2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Luke: An Autonomous Robot Photographer

Manfredas Zabarauskas, Stephen Cameron
Department of Computer Science, University of Oxford, UK

***Abstract*— An autonomous robot photographer should move around a crowd, avoiding collisions while taking reasonable pictures of the subjects within it. We present a system based on a low-cost Turtlebot platform, the Kinect sensor, and a consumer-grade stills camera, that fulfills this task while applying several rules of general photograph composition. The subsumption control system was built using the ROS platform, and novel software components include the crowd head detection, the photograph composition procedures, the subject-interface, and uploading the photographs to a social media website. The result is a system that shows good performance in a public environment, as evaluated by independent human judges; performing a statistical analysis of these results against other published work verified the quality of the resulting photographs.**

I. INTRODUCTION

Within the field of autonomous robotics (and the variety of its application areas), robot photographers serve as excellent low-cost research platforms. They encompass a number of challenges common in robotics research, like task and path planning, locomotion and navigation (including obstacle avoidance), and human subject detection/ tracking.

Robot photographers also include multidisciplinary challenges, like automatic photograph composition (which requires computational understanding of the aesthetics of photography) and Human-Robot Interaction (HRI). As pointed out by Byers et al. [9], robotic photographer platforms are particularly well suited for HRI research, since the general public can easily grasp the overall concept ("it's a robot whose job is to take pictures"), and thus tend to interact with the robot naturally.

Here we describe a system that uses a single Microsoft Kinect sensor for obstacle avoidance and people detection, coupled with a consumer "point-and-shoot" camera for taking the final images, and which outperforms earlier robot photographer approaches.

II. RELATED WORK

The earliest implementation of an end-to-end mobile robot system capable of taking well-composed pictures is described by Byers et al. [9], which uses color to isolate subject faces. Campbell and Pillai [10] relies on optical flow, and Ahn et al. [1] used the Viola-Jones detector [32] and incorporated rules for photograph composition. Kim et al. [23] proposed a mobile photographer robot which uses a combination of sound and skin tone to detect subjects, and also used a number of composition rules. Gadde and Karlapalem [17] present a stationary robot based on a humanoid NAO platform [20], which can take pictures of static scenes containing both human and non-human subjects using three aesthetic criteria.

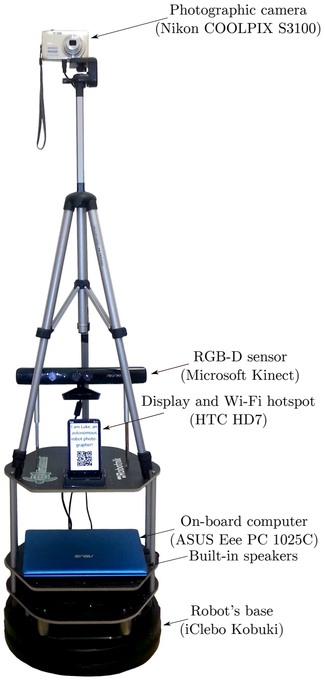

Fig. 1. Luke, an autonomous event photographer robot.

III. PROPOSED ROBOT PHOTOGRAPHER'S HARDWARE

The developed autonomous robot photographer, Luke (Fig. 1), is built using iClebo Kobuki's base, which has a 4kg payloads, maximum velocity of 65cm/s, and a base integrated with the Turtlebot 2 open robotics platform. For its vision, Luke uses a Microsoft Kinect RGB-D sensor, which provides both a 10-bit depth value and VGA resolution color at 30 FPS. The sensor has a combined 57° horizontal and 43° vertical field-of-view. The Kinect was attached to the Turtlebot's base at an inclination of 10° in order to be able to track upright standing humans at a distance of 1.5–2m. Since this limits low obstacle detection abilities, the linear velocity of the robot is limited to 10cm/s and the bumpers on Kobuki's base are used to provide graceful recovery in the case of collision with a low-lying obstacle.

To take the photographic pictures Luke uses a simple point-and-shoot Nikon COOLPIX S3100 camera, which has a maximum resolution of 14 megapixels, a built-in flash, and supports automatic exposure/ISO sensitivity/white balance settings. This camera is mounted on a lightweight, aluminium König KN-TRIPOD21 tripod (weighing 645g), which is attached to the top mounting plate of the robot. The overall size of the robot is approximately 34cm×135cm×35cm (W×H×D).

For Luke's state externalization, an HTC HD7 smartphone

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

with a 4.3 inch LCD display was mounted onto the robot. The display has a resolution of 480×800 pixels, and is used to display Luke's state messages and to show the QR (Quick Response) codes containing the URLs of the pictures that Luke takes and uploads to Flickr. The smartphone also serves as a wireless hotspot, providing a wireless network connection between Luke's on-board computer and a monitoring/debugging station. Furthermore, it provides the internet connection to the on-board computer (for photo uploading to Flickr) by tethering the phone's 3G/EDGE connection over Wi-Fi.

The on-board ASUS Eee PC 1025C netbook has an Intel Atom N2800 1.6GHz CPU and 1GB RAM, providing a battery life of around 3 hours and weighs just under 1.25kg. It is running the Groovy Galapagos version of the Robot Operating System framework (ROS, [29]) on a Ubuntu 12.04 LTS operating system. All processing (including obstacle avoidance, human subject detection, photographic composition evaluation and so on) is done on this machine.

The robot contains two major power sources: a 2200mAh lithium-ion battery which is enclosed in the Kobuki's base, and a 5200mAh lithium-ion battery installed in the on-board netbook. The former powers the robot and also provides some power to the Kinect, whilst the other components are powered by the netbook. During the empirical tests of the fully-powered robot, the average discharge times for the netbook's/Kobuki base's batteries were 3h6m/3h20m respectively.

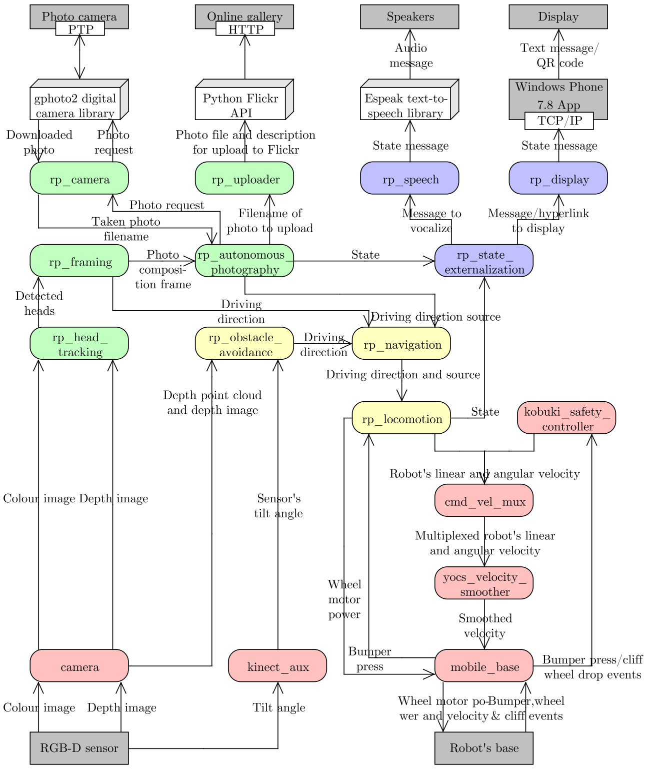

Fig. 2. A simplified Luke's architectural design diagram, showing ROS nodes (red, green, blue and yellow) together with I/O devices (gray rectangles), and the data that is being passed between them (text on the arrows). All nodes with prefixes *rp_* (green, blue and yellow) are the results of the work presented in this paper, the red nodes are parts of Kobuki/ROS/GFreenect/Kinect AUX libraries. Yellow, green and blue nodes represent the first, second and third Luke's competence levels (corresponding to obstacle avoidance, human tracking/photograph taking and state externalization behaviors).

IV. ROBOT PHOTOGRAPHER'S SOFTWARE

A. Architectural Design

Luke's software uses Brooks' [5] hierarchical levels-of-competence approach. Each of the layers in Luke's software hierarchy is based on the behaviors that Luke can perform:

- The base layer allows Luke to aimlessly wander around the environment, while avoiding collisions.
- The second layer suppresses the random wandering behavior at certain time intervals (adhering to what [5] called a *subsumption* architecture), and enables Luke to compose, take and upload photographs.
- The final layer enables Luke to externalize his state i) visually, by showing text messages/QR codes on the attached display, and ii) vocally, by reading state messages out loud using text-to-speech software.

This architecture is illustrated in Fig. 2, and key implementation details are summarized below.

B. Layer I: Random Walking with Collision Avoidance

Luke's capability to randomly wander in the environment without bumping into any static or moving obstacles is implemented in three ROS nodes: *rp_obstacle_avoidance*, *rp_locomotion* and *rp_navigation*. Obstacle detection and avoidance are based on [3], chosen due to its computational efficiency and suitability for the random navigation mode which Luke uses to wander around in the environment. It consists of three main steps:

1) The input point cloud is obtained using the GFreenect library [27], and subsampled using a voxel grid filter.
2) The Kinect's tilt angle is provided by the Kinect AUX library [13] which returns the readings from the Kinect's accelerometer at 20Hz, and with a flat-floor assumption used to tweak the subsampled point cloud.
3) The region of interest (ROI) in front of the robot (defined by the user) is isolated from the transformed point cloud and a moving average of the ROI's size calculated. A positive average size generates a turn direction, otherwise the robot moves forward.

To prevent the robot from getting stuck in an oscillating loop when facing a large obstacle, it is prohibited from changing the direction of the turn once it has started turning, as suggested in [3]. Also, an improvement from [28] is used whereby if the unfiltered point cloud is small then it is assumed that the robot is facing a nearby large obstacle, and a turn directive is issued.

C. Layer II: Taking Well-Composed Photographs of Humans

Luke's second major behavioral competence involves his ability to i) track humans in an unstructured environment, ii) take well-composed pictures of them, and *iii*) upload these pictures to an on-line picture gallery. This competence layer is implemented by five ROS nodes: *rp_head_tracking*, *rp_framing*, *rp_camera*, *rp_uploading* and *rp_autonomous_photography*.

The head detection and tracking node (*rp_head_tracking*) is the most sophisticated node in Luke's ROS graph. For subject head detection it uses a knowledge-based method by Garstka and Peters' [18], which we extended to cope with multiple people present in the image. To improve the head detection results, it employs one of the two skin detectors: a Bayesian skin detector by Jones and Rehg [22], and an adaptive skin detector based on a logistic regression classifier with a Gaussian kernel, and trained on an on-line skin model obtained from the face regions detected using the Viola and Jones [32] detector. Finally, to exploit the spatial locality of human heads over a sequence of frames, this node uses a depth-based extension of the continuously-adaptive mean-shift algorithm by Bradski [4]. Our multiple person extension still scans through a blurred and depth-shadow-filtered depth image one horizontal line at a time, from top to bottom. However, instead of keeping a single potential vertical head axis, a set of vertical head axes $\{H_1, ..., H_k\}$ is constructed. Each head axis is represented by $H_i = \left(\overline{d}_i, (X, Y)_i\right)$, where $\overline{d}_i$ is the average candidate head distance from the sensor, and $(x, y) \in (X, Y)_i$ are the image points on the vertical axis. When a new arithmetic mean of left and right lateral gradients $\overline{x}(y)$ is calculated in the original algorithm, the extended method searches for the head axis H_j such that the last added point $(x', y') \in (X, Y)_j$ is within 5cm distance from the point $(\overline{x}(y), y)$, in which case $(X, Y)_j$ is updated by adding the point $(\overline{x}(y), y)$, and the average head distance $\overline{d}_j$ is recalculated.

A vertical head axis H_i is classified as a detected candidate head if it is closer than 5m, is between 20–30cm tall, and is rotated by less than 35°. A few examples of multiple head detection using this method are shown in figure 3.

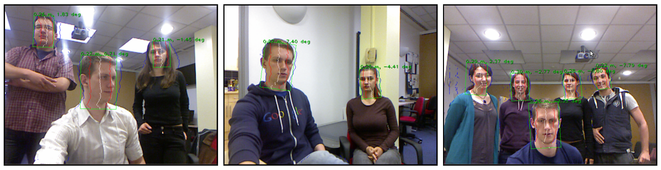

Fig. 3. Multiple head detection examples using the proposed extension of Garstka and Peters' [18] method.

The detected candidate heads are verified using one of two skin detection methods: a Bayesian classifier trained off-line on a very large scale skin/non-skin image dataset [22], and an on-line skin detector trained using skin histograms obtained from a small set of face detections using Viola and Jones [32] detector. In the first case a histogram-based Bayesian classifier similar to [22] is implemented and trained off-line on a large-scale Compaq skin dataset containing nearly a billion pixels, each hand-labelled as skin or non-skin. If this skin classification method is used, then a given candidate head detection is accepted/rejected based on the proportion of skin-color pixels in the corresponding RGB image region. In the second case, many faces are detected in the initialization stage using the frontal and profile face detectors using [32]. For each of the detected face rectangles, a binary mask is applied to segment the image into face oval/background regions, and the pixel hue histograms are assembled in each of the regions. Then these histograms are used as feature vectors in kernel logistic regression (KLR) classifier training. When this classifier is trained, the depth-based head detections can be verified by applying the same oval binary mask to the detected head rectangle, constructing a hue histogram from the face region, applying the KLR classifier, and thresholding.

To further reduce the computational complexity requirements of head/skin detection methods described above, a depth-data based extension of the continuously adaptive mean-shift tracking algorithm (CAMShift, [4]) is employed to exploit the spatial locality of humans over a sequence of frames. While the original CAMShift algorithm uses the probability distribution obtained from the color hue distribution in the image, in this project it is adapted to use the depth information. In particular, the constraints that [18] use to reject local horizontal minima which could not possibly lie on the vertical head axis are used to define the following degenerate head probability:

$$\Pr(\textit{head}\,|(x,y)) = \begin{cases} 1, & \text{if pixel } (x,y) \text{ is a local minimum} \\ & \text{in depth image and it satisfies the} \\ & \text{inner/outer head bound constraints,} \\ 0, & \text{otherwise,} \end{cases}$$

which is then tracked using CAMShift.

The second most important node in Luke's "picture taking" behavioral capability layer is the photograph composition and framing (*rp_framing*) node. This node works as follows. First of all, it subscribes to the locations of detected/tracked human subject heads in the Kinect's image plane, published by the *rp_head_tracking* node. Then this node maps the head locations from Kinect's image plane to the photographic camera's image plane and calculates the ideal framing based on the framing rules described by Dixon et al. [12]. If the calculated ideal frame lies outside the current photographic camera's image plane, a turn direction is proposed; otherwise, the ideal frame location is published over the */rp/framing/frame* topic.

In order to map the locations of detected heads from Kinect's to photographic camera's image plane, the rigid body translation vector is first established between the Kinect senor and the photographic camera. Then, the photographic camera is undistorted using a "plumb bob" model proposed by [6]. This model simulates (and hence can be used to correct) both radial distortion caused by the spherical shape of the lens, and the tangential distortion arising from the inaccuracies of the assembly process. Finally, the approach of [34], [35] is used to estimate the camera intrinsics (*viz.* camera's focal length and principal point offsets). Then, any point in Kinect's world space can be projected into the photographic camera's image space. Using this approach, the 3D locations of the detected heads (provided by the *rp_head_tracking* node) are projected onto the photographic camera's image plane. Then, based on these locations the ideal framing for the photographs is calculated using the photograph composition heuristics proposed by [12]. These heuristics are based on the following four photographic composition rules [21]:

- *Rule of thirds*, which suggests that the points of interest in the scene should be placed at the intersections (or along) the lines which break the image into horizontal and vertical thirds.
- *No middle* rule, which states that a single subject should not be placed at a vertical middle line of the photograph.
- *No edge* rule, which states that the edges of an ideal frame should not be crossing through the human subjects.
- *Occupancy* ("empty space") rule, which suggests that approximately a third of the image should be occupied by the subject of the photograph.

Given these rules, [12] define three different heuristics for single person and wide/narrow group picture composition. In order to choose which heuristic will be used they employ an iterative procedure. It starts by identifying a human subject closest to the center of the current image and calculating the ideal framing for this person using the single person composition heuristic. If this frame includes other candidate subjects, the group framing rules are applied iteratively on the expanded candidate set, until no new candidates are added.

After an ideal frame F is calculated, the *rp_framing* node calculates the overlap coefficient between the part of the frame visible in the current image and the whole frame. If this exceeds a given threshold and the visible part of the frame exceeds the minimal width/height thresholds $\theta_w \times \theta_h$, the node considers that the satisfying composition has been achieved and publishes the position/size of the ideal frame over the */rp/framing/frame* topic. Otherwise, the framing node determines the direction the robot should turn in order to improve the quality of the composition[1] and publishes these driving directions over the */rp/framing/driving_direction* topic. In order to prevent the robot from getting stuck indefinitely while trying to achieve an ideal framing, a decaying temporal threshold for the minimum required overlap is also used. In the current robot photographer's implementation, the framing time limit is set to 60 seconds, the maximum deviation from the ideal overlap is set to 50% and the minimum visible frame size thresholds $\theta_w \times \theta_h$ are set to 2160×1620 pixels.

The *rp_autonomous_photography* node coordinates the actual photograph taking/uploading process, and divides the robot's control time between the obstacle avoidance (*rp_obstacle_avoidance*) and framing (*rp_framing*) nodes. The photograph taking node (*rp_camera*) acts as an interface between other ROS nodes and the physical Nikon COOLPIX S3100 camera that Luke uses to take pictures using the *libgphoto2* API for the open-source gPhoto^2 library [2], which in turn connects to the camera using the Picture Transfer Protocol (PTP). This node provides access to the camera for the rest of the Luke's ROS graph by exposing a ROS service at */rp/camera/photo*. Any other ROS node can send an empty request to this service, which *rp_camera* node transforms into the photo capture request for the *libgphoto2* API. This request triggers a physical camera capture, storing the taken picture in the camera's built-in memory. After the picture is taken, *rp_camera* node moves the picture from the camera's memory to the on-board computer and returns the string file name of the downloaded picture via the service response. The photograph uploading node (*rp_uploader*) uses the Python Flickr API [31] to upload image files to an online Flickr photo gallery. It exposes the Flickr API to the rest of ROS graph by providing the */rp/uploader/upload* ROS service.

D. Layer III: Externalization of the Current State via Vocal and Visual Messages

Luke's third and final behavioral competence involves its ability to externalize its current state via synthesized voice messages (played over the on-board computer's speakers), and text messages/QR codes (shown on the display of the attached HTC HD7 phone). The state externalization node (*rp_state_externalization*) subscribes to the status outputs from all major nodes in Luke's ROS graph, in particular, the lomotion (*rp_locomotion*), head tracking (*rp_head_tracking*), framing (*rp_framing*) and photography process control (*rp_autonomous_photography*) nodes. In order to produce the robot's state messages (which are later vocalized/displayed by *rp_speech* and *rp_display* nodes) the state externalization node uses a table of pre-defined text messages, indexed by the states of four major nodes listed above. If the table contains more than one message for a given collection of states, then the message to be produced is chosen uniformly at random from the matching messages. To show the messages generated by the *rp_state_externalization* node, a Windows Phone OS app connects to the *rp_display* node over TCP and renders received text messages in full-screen mode. If a hyperlink is present within the received text message then this app also generates and renders a QR (Quick Response) code. To vocalize the text messages sent by *rp_externalization* node, the *rp_speech* node uses an open-source eSpeak [14] speech synthesis engine, which in turn is configured to use a formant synthesis based approach as described by Klatt [24]. Since this method does not need a database of speech samples and uses computationally cheap digital signal filters, the resulting text-to-speech engine is both memory and CPU efficient, making it highly appropriate for the use in a mobile robot.

E. Computational Resource Usage of Individual Nodes

In order to illustrate how CPU resources are used by the individual Luke's nodes, their usage statistics were gathered over a ten minute sequence of Luke's operation. The obstacle

[1] Based on the position of the ideal frame's center *w.r.t.* the image's center.

detection/avoidance (*rp_obstacle_avoidance*) and head detection/tracking (*rp_head_tracking*) nodes use the largest amount of the computational resources (approximately 40%/25% of the CPU respectively).

TABLE I
STATISTICS OF LUKE'S DEPLOYMENT AT AN OPEN-DAY EVENT

Day	Total robot's presence time	Total duration of robot's activity	Number of photos taken	Average time between pictures
1	5h26m25s	2h11m34s	57	2m18s
2	5h10m16s	1h25m59s	46	1m52s
Total	10h36m41s	3h37m33s	103	2m07s

V. INSIGHTS FROM ROBOT PHOTOGRAPHER'S DEPLOYMENT IN REAL-WORLD

In June 2013 the robot was deployed at a public event attended by more than four hundred people over two days, in a large area primarily illuminated with natural lighting. For its successful operation, Luke had to avoid both static obstacles (chairs, tables, presentation stands, walls, *etc*.) and dynamic obstacles (*viz*. attendees, randomly milling about). In order to ensure Luke's safe operation, its linear velocity was limited to 10cm/s, and the angular velocity was limited to 0.5rad/s. At this speed, the obstacle detection and avoidance method described in section IV performed flawlessly, allowing Luke to avoid any collisions during its operation. The only times when human supervision was required were when Luke was about to wander out into the corridors leading away from the area in which it operated; in these cases Luke was directed back by blocking its path and forcing it to turn around; such interventions were required around once in every thirty minutes of Luke's operation.

Due to its relatively slow speed (and noisiness of the environment, which was burying Luke's audio messages), the robot often was able to take candid photographs of the attendees while remaining unnoticed (just like a human event photographer would). However, some people indicated that they found the presence of the robot somewhat unsettling or creepy because of its ability to track people and navigate the environment autonomously. This could have been caused by the fact that Luke has a very limited set of reactive interaction skills with humans: most of the times the HRI was limited to the attendees blocking the robot's path, and Luke changing his driving direction to avoid them. At this point, the attendees would notice the message on Luke's status display that he is "looking around for good picture locations" (or similar), and leave him alone.

Over the total ten and a half hour period of Luke's presence, the robot spent three hours and thirty-seven minutes actively taking pictures, *viz.* during the breaks between various events. In total Luke took 103 pictures, approximately one picture every two minutes (some examples of pictures taken are displayed in Fig. 5). The breakdown of these statistics for each day is shown in table I.

In order to quantitatively evaluate the quality of the pictures taken by Luke, sixteen people (unrelated to the project) were asked to evaluate all 103 pictures on the following Likert scale:

Very bad (1), *Bad (2)*, *Neutral (3)*, *Good (4)*, *Very good (5)*.

To reduce the combined perceptual correlations between the neighboring photographs, all pictures were shown in a randomized order for each participant of the experiment.

Using an online tool built on purpose for this task, a set of 1,648 ratings was collected (*i.e.* sixteen ratings on the five-point Likert scale for each picture). To better understand how reliable are the collected ratings, the inter-rater agreement was measured using the weighted Cohen's kappa statistic [11], and a simple percentage agreement.

A. Statistical Comparison to Ratings Obtained by Earlier Robot Photographers

Most of the robot photographer development approaches described in the literature and listed in section II either did not provide thorough quantitative evaluations of the obtained photographs (including Campbell and Pillai [10], Kim et al. [23] and Shirakyan et al. [30]) or the approaches were not directly comparable to the one used in this project (*e.g.* the robot photographer developed by Gadde and Karlapalem [17] was stationary and took pictures only of static scenes).

For this reason, the results obtained by Luke were compared to the results obtained in the approaches described by Byers et al. [9] and Ahn et al. [1], who provided the quantitative evaluations of the pictures that their robots took on the Likert scales.

To that end, the comparison of photograph proportions in each of the rating categories for each robot photographer approach are given in table II and figure 4, and the statistical summary of the results (using both-parametric and non-parametric statistics) is provided in table III.

TABLE II
COMPARISON OF PICTURE PROPORTIONS IN RATING CATEGORIES

Authors	Very bad (1)	Bad (2)	Neutral (3)	Good (4)	Very good (5)
Byers et al. (2003)	18.0%	25.0%	28.0%	20.0%	9.0%
Ahn et al. (2006) †	6.7%	23.5%	32.7%	26.0%	11.1%
This paper (2014)	4.7%	14.6%	25.5%	33.5%	21.7%

† Ahn et al. [1] used the following Likert scale: "*Very poor*", "*Poor*", "*Normal*", "*Nice*", "*Very nice*".

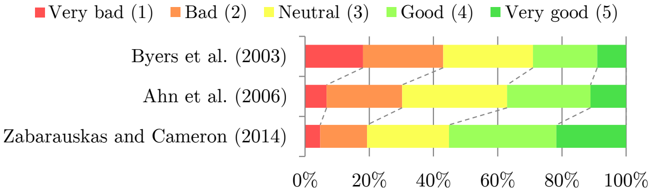

Fig. 4. Visual summary of the proportion of pictures in each of the rating categories in three different robot photographer approaches.

As shown in table II and Fig. 4, more than half of all pictures taken by Luke were evaluated by humans as being either "good" or "very good", and more than 80% were evaluated as "neutral" or better, significantly outperforming Byers et al.'s [9] and Ahn et al.'s [1] robots.

TABLE III

Comparison of Parametric and Non-Parametric Statistics of the Pictures Taken by Different Robot Photographers

Authors	Central tendency			Variability	
	Mean	Median	Mode	Standard deviation	Inter-quartile range
Byers et al. [9]†	2.77	Good (3)	Good (3)	1.22	2
Ahn et al. [1]	3.11	Good (3)	Good (3)	1.09	2
Zabrauskas and Cameron	3.53	Very good (4)	Very good (4)	1.12	1

† Since Byers et al. mention that around 2,000 photographs were evaluated (without specifying the exact number), the comparisons in the following sections assume an exact rating count of 2,000.

To verify the statistical significance of the obtained results, both parametric and non-parametric methods were employed to reject the hypothesis that the real rating means (and therefore the underlying qualities of the pictures taken by each robot) are equivalent, and the differences observed in the individual rating sets (as summarized in table II/figure 4) arise purely by chance.

First, a parametric test based on one-way analysis of variance (ANOVA, [16]) was performed on all three rating sets simultaneously. In order for the ANOVA test to be applicable (besides the independence assumption described above), the responses ideally should be normally distributed. While it is not obvious whether the rating sets satisfy this condition, they are also not extremely skewed (the kurtosis of each set respectively is -0.614, -0.71 and -0.92). Due to the fact that ANOVA test is not extremely sensitive to non-normality, it is still considered to be applicable. However, a standard ANOVA test also relies on homoscedasticity[2] of the rating sets. It is unlikely that the homoscedasticity condition holds in this situation, and to mitigate this problem, an ANOVA-based $F^\star$ test [7] [3] was used, which is designed for sample sets with non-equal variance (and is less sensitive to non-normality than the regular ANOVA test), as shown in table IV.

Due to some degree of controversiality regarding the use of ANOVA with data obtained using Likert scales, the same hypothesis was tested using a non-parametric version of one-way analysis of variance by [25]. This test does not assume an underlying normal distribution or the homogeneity of variances, and the result is also shown in table IV.

Since both parametric and non-parametric tests confirm statistically significant differences between the rating set means simultaneously (at $p = .0001$ confidence level), but they do not specify whether (and which) individual pairs of rating sets come from different underlying distributions, a number of pair-wise post-hoc tests were performed. First of all, each pair of rating sets was compared using a version of parametric Student's t-test [19]. This test can be used to reject the null hypothesis that the underlying populations from which the two rating sets have been sampled actually have equal mean ratings, and the observed differences arise purely by chance. However, this test also assumes homoscedasticity, which cannot be easily shown for the rating sets. While it is relatively insensitive to violations of this assumption when the sample sizes are equal, it is not very robust in presence of unequal variances and unequal sample sizes. Welch [33] proposes a version of Student's t-test, which is insensitive to unequal variances even for different-size samples. To avoid inflating the Type I error by performing repeated null hypotheses tests, the Bonferroni correction [15] was used. In particular, since three tests are required to check each pair of the rating sets for the equality of their means, the p-value of 0.0001 (as used above) is discounted by a factor of three, *viz.* $p = 0.0001/3 = 0.0000333$ and thus $\alpha = 99.9967\%$. For this level of statistical significance, Welch's t-test rejects the null hypotheses of mean equality for each of the rating set pairs.

Since Welch's t-test interprets the Likert scales as interval data (which is somewhat controversial, as mentioned above), the same post-hoc tests were repeated using a non-parametric analogue of the t-test as proposed by Mann and Whitney [26]. In contrast to Welch's t-test, Mann-Whitney's U test[4] does not assume interval data (*i.e.* it can be straight away applied to ordinal data, like Likert scales). This test can be used to refute a similar null hypothesis as the Welch's t-test (*viz.* the assumption that the two rating sets come from the same distribution). Under the same Bonferroni correction, p-value of $0.0001/3$ is chosen for each test of a rating set pair. For this level of statistical significance, Mann-Whitney U test also rejects the "equal distribution" null hypotheses for each of the rating set pairs.

TABLE IV

Statistical Significance of Similarity b/t All Rating Sets

Statistical test	Test score	Critical score ($p = .0001$)	Null hypothesis	Decision
Parametric (Brown-Forsythe)	$F^\star = 199.8352$	$F^{\text{crit}} = 9.2305$	"Means of underlying rating populations are equal."	Rejected
Non-parametric (Kruskal-Wallis)	$K = 353.1212$	$K^{\text{crit}} = 18.4207$	"Mean ranks of underlying rating populations are equal."	Rejected

VI. Summary and Conclusions

This paper described RGB-D data-based solutions for the main autonomous robot photographer challenges. The described methods were implemented within an open-source Robot Operating System framework [29], and achieved sufficient performance for real-time application on a modest configuration on-board netbook[5]. To test this software in real-world situations, a physical robot has been built using an open-reference Turtlebot platform (available off-the-shelf, or as a construction kit), a simple point-and-shoot photographic

[2] Also known as the homogeneity/equality of variances.

[3] $F^\star$ test is used for the equality of *means* and should not be confused with a different [8] test for the equality of *variances*.

[4] Also known as Wilcoxon rank-sum test.

[5] The source code of the implemented robot photographer is available at `http://zabarauskas.com/robot-photographer`.

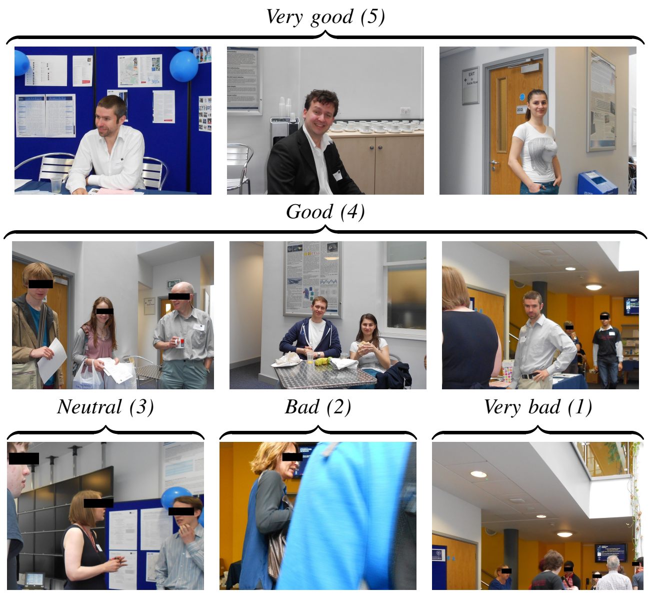

Fig. 5. Luke's photographs with the largest rating count in each category.

camera and a low-cost RGB-D Microsoft Kinect sensor. The developed autonomous photographer robot "Luke" has been deployed in an unstructured open-day event. During this event, Luke took more than a hundred pictures, shooting a new photograph roughly every two minutes of its operation. Most of the subjects accepted Luke's presence, with just a few showing concern. All taken pictures were evaluated by sixteen judges (unrelated to the project), yielding a total of 1,648 ratings which were determined to be sufficiently reliable by inter-rater agreement measurements. More than half of Luke's pictures were rated by judges as being "good" or "very good" on a five-point Likert scale, markedly exceeding the results reported in [9] and [1]. The statistical significance of these results has been confirmed by both parametric and non-parametric tests at the $p = .0001$ significance level.

References

[1] Hyunsang Ahn, Dohyung Kim, Jaeyeon Lee, Suyoung Chi, Kyekyung Kim, Jinsul Kim, Minsoo Hahn, and Hyunseok Kim. A Robot Photographer with User Interactivity. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5637–5643, 2006.

[2] Christophe Barbé, Hubert Figuière, Hans Ulrich Niedermann, Marcus Meissner, and Scott Fritzinger. gPhoto2: Digital Camera Software, 2002.

[3] Sol Boucher. Obstacle Detection and Avoidance using TurtleBot Platform and XBox Kinect. Technical report, Rochester Institute of Technology, 2012.

[4] Gary R Bradski. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal*, (Q2):214–219, 1998.

[5] R Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[6] Duane C Brown. Decentering Distortion of Lenses. *Photometric Engineering*, 32(3):444–462, 1966.

[7] M B Brown and A B Forsythe. The Small Sample Behavior of Some Statistics Which Test the Equality of Several Means. *Technometrics*, 16(1):129–132, 1974.

[8] Morton B Brown and A B Forsythe. Robust Tests for Equality of Variances. *Journal of the American Statistical Association*, 69:364–367, 1974.

[9] Zachary Byers, Michael Dixon, Kevin Goodier, Cindy M Grimm, and William D Smart. An Autonomous Robot Photographer. *Intelligent Robots and Systems*, 3:2636–2641, 2003.

[10] J. Campbell and P. Pillai. Leveraging Limited Autonomous Mobility to Frame Attractive Group Photos. In *IEEE International Conference on Robotics and Automation*, pages 3396–3401, 2005.

[11] J Cohen. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychological Bulletin*, 70(4):213–220, 1968.

[12] Michael Dixon, C Grimm, and W Smart. Picture Composition for a Robot Photographer. Technical report, Washington University in St. Louis, 2003.

[13] Ivan Dryanovski, William Morris, Stéphane Magnenat, Radu Bogdan Rusu, and Patrick Mihelich. Kinect AUX Driver for ROS, 2011.

[14] Jonathan Duddington. eSpeak: Speech Synthesizer, 2006.

[15] O. J. Dunn. Multiple Comparisons Among Means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[16] R A Fisher. On a Distribution Yielding the Error Functions of Several Well-Known Statistics. In *Proceedings of the International Congress of Mathematicians*, pages 805–813, 1924.

[17] Raghudeep Gadde and Kamalakar Karlapalem. Aesthetic Guideline Driven Photography by Robots. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 3, pages 2060–2065, 2011.

[18] Jens Garstka and Gabriele Peters. View-dependent 3D Projection using Depth-Image-based Head Tracking. In *Proceedings of the 8th IEEE International Workshop on Projector-Camera Systems*, pages 52–58, 2011.

[19] W S Gosset. The Probable Error of the Mean. *Biometrika*, 6(1):1–25, 1908.

[20] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The NAO Humanoid: a Combination of Performance and Affordability. In *Computing Research Repository*, pages 1–10, 2008.

[21] Tom Grill and Mark Scanlon. *Photographic Composition*. American Photographic Book Publishing, Orlando, FL, 1990.

[22] Michael Jones and James M Rehg. Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[23] Myung-Jin Kim, Tae-Hoon Song, Seung-Hun Jin, Soon-Mook Jung, Gi-Hoon Go, Key-Ho Kwon, and Jae-Wook Jeon. Automatically Available Photographer Robot for Controlling Composition and Taking Pictures. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6010–6015, 2010.

[24] D H Klatt. Software for a Cascade/Parallel Formant Synthesizer. *Journal of the Acoustical Society of America*, 67(3):971–995, 1980.

[25] William H Kruskal and Allen W Wallis. Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[26] Henry B Mann and Donald R Whitney. On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[27] OpenKinect. GFreenect Library, 2012.

[28] Brian Peasley and Stan Birchfield. Real-Time Obstacle Detection and Avoidance in the Presence of Specular Surfaces using an Active 3D Sensor. In *2013 IEEE Workshop on Robot Vision (WORV)*, pages 197–202, January 2013.

[29] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: an Open-Source Robot Operating System. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, 2009.

[30] Greg Shirakyan, Loke-Uei Tan, and Hilary Davy. Building Roborazzi, a Kinect-Enabled Party Photographer Robot, 2012.

[31] Sybren A. Stüvel. Python Flickr API, 2007.

[32] Paul Viola and Michael Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[33] B L Welch. The Generalization of "Student's" Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1–2):28–35, 1947.

[34] Zhengyou Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, pages 666–673, 1999.

[35] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.