream. Thus it is obvious that less data bandwidth is required to transmit the binary mask stream as for *YCbCr* digital video format.

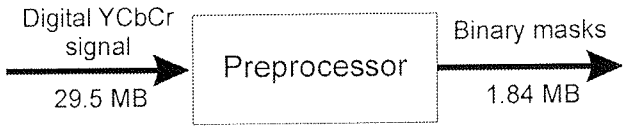

Figure 4: Color and spatial filtering stage.

The preprocessing stage includes the already mentioned digital video format conversion, and the color and spatial filter.

3.2. Skin color filter

An employed nonlinear parametrical digital color filter is based on the given mathematical model /35/ where the suggested logical units were used. In order to lower the logical and computing complexity of the digital filter the given structure of common logical unit *LU* was optimized for each of the given threshold constraints within the mathematical model in association with their computing complexity /35/.

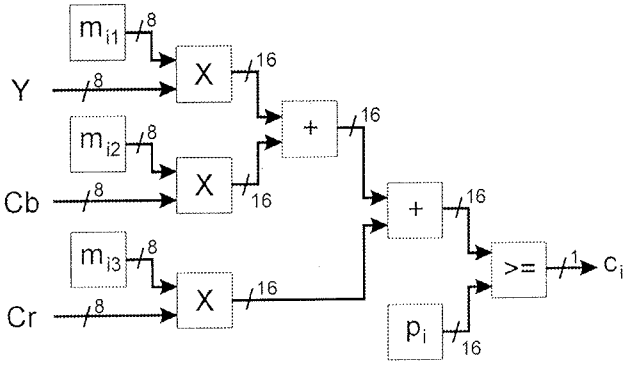

Figure 5: Structure of common logical unit LU.

Figure 5 presents a general design of a logical unit *LU* that features computing for all conditions c_i within the mathematical model of the filter /35/. Mathematically the logical unit expression can be written as a scalar product of the two vectors

$$c_i \Rightarrow [m_{i1} \quad m_{i2} \quad m_{i3}] \cdot [Y \quad Cb \quad Cr]^T \geq p_i. \quad (3.1)$$

The values m_{ij} present the parameters of the digital nonlinear filter and the *YCbCr* triple the input pixel sample where p_i is the given threshold value for the current condition c_i. On its input the logical unit accepts values of the individual image points within the *YCbCr* color space where each particular component of the *YCbCr* triple is given as an 8-bit value. Thus the color of each input point is given as 24-bit color depth. The *Y* component is given as value within the interval [0, 255] while chrominance components *Cb* and *Cr* are given as values within the interval [-128, 127].

Each individual input data component is multiplied by the appropriate parameter m_{ij} which is given as an 8-bit value. Used multipliers perform signed integer multiplication where, in the next step, the adders are forming partial sums that are compared to an adequate parameter p_i. The comparison is carried out by a 16-bit comparator that passes the binary output information to the output logical function of the digital filter. The parameters m_{ij} and p_i are implemented as registers within the digital filter and are able to memorize current setup values. These values can dynamically change during runtime.

The architecture of the given digital color filter is based on parallel configuration where appropriate condition values are computed in a parallel fashion. There are six conditions needed for the mathematical model. Therefore for each individual image pixel within the input digital video stream given conditions are computed where the output function of the color filter defines the logical state that represents the skin feature within the binary mask stream.

3.3. Automatic adjustment to scene luminance

The described digital nonlinear color filter has the ability to modify its parameters dynamically. The given parameters

are definable using the shown mathematical digital filter model. Thus it is possible to automatically adapt the parameters regarding scene luminance within the input image stream *i(t)*. It is reasonable that the parameters of the filter have to be corrected for larger changes in scene luminance, to achieve regular skin features extraction. It was found that within the changes in scene luminance it is necessary to modify the threshold values for saturation s_{max} and value v_{max} as the hue in such cases shows more or less constant value except when the brightness is very high.

Within the input sequence image, represented by the input image stream *i(t)* the color stimulus, in general, is non-uniform over the entire image area. Thus it is reasonable to obtain a general estimation for the brightness and saturation of the entire image area. As the input stream is represented within the *YCbCr* color scheme it is possible to use the *Y* component for an adequate estimation of brightness, whilst chrominance components *Cb* and *Cr* can be used for an adequate estimation of saturation. The average values for image brightness $\bar{v}$ and saturation $\bar{s}$ can, therefore, be computed as

$$\bar{v} = \frac{1}{N}\sum_{i}^{N} Y(i) \text{ and } \bar{s} = \frac{1}{N}\sum_{i}^{N} S(Cb[i], Cr[i]) \quad ,(3.2)$$

where *N* stands for the number of image points, *i* denotes the current coordinate of a point and *S(Cb,Cr)* is the saturation function indirect to chrominance.

The estimation for saturation $\bar{s}$ remains constant for global changes of scene luminance as a reflection of changes in the brightness of the light sources whilst the input image stream shows changes in estimation for value $\bar{v}$. The average estimation for saturation $\bar{s}$ mostly depends on the spectrum of the light stimulus within the scene. This shows that the average saturation estimation can be used to correct the saturation threshold of the digital filter and furthermore the average estimation value can be used to correct the value threshold.

The digital filter's threshold values can be modified dynamically in real time regarding the capture interval *T*. Thus both estimations become time-dependent as their values within the previous capture interval *T-1* can be used to correct the current threshold values at interval T. Consequently the threshold values v_{max} and s_{max} become time-dependent as their values are altered at the beginning of the capture interval *T*. The given implication can be written as

$$\{\bar{v}(T-1), \bar{s}(T-1)\} \Rightarrow \{v_{\max}(T), s_{\max}(T)\}. \quad (3.3)$$

The functional dependence of a given implication can be gathered using large sample set of test images under different lightning conditions where the appropriate values can be estimated to assure constant filtering quality.

3.4. Spatial filter

The introduced spatial filter features a median translation function within the binary mask stream. Spatial filtering is employed in order to remove singularities within the binary mask stream, caused by image noise, and to link small homogenous regions of skin features into larger skin formations. Skin formation *F* is the union of those skin features or elements of binary mask that are combined by spatial connectivity. Two elements of binary mask are spatially connected if a path between them exists where the characteristic function stays constant. If spatial connectivity is expressed as:

$$c(x,y,T) = \sum_{i=-1}^{i=1}\sum_{j=-1}^{j=1} m([x+i],[y+j],T), \quad (3.4)$$

then the element *m(x,y,T)* is spatially connected with one of its eight neighbors if *c(x,y,T)>1*. As noise within the binary mask not only exists as a singular skin feature but also as small skin formations it is reasonable to remove small skin formations or spatially link them to larger skin formation with the use of a spatial median filter *MF*. The translation function of the median filter within the level of binary masks can be expressed as:

$$MF = \begin{cases} 1, & \sum_{i=-\frac{(k-1)}{2}}^{i=+\frac{(k-1)}{2}} \sum_{j=-\frac{(l-1)}{2}}^{j=+\frac{(l-1)}{2}} m([x+i],[y+j],T) \geq med \\ 0, & \sum_{i=-\frac{(k-1)}{2}}^{i=+\frac{(k-1)}{2}} \sum_{j=-\frac{(l-1)}{2}}^{j=+\frac{(l-1)}{2}} m([x+i],[y+j],T) < med \end{cases}, \quad (3.5)$$

where *k* and *l* are positive odd numbers that define the size of the treated region within the binary mask. The adequate median value *med* can be expressed as

$$med = \frac{k \cdot l}{2} + 1. \quad (3.6)$$

As an example a digital spatial filter with size of translation matrix 3 x 3 is shown in Figure 6.

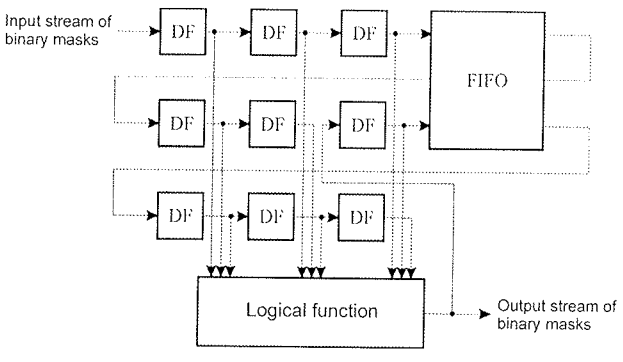

Figure 6: Architecture of spatial filter with size of translation matrix 3 x 3.

The logical function of a given spatial filter can be expressed in terms of partial sums by column or rows. The FIFO memory is employed in order to memorize those particular binary mask lines needed to gain proper spatial filter operation.

4. Initialization, synchronization and data path

Because *FPSLIC* and *FPGA* logical devices are *SRAM* based they need to be initialized with the appropriate configuration bitstream after power-up or master reset signal has been received. Therefore a serial *EEPROM0* configuration memory *AT17LV010* with capacity of *128 kB* was used /16/. Both devices are configured from the same configuration memory in master/slave fashion. The bitstream of the *FPSLIC* device also includes the firmware or the program memory of the embedded *AVR* microcontroller.

The firmware of the embedded AVR microcontroller is started after successful configuration of the logical devices. The first step of the *AVR* program is to configure the digital video decoder. This is done by uploading the digital video decoder firmware which is written in the second serial *EEPROM1* configuration memory *AT17C256* with a capacity of *32 Kb* /16/. The host peripheral interface *HPI* /17/ connected to the embedded *AVR* core is used for loading the program data and appropriate digital video decoder configuration. As the digital video decoder is being configured, the *AVR* program configures the time-sequential analog video multiplexer. The connection between the embedded *AVR* core and the video multiplexer is done by using one of two asynchronous serial ports. The second asynchronous serial port is used for direct communication between the embedded *AVR* core and a standard personal computer. This makes it possible to dynamically change capture parameters, digital color filter parameters or gather status values in runtime during operation. This hardware feature is useful during application development and debugging, or similar.

The master synchronization signal is generated by a controllable analog video multiplexer where both analog cameras are synchronized with each other. The analog multiplexer can receive control commands such as full/half frame auto-switching where an analog time-sequential and spatially superimposed video signal is generated. The video signal can also be configured as a single left/right camera signal. All commands to the video multiplexer synchronization controller are executed within the blanking of the video signal so that no synchronization errors can occur. Each time the video signal is switched between left and right camera signals, a special synchronization signal is send to the embedded *AVR* core over the serial link. This signal informs the *AVR* core to add framing to the data path.

As already stated the data path begins at the analog video multiplexer where the time-sequential video signal is generated. This signal is captured by the digital video decoder and passed to the digital preprocessor. The logical circuit within the *FPSLIC* device receives the digital video signal and extracts skin features. Skin features in the form of a binary mask stream are transferred to the second logical circuit within the *FPGA* device where spatial filtering is performed. The skin formations are created in such a way and then transferred to the *DSP* system.

There are three different communication channels implemented between the *DSP* and digital preprocessor. Two of them use *McBSP* (Multi-channel Buffered Serial Port) interfaces of the *DSP* /19/. These two ports need syn-

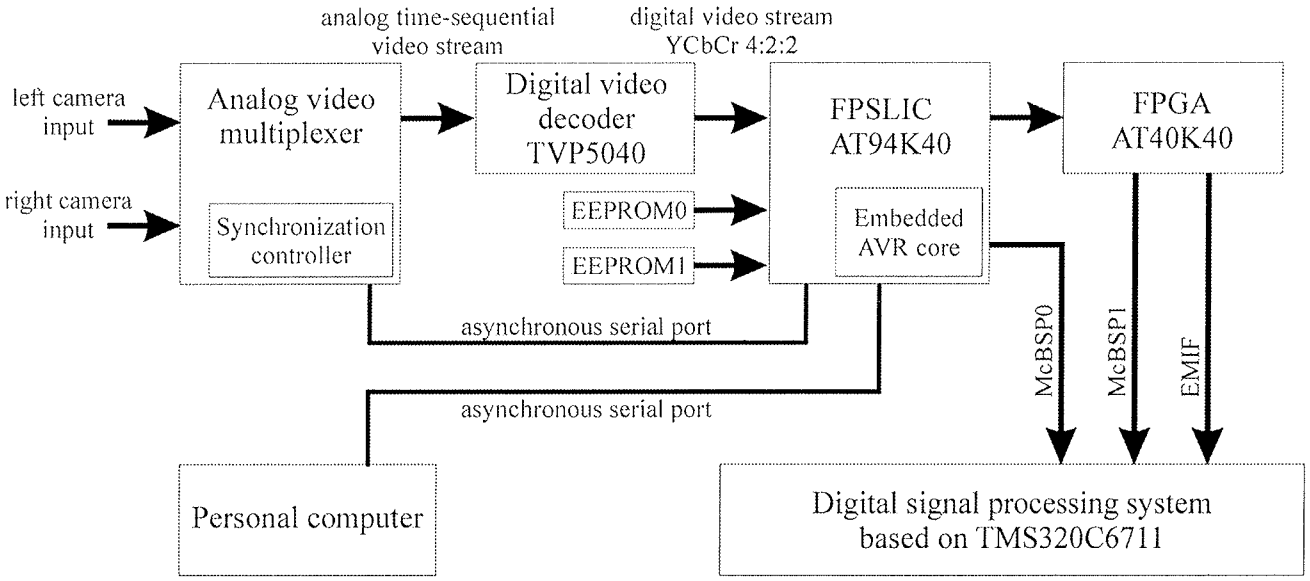

Figure 7: Block scheme of the capture unit.

chronous serial transmitter/receiver logic which is implemented within the *FPGA* and *FPSLIC* logical circuits. The *McBSP0* port is used for the transfer of the binary mask stream. This serial port is configured to transfer 32-bit long words where *24* framing signals are required to transfer *768* bits of the complete binary mask horizontal line. The *DMA* (Direct Memory Access) of this serial communication channel is configured to receive and transfer the binary mask stream to the main memory of the *DSP* system where complete framing of the binary mask is added. Motion extraction has been added within each of the DMA cycles where implication of the logical operator *LO* has been used. Each of the 32-bit long words received for the current binary mask *M(T)* is used to compute the current dynamic binary mask *MM(T)*. This means that, the dynamic binary masks are computed in-time with binary mask stream capturing.

The second *McBSP1* port of the *DSP* is used to send/receive configuration data as, for example, the parameters of skin feature extraction within the digital color filter. The second serial communication channel is bound to the embedded *AVR* core where the embedded firmware receives the configuration data and applies those configuration changes into logical units *LU* registers used to store the parameter values of the skin feature extraction process.

Automatic adjustment to scene luminance is solved over the second serial communication channel. The embedded *AVR* core accumulates the current average values for value $\bar{v}$ and saturation $\bar{s}$ within the current captured frame and passes them to the *DSP* system. The *DSP* program calculates new color filter parameters and sends them back to the *AVR* core where these new calculated values are applied to the color filter logic.

The third communication channel between the logical preprocessor and the *DSP* can be configured using the *EMIF* /19/ (Extended Memory Interface) of the *DSP*. The *EMIF* interface features 32-bit parallel access to the logic circuit within the *FPGA* device for read/write operation. This feature of the hardware design can be used within applications where larger data bandwidth is required. Within the *DSP* system the *EMIF* interface can also be configured for *DMA* cycles and can be used for non-filtered image transfers.

5. Efficiency estimation for given hardware structure

In order to gain real-time processing within the given system, the available time to perform all required algorithms is *80 ms* when *PAL* video standard is employed. A time delay of *40 ms* is present between the left and right captured images due to the employment of a time-sequential video system. To achieve efficiency estimation for the given hardware structure, a test bed was created that runs on a standard personal computer within the MS Windows operating system and has been developed with MS Visual Studio 6.

Some additional processes have been added to estimate time performance requirements for a particular algorithm within the given functionality. The current test bed features definition for the spatial positions of those skin formations stereo pairs that represent the nearest skin-colored subject within the captured scene. The algorithm that defines the spatial position uses the energy projection approach within the left and right dynamic binary mask. As energy projections are computed the set of potential stereo skin formation pairs is generated where the best match is chosen by the stereo correspondence algorithm that uses the stereoscopic and epipolar constraints of the stereo parallel camera setup /3/.

In such a way the combined processing path of the complete test bed functionality was divided into particular processes:

- Process of color and spatial filtering that includes skin feature labeling and the creation of homogenous skin formations.
- Process of dynamic binary mask creation with employment of a logical operator.
- Process of energy projections computation.
- Process of defining the spatial positions of skin formations that includes motion analysis and solving stereo correspondence problem.
- Process of frame backup that is required to gain dynamic binary mask creation.

Table 1 shows the gathered performance requirements measured using a personal computer where the times are given as a proportion of the disposable processing power.

Table 1: Relative times required to perform specific process.

Process	Required time
Color and spatial filtering	0.6386
Dynamic binary masks creation	0.3456
Energy projection computation	0.0005
Definition of spatial position	0.0004
Backup current frames	0.0007

The time required to define spatial position depends of the number of skin formations found or present within the captured scene. It can be seen (table 1) that most of the available processing time is required for color and spatial filter. About *35 %* of available time is required for the creation of dynamic binary masks, where other *1.6 %* is required for other operations that include energy projection computation, spatial position definition and backup of current frames.

Color and spatial filtering is performed using the suggested hardware architecture within the preprocessing stage, thus available processing time is increased by about *60 %*.

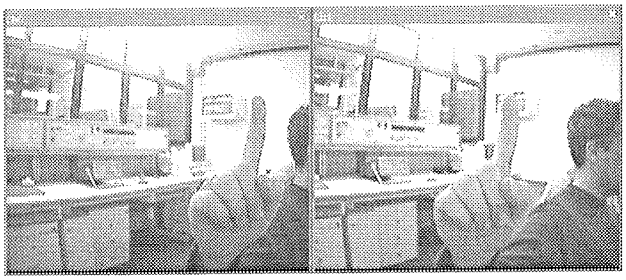

Figure 8: Non-filtered stereo image of input sequence.

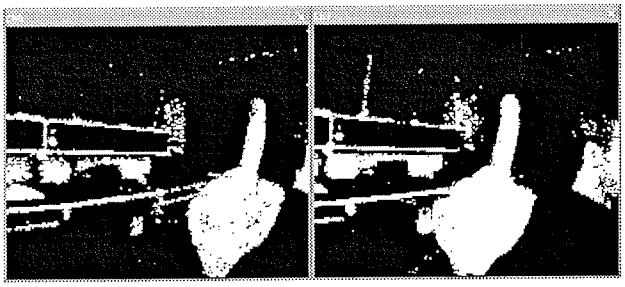

Figure 9: Appurtenant binary masks without employment of spatial filtering.

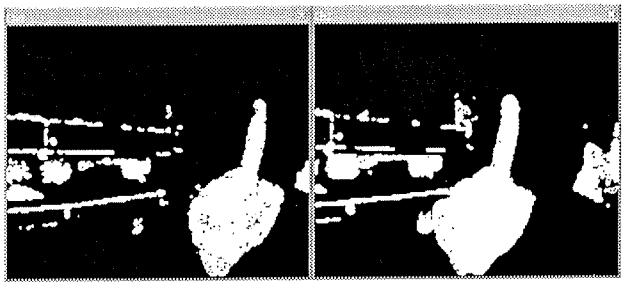

Figure 10: Appurtenant binary masks with employment of spatial filtering.

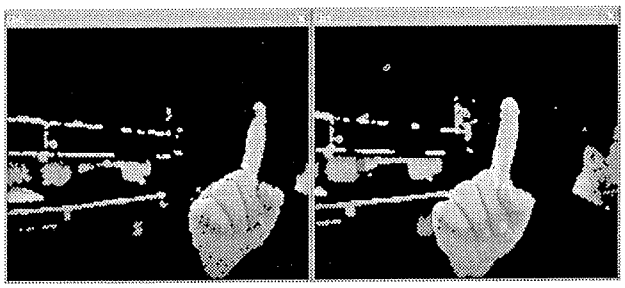

Figure 11: Placing the binary masks onto the original input stereo image.

The original stereo image of the input sequence is given in figure 8. Figure 9 shows the appurtenant binary masks where no spatial filter has been employed. In contrast, figure 10 shows the same binary masks with the employment of spatial filtering with size of translation matrix 5 x 5. Figure 11 presents the implication of the binary masks onto the original input stereo image where they are spatially superimposed.

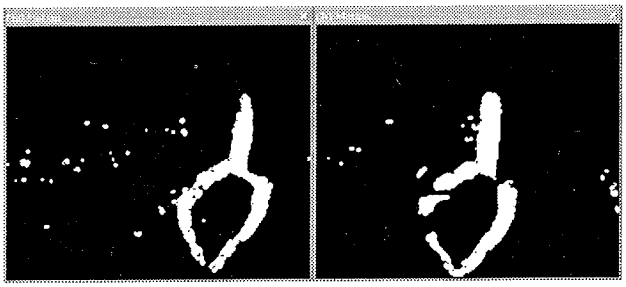

Figure 12: Dynamic binary masks $MM_L(T)$, $MM_R(T)$ *where the input sets of their elements are the binary masks* $M_L(T-1)$, $M_R(T-1)$.

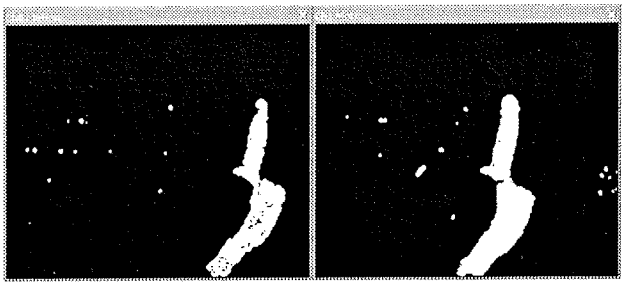

Figure 13: Dynamic binary masks $MM_L(T)$, $MM_R(T)$ *where the input sets of their elements are the binary masks* $M_L(T)$, $M_R(T)$.

Figures 12, 13 show two different approaches for motion extraction using the same logical operator. It can be seen that the skin formations are not fully extracted due to the slow-motion of the controlled subject regarding capture period *T*. Thus the quality of motion detection mostly depends on the velocity of the subjects in motion within the scene and the capture period *T*. The capture period was 6 stereo frames per second for the given results as they were captured within the test bed software.

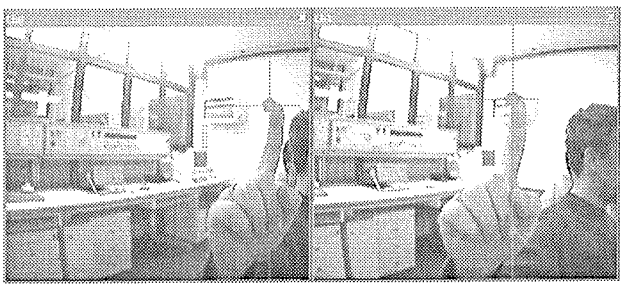

Figure 14: Definition of the navigation point's spatial position.

The definition of the navigation point's spatial position is shown in figure 14 where the vertical lines denote the position of extreme value within the appropriate energy projections whilst the horizontal lines denote the boundary frames B_L, B_R for given skin formations (figure 15).

Figure 15 shows vertical errors λ_L, λ_R of the adequate boundary frames B_L, B_R for given skin formations caused by motion parallax due to time delay in time-sequential ster-

eo capturing. In such cases the navigation point coordinates in the vertical direction can be computed as the average value between appropriate navigation point projections to the left and right image planes.

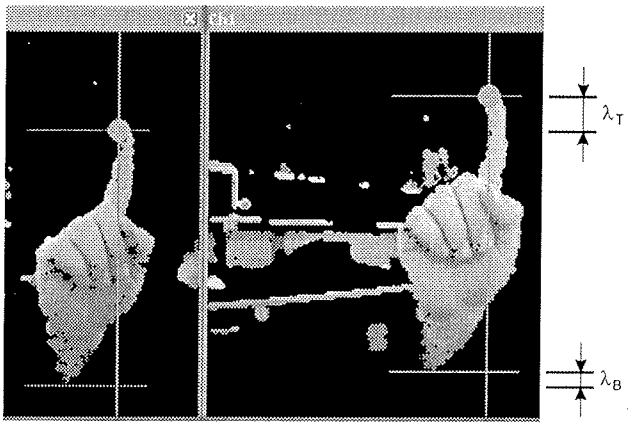

Figure 15: Vertical errors within the boundary frames B_L, B_R caused by motion parallax.

6. Conclusion

It has been shown that with the suggested time-optimized labeling or skin feature extraction within the captured stereo input image sequence performed by nonlinear parametrical digital color filter, it is possible to create binary masks in real-time. It is possible to create skin formations and remove noise singularities by spatial dependence examination between skin features within a small region of the binary mask. It has been shown that with dynamic binary mask computation with a suggested logical operator, it is possible to define those regions within the image sequence that are exposed to changes due to the motion of skin-colored subjects and objects within the captured scene.

The presented color and spatial filtering can also be foreseen within other vision-based applications that require the extraction of human parts from real-time image sequence such as a human face for example. In further work we suggest detailed analysis of automatic color filter adaptation to different types of skin. A more exact definition of skin-color area that would be more specific to user's skin types would gain more accurate extraction of skin features and thus, more precise definition of skin formations within the image sequence. Regions could be predicted where occlusion of dynamic and static skin formations could occur using appropriate scene complexity estimation. Therefore the employment of a two-dimensional model of the control subject could solve the problems of spatial positional definition within those regions.

Literature

/1/ G. Sharma. Ed. Digital Color Imaging Handbook. Xerox Corporation, CRC Press LCC, New York, 2003.

/2/ S. W. Perry, H. S. Wong, L. Guan. Adaptive Image Processing. A computational Intelligence Perspective. CRC Press LCC, New York, 2002.

/3/ R. Hartley, A. Zisserman. Multiple View Geometry in computer vision. Cambridge University Press, UK, 2000.

/4/ L. Guan, Ed. Multimedia Image and Video Processing. Boca Raton: CRC Press LCC, New York, 2001.

/5/ J. C. Russ. The Image Processing Handbook - Fourth Edition. Materials Science and Engineering Department, North Carolina State University, CRC Press LCC, North Carolina 2002.

/6/ Palmer, Steven. Vision Science. The MIT Press, Cambridge, MA, 1999.

/7/ I. Kramberger, M. Solar. DSP Acceleration Using a Reconfigurable FPGA. Proc. of IEEE International Symposium on Industrial Electronics - ISIE '99, Bled - Slovenija, 1999.

/8/ New, Bernie. A distributed arithmetic approach to designing scaleable DSP chips. EDN, pp. 107-114, 1995.

/9/ Atmel. Recommended Design Methods. Atmel Corp., September, 1997.

/10/ Atmel. Implementing Cache Logic with FPGAs. Atmel Corp., September, 1997.

/11/ Atmel. Implementing Bit-Serial Digital Filters. Atmel Corp., September, 1997.

/12/ Atmel. Implementing FreeRAM inside the FPGA or AT94K Series FPSLIC Using VHDL with IP Core Generator. Atmel Corp., 2001

/13/ Atmel. AT94K Series Field Programmable System Level Integrated Circuit. Atmel Corp., 2001.

/14/ Atmel. 8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash - AT90S8515. Atmel Corp., 2001.

/15/ Atmel. 5K - 50K Gates Coprocessor FPGA with FreeRAM™. Atmel Corp., 2002.

/16/ Atmel. FPGA Configuration EEPROM Memory. Atmel Corp., 2003.

/17/ Texas Instruments. TVP5040 NTSC/PAL Digital Video Decoder With Macrovision (TM) Detection. Data Manual, DAV Digital Video/Imaging, May 2001.

/18/ Texas Instruments. TVP5031/40/5145EVM User's Guide. DAV Digital Video/Imaging, September 2001.

/19/ Texas Instruments. TMS320C6711 Floating-Point Digital Signal Processors. Texas Instruments Incorporated, 1999, Revised 2003.

/20/ D. Bell. TMS320 Cross-Platform Daughtercard Specification - Revision 1.0. Texas Instruments Incorporated, Application report, 2000.

/21/ MathWorks, Inc.. Image Processing Toolbox User's Guide. Mathworks, 2001.

/22/ MathWorks, Inc.. Signal Processing Toolbox User's Guide. Mathworks, 2001.

/23/ MathWorks, Inc.. Statistics Toolbox User's Guide. Mathworks, 2001.

/24/ Y. Wu, Y. Shan, Z. Zhang, S. Shafer. Visual Panel: From an ordinary paper to a wireless and mobile input device. Microsoft Research, October 2000.

/25/ J. R. Ohm, K. Grunberg, E. M. Izquiredo, M. Karl. A realtime hardware system for stereoscopic videoconferencing with viewpoint adaptation. Heinrich Hertz Institute, Image Processing Department, Germany, 2000.

/26/ J. Keith. YCbCr to RGB Considerations, Converting 4:2:2 to 4:4:4 YCbCr. Intersil Application Note March 1997 AN9717.

/27/ M. Suen, R. Kleihorst, A. Abbo, E. C. Solal. Real time skin-tone detection with a single digital camera. Philips Research Laboratories, Eindhoven, Netherland. 2001.

/28/ S. Ahmad. A usable real-time 3D hand tracker. In proc. IEEE Asilomar Conf., 1994

/29/ J. Crowley, F. Berard, J. Coutaz. Finger tracking as as input device for augmented reality. In Proc. Int'l Workshop on Atomatic Face and Gesture Recognition, pages 195-200, Zurich, 1995.

/30/ R. Kjeldsen, J. Kender. Finding skin in color images. In Proc of Second International Conference on Automatic Face and Gesture Recognition, pages 312-317, 1996.

/31/ M. Jones, J. Rehg. Statistical color models with application to skin detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab., 1998.

/32/ Raja, S. McKenna, S. Gong. Colour model selection and adaptation in dynamic scenes. In Proc. European Conf. Computer Vison, pages 460-475, 1998.

/33/ A. Wu, M. Shah, N. V. Lobo. A Virtual 3D Blackboard: 3D Finger Tracking using single Camera. Department of Computer Science, University of Illinois, Urbana, School of Computer Science, University of Central Florida, 1999.

/34/ N. Herodotu, K. N. Plataniotis, A. N. Venetsanopoulos. Automatic location and tracking of the facial region in color video sequences. Signal Processing: Image Communication, 14, 359, 1999.

/35/ I. Kramberger, Z. Kačič. Implementation of parametrical nonlinear digital filter for skin features identification in digital image using FPSLIC technology. Informacije MIDEM, št.: 3, 2003.

Dr. Iztok Kramberger, Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia.

prof. Dr. Zdravko Kačič, Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia.

University of Maribor, Faculty of Electrical Engineering and Computer Science Smetanova ulica 17, 2000 Maribor, Slovenia

Prispelo (Arrived): 21.06.2003 Sprejeto (Accepted): 26.08.2003