Automatic Functionality Verification of Hybrid Set-Top Boxes with Dynamic User Interface

Matija Pul, Vukota Peković, Mario Vranješ, *Member, IEEE*, Ratko Grbić

***Abstract*—Modern set-top-box (STB) user interfaces (UI) contain increased amount of animations/effects compared to standard UIs, thus demanding a new approach for accurate and cost-efficient STB functionality verification. In this paper, a fully-automated system for functional failure detection in hybrid STBs with dynamic UI is proposed. It consists of image capturing unit, testing tool and hybrid STB being verified. The automatic verification of hybrid STB functionality is performed by comparing the captured image of the dynamic UI to the referent image captured from referent STB. The system incorporates a new algorithm for image similarity measurement, adjusted for image alterations which often appear in dynamic UIs. Algorithm utilizes image feature extraction and comparison to determine captured and referent image similarity and to make decision regarding STB functionality. The proposed system was compared with existing fully-automated system for functionality verification of a STB device with static UI. Tests were performed on both computer generated and live stream cases. For this purpose a new Computer Generated Image Alterations Database (CGIAD) is created and is freely available. The results show that the proposed system achieves higher performance and robustness, in terms of measuring image similarity despite alterations caused by dynamic UI effects.**

***Index Terms*—Automated functionality verification, functional failure detection, hybrid STB, image similarity**

I. Introduction

PROGRESS in fields of both hardware (HW) and software (SW) allowed for better operating systems which can support more complex interfaces and new ways of interaction with the user, such as speech recognition and internet browsing [1]–[4]. Consequently, TV systems (and Set-Top-Box-es (STB) as their part), being one the most expanding consumer electronic field, have an endless demand for cutting-edge functionality verification (FV) systems [5], [6].

Manufacturers are required to perform FV of their products in order to provide their consumers with efficient and reliable products. As a result of increasing hybrid STB complexity, manual FV has become inefficient due to its requirement of considerable human effort and time [7]. Hybrid STB offers many new functionalities (video on demand, internet applications, video telephony, surveillance, gaming, shopping, dynamic user interface (UI), etc.) that increase time and difficulty for their performance evaluation, in comparison to regular STB with singular TV delivery methods and elementary functionalities. Furthermore, hybrid STBs with dynamic UI contain animations and effects, making the UI appear more interactive and interesting but also more hectic. Due to that reason, FV is performed by automated systems which operate without the requirement of human interaction and are capable of testing multiple devices simultaneously. Automated systems perform FV with reduced resource consumption, while maintaining high verification reliability.

STB functionality issues can occur as a consequence of HW and/or SW problems. SW problems often manifest in a form of deadlocks, memory leaks or task starvation, while HW problems manifest in other, usually harder to detect, forms. In order to perform complete FV, it is necessary to verify that the device under test (DUT) is not experiencing any HW or SW problems. Due to the fact that HW and SW in hybrid STB are tightly coupled, any HW or SW problem is expected to manifest itself on the device output in a form of unwanted image alteration. Therefore, the best approach for functional failure detection is the performing digital image analysis on device output, which facilitates detection of both SW and HW functional failures [8]–[11]. The idea is to analyze the images displayed by the hybrid STB under test in order to detect unwanted image alterations caused by functional failures within the hybrid STB. There are many related works published in the field of image artifacts detection describing their potential for measurement of unwanted image alterations [12]–[15]. As explained in [13], objective methods for image quality assessment are classified by available referent image information. Full reference (FR) methods compare original image with the test one, reduced reference (RR) methods use only representative features of the original image when comparing it to the test image and no-reference (NR) methods perform assessment without having any information about the

This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, under grant number: TR36029 and partially by J.J. Strossmayer University of Osijek business fund through the internal competition for the research and artistic projects “IZIP-2016”.

M. Pul is with RT-RK Institute for information technologies, Osijek, 31000, Croatia. (e-mail: matija.pul@rt-rk.com).

V. Peković is with RT-RK Institute for computer based systems, Novi Sad, Serbia (e-mail: vukota.pekovic@rt-rk.com).

M. Vranješ is with the Faculty of electrical engineering, computer science and information technology, Osijek, 31000, Croatia, Department of Communications (e-mail: mario.vranjes@ferit.hr).

R. Grbic is with the Faculty of electrical engineering, computer science and information technology, Osijek, 31000, Croatia, Department of Computer Engineering and Automation (e-mail: ratko.grbic@ferit.hr).

original image. When performing image alteration measurement, there must be no outside factors (faulty equipment such as cables or STB remote control devices) that could potentially cause additional image alterations. This is necessary in order to be sure that all detected image alterations are caused by the DUT. NR and RR methods implement a potential risk of false image degradation detection due to not having complete information about what the test image should look like. On the other hand, due to having access to original/referent image information, FR methods guarantee highest accuracy for image alteration measurement, which makes them most suitable for hybrid STB functional failure detection. Original/referent image is captured from device of same type as DUT, which is considered to be functioning properly.

Image alteration measurement only covers one aspect of the functionality failure detection system. In order to perform complete device verification, the system is required to execute multiple tests on DUT, which would cover all possible cases and scenarios in which such device could experience functionality issues. Such automated systems are thoroughly explained in previously mentioned papers regarding automated FV of TV sets [5], [6], [8], [9]. Their testing methodology proved to be highly efficient in functionality failure detection but it has shown a high level of inaccuracy when applied to hybrid STB with dynamic user interfaces (UI). When analyzing dynamic images containing rotated, scaled, translated or occluded (partially visible) objects, their systems would detect false functionality failures due to having no tolerance towards image alterations which was set in order to maximize system precision. Such tolerance levels are optimal for static images but leave little space for image alterations caused by animated objects, which are quite common in dynamic UI.

In order to perform FV, the algorithm has to be sensitive enough to detect these unexpected image alterations while simultaneously ignoring insignificant alterations such as those caused by noise. Natural approach to detect image similarity is to use state-of-the-art image quality assessment (IQA) algorithms. However, these methods tend to be too sensitive for efficient STB FV, especially in the case of dynamic UI [9]. Existing solutions for FV of STBs such as systems based on Picture Block Compare (PBC) algorithm [8], [9], [11] outperform other state-of-the-art IQA algorithms (like PSNR, SSIM [16] and VIF [17]) when considering image similarity measurement problem. However, PBC algorithm has a high sensitivity towards drastic image alterations due to being designed for static UI images, where such alterations are not expected. Therefore, common occurrences in dynamic UIs, such as changes of object color or shape caused by animations, are likely to negatively affect PBC efficiency and thus overall FV system performance.

In this paper, the automated system and framework for functional failure detection in hybrid STB with dynamic UI is proposed. The proposed system is Black Box Testing (BBT) system and incorporates FR approach to FV. It is solely based on analysis of output images from referent STB and STB under test. The proposed system utilizes a new Dynamic Image Similarity Measurement (DISM) algorithm capable of comparing dynamic images. To the best of our knowledge, there are no existing solutions for image similarity measurement, with focus on dynamic image comparison, considering hybrid STB functionality failure detection. Because of that, the proposed system which utilizes DISM algorithm was compared with another system of similar design which utilizes PBC algorithm.

System proposed in this paper was experimentally evaluated and compared with the PBC-based system for automatic STB FV [8] on both computer generated and live stream cases. The comparison consists of image similarity measurements performed on both static and dynamic objects. For the purpose of algorithms testing, a new database called Computer Generated Image Alterations Database (CGIAD) that contains 10000 images is created and is freely available to the research community at http://www.rt-rk.com/other/CGIAD.html [18]. The results show that the proposed system utilizing DISM algorithm outperforms the other system utilizing PBC algorithm.

The rest of the paper is organized as follows. Section II describes the general system framework with detailed system description. The proposed DISM algorithm is explained in Section III. Section IV presents experimental results, whereas the conclusion is given in Section V.

II. General Description Of The Hybrid Stb Functional Failure Detection System

The proposed system for functional failure detection (caused by various reasons such as HW defects or SW related issues) of the hybrid STBs is designed as a BBT system based on output image analysis. Since the DUT is treated as a black box, output image is the foremost source of information available for analysis. Therefore, the proposed system is designed under the assumption that a functional failure of DUT would manifest itself on the output image. The main idea is to excite the DUT into a desired state and subsequently capture an output image in order to verify device functionality. Image captured from DUT is compared to referent image captured from a referent hybrid STB, which is considered to be functioning properly. Image comparison is performed by the new proposed DISM algorithm for dynamic image comparison, which outputs image similarity value used to make the DUT functionality decision. The proposed system performs fully automated functional failure testing which is crucial for reduction of manufacturing cycle time, while also providing a more detailed analysis with higher reliability due to elimination of human factor. Human interaction is necessary only for reference image validation and test case generation.

Hybrid STB testing station is a portable and modular Windows based embedded PC system, which is intended to be used by the DTV set and STB manufacturers for R&D, production and field device FV. This system consists of three main components: (i) Image capturing unit (ICU); (ii) Testing tool (TT), and (iii) DUT, as can be seen in Fig. 1. The DUT is controlled by the signals sent from IR transmitter. IR transmitter is connected to the PC system which is running the

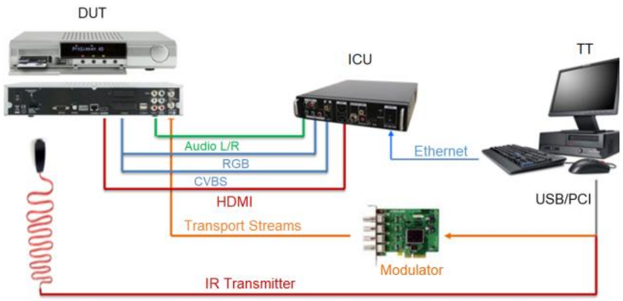


Fig. 1. General description of the proposed automated functionality failure testing system for hybrid STB with dynamic UI.

TT. TT sends commands to DUT navigating it towards dynamic UI position which is currently being tested. After successfully navigating to correct position, ICU captures an image of current UI screen via video source cable (HDMI, CVBS, SCART, and S-VIDEO) and forwards it to TT via Ethernet cable. TT runs the proposed DISM algorithm, which compares captured test image with its corresponding referent image and determines if DUT is functioning properly.

Detailed description of the proposed system for hybrid STB functional testing is depicted in Fig. 2. TT communicates with the system via Tester communication unit (TCU) which consists of two main components: (i) Use case scenario database (UCSD) and (ii) System user interface (SUI). UCSD contains test scripts with commands necessary for test execution. SUI is used for communication with the user, mainly to display current test progress and results. Processing unit (PU) contains all components necessary for test case execution. When test script is initialized, UCSD instructs System control logic unit (SCLU) to perform specific operations. Based on given instructions, SCLU either sends navigation instructions to DUT through the DUT control unit (DCU), captures a test image via ICU or commands Functionality evaluation unit (FEU) to execute the proposed DISM algorithm. FEU is tasked with algorithm execution and database management. All referent images used in current test case scenario are stored in Referent image database (RID). The proposed DISM algorithm is called from Algorithm database (AD), where all the files necessary for algorithm execution and configuration are stored. When executed, the algorithm compares captured test image to referent image from RID and stores similarity result into Result database (RD). This way results remain available if further analysis is required.

III. The Proposed Dynamic Image Similarity Measurement Algorithm

This section describes the proposed DISM algorithm, which is based on image feature extraction and comparison, with focus on measuring the similarity of descriptors of key points detected on test and referent images. Test image is captured from the hybrid STB under test and the referent image is captured from the referent hybrid STB (considered to be functioning properly).

Dynamic image comparison represents a rather new but significant issue in the field of computer vision. Such images are difficult to compare due to their tendency of having dynamic content which changes over time, while preserving image context, example shown in Fig. 3. Human brain compares dynamic images with ease because it can recognize the context while ignoring lesser changes in details. On the other hand, pixel-by-pixel image comparison methods (like [11]) compare images based only on their pixel values. Such methods are ineffective for dynamic image comparison because of their high sensitivity to changes. This is exactly the reason why image feature extraction was chosen as basis for DISM algorithm. It provides a way to describe image context based on image content, which can then be used for dynamic image comparison.

Image feature extraction is a field of science on its own and contains various methods for describing image features. Method of image feature extraction is often chosen based on image context. For example, if working with images containing people, then it would be best to use method of image feature extraction which is designed to recognize features such as human faces and clothes with high precision. In other cases, if there is no way of knowing beforehand what the context of images would be, a more generic image feature extraction method would prove to be more beneficial. Since hybrid STB manufacturers have no limitations but their own imagination on how to design UI, it is better to use a generic method of image feature extraction.

DISM algorithm utilizes key points as a method of image feature extraction. Key points are distinguishable areas of the image described by their surrounding environment. Key point descriptor contains information about the environment around its appropriate key point. It is often presented as an n-dimensional vector where n presents the number of attributes

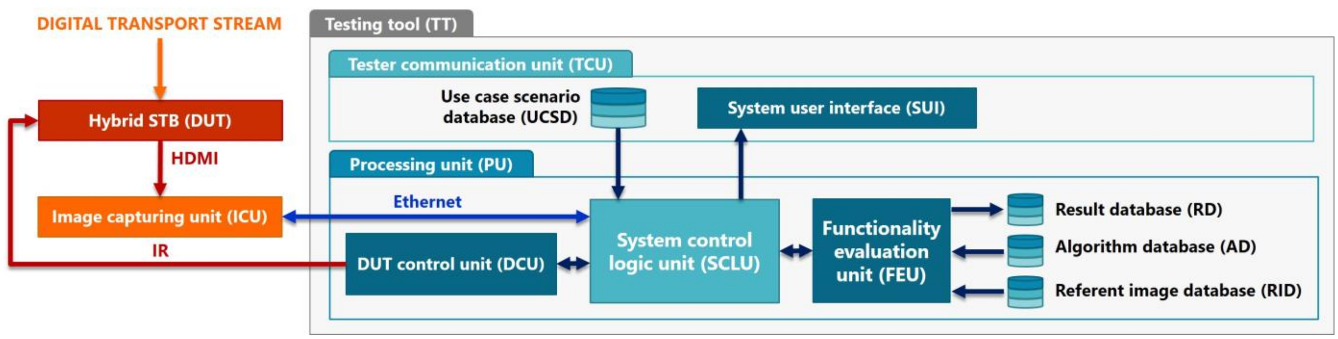
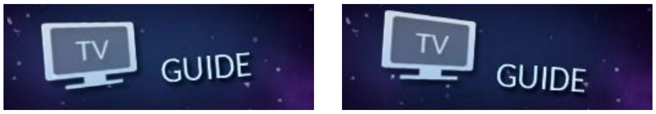


Fig. 2. Block diagram of the proposed system for hybrid STB functional testing.



(a) (b)

Fig. 3. An example of a single animated object captured at two different moments: (a) moment 1, (b) moment 2. The object contains the following animations: rotation, scaling, translation and occlusion.

used to describe the environment. As long as most of the environment around the key point remains unchanged, the same will be true for its description. It is crucial for the key point description to be invariant to transformations such as rotation, translation, scaling and occlusion, often present in dynamic images. This makes key points suitable for dynamic image comparison because of their tolerance to image alterations often caused by animations and various visual effects.

Fig. 4 presents the flow diagram of DISM algorithm, in which key points, extracted from test and referent images, are matched and used to determine quantitative measure of image similarity. According to predefined threshold, image similarity measure determines the acceptance/rejection of the tested hybrid STB (DUT) FV. In order to make a binary decision, it is necessary to verify every single functionality of the DUT, such as opening a specific application, playing the selected media file or specific GUI functionalities. For instance, verifying if object has appeared on the correct location in the right moment is an example of a common test case. When testing DUT functionality which affects only a certain region of the captured image, the algorithm applies region-of-interest (ROI) extraction in order to increase precision and prevent undesirable key points from impacting the verification result. ROI extraction is performed during the image preparation stage.

Key point processing comprises of key point detection and filtering based on their quality. There are many algorithms that could be used to extract key points, like Scale-Invariant Feature Transform (SIFT) [19] and Speeded-Up Robust Features (SURF) [20]. Although these algorithms are used in many researches dealing with image processing [21]–[24], Oriented FAST and Rotated BRIEF (ORB) algorithm was chosen because it is an open-source algorithm and achieves high performance. More information about ORB can be found in [25] and about its various applications in [26]–[29]. ORB inspects image for local extremes which are considered as potential key points. Detected candidates are tested in order to determine their stability in terms of robustness towards changes in their surroundings. Candidates which proved to be stable are classified as key points and proceed to have their descriptors generated. As a result, ORB algorithm generates an array of key points suited for further analysis.

Due to the abundance of generated key points, it is possible to increase algorithm precision by filtering out all key points with lower stability and poor location along edges. ORB assigns a stability value S_{KP} to each key point it generates, which can be used as a basis for filtering. Threshold T_S for stability filtering should be adjusted to the content of ROI. For example, T_S can be higher for ROI containing text and thus producing higher number of stable key points. On the other hand, for ROI containing shapes without text, thus producing lower number of stable key points, T_S can be lower. In order to increase precision, the DISM algorithm discards all key points whose stability value S_{KP} is lower than T_S. Fig. 5 shows the effect of filtering based on stability being used for removal of low-stability key points detected on background objects (unimportant flying particles) as well on foreground objects (context), where each circle represents a single key point.

After key point filtering, a single descriptor is computed for each of the remaining key points. ORB algorithm computes descriptors based on light intensity gradients of a grayscale image. When comparing images, there are two sets of key points: key points detected on the referent image (set A) and key points detected on test image (set B). Goal is to find a match for each key point from set A in set B. In order to achieve this, each key point from set A is compared to each key point in set B. This process is performed under the assumption that the key point in set B with the highest similarity to the currently observed key point from set A is the correct match. Descriptors are used as a basis for key point comparison. In DISM algorithm, similarity measure between two descriptors $\vec{u}$ and $\vec{v}$ is measured using the Euclidian distance between two vectors in space:

$$d(\vec{u},\vec{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 \dots (u_n - v_n)^2} \tag{1}$$

The smaller the distance d, the greater the similarity between descriptors. Threshold T_D for descriptor quality filtering can be adjusted to achieve high precision matching. All matches with

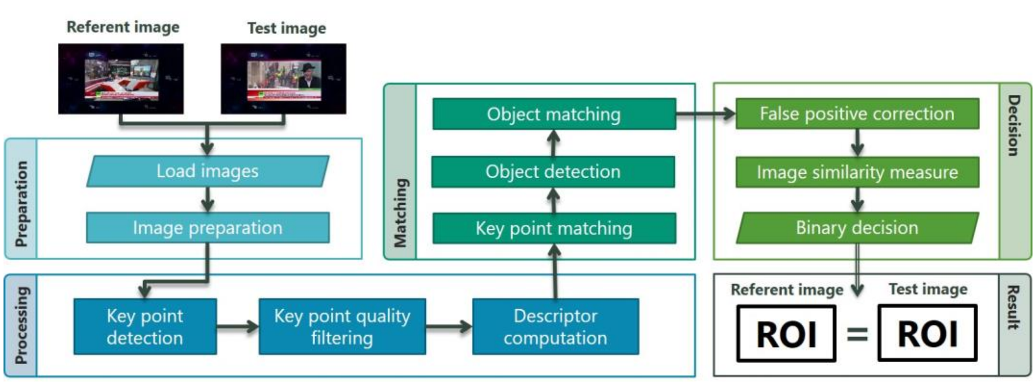


Fig. 4. The proposed Dynamic Image Similarity Measurement algorithm flow diagram for image similarity measurement.

(a) (b) (c)

Fig. 5. Result of key points filtering based on their stability: (a) referent image (b) referent image with detected key points (c) referent image with detected key points remaining after filtering based on their stability.

descriptor distance d higher than T_D will be disregarded.

Main problem with key point matching are false positive matches. False positive detection happens when a wrong key point deforms itself enough to produce a better match with a referent key point than its actual true match does. This often happens on images that contain text, because same letters can appear in very similar surroundings causing detection of many similar key points. In order to prevent false positive matches from decreasing result accuracy, the algorithm applies an object detection method. In this case, object detection refers to the process of assigning each key point an object index. In order to do this, the algorithm detects clustered key points and classifies them as a single object by assigning each key point, in a single cluster, the same object index which is unique for each object. As a result, objects are perceived as clusters of key points with the same object index. The following statement is true for each key point belonging to the same object:

$$\exists d \in D: d < d_{max}, \tag{2}$$

where D represents a set of distances d between observed key point and every other key point in that object, and d_{max} represents the predefined maximum allowed distance. This means that for each key point inside the object there is at least one other key point in a radius of d_{max}, provided that D contains more than one key point. Solitary key points without any adjacent key points within radius of d_{max} will still be given a unique object index. It is possible to control the size of clusters classified as objects by adjusting the maximum allowed distance. For example, if image contains a single sentence and the current d_{max} results in every word classified as an object, then increasing the distance would result in entire sentence being classified as a single object, while lowering the distance would cause every letter to be classified as an independent object.

After the object detection is completed and every key point is assigned to a certain object, it is possible to remove false matches. First step is to pair objects from referent image with objects on test image. This is performed by observing matches for each object from referent image O_R and pairing it with an object from test image O_T on which it scored most matches:

$$\forall O_R \exists! \, O_T: M_T = max(N), \tag{3}$$

where M_T represents the number of matches scored on the observed object from test image and N represents the set containing numbers of matches scored on each object from test image by the currently observed object from referent image. Object comparison is performed under an assumption that all key points from the observed referent object can be detected only on a single object from the test image. This means that every referent object key point match detected on a key point which doesn't belong to the test object paired with the observed referent object is considered to be a false match. What remains is to go through each match M for each pair and discard all false matches so that only matches between paired objects M_P remain. The following must be true for all matches:

$$\forall M, M \in M_P \tag{4}$$

This removes the majority of false positive matches, significantly reducing their impact on algorithm accuracy. Remaining matches are considered to be positive and count towards image similarity measurement.

Based on the number of positive matches, the algorithm determines image similarity measurement. In order to make a binary decision on the hybrid STB functionality, an application-specific threshold T is applied on the image similarity measurement.

Final image similarity is determined based on success of key point detection, which is calculated according to:

$$P[\%] = \frac{N_{matches}}{N_{referent}} \cdot 100, \tag{5}$$

where $N_{matches}$ represents the number of referent image key points detected on test image after running algorithm and $N_{referent}$ represents the number of key points detected on referent image. If the result from (5) surpasses the application-specific given threshold T, then the hybrid STB is considered to be functioning properly. The algorithm is expected to perform equally for static and dynamic images. Since static images do not contain any animations and have little alterations, the image similarity measurement result is expected to be much higher with regards to dynamic images.

IV. Experimental Results

This section demonstrates the performance evaluation of proposed DISM algorithm for the most common dynamic UI object transformations. Used sample data set is diverse in terms of object transformation types and intensities. Testing data set consists of two groups of images: group with computer-generated alterations designed to thoroughly test algorithm robustness to certain transformations (CGIAD) and a group consisting of images captured from a real hybrid STB sequence portraying a dynamic UI. To the best of our knowledge there are no existing freely publicly available algorithms for image similarity measurement with focus on dynamic image comparison. Therefore, image similarity measurement algorithm from [9], [11], called PBC, was used for performance comparison and its default block size of 32 × 32 pixels is used in this experiments. Similar to DISM, PBC also outputs an image similarity measurement value in a range of 0 to 100, where 100 means the images are completely the same, while also using a predefined threshold lower than 100 in order to make a binary decision regarding the functionality verification.

A. *Test case database description*

As have been previously mentioned in the introduction, new database called CGIAD is created in order to test the performance of the proposed DISM algorithm. CGIAD is comprised of 10000 images, 950 × 725 pixels in size, 24 bits/pixel. Each of these images contains a single object which is experiencing at least one out of four possible transformations (rotation, scaling, occlusion and translation). The group is divided into 10 separate test groups containing 1000 images each. Each test group has its own unique pair of objects, referent object and object used for true negative tests. Pair of objects used in first test group can be seen in Fig. 6. Referent images used in these tests contain only the referent object without any transformations. Goal is to check if the algorithm can recognize the referent object on the test image despite its alterations caused by transformations. Out of 1000 test images in each test group, 500 images are used for true positive and 500 for true negative detections. Every true positive test image contains its corresponding referent object and has its true negative counterpart image with its corresponding true negative object which is experiencing the same transformations as the object on true positive image. Out of those 500 images, 50 are experiencing object rotation, 50 are experiencing object scaling, 50 are experiencing object occlusion, 50 are experiencing object translation and 300 are experiencing combined object transformations, respectively. Each image contains transformation types and intensities in its name. Transformations listed in the image name are as follows: rotation R, scaling S, occlusion O, horizontal translation and vertical translation T, respectively. For example, the following image name "Test_Group_1_TP_R_-10_S_-10_O_5_T_-10_10_NBR_583" means the image belongs to the test group 1

Fig. 6. Objects used in the first test group containing computer-generated alterations: (a) referent object and (b) object used for true negative tests.

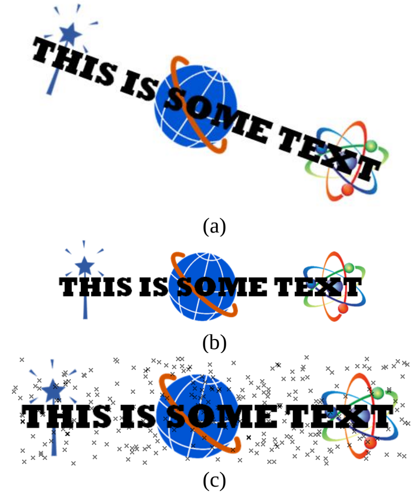


Fig. 7. Example of object experiencing the following transformations: (a) 20° rotation, (b) 20% downscaling and (c) 10% particle occlusion.

and is used in a true positive test, object is rotated for 10° counterclockwise, downscaled for 10% of its size, occluded for 5% of its surface, horizontally translated for 10 pixels to the left and vertically translated for 10 pixels downwards and the image unique number is 583. CGIAD and all information about it can be found at [18].

Object rotation (expressed in degrees) often causes severe changes to object shape on pixel level, as can be seen in Fig.7 (a). Intensity of pixel deformation mostly depends on the algorithm performing the transformation but also on the image resolution due to the pixel grid limitations. Similar to rotation, scaling causes drastic changes to object shape on a pixel level, depending on the scaling algorithm interpolation complexity. Object scaling example is shown in Fig. 7(b). Unlike rotation and scaling, object occlusion (expressed in percentage of image covered with obscuring particles) is not performed on the object itself and does not change its entire composition. It mostly causes changes to the environment around the object and often affects only minor areas of the object itself, as demonstrated in Fig. 7(c). This effect was added to the test because modern dynamic UIs tend to have full screen animations such as falling snowflakes, which could occlude the object in the moment test image was taken. One of the main characteristics of dynamic UIs are moving objects, which is why vertical and horizontal translation (expressed in pixels) were added to the test. During translation, object remains the same but the environment changes, which proved to be quite problematic for pixel compare based algorithms. This was the main incentive for development of the new image similarity detection algorithm which is robust to basic object animations such as translation.

Second group is comprised of 50 images but in this case test images were captured from a real hybrid STB with dynamic UI, see Fig. 8. Unlike the previous group, all test images in this group contain the same object as their corresponding referent image. This group serves as a live test and contains animated objects which are experiencing one or more object transformations (rotation, scaling, occlusion or translation) at the same time. It includes image similarity measurement of highly animated objects, static objects and full screen images

Fig. 8. Test images used in the second test group: (a) “Main_menu_selected_tv_guide_test_1”, (b) “Main_menu_selected_search_test_1”, (c) “Main_menu_selected_media_test_1”, (d) “Main_menu_selected_apps_test_1”, (e) “Channel_installation_test_1”, (f) “Guide_Saturday_test_1”, (g) “Guide_morning_test_1” and (h) “Guide_poster_test_1”.

which are captured in real test cases.

When testing hybrid STB functionality, it is required to verify every option it has to offer. In order to establish full test automatization, the testing tool must be capable of successfully navigating through the UI. This requires successful detection of both static and dynamic objects. Therefore, this test group is designed to test those capabilities. An example of test images containing a dynamic object with combined transformations has been previously shown in Fig. 3, while Fig. 8 (f), (g) and (h) show an example of static objects used in this test group. Note that these static objects are only static in their own behavior. This means that the rest of the UI is still animated in a way which might affect static objects. For example, moving particles or light rays which cover the entire screen, appear from any direction and over any object. Such animations do not interact with the object but can change the perception of its properties in the moment the test image was taken.

B. Performance evaluation results

Similarity estimations are displayed as a percentage in a range from 0 to 100, where 0 represents the lowest possible similarity and 100 represents the maximum possible similarity (test image is the same as the referent image). Experimental results for computer generated alterations are shown in a form of a line graph for improved readability (Fig. 9.). The graph shows true positive and true negative similarity estimations for both DISM and PBC.

DISM algorithm parameters used for computer generated alteration are: $T_S = 0.001$, $T_D = 220$ and $d_{max} = 10$. This test was designed to test algorithm robustness when objects are submitted to various alterations. True positive tests are expected to achieve high image similarity measurement result while true negative tests are expected to yield a low image similarity measurement result, as can be seen for DISM results in Fig. 9.

PBC algorithm achieved low similarity measurement results for true positive tests, except for images containing only object occlusion. In our test cases, the object occlusion is simulated by randomly generating particles 5 × 5 pixels in size over the image. This process continued until required percentage of image was covered in particles. Since PBC is based on splitting image to blocks and subsequently performing block comparison, it has a certain level of tolerance for allowed block difference. Particle size used in performance test was small enough to fall within allowed tolerance for used default block size resulting in a slightly higher similarity score.

True negative tests contain the same set of transformations used for true positive tests. This allows for algorithm performance comparison in cases when the test image contains the desired object and when it contains some other object with vaguely similar properties (see Fig. 6 (a) and (b)), all while experiencing various transformations.

Results show that PBC scored lower measurement results than DISM for true negative tests. While scoring low in these tests is desirable, it is also worth noticing that PBC scored roughly the same results as it did for most true positive tests. This means there is no valid threshold on which it could make a binary decision regarding the images being same or different.

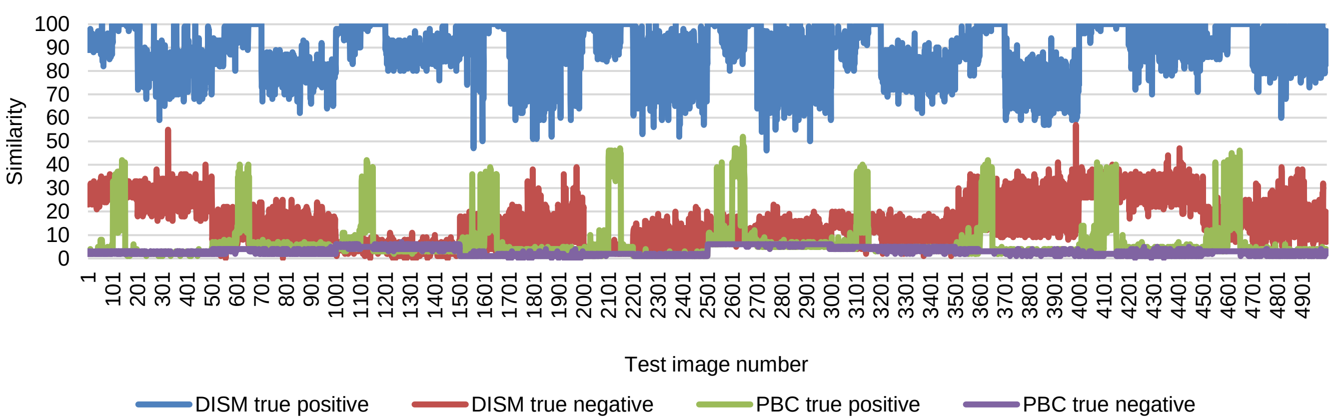


Fig. 9. Performance comparison of image similarity assessment for various object transformations

On the other hand, DISM scored higher results than PBC for true negative tests. The reason for this is because the object pairs used in these tests contain objects which are quite different but still share some resemblance, such as letters and shapes, making them somewhat similar. When comparing true positive and true negative test results, it is clear that DISM scored much higher similarity measurement results when the test image contained the correct object. It is also worth noticing that the difference between true positive and true negative test results is significant enough to determine a valid threshold for a binary decision. In this case the value of such threshold is set to 50, meaning if the algorithm scored higher or equal to 50 then the referent and the test image are the same and vice versa. With threshold set to 50, DISM had 2 false positive and 2 false negative results on a set of 10000 images, making it 99.96% accurate. In the context of testing hybrid STB functionality, precision must be 100%, i.e. number of false positive results must be zero. This can be achieved by setting the threshold to 60, which would also reduce accuracy to 99.66%.

There is no strict correlation between specific transformation and its impact on the results of both tested algorithms. This is because the amount of object deformation depends on its characteristics such as shape and size. For an example, a rectangular object will get slightly deformed when rotated while a round object might experience little to no deformations at all. Each of applied transformations changed some pixel values on the image which caused PBC algorithm to score poorly because of its high sensitivity to pixel change. Unlike PBC, DISM algorithm observes only the object and its features, so the algorithm provides high similarity results if the observed test object has the same characteristic features as the object on referent image.

DISM algorithm parameters used for images captured from a real hybrid STB with dynamic UI are: $T_S = 0.001$, $T_D = 250$ and $d_{max} = 10$. Observing the results for dynamic objects shown in Table I, it is clear that PBC underperformed and achieved worse results than the proposed DISM algorithm for animated objects. Substantial amount of alterations caused by animations, to the object and its environment, proved to be exceptionally problematic for PBC. Increase in transformation is inversely proportional with the measured similarity percentage. While DISM achieved better results for static objects, it is also important to notice that PBC achieved satisfying results. This was expected since PBC was designed for testing functionalities of static UIs. The results prove that the proposed DISM algorithm can be used reliably for FV of hybrid STBs with dynamic UI while using a single fixed threshold for all test cases.

V. Conclusion

The proposed system for automated functional failure detection in hybrid STB with dynamic UI provides a more detailed analysis with higher reliability, while reducing manufacturing cycle time. Functional failure detection is performed by image similarity measurement where test images are captured from DUT and compared to referent image captured from referent device of same type considered to be

TABLE I
PERFORMANCE COMPARISON OF IMAGE SIMILARITY ASSESSMENT FOR HYBRID STB DYNAMIC USER INTERFACE

	Test image name	**PBC**	**DISM**
Dynamic image single object comparison, combined transformations	Main_menu_selected_tv_guide_test_1	7	**90**
	Main_menu_selected_tv_guide_test_2	7	**89**
	Main_menu_selected_tv_guide_test_3	7	**85**
	Main_menu_selected_tv_guide_test_4	8	**100**
	Main_menu_selected_tv_guide_test_5	11	**100**
	Main_menu_selected_tv_guide_test_6	9	**85**
	Main_menu_selected_tv_guide_test_7	8	**94**
	Main_menu_selected_tv_guide_test_8	7	**89**
	Main_menu_selected_tv_guide_test_9	9	**100**
	Main_menu_selected_tv_guide_test_10	9	**99**
	Main_menu_selected_tv_guide_test_11	10	**97**
	Main_menu_selected_search_test_1	8	**100**
	Main_menu_selected_search_test_2	8	**100**
	Main_menu_selected_search_test_3	9	**98**
	Main_menu_selected_search_test_4	10	**100**
	Main_menu_selected_search_test_5	9	**100**
	Main_menu_selected_search_test_6	8	**100**
	Main_menu_selected_media_test_1	8	**100**
	Main_menu_selected_media_test_2	41	**100**
	Main_menu_selected_media_test_3	8	**100**
	Main_menu_selected_media_test_4	8	**95**
	Main_menu_selected_media_test_5	8	**96**
	Main_menu_selected_media_test_6	8	**97**
	Main_menu_selected_media_test_7	8	**100**
	Main_menu_selected_apps_test_1	14	**100**
	Main_menu_selected_apps_test_2	11	**97**
	Main_menu_selected_apps_test_3	10	**95**
	Main_menu_selected_apps_test_4	11	**93**
	Main_menu_selected_apps_test_5	11	**86**
	Main_menu_selected_apps_test_6	11	**88**
	Main_menu_selected_apps_test_7	10	**97**
	Main_menu_selected_apps_test_8	11	**91**
	Main_menu_selected_apps_test_9	10	**96**
Full screen	Channel_installation_test_1	78	**100**
	Channel_installation_test_2	84	**100**
	Channel_installation_test_3	85	**100**
Dynamic image single object comparison, no transformations	Guide_Saturday_test_1	**100**	**100**
	Guide_Saturday_test_2	80	**100**
	Guide_morning_test_1	**100**	**100**
	Guide_morning_test_2	98	**100**
	Guide_morning_test_3	**100**	**100**
	Guide_poster_test_1	96	**100**
	Guide_poster_test_2	83	**100**
	Guide_poster_test_3	50	**100**
	Guide_poster_test_4	**100**	**100**
	Guide_poster_test_5	67	**100**
	Guide_poster_test_6	57	**100**
	Guide_poster_test_7	68	**100**
	Guide_poster_test_8	61	**100**
	Guide_poster_test_9	52	**100**

working properly. This allows for a BBT system which requires no insight into the content of the device under test. An important part of the proposed system, the new DISM algorithm is robust to various image deformations caused by animations and effects from dynamic UIs such as translation, rotation, scaling and occlusion. The comparison of the proposed DISM algorithm to the only existing publicly available PBC algorithm for static image similarity measurement showed the DISM superiority over PBC when using for hybrid STB automated functional failure detection. In future work, the possibility of DISM algorithm usage in other applications will be examined.

REFERENCES

[1] S. Pravin and R. BalaKrishnan, "Set top box system with android support using Embedded Linux operating systempaper," *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, Nagapattinam, Tamil Nadu, 2012, pp. 474-478.

[2] K. Fujita, H. Kuwano, T. Tsuzuki, Y. Ono, and T. Ishihara, "A new digital TV interface employing speech recognition," *in IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 765-769, Aug. 2003, DOI: 10.1109/TCE.2003.1233816.

[3] D. Grbić, M. Vranješ, B. Kovačević, and M. Milošević, "Hybrid electronic program guide application for digital TV receiver," *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, 2017, pp. 177-180, DOI: 10.1109/ICCE-Berlin.2017.8210622.

[4] D. Grbić, J. Vlaović, M. Vranješ, and T. Bartulović, "Proposal of application format for hybrid digital TV developed for cost effective STBs," *2016 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, Novi Sad, 2016, pp. 55-58, DOI: 10.1109/ZINC.2016.7513654.

[5] N. Nikić, M. Herceg, V. Peković, and N. Šoškić, "System for DASH support verification in HbbTV environment," *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, 2017, pp. 212-216, DOI: 10.1109/ICCE-Berlin.2017.8210630.

[6] M. Plenković, M. Herceg, V. Peković, and S. Novak, "Coupling of HbbTV test system with automatic testing subsystem," *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, 2017, pp. 254-257, DOI: 10.1109/ICCE-Berlin.2017.8210643.

[7] D. Marijan, N. Teslic, M. Temerinac, and V. Pekovic, "On the effectiveness of the system validation based on the black box testing methodology," *2009 IEEE Circuits and Systems International Conference on Testing and Diagnosis*, Chengdu, 2009, pp. 1-4, DOI: 10.1109/CAS-ICTD.2009.4960847.

[8] M. Katona, I. Kastelan, V. Pekovic, N. Teslic, and T. Tekcan, "Automatic black box testing of television systems on the final production line," in *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 224-231, February 201, DOI: 10.1109/TCE.2011.5735506.

[9] D. Marijan, V. Zlokolica, N. Teslic, V. Pekovic, and T. Tekcan, "Automatic functional TV set failure detection system," in *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 125-133, February 2010, DOI: 10.1109/TCE.2010.5439135.

[10] V. Pekovic, V. Zlokolica, J. Zloh, M. Katona, and N. Teslic, "Automated system for testing and verification of control access kernel functionality in set-top boxes," *2012 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2012, pp. 25-26, DOI: 10.1109/ICCE.2012.6161721.

[11] D. Marijan, V. Zlokolica, N. Teslic, and V. Pekovic, "Quality assessment of digital television picture based on local feature matching," *2009 16th International Conference on Digital Signal Processing*, Santorini-Hellas, 2009, pp. 1-6, DOI: 10.1109/ICDSP.2009.5201147.

[12] H.-S. Han, D.-O. Kim, and R.-H. Park, "Structural information-based image quality assessment using LU factorization," in *IEEE Transactions on Consumer Electronics*, vol. 55, no. 1, pp. 165-171, February 2009, DOI: 10.1109/TCE.2009.4814430.

[13] M. Shahid, A. Rossholm, B. Lövström, and H.-J. Zepernick, "No-reference image and video quality assessment: a classification and review of recent approaches," *EURASIP J. Image Video Process.*, vol. 2014, no. 1, p. 40, 2014, DOI: 10.1186/1687-5281-2014-40.

[14] Q. Fan, W. Luo, Y. Xia, G. Li, and D. He, "Metrics and methods of video quality assessment: a brief review," *Multimed. Tools Appl.*, pp. 1–15, Jul. 2017, DOI: 10.1007/s11042-017-4848-x.

[15] A. B. Lamb and M. Khambete, "No-reference perceived image quality measurement for multiple distortions," *Multimed. Tools Appl.*, pp. 1–23, May 2017, DOI: 10.1007/s11042-017-4761-3.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004, DOI: 10.1109/TIP.2003.819861.

[17] H. R. Sheikh and A. C. Bovik, "A visual information fidelity approach to video quality assessment," in *Proc. Int. Workshop on Video Proc. and Quality Metrics for Consum. Electron.*, Scottsdale, AZ, USA, 2005, pp. 23–25.

[18] "Computer Generated Image Alterations Database (CGIAD)." [Online]. Available: http://www.rt-rk.com/other/CGIAD.html. [Accessed: 07-Sep-2018].

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, DOI: 10.1023/B:VISI.0000029664.99615.94.

[20] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008, DOI: 10.1016/j.cviu.2007.09.014.

[21] K. M. Singh, "A robust rotation resilient video watermarking scheme based on the SIFT," *Multimed. Tools Appl.*, pp. 1–26, Sep. 2017, DOI: 10.1007/s11042-017-5213-9.

[22] Z. Elleuch and K. Marzouki, "Multi-index structure based on SIFT and color features for large scale image retrieval," *Multimed. Tools Appl.*, vol. 76, no. 12, pp. 13929–13951, Jun. 2017, DOI: 10.1007/s11042-016-3788-1.

[23] S.-H. Nam, W.-H. Kim, S.-M. Mun, J.-U. Hou, S. Choi, and H.-K. Lee, "A SIFT features based blind watermarking for DIBR 3D images," *Multimed. Tools Appl.*, pp. 1–40, May 2017, DOI: 10.1007/s11042-017-4678-x.

[24] R. Jin and J. Kim, "Tracking feature extraction techniques with improved SIFT for video identification," *Multimed. Tools Appl.*, vol. 76, no. 4, pp. 5927–5936, Feb. 2017, DOI: 10.1007/s11042-015-2694-2.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision*, Barcelona, 2011, pp. 2564-2571, DOI: 10.1109/ICCV.2011.6126544.

[26] Y. Zhu, X. Shen, and H. Chen, "Copy-move forgery detection based on scaled ORB," *Multimed. Tools Appl.*, vol. 75, no. 6, pp. 3221–3233, Mar. 2016, DOI: 10.1007/s11042-014-2431-2.

[27] Y. Sugama and T. Murase, "Projection-based AR book system involving book posture detection and robust page recognition," *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2017, pp. 13-14, DOI: 10.1109/ICCE.2017.7889211.

[28] W.-H. Peng, M.-Y. Lee, T.-H. Li, C.-H. Huang, and P.-C. Lin, "Performance comparison of image keypoint detection, description, and matching methods," *2016 IEEE 5th Global Conference on Consumer Electronics*, Kyoto, 2016, pp. 1-2, DOI: 10.1109/GCCE.2016.7800416.

[29] H. Javidnia and P. Corcoran, "Real-time automotive street-scene mapping through fusion of improved stereo depth and fast feature detection algorithms," *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2017, pp. 225-228, DOI: 10.1109/ICCE.2017.7889293.

Matija Pul received the M.D degrees in electrical engineering from Faculty of electrical engineering, computer science and information technology in Osijek, Croatia in 2016. He works in R&D at Institute RT-RK Osijek LLC for information technologies. His research interests include image/video processing, object detection and embedded systems.

Vukota Peković received M.S. degree in computer science and automation from University of Novi Sad, Serbia in 1996. He is currently a Head of BBT Department at RT-RK Company. He has a particular interest in system software design, automated

test system design and embedded software design.

Mario Vranješ (S'08–M'12) received the M.S, and Ph.D. degrees in electrical engineering from Faculty of electrical engineering in Osijek, Croatia in 2006, and 2012 respectively. He is an Assistant Professor with the Dept. of Communications at the same institution. His research interests include mobile communications, image/video processing, transmission.

Ratko Grbić received his B.Sc. and Ph.D. degrees in 2005 and 2013 from the Faculty of Electrical Engineering, University of Osijek, Croatia. He is an assistant professor at the same institution. His research interests include process modelling and monitoring, image processing and computer vision.