Relational Database for Visual Data Management

Dale Lord (Student) and Kurt Kosbar (Advisor)

Telemetry Learning Center
Department of Electrical and Computer Engineering
University of Missouri – Rolla
Rolla, MO 65409-0040

ABSTRACT

Often it is necessary to retrieve segments of video with certain characteristics, or features, from a large archive of footage. This paper discusses how image processing algorithms can be used to automatically create a relational database, which indexes the video archive. This feature extraction can be performed either upon acquisition or in post processing. The database can then be queried to quickly locate and recover video segments with certain specified key features.

Keywords: Feature Extraction, Data Management, Indexing

INTRODUCTION

Many telemetry systems include a video signal as well as sensor data. This video is often recorded to catch many of the environmental aspects that were not seen in the sensor data. It is often difficult to tell what is contained in each video segment, and it may be necessary to manually label the video clips. This paper introduces a method for automating this process of labeling the video clips by characteristics of the video, or what is contained in the video frames.

The first step in processing the video frames is to determine the motion of the camera and the motion of objects in the video. Without other sensors, encoding information about this motion may be an error prone task. However in limited domains, motion estimation can be done directly from the video. If this information is not available the processing can still be achieved, but at a much slower rate.

Once the parameters of motion are estimated, they are used to transform the location of elements in the view, by using view morphing. The new locations are verified with the new frame of video, which eliminates searching or detecting elements in the new frame. When new view information is obtained, the region needs to be searched for image elements. These elements can be any item or feature in the image which has unique location tracking ability.

MOTION ESTIMATION

Ideally, motion information can be provided from other sensors attached to the camera or the platform the camera is affixed. When the camera motion cannot be calculated using other sensors a procedure for estimating the camera motion from the view is needed.

The image view is divided into nine parts, as enumerated in Figure 1, and each part is searched for an image feature which is unique spatially.

1	2	3
4	5	6
7	8	9

Figure 1: Image Segments

This spatially unique feature is tracked from image frame to image frame to establish the feature's motion. Since there will be movement in the image, independent of the camera motion, fuzzy logic from each of the image segments is used to estimate the view motion. The expected image element motion may be summarized in a table similar to Figure 2.

	Segment								
Motion	1	2	3	4	5	6	7	8	9
Forward / Zoom	UL	U	UR	L	-	R	DL	D	R
Reverse / Zoom out	DR	D	DL	R	-	L	UR	U	L
Move left / Pan left	R	R	R	R	R	R	R	R	R
Move right / Pan Right	L	L	L	L	L	L	L	L	L
Move up / Pan up	D	D	D	D	D	D	D	D	D
Move down / Pan down	U	U	U	U	U	U	U	U	U

Figure 2: Feature Motion (U=up D=down L=left R=right)

Quantifying the motion is difficult since the (generally) poor resolution and the need for additional geometry to calculate real distances or angles. Various techniques are available to estimate these geometries including stereoscopic vision [15] and view interpreting [13]; however they are beyond the scope of this paper.

VIEW TRACKING

With the motion of the camera, and some geometry of the scene, one can predict how locations change in the camera view [10][11]. This can be useful to reduce the amount of reprocessing that is required, since only new areas of the view need to be processed in each frame. Also, the motion of moving objects in the view can be more easily detected, since the scene motion caused by the camera movement can be removed.

Common platform and camera motions result in transformations of the image including zooming in, which is similar to moving forward, zoom out, which is similar to moving backward, turning which is similar to a camera pan, and camera tilt. The zoom in and moving forward loose view information from all edges of the image however the detail of the scene increases. While zoom out and move backward give new information at the edges of the image and reduce the detail of the previous view. These both can be approximated by a simple scaling operation shown in equation (1):

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

Where $s \in [0,\infty)$ is the scaling factor. A value greater than one implies the platform is moving forward or zooming in. For values between zero and one, the platform is moving backward, or zooming out.

The turning of the robot, or a pan of the camera, not only moves scene information horizontally, but also has some scaling effect - depending on the angle. This effect can be represented by the perspective transformation of equation (2)

:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} \cos(\theta_v) & 0 & -\sin(\theta_v) \\ 0 & 1 & 0 \\ \sin(\theta_v) & 0 & \cos(\theta_v) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{2}$$

The x_i and y_i are image locations, and z_i represents an approximation of distance to the object in the image. The resulting 3D point $P_t=(x_t,y_t,z_t)$ can be reprojected back to the image plane by equation (3):

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} z_i / z_t & 0 \\ 0 & z_i / z_t \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{3}$$

Similarly for the camera motion θ points (n_x,n_y,n_z) in the new image B (see Figure 3) can be partially formed from the projection of A onto B.

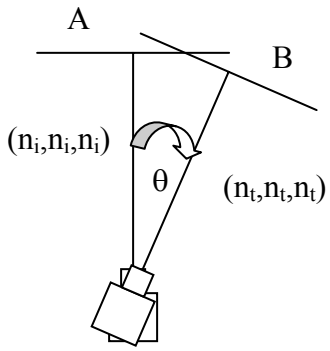

Figure 3: Camera motion of turn or pan

The operation of tilting the camera by ϕ has a similar transformation shown in equation 4.8:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\sin(\phi_v) \\ 0 & \cos(\phi_v) & 0 \\ 1 & \sin(\phi_v) & \cos(\phi_v) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{4}$$

Every object in the cameras view can be transformed to the new locations using this projective geometry.

As a demonstration, an image that would show the worst-case, is transformed below, in figures 4 through 7. This view is called worst-case, because the scene depth is far from constant, which is the assumption used in the equations

Figure 4: Original Image

Figure 5: Forward, or Zoom

A camera pan right of twenty degrees and a tilt up of twenty degrees are shown below

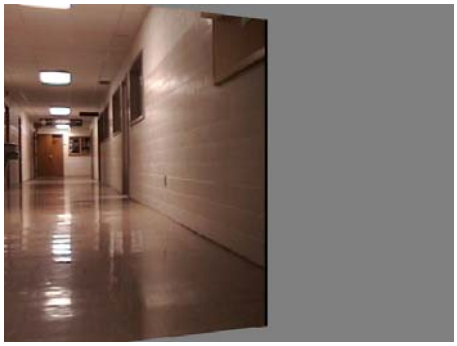

Figure 6: Pan

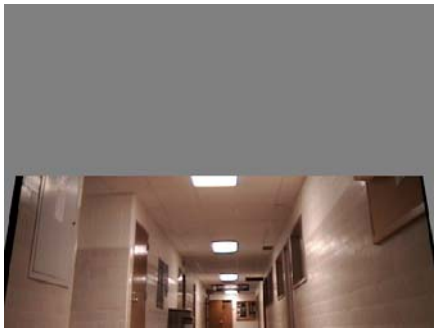

Figure 5: Tilt

Note: Areas in gray are unable to be determined from the previous view.

COLOR SPACE TRANSFORM

Image data can include color information about the scene. This color data, makes color recognition a relatively simple task of reading this data at a specified location, and reporting the color. The first issue, however, is the color-space model used. Probably the most commonly used color-space is the RGB space. This is the color-space used for CRT displays, and many image raw storage formats. There are three values which indicate the amount of red, green and blue color in a pixel. This space is convenient because commercial capture cards are capable of producing this color-space. Sending the image to a screen is also simple, since the values just need to be copied to the display buffer. Unfortunately, this scheme is not particularly useful to classify color, nor is it

easy to account for lighting changes in the image. Different lights can have different emission spectrums affecting the color of objects in the view, however the most common differences in lighting is the changes in intensity and saturation caused by glair and shadows.

A color-space conceived to more naturally fit how people talk about color is the HSV representation. In this color model there are components for Hue, Saturation and Value. Hue values are arranged in a manner similar to the frequency spectrum of light. Saturation represents how much color verses gray scales are in each pixel and Value represents the overall brightness of the pixels. While this representation is useful for classification it would have to be computed in software for each pixel in the image.

A color space representation that is similar to the HSV representation and that is native to NTSC cameras and capture cards is the YCrCb (YIQ) color space. This color model has luminance (Y) which is similar to the value component of HSV. However color and saturation are represented by Cr (I) and Cb (Q). I is the orange-blue axis, and Q is the purple-green axis.

The last problem with color recognition is deciding what color name goes with what part of the color spectrum. This varies from person to person so for simplicity data was recorded for various colors and statistical thresholds were established between green, red, yellow, orange, purple and blue.

SCENE CHARACTERISTICS

General information about the video can be obtained by drawing simple conclusions from these aforementioned image transforms and motion estimations. The frames of the video are sampled at regular intervals and the transforms and estimations are done to get descriptors of the video.

For example, video that was sampled outdoors can give an indication of whether the video was taken during the day or at night. A video that was taken at night would generally have large segments of dark scene and could be identified accordingly. While well lit or daylight scenes can also be identified by the high average brightness value component.

Depending on the domain there may be other conclusions that can be drawn by the transformed average scene hues. For example if an outdoor scene has mostly shades of green in the view the video could be characterized as foliage. Footage with large amounts of blues could be characterized as sky or water.

Motion estimation also gives a descriptor for the video segments. By estimating the camera motion the video can be labeled according to how the camera was moved during the video taping. With view tracking, it is possible to account for the motion in the view that was caused by the camera movement. This leaves the remaining motion in the view to be the motion of various objects moving around in the scene.

IMAGE ELEMENTS

Gaining specific information about objects in the scene can take a significant amount of processing time. A method of image processing, which is useful to get quick information about the scene, is image element detection. Much of the raw image data of a normal scene is redundant because many objects have a similar visual texture throughout the object. It is more efficient to concentrate on image areas that are unique, like the edges or corners of objects.

Information theory [6] contends that there is more information in less probable events. In order to maximize the information gained from the least amount of processing time, searching for an extremely rare event would provide the most information. However if the event is not likely to occur within a given sample then no information is gained. So a balance is needed between the information gained and the probability of occurrence.

In a natural scene there are often large areas of similar image texture so more information is gained by looking at the edges between these areas. And even less likely to occur are locations where one or more edges intersect. These intersections, called image elements in this paper, occur often enough for a good balance in information gain.

The implementation of image elements not only uses edges but also uses image color. Each element is defined by two templates. Each template is a two dimensional array of values corresponding to the edge strength and color. In the templates a reserved value of zero is used to represent an image value that has no weight in the template matching or commonly referred to as a "don't care". The first template is a color matching template. Different areas of the template can represent different colors expected of the feature. The second template is an edge matching template. This edge template is matched to the edges of the image. These edges are obtained using the Sobel [4] operator on the video frames. This edge template is useful to get a match in areas where lighting changes can cause widely varying intensities and colors.

The image elements of non moving image regions are identified and recorded in the database. Each image element is related to the other image elements by the physical distance between them. In the case where the image texture is fairly uniform between the image elements the image between the elements is recorded along with the elements and their physical relationship between each other.

For example, in an indoor environment, the four corners of a door are usually separated by uniform image texture. The corners are also generally separated by around three feet horizontally and around seven feet vertically. If a handle could be identified, the process could automatically identify this region of the video image with the label "door". However even if the final step of associating a label with the elements and image region is not done, it is still useful to have the image region stored in the database with the elements. A manual process of looking through these images is still faster than viewing the original video footage. To aid future searches, the user can associate the label "door" with the image and image elements. This would allow upcoming searches to be performed using the key word "door".

RELATIONAL DATABASE

The objective of this work is to have the ability to perform searches for segments of video which match certain criteria. The system is developed to build the database of features and images that can be searched either by browsing the contents or by searching by key words. This describing data is deposited in a relational database which contains sample images of features in the image as well as image labels whenever possible. The industry standard SQL (Structured Query Language) is used to build indexes and perform queries on the data to find the video segments desired.

Identifying different objects in the scene can be done for a select few objects. And the code for each object is fairly specific to detecting the object. Detecting a door, for example, is finding a shape of approximately three feet by seven feet; hinges and a handle indicate a possible door. As an interim solution to having software to identify every possible object in the video, the image element identifying code stores sample images of connected image elements. When a specific object is required to be in the image a process of browsing through the stored image elements is done to find the elements identification number. Then searches on the video archive can be done similar to the following:

Select video, image, day_type from video_archive where day_type=”night” and image=42;

CONCLUSION

While complete image understanding is still not viable with automated computer processing, it is still possible to gain much of the functionality of indexing video images by the contents of the video. Processes can categorize image elements as well as image characteristics with image processing techniques. These features placed in a relational database can be indexed and queried for future retrieval. As image processing hardware and software advance new detected features and objects can be placed into this database allowing for further specific searches through the video archive.

REFERENCES

[1] “Color Discrimination” (viewed 11-6-2002)
http://www.ergogero.com/FAQ/Part2/xfawPart2.html

[2] “Color Spaces” (viewed 11-6-2002)
http://developer.apple.com/techpubs/mac/ACI/ACI-48.html

[3] Gonzalez, Rafael C., Woods, Rafael C. "Digital Image Processing Second Edition" Prentice Hill New Jersey

[4] Horn, Paul, Klaus, Berthod “Robot Vision,” The MIT Press McGraw-Hill Book Company 1997 ISBN 0-262-08159-8

[5] Jack, Keith “Video Demystified” High Text Interactive 1996 ISBN 1-878707-X

[6] Luger, George F and Stubblefield, William A “Artificial Intelligence – Structures and Strategies for Complex Problem Solving,” Addison-Wesley 1998 ISBN 0-805-31196-3

[7] Manian, Vidya, Vasquez, Ramon and Katiyar, Praveen “Texture Classification Using Logical Operators,” IEEE Transactions on Image Processing vol. 9 no. 10 October 2000

[8] Mirmehdi, Majid and Petrou, Maria “Segmentation of Color Textures,” IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 22 no. 2 February 2000

[9] Paschos, George “Perceptually Uniform Color Spaces for Color Texture Analysis: An Empirical Evaluation,” IEEE Transaction on Image Processing vol. 10, no. 6 June 2001

[10] Seitz, S. M. and Dyer. C. R., “Toward Image-Based Scene Representation Using View Morphing” Proc. 13th Int. Conf. Pattern Recognition, Vol. I, Track A: Computer Vision, 1996, 84-89.

[11] Seitz, S. M. and Dyer, C. R., “View Morphing” Proc. SIGGRAPH 96, 1996, 21-30

[12] Spong, Mark W. and Vidyasagar, M. "Robot Dynamics and Control" John Wiles & Sons New York

[13] Steinhage, V., “Verification of the general position assumption in the practice of stereovision” Pattern Recognition, 1992 . Vol.1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, 30 Aug.-3 Sept. 1992 Page(s):746 - 749

[14] Uchiyama, Toshio, Mukawana, Naoki and Kaneko, Hiroshi “Estimation of Homogeneous Regions of Segmentation of Textured Images,” IEEE Transactions on Pattern analysis and Machine Intelligence 2000

[15] Waldmann, J.; Merhav, S.; “Fusion of stereo and motion vision for 3-D reconstruction” Pattern Recognition, 1992 . Vol.1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on 30 Aug.-3 Sept. 1992 Page(s):5 - 8