

Implementation of Vision-based Object Tracking Algorithms for Motor Skill Assessments

Beatrice Floyd and Kiju Lee

Department of Mechanical and Aerospace Engineering
Case Western Reserve University
Cleveland, Ohio 44106

Abstract—Assessment of upper extremity motor skills often involves object manipulation, drawing or writing using a pencil, or performing specific gestures. Traditional assessment of such skills usually requires a trained person to record the time and accuracy resulting in a process that can be labor intensive and costly. Automating the entire assessment process will potentially lower the cost, produce electronically recorded data, broaden the implementations, and provide additional assessment information. This paper presents a low-cost, versatile, and easy-to-use algorithm to automatically detect and track single or multiple well-defined geometric shapes or markers. It therefore can be applied to a wide range of assessment protocols that involve object manipulation or hand and arm gestures. The algorithm localizes the objects using color thresholding and morphological operations and then estimates their 3-dimensional pose. The utility of the algorithm is demonstrated by implementing it for automating the following five protocols: the sport of Cup Stacking, the Soda Pop Coordination test, the Wechsler Block Design test, the visual-motor integration test, and gesture recognition.

Keywords—Vision-based Object Tracking; Motor Skill Assessment; Multi-marker Tracking; Computer-based Assessment.

I. INTRODUCTION

Assessment of upper extremity motor skills often involves manipulating physical objects, hand drawing and writing, or performing specific gestures [1] - [7]. Early assessment of such skills can potentially lead to early diagnosis of any deficits and thus result in better treatment outcomes in the long term [1], [8]. For example, motor skills deficiencies can be observed and are symptomatic of a learning or developmental disability, a traumatic brain injury, and normal aging. Assessment of such skills by a human clinician may encounter several challenges such as high cost [9]; time constraints [3]; and inconsistent professional awareness and expertise in diagnosis [10], [11]. Advancement in computing and sensing technologies have enabled automation of such assessment tasks previously conducted by human administrators. Automation does not only improve the accuracy and efficiency of tasks, but also can accomplish tasks that were previously impossible using human skills alone [12]. The assessment of motor skills would in particular benefit from automation. This is partly because of the increased accuracy, efficiency, and consistency of the measurements, but more notably automation can result in quantitative information that would not be possible from traditional manual assessment methods. For example, the Box and Block Test of Manual Dexterity (BBT) could be automated by installing RF readers in the two boxes and embedding RF tags in all the blocks [13], [14]. The system was able

to automatically sense when the blocks were placed in either of the boxes based on the relative signals from the two readers [15], [16]. This resulted in the same assessment data as the manual assessment while being more time efficient and collecting more data about the blocks movements.

Automation of upper extremity motor skill assessment that involves object manipulation can be realized in two ways: i) by employing *active* objects with embedded sensing and communication capabilities or ii) using *passive* objects with an external sensing device(s). It may be a combination of the two. Over the past decade, a variety of active objects have been developed for a broad range of education, entertainment, and research purposes [17]. Several studies have used the sensor-embedded blocks for measuring three-dimensional (3D) spatial cognitive abilities by observing construction patterns and performance [18]-[20]. Learning Block is a digitally augmented physical block system enriched with a speaker and LED display [21]. It aims to function as a playful learning interface for children via embedded gesture recognition. Another interesting application is the use of a sensor-embedded block system, called Navigation Blocks, for tangible navigation of digital information through tactile manipulation and haptic feedback [22]. Tangibles is also an active object system designed for tangible manipulation and exploration of digital information [23]. There are also block systems integrated with sound feedback. For example, AudioBlocks and Block Jam features an augmented sound feedback mechanism to enable users to design musical sequences by manipulating the tangible objects with visual and sound feedback [24], [25]. Multi-agent autonomous interactive blocks and games were developed specifically for behavioral training of children with an autism spectrum disorder [26].

Most of the existing work on object manipulation and gesture detection using passive objects has been geared toward vision-based approaches [27], [28]. For example, a depth-sensing camera was used to build a height map of the objects on an interactive tabletop platform for recognizing objects and detecting interaction between the player and the objects [29]. PlayAnywhere is a projection-vision system that can detect hover and touch by a human finger on a tabletop with a projected image [30]. Another interesting system is called TouchSpace, which is a game environment that combines reality with a virtual game environment based on ubiquitous, tangible, and social computing [31], [32]. Vision-based systems, compared to the methods using active objects, allow flexibility in the game or test design and the types of applications. However,

most of these algorithms are computationally expensive [30]. In addition, sensing is limited to the vision range unless additional sensing devices are used. Using active objects with embedded sensors or combining the two approaches may overcome the limitations of a vision-only method, but the hardware can be costly, in particular if a large number of objects are employed, and it is difficult to make a versatile method due to inflexibility of the hardware [33], [34].

This paper presents an algorithm designed for assessing object manipulation skills and hand gestures using a single standard webcam. No additional equipment other than a webcam is required. The algorithm is based on color thresholding for initial localization and morphological operations to find the object's edges. The corners are then identified by transferring the edges and used for pose estimation in real time. The result is the three-dimensional (3D) pose of the object which can be used for test automation and additional behavioral assessments. The algorithm described here is for tracking well-defined objects or markers rather than directly tracking hands and arms to simplify the computational complexity. Tracking hand motions would give a lot of interesting information about the person's upper extremity motor skills as explored by other researchers [35]. However, it is not necessary or ideal for object-based motor skill assessments for several reasons. First, the assessments being automated do not rely on hand position information, but instead on the resulting position of objects. Thus it would be counterproductive to track the hands since it would add another layer of complexity to determine the objects position relative to the hands. Second, our goal is to make this method work in real-time on a computer with a normal computing capability. The objects with simple, known shapes can be tracked without requiring heavy computations in contrast to hands with irregular shapes. The utility of the algorithm is demonstrated by the following four applications: the sport of Cup Stacking, the Soda Pop Coordination test, the Wechsler Block Design test, and a simple hand gesture test.

II. THE ALGORITHM

A. Overview

Assessments of upper extremity motor skills often involves a set of objects that are manipulated by the person being evaluated or a sequence of tasks, such as extending the shoulder and twisting elbow [4], [5], [13], [36], [37]. Resulting measurements include the time for completion, accuracy, and extension/flexion range of each motion. Our approach aims to automate the evaluation process with real-time data collection by employing vision-based techniques using a standard webcam. The algorithm first identifies specific objects within a field of view, projects their position into 3D space based on known shape information, and then tracks them in real time. To simplify the processing time, the algorithm targets tracking objects being manipulated by or attached to human hands instead of directly tracking the hands. The output of this algorithm is the 3D pose of each object. The only requirements are that the item must have straight edges and it must be distinguishable from the rest of the environment by either its shape or color. Shape and color form a two-tiered classification structure that determines whether objects within a video frame are items of interest. These values can be altered depending on applications via calibration.

A major advantage of the presented algorithm over similar approaches [38], [39] is that it is versatile. The algorithm works with a variety of different markers without requiring reprogramming. The limiting factors are that the markers must be unique in the environment to avoid false detections. A detailed description of the algorithm is provided in the following subsections. Section II-B describes the item localization method based on the two-tiered classification scheme used to identify items of interest within the image and to detect the corner locations. Section II-C presents the pose estimation method using the corner points to project the item from the 2D image frame into the 3D real world frame using object shape information and internal camera properties. This process is called pose estimation and is a technique for extracting 3D information from a single camera frame. Lastly, Section II-D describes the camera calibration needed to introduce the algorithm to new markers and determine internal camera properties necessary for pose estimation. The codes were written in C++ and utilized OpenCV for computer vision implementations. The captured images are in the RGB format with a resolution of 640×480 pixels. Post-processing of the data for some applications was performed in MATLAB.

B. Item Localization

Localization is the process of identifying the location of the target item(s) in the image frame. We employ a two-tiered classification approach for localizing items of interest. A two-tiered system achieves a high degree of accuracy in identifying items as a result of the two different properties that are required to detect a matched item. Color and shape are used as the distinguishing properties. Color indicates the normalized color of the item within a certain color range. Shape is defined as the number and relative positions of an item's corner points. Explicitly, the algorithm searches images for items with a known normalized color, and then locates the edges of the items using morphological operations on the color regions. Those edges are then traversed to locate the items' corners. The resulting corners are the outputs of localization and can be used to estimate the 3D positions of the items using known shape information.

1) *Color classification*: The first step of localization is to segment the image in order to identify what parts of the input image could potentially be items of interest. Normalized color initially distinguishes the potential object regions within the image. It was chosen as the distinguishing property because it is not affected by adverse lighting conditions and represents the inherent color of an object [40]. Color normalization compensates for the intensity changes in lighting by forcing all intensity values to sum to 1. The well-known equations for normalizing the color at each pixel location in an image are used:

$$r = \frac{R}{R+G+B}; \quad g = \frac{G}{R+G+B}; \quad b = \frac{B}{R+G+B}$$

where $r + g + b = 1$. The intensity values correspond to the values of the three image planes (red (r), green (g), and blue (b)) that make up an image.

The color of an image is thresholded by examining each pixel's values to determine whether it falls within certain threshold ranges. A binary value of 1 or 0 is assigned based on whether it passes the threshold or not,



Fig. 1: Examples of binary images after color normalization.

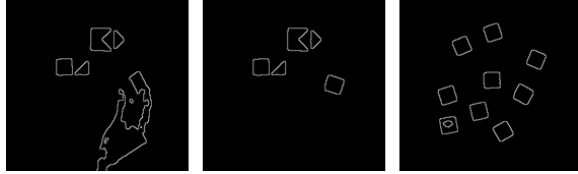


Fig. 2: Examples of edges found using the difference between a morphological dilation and the original image.

respectively. The ranges are defined by minimum and maximum values which are included in the objects color ($\{r_{min}, r_{max}\}, \{g_{min}, g_{max}\}, \{b_{min}, b_{max}\}$). These ranges can be easily identified for an object of interest by normalizing its color and finding the minimum and maximum color values for the object. Fig. 1 shows examples of the binary images resulting from color normalization.

The resulting binary image often requires additional processing to increase the accuracy. This process is necessary when color normalization fails to compensate for all imperfect lighting conditions or when the color threshold ranges are not completely accurate in reflecting the item's actual color. One technique is conditional dilation which can be beneficial when the color threshold detects only part of an item. It detects the rest of the item by expanding its area until it reaches the item's edges. The morphological operator of dilation is applied to the binary image but the results are only kept if the color values are close to the values of their neighboring pixels. Edges of objects are distinguishable by the dramatic change in the color range. Color values remain similar within the same item, but once dilation approaches an edge the values start to change quickly and exceed the acceptable range by the conditional dilation operator. In order to use this operator, the item's edges must be well defined.

2) *Shape classification*: The next stage of classification takes the outputs from color classification and further narrows down the regions of the image to detect the items of interest. Since color classification results in defined areas, the next step would often be blob detection. However, this is not the most convenient method in this case. Instead, the edges of the items are found and traversed in order to locate the corners of the item. The corners are then used to classify the item's shape as defined by the number of corners and their spacing. This method is chosen because it gives accurate positions of the corners used for classification that are essential for finding the object's 3D position. A morphological operator is used to find the object's edges through two steps. First, the color thresholded image is taken and a dilation operator is applied. The difference is then taken between the original image and the dilated image. A dilation operator expands the colored regions

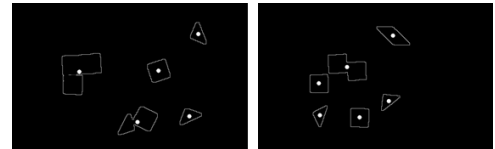


Fig. 3: Two images of blocks overlaid with centroids found after first traversal of each object's edges.

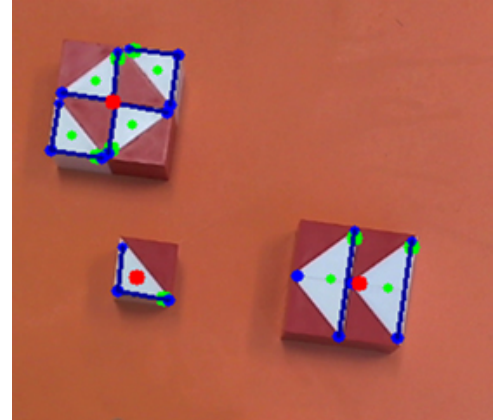


Fig. 4: A processed image showing the centroids of individual blobs and starting points (in green), overall centers (in red), and corners and block edges used to calculate tilt (in blue).

outward by operating on the image using a 3×3 rectangular structuring element. The result from taking the difference is an image containing only the edges of the colored areas. The edges are guaranteed to be 1-pixel thick, 4-connected, and form a closed loop. These three properties make them easy to traverse. The examples of these ideal edges are shown in Fig. 2.

The identified edges are traversed with the aim of pinpointing the location of the corners (Fig. 3). It takes three traversals of an edge set to find these corners. On the first traversal, spurs are removed, the object's centroid is found, and the edges are put in a stack so they can be accessed easily on the second two traversals. The edges are traversed by first finding a point on the image that is part of an edge. The next point is found by checking the four coordinate directions in the order of up, right, down, and left and then moving in the first found direction that is part of the edge. Each point previously visited is added to a stack of edges so that it can be referenced later and the location in the image is blacked out so that it is no longer recognized as an edge. A spur is recognized to exist on the edge when a point, that is not the beginning, is found to have no neighbors. At that point, the path is retraced by popping values from the edge stack until a new point is found that has a neighbor, meaning that it originally had two neighbors. An edge is considered complete when it loops back to its starting location. The centroid of each region is calculated by keeping a running average of pixel locations.

After the first traversal of an object all the edge points are conveniently in a stack and the centroid has been calculated. The next traversal is used to find a point on the edge that is guaranteed to be a corner. Since the objects have straight

edges, the edge location farthest from the object's centroid is guaranteed to be a corner point. This point is found by calculating the distance between the center and every point along the edge. The point that has the greatest calculated distance will be the corner point of interest. The last traversal finds all additional corners. They are found by moving along the edge and calculating the slope for each point edge. A constant slope designates the straight edge of an object and a rapid change in slope indicates a corner. After the corners are found they are amended by finding the intersection of lines fitted to the edges on either side of each corner. The resulting corner points are finally classified. If the number of points for an item is not equal to the expected number of corner points or the spacing of corners is not similar to that of the known shape, then the item can be concluded to not be an item of interest. For example, if the item of interest is a square then in order to be an object of interest there must be four evenly spaced corners. If it is found to be an object of interest then pose estimation can be used to get the object's 3D position. Fig. 4 shows an example of the processed image.

C. Position Estimation using Shape Information

The corners of the object, found through item localization, are used to estimate the position of the object in 3D space using known information about the object's shape and internal camera properties. The internal camera properties determine the perspective with which the camera views an object. By comparing the actual shape of an object with its warped shape within the camera frame, its pose relative to the camera can be determined. However, the relationship is nonlinear and typically cannot be solved directly. This can be circumvented using a variety of methods, including making additional assumptions about the object position or iterating to find the best values instead of solving directly. In an image frame, objects can be scaled and their perspective can be altered due to their relative position and orientation to the camera. If an assumption is made that the object's deformation in the image is either due to scaling or perspective then the equations can be simplified greatly. If it is assumed that the object has only been scaled, then the distance to the camera for all points on the object will be the same. At least two points are required to solve this system of equations, but the calculations can be made more accurate if more than two points are known.

The follow equation converts the $\{x, y, z\}$ image coordinate system to the $\{X, Y, Z\}$ real world coordinate system. The relationship can be described using their simple geometric relationship as shown in Fig. 5, given that the camera's focal length is broken down into f_x and f_y and the center of the image is at c_x and c_y . The Z axis is perpendicular to the image frame and both X and Y are parallel. The equations for this relationship are provided below and are rearranged so that they solve for the real world values. The two dimensions are independent and can thus be treated separately.

$$\begin{aligned} x &= f_x \frac{X}{Z} + c_x \rightarrow X = (x - c_x) \frac{Z}{f_x} \\ y &= f_y \frac{Y}{Z} + c_y \rightarrow Y = (y - c_y) \frac{Z}{f_y}. \end{aligned} \quad (1)$$

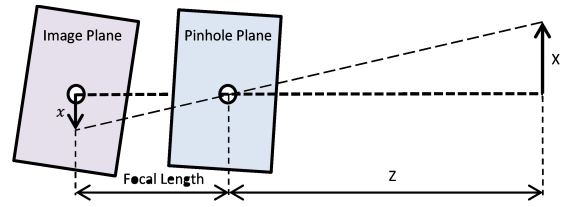


Fig. 5: The pinhole camera model in one dimension used for camera calibration.

There are three unknowns in (1), $\{X, Y, Z\}$. This problem is solved by using two points with a known relationship between each other and that are at the same distance from the camera. This provides a known distance between the points represented by the equation, $(\Delta X)^2 + (\Delta Y)^2 + (\Delta Z)^2 = d^2$ for a known d . The second simplification is that the points are at the same distance from the camera such that $Z_1 = Z_2 = Z$ since the Z -direction is perpendicular to the image. Under these assumptions, the real world coordinate is calculated by

$$\begin{aligned} X &= (x - c_x) \frac{\sqrt{\frac{d^2}{\left(\frac{\Delta x}{f_x}\right)^2 + \left(\frac{\Delta y}{f_y}\right)^2}}}{f_x} \\ Y &= (y - c_y) \frac{\sqrt{\frac{d^2}{\left(\frac{\Delta x}{f_x}\right)^2 + \left(\frac{\Delta y}{f_y}\right)^2}}}{f_y} \\ Z &= \sqrt{\frac{d^2}{\left(\frac{\Delta x}{f_x}\right)^2 + \left(\frac{\Delta y}{f_y}\right)^2}} \end{aligned} \quad (2)$$

where

$$\begin{aligned} \Delta X &= (x_1 - x_2) \frac{Z}{f_x} = \Delta x \frac{Z}{f_x} \\ \Delta Y &= (y_1 - y_2) \frac{Z}{f_y} = \Delta y \frac{Z}{f_y} \\ \Delta Z &= Z_1 - Z_2 = 0. \end{aligned}$$

D. Calibration

Calibration is an essential process to determine the conditions in which the camera is used and the properties of the item of interest. To locate an object, its color and shape must be known. The internal properties of the camera must also be quantified to determine how it views the item and to project the item from a 2D camera space into a 3D real space. The internal camera properties, or intrinsics, determine how a 3D object is projected into the 2D camera plane. The intrinsics includes the focal length and image center and are different for every camera. A camera can be represented by the pinhole camera model in which the light that the camera captures goes through a pinhole and is then projected onto the image plane. The focal lengths, f_x and f_y , are the distances in the x and y directions between the pinhole and the image center. The image center $\{c_x, c_y\}$ is the location of the pinhole projected onto the image frame. The geometric relationships between the 2D image and the 3D space are previously provided in (1). A commonly used object for determining the camera intrinsics is

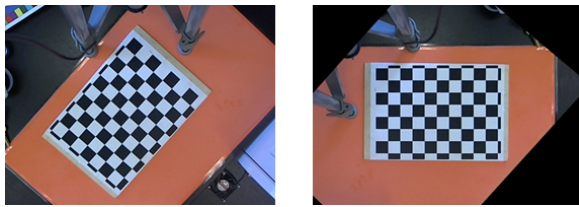


Fig. 6: A sample frame from a webcam showing the experimental set-up from a near-vertical camera view (left) and the transformed image to compensate for initial camera angle (right).

a checkerboard due to its defined number of points with known spacing. By analyzing the relative position of the checkerboard points within the image, the camera intrinsics can be found. Fig. 6 shows how camera intrinsics can be used to compensate for perspective and orientation undesirabilities. The left image frame shows the camera's perspective on the area and the right shows the perspective altered frame. It has been changed so that the corners of the checkerboard form a perfect square and aligned so that the work area lines up with the camera frame.

III. ALGORITHM IMPLEMENTATION

A. Overview

This section describes the applications of the computer vision algorithm previously described. The algorithm requires a uniquely colored square piece of paper to be placed visibly on the object to be tracked, or the object to contain a surface that is uniquely colored, and a camera to capture its motion. The paper/surface can be any color that is unique in the environment and the only requirement is that it stays visible throughout the motion. The versatility of the tracking algorithm is proven through its application to three different situations. The first is a sport played mostly by elementary school children called Cup Stacking. The second is a motor skill and coordination test developed by Hoeger & Hoeger called the Soda Pop Coordination test. The third is the Wechsler Intelligence Scale that is one of the most widely accepted psychological assessment tool. Among its subtests, we selected the Block Design test for the third application of our algorithm. For these applications, an automatic scoring system is implemented by identifying when certain events occur. In addition to these three specific examples, we also implemented the algorithm for potential applications in visual-motor integration assessment and gesture recognition.

B. Cup Stacking

Cup stacking (also called Sport Stacking) is an activity for individuals and teams in which specialized cups are used to create pyramids of three, six, and ten cups as quickly as possible. It is governed by the World Sport Stacking Association and a variety of studies have been conducted to assess its influence on motor skills [36], [41], [42]. Specifically, a study involving the second graders playing cup stacking for 15 minutes a day for 12 weeks showed that it might improve central processing and perceptual-motor integration skills [36]. Another study involving second and fourth graders playing cup stacking for 10-15 minutes a day for 3 weeks found no

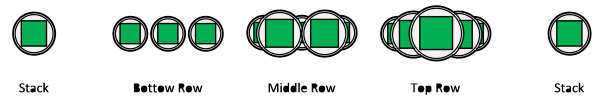


Fig. 7: Top view of the cups with green squares placed on the top illustrating five steps of six-cup stacking.

difference between a control group and a group participating in cup stacking [41]. It is also found that cup stacking is effective in improving hand-eye coordination and reaction time in second graders by playing the game for 20-30 minutes a day for 5 weeks [42]. It is notable that the sequences for stacking have a learning curve so cup stacking cannot be used to directly measure motor skills unless a training period is allowed.

The scoring of cup stacking was automated by placing a marker on the top of each cup. The 3D position of each cup, (X, Y, Z) , was found using pose estimation and then saved for further analysis. A six-cup stack game was employed as shown in Fig. 7. The automatic scoring was performed in real-time by recording when certain key actions occurred. The tasks included when the cups first started to move indicating the start of the activity, when three cups were placed as the base, when two cups were placed on top of the base, when the top cup was placed, and finally when all the cups come back together and stop moving indicating the activity is complete. A measure of the cup placement accuracy is determined by the straightness of the placement of cups in the bottom row and the relative angle between the bottom and middle rows, indicating how precisely the middle is placed relative to the bottom row. The automatic scoring component can easily be evaluated by comparing manually and automatically recorded trials. These values were compared for 91 different times and the resulting correlation is reasonable with an r-squared value of 0.9615 and an average error of 0.35 ± 0.27 seconds.

C. Soda Pop Coordination Test

The Soda Pop Coordination Test is a motor skills test that is a part of the American Alliance for Health, Physical Education, Recreation & Dance (AAHPERD) battery of tests. It is advantageous over similar tests because it uses commonly available materials and is easy to administer [37]. The test uses three soda pop cans and needs six marked locations on the table for the cans to be placed on, as shown in Fig. 8. In basic terms, the test involves flipping the soda cans over one at a time as fast as possible. Specifically, Can A is moved from position 1 to position 2, Can B is moved from position 3 to position 4, and Can C is moved from position 5 to position 6. Then the cans are moved back to their original positions in the reverse order. The hand must start with the thumb facing upward for the first set of movements and downward for the second set of movements. The test is usually scored using the time it takes to go back and forth twice and can be done for either the dominant or non-dominant hand.

The advantage of having an automated system for the Soda Pop Coordination test is that it increases the accuracy of scoring and makes data processing easier. Traditionally, the performance results would be a large amount of hand written

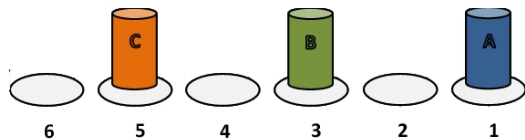


Fig. 8: Starting configuration for the soda pop coordination test with three soda cans and six locations

data (i.e. time, accuracy) that would have to be manually inputted into a computer. By having an automated system, the times are already saved on the computer and the mindless data entry step can be skipped, allowing for fewer opportunities for errors in recording. The Soda Pop Coordination test is usually administered before and after some training regimen to demonstrate how an action has improved a person's abilities [42], [43], [44]. It can also be administered to monitor coordination skills and then compared to or used to create standardized scores [37], [45]. Examples of before and after testing include a study to identify the effects of a 10 week Tai-Chi-Soft-Ball training on the physical functional health of Chinese adults [43], a 5 week study on second graders to identify the effect of 20-30 minutes of sport-stacking on hand-eye coordination and reaction time [42], and a study on the effect of a weight-bearing and water-based exercise program on osteopenic women [44]. Examples of using standardized scores include a study on the elderly, which showed the relationship between heart rate variability and coordination [45].

The test was automated by placing a marker on the top of Can A. The start time is set as the time the marker starts moving and the stop time is set as the time the marker comes back into view and stops moving. The marker will disappear as the cup is turned over and, in order to accommodate false starts, it is assumed to take more than two seconds to complete the test. Additionally, since the test has two sets of back and forth that count as one round, the numbers for the two consecutive times can simply be added together. The system was evaluated by comparing manually and automatically collected data to determine the accuracy and usability. Data was manually and automatically collected for 87 laboratory rounds of the Soda Pop Coordination test. The correlation between the manually and automatically collected data is 0.985 and the average difference in timing is 0.215 seconds.

D. Wechsler Block Design Test

The Wechsler Intelligence Scale for Children (WISC) and the Wechsler Adult Intelligence Scale (WAIS) are widely accepted psychological assessment tests used to measure intelligence in children and adults that were initially developed by David Wechsler in the 1930s [46], [6], [7]. Both scales contain a subtest called the Block Design test that measures a person's non-verbal conceptualization, spatial visualization, and fine-motor control [47]. The Block Design test was first proposed by Kohs in 1923 [48], but has been incorporated in some form into most intelligence tests. The WISC and WAIS subtests themselves involve recreating 2D red-and-white geometric patterns using 3D cubes that have red, white, and red-and-white sides. The patterns can be made up of two, four, or nine blocks and a score is awarded for each pattern

based on the time taken to complete the assembly and whether the final assembly is correct [6], [7]. Typically when this test is administered, a trained professional must be present to walk the testee through the process by keeping track of completion times, recording incorrect answers, scoring the test, and monitoring the test taker for any psychological clues.

In this test, the algorithm was implemented to directly track the blocks instead of placing separate markers on them because the blocks themselves satisfy the requirements for serving as a marker. The blocks have sides that appear as triangles and squares when either white or red color is tracked, as well as being able to form more complex shapes by putting the blocks together. Scoring requires additional considerations because the system must recognize whether a testee has successfully created a pattern using multiple blocks. This means that the resulting position of the blocks must be used to estimate the pattern created by the blocks. The scoring process involves overlaying a grid over the found blocks and determining the color layout within the grid to match to the pattern. The start time of a trial was indicated by the blocks being dispersed throughout the environment and is marked as complete when the blocks form the goal pattern of that trial. If a pattern is not completed successfully, then the time is stopped and marked incomplete when the blocks are dispersed for the next pattern. The score is assigned by the same conventions as in the WISC and WAIS block design tests. The automation was tested by comparing manually and automatically scored tests showing 100% accuracy.

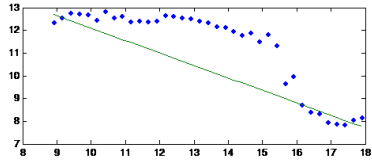
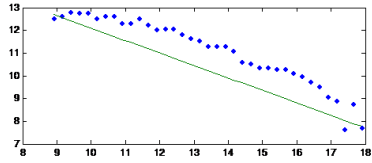
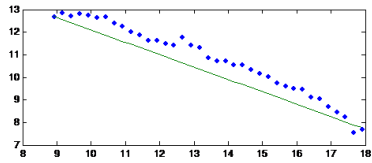
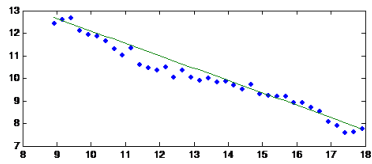
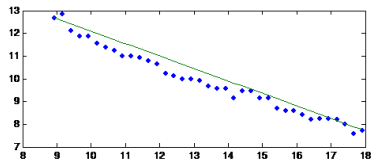
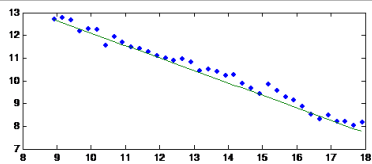
E. Visual-Motor Integration Test

A part of motor skills is reflected by how well a person can trace lines and shapes in 2D. The closeness of a followed path to the ideal path and steadiness of the movements reflect the motor skills of the person in terms of how advanced they are in their motor development or if they have any difficulties with any of their individual joints or muscles. The idea is similar to that of the Beery-Buktenica Developmental Test of Visual-Motor Integration (Beery VMI) where the subject must copy or trace lines and shapes using a pencil [49]. For our demonstration, a cup is used instead of a pencil to trace out a pattern on the table and shapes in the air. The path could be anything as long as its shape is known so that an ideal path is available for comparison. Table I shows six trails of drawing a straight line between two points using a cup as a marker. A correlation between the actual position data and an ideal fitted straight line was analyzed by performing a linear regression between the two. A higher value of r indicates the movement trajectory was closer to the given straight line.

F. Gesture Recognition

Gesture recognition aims to classify the motion that a person is performing [49], [50]. It has a wide range of applications including aids for the hearing impaired, interpreting sign language, lie/stress/emotional state detection, and controls or tools for interaction with virtual environments [49]. A variety of methods can be used to interpret gestures including principal component analysis, the CONDITIONAL DENSITY PROPAGATION (CONDENSATION) algorithm, Kalman filtering and more advanced particle filtering, and hidden Markov models [49]. The goal of this application is to create a simple gesture

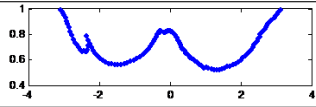
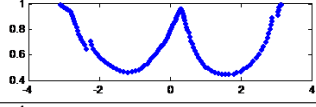
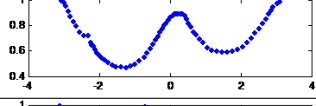
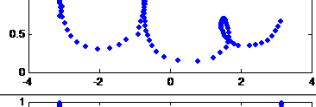
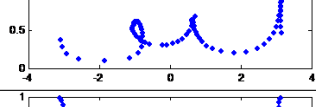
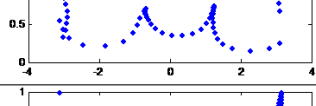
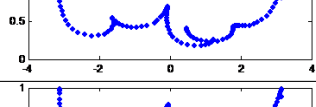
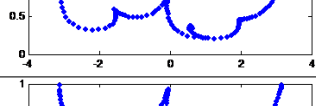
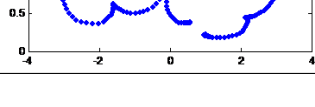
TABLE I: Paths exhibiting a range of different accuracies between two points shown in graphs of points and straight ideal lines along with the calculated correlation values for the match between the two.

Correlation coefficient	Movement trajectory
$r = 0.8539$	
$r = 0.9562$	
$r = 0.9827$	
$r = 0.9930$	
$r = 0.9889$	
$r = 0.9930$	

recognition tool that can identify the motion of tracing the geometry of a shape. It is also desired that it is not affected by the speed or the size of the motion but is simply unique to the shape or form of the motion.

Only a couple of distinct shapes were explored for this section, so a simple method was chosen for recognition. The motions were to draw a circle, triangle, and square in the air and the method used to recognize the shapes was a shape descriptor technique called shape signatures [51]. Shape signatures represent an object's shape as a one dimensional function of its edge points. A variety of different methods can be used to create this function but a common method, which is used here, is the distance of the boundary points and angle relative to their centroid. The signature is made scale invariant by dividing all distances by the maximum distance and is made orientation invariant by finding the angular position of the maximum point and making the function start at this value.

TABLE II: Signatures for the three shapes (circle, square, and triangle) and calculated feature values and logic gate outputs using the threshold values of $P_1 = 0.4$ and $P_2 = 0.6$.

Motion Trajectory	P_1	P_2	Output
	0.5237	0.8350	Circle
	0.4501	0.9350	Circle
	0.4740	0.8977	Circle
	0.1545	0.2119	Triangle
	0.1047	0.3606	Triangle
	0.1569	0.3932	Triangle
	0.1825	0.6952	Square
	0.2155	0.7871	Square
	0.1849	0.9363	Square

The signature can then be analyzed to find the number of corners mapped to a function. In this case, the signature is simply examined at key locations that distinguish the different shapes. The first key feature is the relationship between the minimum and maximum value of the radius (R_{min}, R_{max}). This distinguishes circles (or in this case ellipses) from the squares and triangles. Unless the eccentricity is high, the ratio will be significantly higher for circles than for the other two. The second feature is the behavior of the shape at an angle of zero. Circles have extreme points at angles of π and 0, so the behavior at 0 should be high. Squares have extreme points at $\pi, \pi/2, 0,$ and $\pi/2$, so the behavior at 0 should be high. Finally, triangles have extreme points at $\pi, 2\pi/3,$ and $-2\pi/3$, so the behavior at 0 should be low. For these three shapes, two features effectively take care of all possible cases. If more shapes need to be identified then additional features would become necessary, but would be easy to add to the current framework. Table II shows recognition results for each shape. P_1 and P_2 are calculated by

$$P_1 = \frac{R_{min}}{R_{max}}; \quad P_2 = \frac{R_{center}}{R_{max}}$$

and the shape is recognized as a *circle* if $P_1 > 0.4$ and $P_2 > 0.6$, a *triangle* if $P_1 < 0.4$ and $P_2 < 0.6$, or a *square* if $P_1 < 0.4$ and $P_2 > 0.6$.

IV. CONCLUSION AND DISCUSSION

This paper presented an integrated low-cost, real-time vision processing algorithm that can be used for a variety of assessment tests for upper extremity motor skills that involve object manipulation. While individual layers of the algorithm utilize existing techniques, the main contribution of this paper lies in the proper integration of these techniques keeping the computational cost low for target clinical and educational applications. The algorithm was implemented in four well-known games/tests and a simple gesture recognition application for demonstrating its potential utility. When such motor assessment tests need to be periodically administered to an individual or to a large group of people, automating the entire process can significantly reduce the time, cost, and labor intensity while also improving the quantity and quality of the measurable data. The specific applications presented in this paper were carefully selected to cover a broad range of motor skill assessment tests so that one can easily take it into use.

The presented algorithm requires comparison with other vision-based object tracking algorithms to prove its time efficiency. To further improve the versatility of the algorithm, another layer of prior image processing can be added for automatically determining the color threshold range instead of using a pre-defined value so that any arbitrary objects can be detected and tracked as long as they are distinguishable from the environment. In addition, benefits expected by the algorithm implementation needs to be verified through human subject studies involving non-technical administrators (e.g. teachers, parents, and clinicians) and potential testees (e.g. students, children with varying cognitive/motor skills, and older adults). Our ongoing work involves human subject evaluation and cost analysis in addition to continuous improvements in the algorithm.

REFERENCES

- [1] G. H. Noritz, and N. A. Murphy, "Motor delays: early identification and evaluation," *Pediatrics*, 131(6): e2016-e2027, 2013.
- [2] American Academy of Pediatrics, Committee on Children with Disabilities, "Developmental surveillance and screening of infants and young children," *Pediatrics*, 108: 192-196, 2001.
- [3] Division of Health Policy Research, "Identification of children \geq 36 months at risk for developmental problems and referral to early identification programs," *American Academy of Pediatrics, Periodic Survey of Fellows.*, 2003.
- [4] Beery-Buktenica Developmental Test of Visual-Motor Integration, 6th Edition (BEERY VMI), Pearson Education, Inc.
- [5] C. DeMatteo, M. Law, D. Russell, N. Pollock, P. Rosenbaum and S. Walter, "QUEST: Quality of Upper Extremity Skills Test," *Canchild Centre for Childhood Disabilities Research*, Hamilton, Ontario, Canada, 1992.
- [6] D. Wechsler, "Manual for Wechsler Adult Intelligence Scale - Revised", The Psychological Corporation, New York, 1981.
- [7] D. Wechsler, "Wechsler Intelligence Scale for Children - Revised", The Psychological Corporation, New York, 1971.
- [8] L. First and J. Palfrey, "The infant or young child with developmental delay," *The New England Journal of Medicine*, 330(7): 478-483, 1994.
- [9] F. P. Glascoe, M. Foster and W. Wolraich, "An economic analysis of developmental detection methods," *Pediatrics*, 99: 830-837, 1997.
- [10] C. A. Brogan, "The pathway to care for children with autism spectrum disorders aged 0 to 12 years," Glasgow: National Autistic Society Scotland, p. 2001.
- [11] T. H. Lee, C. M. Blasey and J. Dyer-Friedman, "From research to practice: Teacher and pediatrician awareness of phenotypic traits in neurogenetic syndromes," *American Journal on Mental Retardation*, vol. 10, no. 2, pp. 100-106, 2005.
- [12] R. Parasuraman and V. Riley, "Humans and Automation: Use, misuse, disuse, abuse," *Human Factors*, vol. 39, no. 2, pp. 230-253, 1997.
- [13] V. Mathiowetz, G. Volland and K. Weber, "Adult Norms for the Box and Block Test of Manual Dexterity," *American Journal of Occupational Therapy*, vol. 39, pp. 386-391, 1985.
- [14] J. Desrosiers and G. Bravo, "Validation of the Box and Block Test as a measure of dexterity of elderly people: reliability, validity, and norms studies," *Arch Phys Med Rehabil*, vol. 75, pp. 751-755, 1994.
- [15] C. Hekimian-Williams, "Accurate Localization of RFID Tags Using Phase Difference," in *Proceedings of the IEEE International Conference on RFID*, Orlando, FL, 14-16 April 2010.
- [16] A. W. Reza and T. K. Geok, "Objects tracking in a dense reader environment utilizing grids of RFID antenna positioning," *International Journal of Electronics*, vol. 96, no. 12, pp. 1281-1307, 2009.
- [17] D. Jeong, E. Kerci, and L. Lee, "TaG-Games: Tangible Geometric Games for Assessing Cognitive Problem-Solving Skills and Fine Motor Proficiency," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligence Systems*, pp. 32-37, 2010.
- [18] L. Buechley and M. Eisenberg, "Boda Blocks: A Collaborative Tool for Exploring Tangible Three- Dimensional Cellular Automata," in *Proceedings of the 8th International Conference on Computer Supported Collaborative Learning*, pp. 102-104, 2007.
- [19] R. Watanabe, Y. Itoh, M. Asai, Y. Kitamura, F. Kishiro, and H. Kikuchi, "The Soul of ActiveBlock: Implementing a Flexible, Multimodal, Three-Dimensional Spatial Tangible Interface," in *Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2004.
- [20] D. Anderson, J. L. Frankel, J. Mark, D. Leigh, K. Ryall, E. Sullivan and J. Yedidia, "Building Virtual Structures with Physical Blocks," in *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, 1999.
- [21] L. Terrenghi, M. Kranz, P. Holleis, and A. Schmidt, "A cube to learn: a tangible user interface for the design of a learning appliance," *Pers Ubiquit Comput*, (10): 153-158, 2006.
- [22] K. Camarata, E. Y. Do, M. D. Gross, and B. R. Johnson, "Navigational Blocks: tangible navigation of digital information," in *Proceedings of International Conference on Computer Human Interaction (CHI)*, 2002.
- [23] M. G. Gorbet, M. Orth, and H. Ishii, "Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topograph," in *Proceedings of CHI*, pp. 18-23, 1998.
- [24] B. Schiettecatte, and J. Vanderdonck, "AudioBlocks: a Distributed Block Tangible Interface based on Interaction Range for Sound Design," in *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI)*, 2008.
- [25] H. Newton-Dunn, H. Nakano, and J. Gibson, "Block Jam: A Tangible Interface for Interactive Music," in *Proceedings of the Conference on New Interfaces for Musical Expression*, pp. 170-177, 2003.
- [26] S. Alers, E. I. Barakova, "Multi-Agent Platform for Development of Educational Games for Children with Autism," in *Proceedings of Games Innovations Conference, Intl. IEEE Consumer Electronics Society*, 2009.
- [27] J. P. Wachs, M. Kolsch, H. Stern, and Y. Edan, "Vision-based Hand-gesture Applications," *Communications of the ACM*, 54(2): 60-71, 2011.
- [28] H. Kato, M. Billingham, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual Object Manipulation on a Tabletop AR Environment," in *Proceedings of IEEE and ACM International Symposium on Augmented Reality*, 2010.
- [29] A. D. Wilson, "Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction," in *Proceedings of IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, Newport, RI, 2007.
- [30] A. D. Wilson, "PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System" *Proceedings of the 18th Annual ACM symposium on User interface software and technology*, 2005.

- [31] A.D. Cheok, X. Yang, Z. Z. Ying, M. Billinghurst, and H. Kato, "Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing," In Proceedings of the 2004 ACM SIGCHI International conference on Advances in Computer Entertainment Technology, pp. 117-126, 2003.
- [32] B. H. Thomas, "A survey of visual, mixed, and augmented reality gaming," Computers in Entertainment Magazine, 10(3), Article 3, 2012.
- [33] D. Jeong, E. Kerci and K. Lee, "TaG-Games: tangible geometric games for assessing cognitive problem-solving skills and fine motor proficiency," IEEE MFI, pp. 32-37, 2010.
- [34] B. Floyd, D. Jeong and K. Lee, "Geometric Games for Assessing Cognitive, Working Memory, and Motor Control Skills," in Tangible, Embedded, and Embodied Interaction, Kingston, ON Canada, 2012.
- [35] R. Y. Wang and J. Popovic, "Real-Time Hand-Tracking with a Color Glove," ACM Transactions on Graphics, vol. 28, no. 3, 2009.
- [36] Y. Li, D. Coleman, M. Ransdell, L. Coleman and C. Irwin, "Sport Stacking Activities in School Children's Motor Skills Development," Perceptual and Motor Skills, vol. 113, no. 2, pp. 431-438, 2011.
- [37] E. Hoeger and S. Hoeger, Principles and Labs for Fitness and Wellness, Belmont, CA: Wadsworth/Thomson Learning, 2004.
- [38] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey," ACM Comput. Surv., vol. 38, no. 4, 2006.
- [39] W. Hu, T. N. Tan, L. Wang and S. Maybank, "A survey on visual surveillance of object motion and behaviors," IEEE Transaction on Systems, Man, and Cybernetics Part C - Applications and Reviews, vol. 34, no. 3, pp. 334-352, 2004.
- [40] M. J. Swain and D. H. Ballard, "Color indexing," Internal Journal of Computer Vision, vol. 7, no. 1, pp. 11-32, 1991.
- [41] M. Hart, L. Smith and A. DeChant, "Influence of Participation in a Cup-Stacking Unit on Timing Tasks," Perceptual and Motor Skills, vol. 101, pp. 869-876, 2005.
- [42] B. Edermann, S. Murray, J. Mayer and K. Sagendorf, "Influence of Cup Stacking on Hand-Eye Coordination and Reaction Time of Second-Grade Students," Perceptual and Motor Skills, vol. 98, pp. 409-411, 2004.
- [43] M. Lam, S. Cheung and B. Chow, "The effects of Tai-Chi-Soft-Ball training on physical functional health of Chinese older adult," J. Hum. Sport Exerc, vol. 6, no. 3, 2011.
- [44] G. Bravo, P. Gauthier, P. Roy, H. Payette and P. Gaulin, "A Weight-Bearing, Water-Based Exercise Program for Osteopenic Women: Its Impact on Bone, Functional Fitness, and Well-Being," Arch Phys Med Rehabil, vol. 78, no. 12, pp. 1375-80, 1997.
- [45] R. H. Wood, J. M. Hondzinski and C. M. Lee, "Evidence of an association among age-related changes in physical, psychomotor and autonomic function," Age and Ageing, vol. 32, no. 4, pp. 415-421, 2003.
- [46] G. Frank, "The Wechsler Enterprise: An Assessment of the Development, Structure, and Use of the Wechsler Tests of Intelligence", Pergamon, Oxford, 1983.
- [47] J. Sattler, "Assessment of children's intelligence", Saunders, Philadelphia, 1974.
- [48] S. Kohs, "Intelligence Measure", Macmillan, New York, 1923.
- [49] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE Transaction on Systems, Man, and Cybernetics Part C - Applications and Reviews, vol. 37, no. 3, pp. 311-324, 2007.
- [50] J. Daugman, "Face and Gesture Recognition: Overview," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 675-676, 1997.
- [51] S. Belongie, J. Malik and J. Puzich, "Matching Shapes," in 8th IEEE International Conference on Computer Vision, Vancouver, Canada, 2001.