Who's Counting?: Real-Time Blackjack Monitoring for Card Counting Detection

Krists Zutis and Jesse Hoey

School of Computing, University of Dundee
{kristszutis,jessehoey}@computing.dundee.ac.uk

Abstract. This paper describes a computer vision system to detect card counters and dealer errors in a game of Blackjack from an overhead stereo camera. Card counting is becoming increasingly popular among casual Blackjack players, and casinos are eager to find new systems of dealing with the issue. There are several existing systems on the market; however, these solutions tend to be overly expensive, require specialised hardware (e.g. RFID) and are only cost-effective to the largest casinos. With a user-centered design approach, we built a simple and effective system that detects cards and player bets in real time, and calculates the correlation between player bets and the card count to determine if a player is card counting. The system uses a combination of contour analysis, template matching and the SIFT algorithm to detect and recognise cards. Stereo imaging is used to calculate the height of chip stacks on the table, allowing the system to track the size of player bets. Our system achieves card recognition accuracy of over 99%, and effectively detected card counters and dealer errors when tested with a range of different users, including professional dealers and novice blackjack players.

1 Introduction

Who's Counting? is a computer vision based software prototype designed to track a live game of casino Blackjack, with the primary goal of detecting players that use the technique of card counting in an attempt to gain an edge over the casino. Blackjack (or *"21"*) is the most popular casino table game in the world, in which the aim is to end the round with more points than the dealer, while remaining equal to or under 21 points. The only proven method of gaining an edge over the house is card counting, in which a player tracks which cards have been played, allowing him or her to make optimised betting decisions. Without card counting, the house has a 0.5% higher chance of winning a round. Although card counting without the use of physical aids or equipment is not illegal, the majority of casino establishments do not allow players to count cards. With the rapid advances in portable technology, it has become easier for the average Blackjack player to card count, for instance through the use of an iPhone application that makes counting cards easier, acquired by 500 new users every day [11]. Combined with an increased awareness of card counting due to movies such as "21" , casinos are forced to deal with more card counting attempts.

This paper describes our research into the use of automated systems for detecting card counting. In a user-centered design process, we surveyed and interviewed casino employees, uncovering requirements for such a system. The main requirements were (1) speed, as blackjack games can be very fast; (2) cost, as casinos are a money-making business; and (3) non invasiveness, as blackjack players and dealers are intolerant of any technological devices on the table or in the chips/cards. Interestingly, these requirements were more important than accuracy, with dealers stating that they would tolerate a small amount of interaction with the system to ensure 100% accuracy.

We describe a real-time (5fps) system to detect card counting from a single overhead stereo camera. The system uses a combination of contour analysis, template matching and the SIFT algorithm to detect and recognise cards. Stereo imaging is used to calculate the height of chip stacks. The outputs of these two algorithms are combined with a temporal analysis to detect if a person is counting cards based on the pattern of their plays and bets over a period of time. The two key contributions of the work are a user requirements gathering and the demonstration of stereo for chip stack value counting.

2 Background

There has been relatively little work on card counting or card game monitoring from cameras. Recent efforts by casinos have focussed on radio-frequency identification tags (RFID) embedded in cards and chips. However, these solutions are prohibitively expensive, and susceptible to fraud. Due to the blind broadcast nature of RFID, experimentation kits have become widely available (from e.g. `ThinkGeek.com`), and RFID systems have been plagued with security concerns that players would be able to broadcast a compromised signal, possibly representing different chip denominations. While encryption within the chips can provide an extra layer of security, this greatly increases the cost of each chip.

Computer vision gives an ideal simple, fast, and inexpensive solution. However, there is little published work in this direction. Clear Deal [4] used a combination of line detection, corner detection and template matching to detect the value of the cards as they are dealt throughout the game. The system analysed the quality of the shuffle carried out by the dealer by comparing the deal across hands, and detected card counting by monitoring game decisions and comparing them with basic strategy. However, this system had no way of monitoring the size or variation of bets placed by the player. Card counting strategy shows that 70-90% [6] of the edge developed by a player is applied by changing the size of the bet as the count fluctuates, whereas the remaining 10-30% of the advantage goes towards the ability to alter game decisions which, therefore, lead to a higher proficiency. Zaworka [12] tracked a Blackjack game by detecting cards and players' chip stacks as they are bet, in real time. Overall accuracy was 97.5% for detecting playing cards and chip stacks, even with occlusion. However, the system only detected the presence, not the values, of cards and chip stacks. The system used an electronic chip tray, whereas ours uses only computer vision. Template matching and a combination of heuristics was used by Zheng [13] to match cards invariant to rotation, but the technique did not handle face cards well, did not model chips or bet sizes, and did not produce a final usable system. The recognition rate was 99.8% over a range of rotations.

There are a few commercial attempts to market systems for card counting monitoring. Tangam Gaming (`tangamgaming.com`) produces an automated card recognition system that requires the use of speciality hardware such as RFID. The MindPlay21 system relied on a range of specialized hardware which included 14 cameras, invisible ink, and RFID tags. Cameras were used to scan the cards as they were dealt, as each card had been marked with a unique barcode painted in special ink [10]. The cost of $20,000 per table, the unreliable components and the slow speed of operation led to the company going out of business in 2005.

Generic object recognition has seen much use in the past decade [9]. There are a number of discriminative approaches proposed, perhaps the most common of which is the use of invariant features [8]. In particular, the scale invariant feature transform (SIFT) is used in many areas of image processing, such as 3D modelling, image stitching and object recognition [7]. SIFT extracts distinct keypoints from an image, which can be compared to other sets of keypoints to look for matches. The keypoints extracted by SIFT are invariant to scale, rotation, and location, with partial invariance to view point angle. Template matching is still in use due to its simplicity and efficiency [2].

3 Requirements Gathering

Our primary goal was to develop a working prototype that would meet the requirements of casino operators, croupiers, and players. Thus, a user-centered design approach consisting of informal interviews and discussions, observations of blackjack games, questionnaires, and consultations with local casino employees and managers uncovered design requirements from an end-user perspective.

The questionnaire contained three sections. The first section aimed to find the level of necessity for the proposed system. The second portion gathered non-functional requirements that would determine the performance aspects of the system, and the final part of the questionnaire was used to extract the usability requirements for the system. We received 7 completed questionnaires.
Our requirements analysis uncovered the following needs:

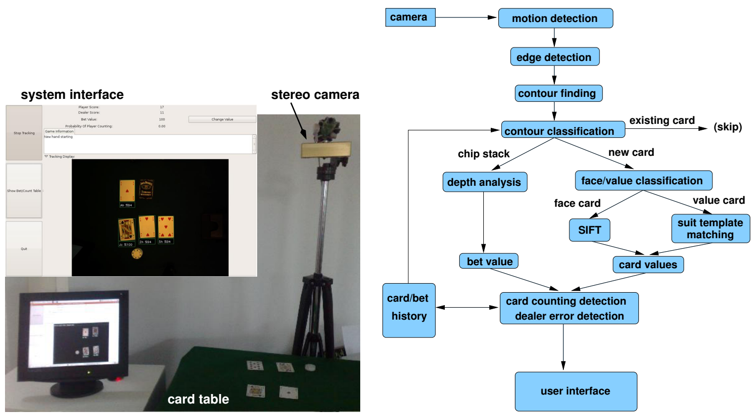
- a necessity for, and an interest in automated card counting detection systems,
- most dealers have suspected a player of card counting,
- error detection features were desired,
- players tend not to play at tables where any technology is visible, particularly if there are cameras at table level.

The requirements analysis also uncovered the following design requirements:

- speed of card detection (2 seconds or less)
- accuracy should be 95% or higher.
- interaction with the system was considered acceptable.
- a graphical user interface (GUI) was preferable to a textual one for croupiers

4 System Description

Figure 1(a) shows a view of the system, showing the overhead stereo camera (Pt. Grey Research 640x480 Bumblebee2 using IEEE 1394 link), the card dealing table, and the croupier's interface. The computer is an Intel Quad Core 3.2 GHz Processor and 4 Gigabytes of RAM running Linux Ubuntu 8.10. The software is written in C/C++, and uses the OpenCV library [1].



(a) (b)

Fig. 1. (a) Who's Counting? system (b) schematic of the software.

Figure 1(b) shows a block diagram of the algorithm used. Briefly, the video stream from the overhead camera is fed to a motion detection algorithm that locates stable images (without hands etc) (Section 4.1). Edges and contours are then extracted from the stable images, and are classified into existing cards, new cards, suit blocks, or chip stacks (Section 4.2). Chip stack contours are then further analysed using stereo to calculate the number of chips in the stack (Section 4.3). Suit blocks are saved for future processing, existing cards are recorded, and new cards are classified into face or value cards (Section 4.4). Face cards are then compared to stored models using the SIFT algorithm (Section 4.4). All suit block contours within a value card are counted using template matching to detect the final value of the card (Section 4.4). All card detections also have an associated confidence measure. Finally the card and bet values are passed to the card counting detection system, which stores the current hand, compares the hand to the history, and relays information to the croupier (Section 4.5).

The system uses multiple parameters and thresholds that are specific to a given setup, and are manually specified. The 'area' parameters are dependent on the distance of the camera from the table surface, and the thresholds for contour extraction are dependent upon lighting levels (assumed constant).

4.1 Motion Detection

Motion is detected using image differencing, and triggers a second phase where the system looks for minimal image differences. The first subsequent image with an absolute difference of less than 8000 pixels is valid. This avoids capturing images with dealers or player's hands. Figure 2(a) shows an example.

4.2 Contour Detection and Classification

Canny Edges are found [3], followed by contour-following analysis in OpenCV [1]. Each contour is classified into one of three categories (card, chip stack or suit

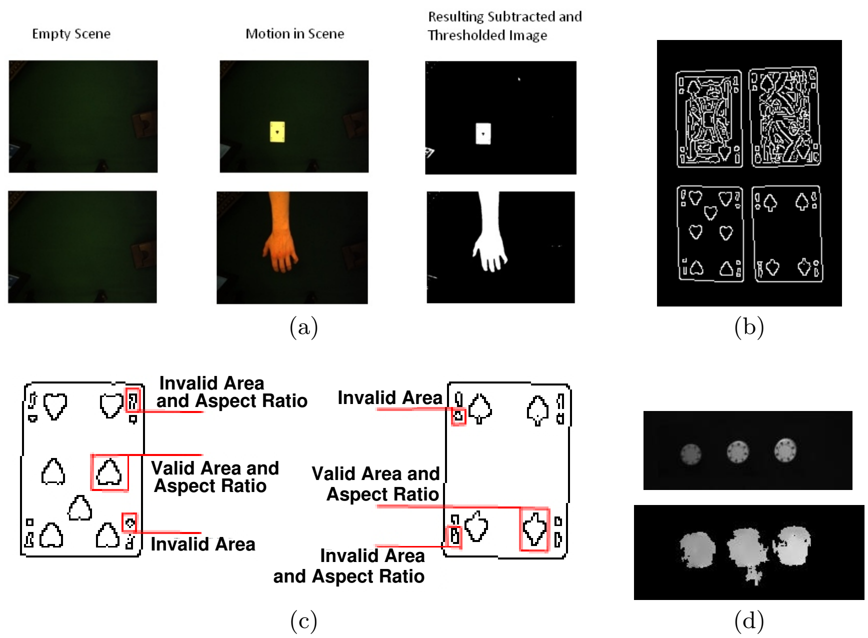


Fig. 2. (a) Example image differencing (b) Binary edge images showing (top row) Face cards with 1407 and 1637 edge pixels and (bottom row) value cards with 630 and 549 edge pixels (c) Binary edge image of a 9 and a 4 of hearts shows valid suit blocks (d) image and disparity map of chip stacks 1, 5 and 10 chips high.

block) using its area, centroid, and geometry. The classification parameters are dependent on the camera height, here we use 86cm. Any contour with 4 vertices and with an area between 4500 and 12000 pixels is classified as a card. Vertices are found using the OpenCV library for contours. These contours are then further classified as existing or new cards, and as dealer or player cards depending on their centroid. Any contour with an area between 70 and 220 pixels, and with an aspect ratio between 0.5 and 1.5 is classified as a suit block (see Figure 2(c)). The centroid of the suit blocks are used to assign them to cards. Finally, any contour with an area between 1000 and 2000 pixels, with an aspect ratio of the surrounding rectangle between 0.8 and 1.2, and with a Hough transform that matches a circle [5] is classified as a chip stack.

Our system does not compensate for unenclosed contours, thus, a full contour around the circumference of the card and each suit block must be detected in order to successfully recognise a card. We found that, due to the clear difference between the suit blocks and card background, the hysteresis thresholds in the Canny edge detector were easily set to ensure full contours.

4.3 Chip Counting using Stereo

In a casino, a bet is placed by stacking chips of the same value, and then placing that stack onto the bet area of the table. Our requirements analysis showed that players were not in favour of cameras at table level, so stereo vision from above was used. A BumbleBee2 stereo camera gave accurate and reliable dense stereo estimates, and a depth resolution that was just adequate to detect an individual chip. The average disparity given by the stereo correspondence algorithm at the chip location given by the contour analysis (see Section 4.2) was calculated as the mode of all disparity values within the chip area. This disparity result was then converted to depth and compared to the pre-calibrated chip stack height values to determine the size of the bet. This calibration was learned from a number of runs of the stereo algorithm on chip stacks of known height.

4.4 Card Classification

Cards were classified as face or value cards for further processing using a simple method of counting the number of edge pixels within the card area and comparing to a threshold (1100 was used in our experiments). This method returned consistent results, an example of which can be seen in Figure 2(d). We now describe how the values of face cards and value cards were determined.

Face Cards We only matched SIFT features on the local areas of each face card: a segmented image size of 100 by 130 was sufficient to find a strong match (100 to 200 keypoints) while maintaining speed. Two keypoints match if the Euclidean distance is less than 0.6. Confidence estimates for face cards were computed by counting how many keypoints the best match had compared to the second best match. A difference of three keypoints was allocated a confidence of 90%, with a further 2% being added for every additional keypoint matched.

Value Cards The value of a card is determined by counting the number of suit blocks that agree with the highest confidence. A fast normalised cross correlation template matching algorithm is used on each candidate suit block, and all those with a correlation of more than 0.4 are kept. Each suit's template is matched twice, once at the original rotation, and the second time with a 180° vertical rotation. The value of the card is then assigned to be the number of matched suit blocks, N. To determine the suit and confidence of the card, suit block i is matched to suit $k_i \in \{h, d, c, s\}$ with maximum correlation r_i. The card's suit, k^*, is then determined as the one with the highest summed correlation, $k^* = \arg\max_k \sum_{i=1}^{N} \mathbb{1}(k_i, k) r_i$, where $\mathbb{1}(k', k)$ is an indicator function returning 1 if $k' = k$ and 0 otherwise. The card's confidence, c^* is the mean correlation after assigning 0% to all suit blocks with $k_i \neq k^*$: $c^* = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(k_i, k) r_i$. It was found this non-linear method yielded more consistent results than a simple maximum of the average of suit correlations.

The area and aspect ratio attributes used to identify candidate contours are invariant to rotation and location, but not the template matching. Automatic rotation of cards to a canonical orientation would help with this problem.

4.5 Card Counting, Dealer Error Detection, and User Interface

To detect card counting, the system looks for correlations between a player's bets and the *card count*, a measure of the likelihood that the next card will have a high value. The *card count* is measured using the "Hi-Lo" method, in which a running count is kept, starting at zero. Each card that is dealt with a value from 2 to 6 adds +1 to the count, 7 to 9 adds 0 to the count, and aces, 10s and face cards all add -1 to the count. If a player increases their bets when this count is high, then card counting is likely. The Hi-Lo card counting strategy was employed because it is the most commonly used method by card counters. However, other methods could also be substituted.

We used a correlation coefficient to calculate the relationship between the sequence of bets b_t and card counts d_t, for $t = 1 \ldots T$ as $r = ((T-1)\sigma_b\sigma_d)^{-1} \sum_{t=1}^{T}(b_t - \bar{b})(d_t - \bar{d})$, where σ_b and σ_d are the sample standard deviations of b and d. A simple threshold on this value is then used to raise an alert to a dealer.

To detect dealer errors, such as paying out a losing hand, our system monitored the payout phase of a round of Blackjack. For any given hand, the game could end in a payout to the player, a loss for the player (removal of the player's chip stack), or a tie (player's chip stack is untouched).

The system required a graphical user interface to show the status of the game and any warning to the dealer. Since the dealer's job is to deal the game of Blackjack, only the basic features are necessary so that the dealer could simply glance at the GUI and ensure that everything was in order. Since testing showed that stereo imaging alone was not reliable enough to calculate the size of the bet placed by the user at all times, an additional input field in the interface allowed the dealer to manually adjust the bet value.

5 Evaluation

Testing was split into three sections, accuracy testing of card recognition and chip stack height, testing of the card counting detector, and user testing to evaluate the system as a whole from the user's perspective.

5.1 Accuracy

To test the accuracy of playing card recognition, each card of two decks was placed onto a green felt lined table ten times. The stereo camera was 86cm above the surface of the table. 8 cards of the same value were then placed onto the table and the results were recorded. This was repeated five times for each set of 8 cards, giving a total of 65 images, with each card being tested ten times. The processing times for a new play was below the required time of 2 seconds.

We first tested the card outline detection and found the system was able to correctly identify cards placed in the scene 100% of the time. We then tested the face/value card differentiation, also finding 100% accuracy. We found 100% accuracy in detecting face cards for both value and suit.

Each value card was tested 40 times, since each card was tested 10 times, and there are 4 suits. The correct value of the card was detected correctly 399 out of 400 times, showing an overall accuracy for card value detection of 99.75%. The results for each suit are Spade: 87%, Club: 98%, Heart: 98%, Diamond: 84%, with an overall accuracy of 92%. However, note that the suit of a card has no impact on the game of Blackjack.

The accuracy of bet value detection was tested using chip stacks of varying heights from 1 to 10 placed directly under the camera. Out of a total 100 chip height calculations, with ten for each height, the correct values were detected 97 times. The first two errors occurred when detecting a 1 chip high stack, and resulted in an underestimated stereo depth calculation. The third error came from unsuccessful chip detection from the image processing algorithm with 9 chips in the stack. The results conclude that out of 100 placed bets, 97% of the bet values were successfully calculated. The stereo algorithm worked 97 of the 99 times it was run, giving an overall accuracy level of 98%. The chip detection algorithm had overall accuracy of 99%.

5.2 Card Counting

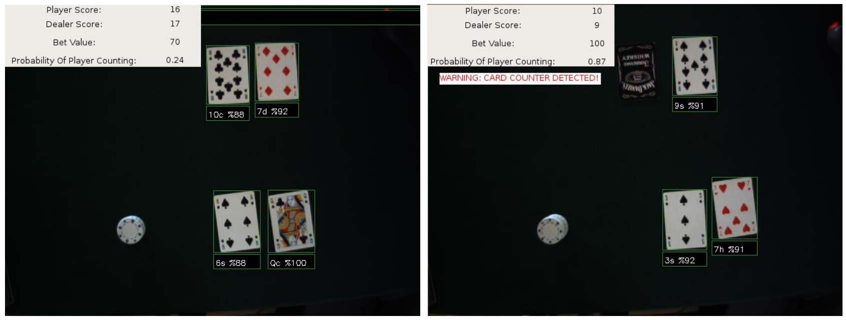
To test the system's ability to detect card counters, 30 games of Blackjack, each with 20 rounds, were played by the authors, for a total of 600 rounds. In fifteen games, the cards were dealt randomly and the player played according to standard rules (no card counting). In the other fifteen games, the player used the "Hi-Lo" card counting system to vary his bets. The card count correlations (r) for each of the games are shown in Table 1. Thus, we see a threshold on r set between 0.45 and 0.62 would be effective. Games 12 and 14 of the card counting tests gave outlying results. These were caused by a continuous negative card count throughout the test period. In an actual situation such as this, a card counter would continue to bet the minimum amount of chips until the cards shifted to a positive count. However, in these tests, the count did not become positive within the twenty test rounds, and resulted in a low correlation score. Figure 3 shows two screenshots of the system in action, with one showing a card counting detection.

card	game number														
counting	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
no	-0.62	-0.21	0.30	0.27	-0.32	0.36	-0.34	0.33	0.43	0.37	0.28	-0.09	0.32	0.14	0.44
yes	0.68	0.91	0.93	0.89	0.77	0.96	0.71	0.90	0.62	0.82	0.86	0.27	0.65	0.32	0.90

Table 1. Results of 30 games showing the correlation r between bets and card counts.

5.3 User Tests

The complete system functionality was tested with users of different backgrounds to evaluate the usability and overall performance. The test group for this study consisted of two professional Blackjack dealers, two average Blackjack players, and four inexperienced players. At the beginning of the test, the background and limitations of the system were described, and example hands were shown to the participants. One participant was then asked to deal a game of Blackjack, while the experimenter played the role of player. Each participant played 15



(a) (b)

Fig. 3. (a) The interface output after ten rounds of Blackjack, when monitoring a non card counting player. The output shows the state of the current round, as well as the correlation coefficient result of 0.24 - a result which is below the card counter threshold of 0.75. (b) System output when monitoring a card counting player for ten rounds of Blackjack. A correlation coefficient result of 0.87 indicates card counting.

hands of Blackjack. As well as being observed during the testing, the participants were interviewed after the tests to find the successful and unsuccessful areas of the system. There was general approval of the system. The system successfully detected card counting, and even withstood an attempt to trick the system by placing extremely large bets during the first few hands, then dropping the bet value and proceeding with normal card counting techniques. The system was still able to detect the card counter in fewer than 20 hands after this attempt.

The dealer error detection was also successful, since it not only detected errors which were purposefully included, but also detected genuine, unintended errors. The system detected incorrect dealer payouts, where the dealer wrongly paid out, or collected the chips when the player had won. However, testing also identified certain areas where the system could be improved. The majority of requests were to update the interface to display information more clearly. Dealers would sometimes adjust the card's position on the table, resulting in the card being recognised twice. This was fixed by adding a buffer area around each card, allowing it to have slight changes in location after it has been placed.

6 Conclusion and Future Work

This paper presented a system for detection of card counting and dealer errors in casino blackjack. The system uses a combination of computer vision techniques to track cards and bets, and uses a correlation algorithm to detect card counting. Our requirements analysis and system implementation have demonstrated the need for, as well as the success of detection of card counters. By introducing a pure computer vision solution, the system showed that a cost effective, discrete system can be used to automate casino surveillance.

Several elements could contribute to future developments of similar applications. Contour analysis was well suited to real-time processing needs, and template matching could be used in real time applications, if the region of interest is localised to a small area. The SIFT algorithm functions well for matching objects even when the database of keys taken from images is limited. Therefore, lowering the image resolution and size before extracting keys for the database results in less keys and a much faster recognition time. A significant result from this research was that stereo imaging can be successfully used to detect the height, and therefore the value of chip stacks. In conducting background research, we found that the ability to identify the value of player bets is a highly desired tool from a casino's perspective. This is a particularly promising area to proceed with further research, as it is a feasible method of monitoring chip values without the need for extra hardware, such as RFID tags. Improvements to the system will be more comprehensive detection of unexpected events, support for mutliple players, dealing with occluded cards, automated card and chip threshold calibration, and application to other games such as Roulette or Three Card Poker.

Acknowledgments. The authors thank the users who participated in the surveys and tests, and the staff of the Dundee casino.

References

1. Gary Bradski and Adrian Kaehler. *Learning OpenCV.* O'Reilly Media, 2008.
2. Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.
3. John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.
4. Wesley Cooper and Kenneth Dawson-Howe. Automatic blackjack monitoring. In *Proc. of Irish Machine Vision Conference*, pages 248–254, 2004.
5. Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.
6. Ken Fuchs. Knock-out blackjack - the theory. koblackjack.com website, April 22 2008. available at: http://www.koblackjack.com/system.html.
7. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
8. Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(27):1615–1630, 2005.
9. Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors. *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*. Springer, 2006.
10. Daniel Terdiman. Who's holding the aces now? Blackjack Universe website article, August 18 2003. available at: http://blackjackuniverse.com/pages/news/mindplay.html.
11. Daniel Terdiman. Casinos on lookout for iphone card-counting app. CNET news article, February 17 2009. available at: http://news.cnet.com/8301-10797_3-10165861-235.html.
12. Arno Zaworka and Stefan Scherer. Machine vision driven real-time black jack analysis. In *Proc. of* 24^{th} *workshop of the AAPR*, 2000.
13. Chunhui Zheng and Richard Green. Playing card recognition using rotational invariant template matching. In *Proc. of Image and Vision Computing New Zealand*, pages 276–281, December 2007.