Computer Vision Based Reliability Control for Electromechanical Dice Gambling Machine

Iztok Lapanja, Miha Mraz, Nikolaj Zimic
Faculty of Computer and Information Science
University of Ljubljana
Tržaška 25, 1000 Ljubljana, Slovenia

Iztok.Lapanja@fri.uni-lj.si

***Abstract* — In our paper we present a complete overview of a reliability control module for an electro-mechanical dice gambling machine based on computer vision principles. We particularly discuss the fundamental, heavily utilized function of color difference, which forms a basis for fast, efficient and parameterized chroma-key applications. We also explain the dice location estimation solution and the final template matching phase of number detection. In the conclusion we discuss some issues and problems which are still left open but should be solved by the project team.**

Keywords: **machine vision, chroma-key, template matching.**

1 INTRODUCTION

In our paper we present the motivation, theoretical background and implementation issues involved in a design of a reliability control system based on machine vision principles. Our system is planned to be a part of an electro-mechanical dice gambling machine [1] production line where it is required that machines exhibit an absolute error-free operation. At the time of our writing the machine vision reliability control module operates as a prototype version.

Although the manufacturer had developed its own reliability tests it became obvious that some independent and functionally separate reliability control system was required to improve on-line fault detection and define a combined system upon which to claim the absolute quality of a product — the dice gambling machine.

We decided to use the machine vision approach for various reasons:

- fast and easy system prototyping,
- high flexibility and adaptability,
- low-cost implementation,
- low-cost maintenance, and
- our experience in the machine vision field.

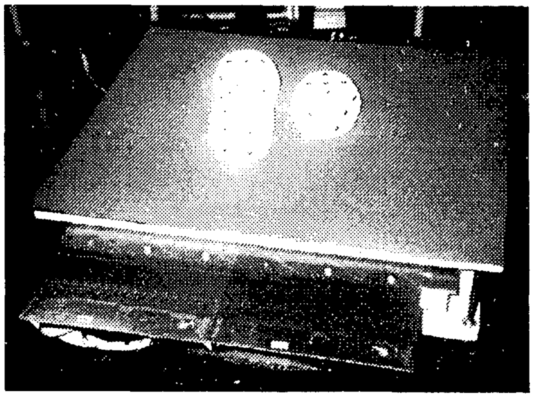

Figure 1: A typical three dice configuration seen from aside the vibration plate of a prototype machine.

Quite certainly, industry level machine vision has to obey the rule of high and deterministic performance at lowest prices. In our case the system has to perform at 100% detection capability. This means that we must always arrive at a conclusion with 100% certainty for each predefined and allowed situation. Typically one recognizes correct and false cases but to operate at 100% one has to introduce undecided, that is, deterministically uncertain case, as well. As will be seen later we actually define three states for our machine vision dice-throw detection — dice can be either recognized with some appropriate number facet, dice can be falsely positioned[1] or the system can not determine what it detects.

In its typical operation the machine operates in a repetead cycle where it accepts the players' bets through the console, starts throwing the dices by shaking the vibration plate, resting the plate, and performing a read-out of dices by the means of a contact-less electronic card device [2,3]. Figure 1 shows a typical three-dice configuration which is the central problem of our discussion.

[1] Is not positioned with exactly one facet on the ground and having the opposite facet unobstructed from above.

As the average time for the electronic read-out is 3.0 seconds, we estimated that the maximal time for the machine vision module to accomplish its task should be half as much.

2 MACHINE VISION READ-OUT OF DICES

The central part of our submission deals with theoretical and implementational details of our machine vision detection system. In particular we discuss our top-down hierarchical image analysis approach where we use

- *low level color* images for dice location estimations by applying the color chroma-key operator and axis-projection estimations [4], and
- *high level monochrome* images to determine the relevant detected situation.

For high level image processing we use a kind of a pattern template-matching technique to automatically devise a procedure that utilizes a 100% deterministic computational structure which produces reliable results.

2.1. Problem Statement

In Figure 2 we provide the snap-shot and dimensions of white dices with black spots denoting numbers. When the machine is being tested for reliability it produces dice configurations normally as in a real game—then it uses its internal detection mechanisms to collect data which is compared to data obtained through our vision system. The flat vibration plate where the dice configurations are produced is covered by greenish cloth and has a circular form in a working machine with a diameter of 50 centimeters.

This allows for use of color image acqusition as it enters more information to the actual processing stages of our machine vision module. It should also be noted by the reader that for high level monochrome images the relevant features are dice spots and that they should cover an image area of more than 1 pixel in order to be actually processed with enough precision.

If the dice throw in question is detected equally by both procedures then it is marked as OK, otherwise an image of the relevant throw is saved for later analysis marked as MISMATCH. If the throw

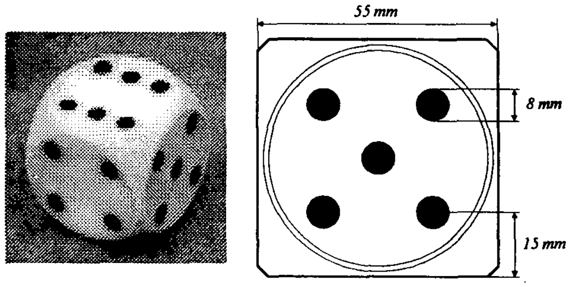

Figure 2: A close snap-shot and dimension specifications of a white dice with black spots etched to the hard-plastic facets.

is false due to a burning dice (see introductory discussion) it must be detected as such by both systems and marked BURNED, or else we again have a MISMATCH situation.

2.2. Color Chroma-Key

The fundamental operation used throughout our machine vision module is a color chroma-keying which is based on a color difference function. As we have reported on our color difference and chroma-key function in [6,7] already let us present them here only briefly.

To start with, we convert the ordinary RGB color triplet to an HSV color space with hue, saturation and value components by the well-known conversion procedure, also described in [5]. Having done that we are ready to define a procedure to calculate the color difference (or similarity if we change a point of view) between two colors (HSV color triplets) ***a*** and ***b***. First, HSV components of some color *c* are converted to their Cartesian XYZ values, if we consider an inverted HSV piramide to have its peak in the 3D Cartesian origin, by:

$$c_X = c_V\, c_S \sin(2\pi\, c_H)$$
$$c_Y = c_V\, c_S \cos(2\pi\, c_H)$$
$$c_Z = 0.8\, c_V$$

It is our empirical finding that it is appropriate to additionaly scale the value component (along the Z axis) by 0.8 to lower the impact of brighter/darker variations of colors as compared to other two components.

Finally, the color difference value is obtained by:

$$d_{XYZ} = (a_X - b_X)^2 + (a_Y - b_Y)^2 + (a_Z - b_Z)^2,$$
$$d(a,b) = e^{-d_{XYZ}\, f_{base}},$$

where the parameter f_{base} serves as a sensitivity adjustment control. The smaller the value the lesser the actual discrimination between colors a and b. The chroma-key principle uses this function to calculate the difference between some defined color c and pixel color triplets over some image area. This produces the image mask with masking values between 0.0 and 1.0 for background and non-background information, respectively.

2.3. Dice Location Processing

Let us move on to the problem of fast dice location processing where we use two examples of dice configurations in Figure 3. Low level image size is set to 66 times 66 pixels. To estimate the dice center locations we

- apply the above chroma-key principle to eliminate the background color information, and
- use some smoothing (convolution with fixed window calculations) of two one-dimensional vectors which represent mask values projected along each image axis, supplemented with approximate maxima finding.

Figure 4 shows the image masks and axis projection plots for examples from Figure 3. For the chroma-keying we use an averaged background color together with an averaged parameter f_{base} which, under required lighting conditions, removes (keys out) the background to a satisfactory degree. The actual average removal ratio is later used to estimate the removal ratio threshold. So we can detect if the lighting conditions are changed in the chroma-key phase. The size of the moving convolutional window is also preset to a fixed number of pixels as the camera camera distance and dice sizes (measured in pixels) practically do not change. Plots in Figure 4 suggest that some smoothing has to be applied.

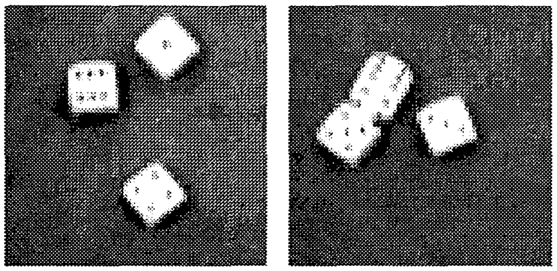

Figure 3: Two examples (low level images) of dice configurations with the right representing a throw with a *burning dice*.

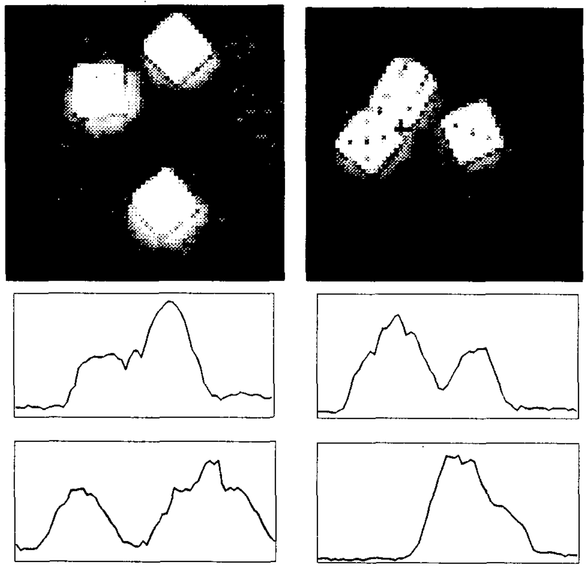

Figure 4: Image masks computed from low level images from Figure 3 together with the corresponding X and Y projection plots. For the purpose of example we applied a chroma-key with slightly different parameters for the left and right image.

We convolve the projected mask values by a predefined bell-shaped convolutional vector and obtain a function which can have from 1 to 3 peaks (the possible number of approximate dice locations). To obtain the estimating intervals of possible dice locations we devised the heuristic procedure which uses threshold cuts as the following pseudo-code explains. In the pseudo-code we use italics to denote non-programatic explanations.

```
for threshold values
  cut = threshold value
  npeaks = 1, reg[npeaks] = largest
  in = true
  for v[i] {
    if (v[i] > cut && in) {
      reg[npeaks][1]= v[i]
      in = false
    }
    else
    if (v[i] < cut && !in) {
      reg[npeaks][2] = v[i]
      npeaks++
      in = true
    }
    if (npeaks > 3)
      stop and use previous peaks data
  }
  if (npeaks < ppeaks)
    stop and use previous peaks data
  previous peaks data = npeaks, reg[][]
}
```

We assume the quantification levels, also called the threshold values, are predefined, the first being the maximal value of vector holding the axis projection.

By sequentially processing both vectors (X and Y axis projection) and combining them properly we have thus obtained as an output up to three image regions which have to be explored in further detail for actual number or erroneous detection.

2.4. Dice Number Detection

As we mentioned in the introduction we use a template matching technique to fine-tune the facet location and detect the number of spots. Here, the templates are actually small, local mask images with predefined color difference parameters for color white of dice facet and color black of dice spots.

The template actually consists of three components: the facet detection mask values, the spot(s) detection mask values, and the circular, truly pixel masking value.

The automated procedure which takes into account a finite number of possible facet orientation precomputes all template masks involved. We therefore have a different number of templates for numbers 1 to 6 due to the difference in their rotational and/or simetry invariance. Quite obviously we need only one template for number 1, half the total turn for numbers 2, 3 and 6, and only a quarter of the total turn for numbers 4 and 5. The templates are again preset in size according to the image acquisition characteristics and the actual patterns are calculated from dice dimension specification (see Figure 2).

An example of an actual template for number 6 used in one line of our development-cycle tests is given in Figure 5. The degree at which the template matches the underlying image is calculated by the same principle as is used in our edge detection method described in [6]. Here, let us briefly point out the major steps:

- We denote the variables and symbols which deal with facet and spot detection by + and -, respectively.
- As the black and white colors are opposite we can use the facet matrix only to calculate the color differences for each (mask blended) pixel of facet mask and underlying image region by:

```
allign template to image region I
for image pixels I[x,y] {
  d⁺[x,y] = v⁺[x,y] * d(white, I[x,y])
  d⁻[x,y] = v⁻[x,y] * d(white, I[x,y])
}
```

- And finally obtain the template matching degree:

```
S⁺ = Sum(d⁺[x,y]), S⁻ = Sum(d⁻[x,y])
match = S⁺ > S⁻ ? S⁺ - S⁻ : 0
```

In the current procedures we use an exhaustive search through the entire template database and pick the best matching as a result.

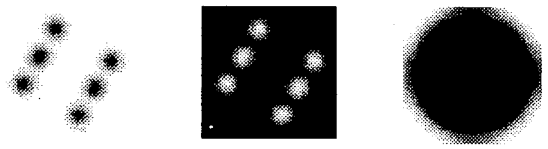

Figure 5: An example of a template for number 6. It consists of three sub-masks for facet, spot and irrelevant pixels, all of size 22 times 22 pixels. If we consider a rotational precision of 36 per total turn, storing a mask value in 1 byte, it means we are dealing with very small template database of less than 36 KB (73 templates times 484 bytes). The compromise however has to be achieved since computation times rise exponentialy.

3 CONCLUSION

Since the production line is still under construction we are not able to provide the reader with an on-line data on reliability tests and performance parameters (as were defined by the manufacturer). The system has to meet stringent time specifications for single test to run as quickly as possible and at the time of our writing the code is non-optimized and runs on an equipment which does not enable cost effectiveness.

In computer vision in general it is hard to provide formal proofs of operation due to large domain spaces and inherently complex, non-sequential processing requirements.

REFERENCES

[1] *Victory dice gambling machine*, Technical Specs & Data, Victory, Sint-Niklass, Belgium.

[2] Urban Bergant, Nikolaj Zimic, *Precise and reliable electronic read-out of mechanical random number generator using dices*, Procs. of CITS, Croatia 1999.

[3] *4002 Read-Only Contactless Identification Device*, Technical Data, EM Microelectronic.

[4] Morton Nadler, Eric P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons, Inc., 1993.

[5] Allan H. Watt, Fundamentals of Three-Dimensional Computer Graphics, Addison Wesley, 1989.

[6] Iztok Lapanja, Nikolaj Zimic, Miha Mraz, Jernej Virant, *Edge Detector: Towards the Solid Ground of an Image Retrieval System*, Procs. of IIS, Bahamas, 1997.

[7] Iztok Lapanja, Nikolaj Zimic, Miha Mraz, *Optimal local chroma-key definition with application to hair extraction from digital images of people*, Procs. of ISPR "In Memoriam Pierre Devijver", Belgium, 1999.