2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

A Saliency-Based Approach to Boost Trail Detection

Pedro Santana, Nelson Alves, Luís Correia and José Barata

***Abstract*— This paper presents a saliency-based solution to boost trail detection. The proposed model builds on the empirical observation that trails are usually conspicuous structures in natural environments. This hypothesis is confirmed by the experimental results, where a strong positive correlation between trail location and visual saliency has been observed. These results are due in part to the proposed extensions to a well known visual saliency computational model. This paper goes further and shows that, with a proper analysis of the saliency information alone, the ambiguity regarding both trail's position and approximate skeleton is reduced to three hypotheses in 98% of the tested natural images. This analysis is performed by a set of agents inhabiting the saliency and feature specific intermediate maps. These agents' behaviours exploit implicit, top-down knowledge about the object being sought in an active way. With the proposed model, computationally demanding accurate trail detectors are able to focus their activity in a fraction of the input image, thus promoting robustness and real-time performance.**

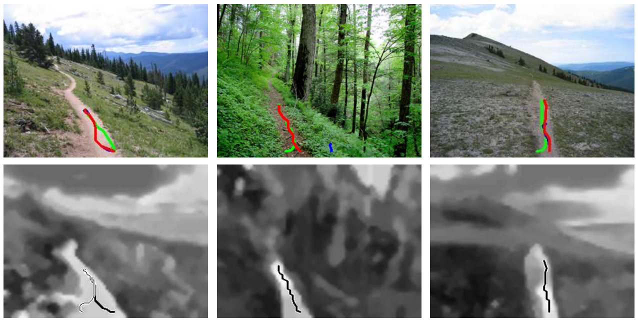

Fig. 1. Input images (above) and respective saliency maps (below), where saliency is represented in grey level. These maps are the superposition of two conspicuity maps, one for colour and the other for intensity channels. Each of these maps is searched for trails by agents (see Section II-B), whose paths are described by the overlaid lines. Thicker lines refer to the most probable trail candidate, which appears in the input image in red. The other coloured lines are from the best agent found on each conspicuity map.

I. INTRODUCTION

Exploiting any sort of structure in outdoor environments is essential for safe robot navigation. An example is the ability of detecting and following trails. Such behaviour reduces the chances of collision with obstacles, in addition to lower the cognitive load associated to path and trajectory planning. This is as true for humans as it is for robots. A practical application of robots with such a feature could be natural parks patrolling, in which robots would be engaged in actively maintaining and cataloguing the environment, and possibly providing support to human hikers.

Current on-board vision-based trail detection methods either assume that the robot is already on trail [1] or that strong edges segment it from the background [2]. Although trails are partially structured, in demanding environments it is difficult to develop assumptions on their appearance or geometry that would facilitate their detection. A stronger, and consequently more computationally demanding approach, is to learn trail models off-line in order to support the scoring of the trail hypotheses generated on-line [3]. In sum, no previous work proposed a solution able to detect trails in a parsimonious way, and at the same time without making hard assumptions.

A careful observation of natural images highlights the fact that trails are structures that easily pop-out. Bearing this some quantitative support, and visual saliency could then be applied to focus the attention of an accurate trail detector in an unbiased way. Experimental results herein presented support this assumption, and furthermore show that saliency information alone provides enough cues for a set of agents to localise trails with an ambiguity of up to three hypotheses, in the vast and diverse used data set. Notably, the robustness of the method is revealed with its ability to detect what we humans would select as the most navigable area, in images where trails are almost indistinguishable or not even present. See Fig. 1 for some representative examples.

To our knowledge, this is the first report on the use of saliency in this task. The method allows focusing resources without the cost and brittleness of obtaining prior knowledge on both appearance and morphology of the sought object. Moreover, since the use of saliency for other tasks is rather common in cognitively rich robots [4], [5], the overhead of saliency computation is diluted over all those modules.

This paper is organised as follows. Section II describes the proposed model. Experimental results with natural images are presented in Section III. Finally, some conclusions are drawn and future work is proposed in Section IV.

II. Proposed Model

This section starts by describing the way saliency is computed, followed by its application to trail detection.

A. Saliency Computation

Saliency computation is about determining which regions of the input image are more conspicuous, i.e. detach from the background, at several scales and feature channels. In

P. Santana is with LabMAg, Computer Science Department, University of Lisbon, Portugal, Pedro.Santana@di.fc.ul.pt. The author's work was supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

N. Alves is with UNINOVA, New University of Lisbon, Portugal and IntRoSys, S.A., nelson.alves@introsys.eu

L. Correia is with LabMAg, Computer Science Department, University of Lisbon, Portugal, Luis.Correia@di.fc.ul.pt

J. Barata is with UNINOVA, New University of Lisbon, Portugal, jab@uninova.pt

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

this paper only intensity and colour channels are used, and saliency follows the biologically inspired model proposed by Itti et al. [6], properly adapted to the task at hand.

Shortly, one dyadic Gaussian pyramid, with eight levels, is computed from the intensity channel. Two additional pyramids, also with eight levels, are computed to account for the Red-Green and Blue-Yellow double-opponency colour feature channels. The various scales are then used to perform centre-surround operations [6]. The resulting centre-surround maps have higher intensity on those pixels whose corresponding feature differs the most from their surroundings. An example is a bright patch on a dark background (on-off), as well as the other way around (off-on). On-off centre-surround operations are performed by across-scale point-by-point subtraction, between a level with a fine scale and a level with a coarser one. Off-on maps are computed the other way around, i.e. subtracting the coarser level from the finer one. Rather than considering the modulo of the difference, as in the original model [6], we consider both on-off and off-on centre-surround maps separately, which has been shown to yield better results [7], [8]. Then, the centre-surround maps are blended to produce two conspicuity maps, one aggregating colour and another aggregating intensity information. Finally, these two maps are blended in a saliency map [6]. Note that all maps are 8-bit grey level images.

When blending maps, the most discriminant ones, i.e. those that highlight a smaller number of objects, are typically promoted by recurring to a normalisation operator. In the original model [6], this is done by scaling a given map X according to the normalisation operator $N(.)$. This operator is defined by the square of the difference between map's global maximum, $M(X)$, and the average of all its other local maxima, $\bar{m}(X)$, i.e. $N(X) = X \cdot (M(X) - \bar{m}(X))^2$. A similar normalisation operator has been proposed by Frintrop et al. [7], [8]. In this case, the uniqueness operator, $W(X) = X/\sqrt{m(X)}$, scales the map X according to the number of its local maxima above a given threshold, $m(X)$. In this paper the threshold is set to its default value, i.e. 50% of the map's global maximum [8]. This method allows, among other things, to account for the proportion of objects competing for attention when determining their saliency.

Common to both methods is the use of local maxima information, which though appealing not always embodies the information intended to capture. Large homogeneous structures for instance, such as the sky, generally encompass only a few local maxima. In this situation, the sky would be undesirably considered highly conspicuous, despite its large foot-print in the whole image. A second aspect is that the two analysed saliency models consider that all pixels contribute equally to the saliency computation. However, excepting for extreme tilt/roll angles, the upper region of the image has little relevant information for trail detection. As a consequence, without a space-variant contribution to the final saliency map, feature maps that are only discriminative in the lower part of the image, and consequently interesting for trail detection, would not be adequately promoted.

In face of these limitations a new normalisation operator is herein proposed. Rather than considering only the map's local maxima when averaging, as it is done in $N(.)$, we propose to use all pixels. Furthermore, the contribution of each pixel to the average is weighted according to its distance from the top row. Formally, let $I(X, c, r)$ return the grey level of the pixel in column c and row r of a given map X, with height $h(X)$. Let $w(X, c, r) = \sqrt{r/h(X)}$ be the weight of pixel at position (c, r). The map's weighted average, m_w, is thus given by

$$m_w(X) = \frac{\sum_{(c,r)\in X} I(X,c,r) \cdot w(X,c,r)}{\sum_{(c,r)\in X} w(X,c,r)} \tag{1}$$

and similarly to operator $N(.)$, the proposed normalising operator, $K(.)$, takes the form $K(X) = X \cdot (M(X) - m_w(X))^2$. To reduce computational cost, the proposed system uses image operators over 8-bit images, whose magnitude is clamped to $[0, 255]$ by thresholding. In addition, prior to normalisation, maps are scaled to cover the interval $[0, 255]$, meaning that $M(X) = 255$ for all cases.

The Receiver Operating Characteristic (ROC) curves depicted in Fig. 2 show that, for the tested dataset (see Fig. 4), the proposed procedure produces consistently a better trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) than the other two methods. The small difference between the ROC curves could suggest that only a small quantitative improvement was obtained with the proposed model. However, the averaging procedure used to build the curves hides the fact that none of the other two methods was able to consistently allocate higher levels of saliency to trail regions than to the background, as often as the proposed one.

Fig. 2 also shows that saliency is considerably correlated with trail location, which is an important contribution by itself. This correlation can also be observed for typical images in Fig. 1. However, the correlation is still lower that the one required for accurate trail detection. That is, there is no single threshold on the saliency map that clearly segments the trail for all images in the dataset. It is thus important to

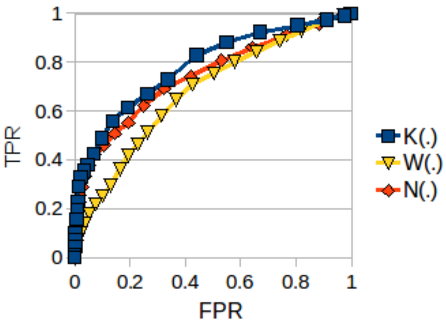

Fig. 2. Normalisation operators comparison. Each plot is the average ROC curve over all images in the dataset, for a given normalisation operator. ROC curves were built by thresholding the final saliency map and comparing the resulting binarised image against the hand-labelled ground-truth of the dataset. All operators result in curves above the line of no-discrimination, $y = x$, thus showing the positive correlation between visual saliency and trail presence. Moreover, the higher area under the curve for the proposed model, $K(.)$, demonstrates that it is the most adequate for the task at hand.

devise a mechanism able to overcome this limitation. As it will be shown in the next section, an agent-based design is the adequate tool for the purpose.

B. Trail Detection Agents

Rather than considering image analysis as information processing, this paper follows the idea of considering it as the result of a sensori-motor coordination process. Under this paradigm, the agent-based approach to image analysis, in particular for object recognition, has been showing promising results [9], [10], [11], [12]. This success story can be in part understood by the fact that agents realise active vision local loops, and thus they exploit all the known advantages of considering perception as an active process [13]. Being this work in line with this novel way of developing robust perceptual systems, its potential success contributes to the body of evidence on the relevance of an agent-based design for perceptual systems.

In a different context from the one considered in this article, i.e. road detection, the agent-based design has already been used successfully [14]. Despite the fact that we focus on trail detection instead of road detection, some additional differences between our work and the one of Broggi & Cattani [14] can be observed. As it will be described, in our case agents inhabit conspicuity and saliency maps, rather than the image space itself. We focus on the structure being sought, i.e. the trail, and not on its boundaries. In addition, we do not make the hard assumption that the robot is on the trail or road.

Our system is composed of a set of agents deployed in each map $m \in \{colour, intensity, saliency\}$, with width $w(m) = 320$ and height $h(m) = 240$. Each agent moves on one of these maps, according to a set of behaviours, in an attempt to follow a given trail hypothesis. Remember that the saliency map is the blend of both colour and intensity conspicuity maps, and maps' pixels are grey level coded. Hence, the brighter the pixel the more salient, in the case of $m \in \{saliency\}$, and the more conspicuous, in the case of $m \in \{colour, intensity\}$.

Agents are deployed with a small offset of the bottom of the map, i.e. at row $r = h(m) - 15$, to avoid any noise potentially present at the map's boundaries. To determine the column where each agent is deployed, the unidimensional vector $\mathbf{v^m} = (v_0^m, \ldots, v_{w(m)}^m)$ is first computed. The element v_k^m of $\mathbf{v^m}$, refers to the average grey level of the pixels in column k, contained between row r and row $r - \delta$, where $\delta = 10$ to avoid deploying agents in columns with spurious highly bright pixels. Formally, $v_k^m = \sum_{l \in [r, r-\delta]} I(m,k,l)/\delta$. Finally, the agent u is deployed in column $c(u) = \arg\max_k v_k^m$, thus compelling the agent to be initiated in the brightest, i.e. most interesting, region according to $\mathbf{v^m}$.

To analyse the second brightest region, an Inhibition-Of-Return (IOR) mechanism is used. This is implemented by zeroing the elements of $\mathbf{v^m}$ that are connected to $v_{c(u)}^m$ through elements with values similar to it. This agent deployment sequence is repeated until one of the following holds: (1) a maximum number of agents, z, has been deployed in the map or (2) the current highest value of $\mathbf{v^m}$, $\max(\mathbf{v^m})$, is below a fraction η of its initial value, i.e. before the first agent was deployed. In this paper $\eta = 0.7$, which avoids the deployment of agents in dark, i.e. uninteresting, regions. Note that although $\mathbf{v^m}$ varies, its time step index has been discarded for sake of clarity.

Let us now describe the behaviour of each deployed agent. For simplicity, agents and map indexes will be discarded in the remainder of this section. That is to say that the following applies to a single agent u allocated to a specific map m.

An agent action is defined in terms of an index, $O = \{1, 2, 3, 4, 5\}$, to the nearest neighbour pixels whereto the agent can move from its current position, $o[n]$, at iteration n (see Fig. 3-Left). To reduce both sensitivity to noise and computational cost, the agent's surroundings are segmented into regions $R_1 \ldots R_5$ (see Fig. 3-Left). The average grey level of a region containing pixel p is given by $A(p)$. For instance, both $A(1)$ and $A(6)$ correspond to the average grey level of the pixels contained within region R_1, as this region is composed of pixels 1, 6 and 11. Thus, regions are indirectly indexed by their encompassed pixels. The grey level of a pixel p is simply given by $I(p)$.

To account for top-down knowledge in the structure of the object being sought, a set of five perception-action rules $B = \{greedy, track, centre, ahead, commit\}$, i.e. behaviours, vote for each possible action, $a \in O$, according to the behaviour-based voting command fusion approach [15]. The most voted action, $a^+[n]$, is selected by the agent as the next motion, which is then used to update its position, $o[n]$,

$$\dot{o}[n] = \Gamma(a^+[n]), \;\; a^+[n] = \arg\max_{a \in O} \sum_{b \in B} w_b \cdot f_b(a,n) \quad (2)$$

where $\Gamma(.)$ transforms a motion index onto pixel coordinates, centred on the current agent's position, w_b is the weight accounting for the contribution of behaviour $b \in B$, described by the evaluation function $f_b(a,n)$ as follows,

$$f_{track}(a,n) = 1 - \frac{\left| A(a) - \sum_{q \in Q} \frac{q}{n-1} \right|}{255} \quad (3)$$

$$f_{ahead}(a,n) = 1 - \frac{|3-a|}{2} \quad (4)$$

$$f_{commit}(a,n) = 1 - \frac{|a^+[n-1] - a|}{4} \quad (5)$$

$$f_{centre}(a,n) = \left| d[n] \cdot \left(\frac{6 \cdot \mathcal{H}(-d_x[n]) - a}{5} \right) \right| \quad (6)$$

$$f_{greedy}(a,n) = \frac{A(a)}{255} \quad (7)$$

where $d[n]$ is computed as described in Fig. 3-Right and $\mathcal{H}(.)$ is the Heaviside function. Q is a set whose elements are scalars with the grey level of the pixels crossed by the agent along its path. Refer to Table I for further details on each behaviour and Fig. 1 for examples of agent typical motions.

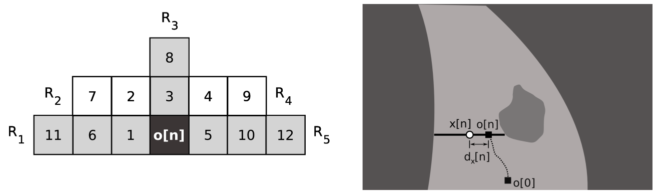

Fig. 3. Trail detection agents. Left: numbers correspond to the pixels' index relative to the current position of the agent, $o[n]$. Regions surrounding $o[n]$ are segmented in $R_1 = \{1, 6, 11\}, R_2 = \{2, 7\}, R_3 = \{3, 8\}, R_4 = \{4, 9\}, R_5 = \{5, 10, 12\}$. Right: illustrating example of key aspects of the *centre* behaviour. The dotted line represents the agent motion since its onset until the current iteration. The pixels composing the thicker horizontal line define the set $S[n]$. The agent will try to approach this line's centroid $x[n]$, represented by the circle, which is deviated from the current agent's position, $o[n]$, by $|d_x[n]|$ pixels.

The best performance has been empirically obtained with the following trade-off, $w_{greedy} = 0.45$, $w_{track} = 0.35$, $w_{centre} = 0.10$, $w_{ahead} = 0.05$, $w_{commit} = 0.05$.

The agent is allowed to move until one of the following stopping conditions is met: (1) a maximum number of α_1 iterations is performed; (2) the agent reaches row α_2 (row zero at image's top); (3) the average grey level of regions $R_1 \ldots R_5$ is below a given proportion $\beta < 1$ of the average grey level of the pixels visited by the agent.

$$\beta \cdot \sum_{q \in Q} \frac{q}{n-1} > \sum_{j=1}^{5} \frac{A(j)}{5} \tag{8}$$

where, $\alpha_1 = 50$, $\alpha_2 = 160$, and $\beta = 0.7$ are empirically defined scalars.

The set of agents deployed in a given map must be ranked to select the one that better represents the trail. Consequently, as soon as one of the previously mentioned stopping conditions is met, the score of the agent is computed,

$$\begin{aligned} s = & \sum_{i=0}^{n} \frac{\mu_1 D'[i] - \mu_2 D''[i]}{n} + \\ & \sum_{q \in Q} \frac{\mu_3 q}{n-1} + \mu_4 d(o[n], o[0]) \end{aligned} \tag{9}$$

where $\mu_1 = 0.01$, $\mu_2 = 0.01$, $\mu_3 = 0.5$, $\mu_4 = 0.5$ are empirically defined scalars and $d(o[n], o[0])$ is the Euclidean distance between the two points. The first term of the score function accumulates the first, D', and second, D'', derivatives of D along the agent's path. This term favours paths where D progressively shrinks towards a vanishing point. The second term promotes agents whose path contains highly bright pixels. Finally, the third term disfavours short paths.

III. Experimental Results

This section presents a set of experimental results obtained with a dataset composed of 50 colour images, with resolution 640×480, obtained from Google (see Fig. 4). The dataset only encompasses images obtained with cameras roughly located at the eyes height, and thus providing a vantage point that would be plausible for a medium-size robot. The trail detector has been implemented without thorough code optimisation, and tested in a Centrino Dual Core 2 GHz, running Linux, and OpenCV for computer vision low-level routines.

Since the output generated by the trail detector is the set of the agents' paths, and not the trail's outline, it is difficult to find a way of comparing the results against some sort of ground truth. The following describes the assumptions taken to assess whether a given agent has been able to represent the trail. Trails are considered correctly detected if the agent is deployed inside the trail and finishes its run also inside, or very close to, the trail. In addition, curves and zigzags described by the agent are considered valid as long as they also stay inside the trail, or very close to its borders.

Table II summarises the results obtained as a function of the maximum number of allowed agents per map $z \in \{1, 3, 5, 7\}$. In a first analysis, success rate is calculated per map. This allows to determine the proneness of each map alone, to provide enough cues for its highest score agent to properly represent the trail. In a second combined analysis, success is obtained when at least one of the three map's best agent, succeeds. In this case, the ambiguity regarding both trail's position and approximate skeleton is of up to three hypotheses, i.e. one per map, in 98% of the tested images. This clearly shows that the proposed method accurately focus agents on the most promising regions of the image.

The obtained results also confirm the positive correlation between saliency and the presence of trails (see Fig. 2). Would this correlation be nonexistent and the trail detection results would be linearly affected by the number of agents. Instead, with a single agent per map, the trails in 90% of the images were properly detected, whereas an increment of only 6% is observed if two additional agents per map are

Behaviour	Voting Preferences
greedy	Regions of highest grey level, under the assumption that trails are salient structures in the input image
track	Regions whose average grey level is more similar to the average grey level of the pixels visited by the agent, under the assumption that trails' appearance is somewhat homogeneous
centre	Regions closer to the centroid, $x[n]$, of the set of pixels, $S[n]$, that: (1) share the row with $o[n]$; (2) display grey levels similar (i.e. within a given margin γ) to the one of $o[n]$; and (3) are connected to $o[n]$ through a set of pixels complying with the first two conditions. The goal is to maintain the agent equidistant to the trail's boundaries, where the deviation to the centroid is given by $d_x[n] = \frac{c(o[n]) - c(x[n])}{D[n]}$ with $D[n] = \lvert S[n] \rvert$. Remember that $c(p)$ returns the column of pixel p. See Fig. 3-Right for an illustration of this process.
ahead	Upwards regions under the assumption that trails appear as vertical elongated structures
commit	Previously selected region, to reduce sensibility to local noise, under the assumption that trails' outline is somewhat monotonous.

TABLE I

Behaviours of trail detection

deployed. An even smaller differential is obtained when we go from three to five agents, namely 2%. Adding more agents reflects in a null gain.

Nr. of Agents	Colour Map	Intensity Map	Saliency Map	Combined
$z = 1$	44 %	64 %	74 %	90 %
$z = 3$	54 %	78 %	82 %	96 %
$z = 5$	58 %	80 %	82 %	98 %
$z = 7$	58 %	80 %	82 %	98 %

TABLE II

TRAIL DETECTION RESULTS

Fig. 4 shows the trail of the best agent per map in the images composing the dataset. These images are very diverse and in some cases no trail can be found altogether, not even by the human eye. The system still produces a correct answer, that is, selects the open region through which the robot would be able to traverse. This is a sign of generalisation capability, which was only possible due to the use of a non-specific detector, as it is the case of saliency.

Hence, even in the most difficult situations, saliency and conspicuity maps were able to maintain a globally coherent description of the environment. However, the existence of local appearance variations requires the system to have a considerable level of robustness in order to be unaffected by those local artifacts. The agent-based approach showed to be that robust, mostly due to the fusion of several behaviours. Moreover, being a purely bottom-up and feed-forward approach, the method is exceptionally fast, taking an average of 1 ms per map. This includes finding all the potential trails, finding their length, and choosing the correct one. An additional cost must be considered, which refers to the computation of the three maps, which takes on average 30 ms. These maps have two remarkable embedded properties: (1) they segment the input image in a very efficient way, and (2) they naturally prioritise the segments according to their conspicuity.

The resulting computation time of the trail detector, i.e. 33 ms, is roughly 30 times lower than the time reported by the successful vision-based trail detector proposed by Rasmussen & Scott [3].

IV. Conclusions

This paper reported for the first time the use of visual saliency in the trail detection problem. Experimental results showed that up to three trail hypotheses are generated by the method, being at least one of them correct in 98% of the cases. Seeing trails as conspicuous parts of the scene allowed the system to generalise. That is, in situations where trails could be hardly identified, even by the human eye, the system reported as trail open regions of the environment, where the robot could easily travel. The proposed normalisation operator for saliency computation played an important role in this achievement. Much of the robustness of the method is due to the use of an agent-based solution. Thus, this paper contributes to the growing evidence of such approach for the development of robust perceptual systems.

The proposed model is innovative on the way top-down knowledge of the object being sought is considered. Typically, visual features are boosted according to the expected object's scale, colour and intensity [16]. In this paper instead, the object's (trail) approximate shape is implicitly considered, by means of feed-forward and, consequently fast, perception-action rules dictating the behaviour of each agent.

We are currently implementing the trail detector under a distributed framework, in line with previous work on object recognition [10], envisaging higher levels of robustness. Despite the system's exhibited robustness, its self-parametrisation will be the focus of future work.

Acknowledgments

This work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

References

[1] C. Rasmussen and D. Scott, "Terrain-based sensor selection for autonomous trail following," in *Proceedings of the 2nd International Workshop on Robot Vision (Robvis 2008)*, 2008, pp. 341–355.

[2] A. Bartel, F. Meyer, C. Sinke, T. Wiemann, A. Nchter, K. Lingemann, and J. Hertzberg, "Real-time outdoor trail detection on a mobile robot," in *Proceedings of the 13th IASTED International Conference on Robotics, Applications and Telematics*, 2007, pp. 477–482.

[3] C. Rasmussen and D. Scott, "Shape-guided superpixel grouping for trail detection and tracking," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008, pp. 4092–4097.

[4] J. Ruesch, M. Lopes, A. Bernardino, J. Hornstein, J. Santos-Victor, and R. Pfeifer, "Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008, pp. 962–967.

[5] J. Moren, A. Ude, A. Koene, and G. Cheng, "Biologically based top-down attention modulation for humanoid interactions," *International Journal of Humanoid Robotics*, vol. 5, no. 1, pp. 3–24, 2008.

[6] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1254–1259, 1998.

[7] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," *Lecture Notes In Computer Science*, vol. LNCS 3663, p. 117, 2005.

[8] S. Frintrop, "Vocus: a visual attention system for object detection and goal-directed search," Ph.D. dissertation, INAI, Vol. 3899, Germany, 2006.

[9] D. Floreano, K. Toshifumi, D. Marocco, and E. Sauser, "Coevolution of active vision and feature selection," *Biological Cybernetics*, vol. 90, no. 3, pp. 218–228, 2004.

[10] Y. Owechko and S. Medasani, "A swarm-based volition/attention framework for object recognition," in *Proc. of the IEEE Computer Vision and Pattern Recognition Workshop (CVPRW)*, vol. 3, 2005.

[11] G. d. Croon and E. O. Postma, "Sensory-motor coordination in object detection," in *Proc. of the IEEE Symp. on Artificial Life (CI-ALIFE 2007)*, 2007, pp. 147–154.

[12] Y. Choe, H. F. Yang, and N. Misra, "Motor system's role in grounding, receptive field development, and shape recognition," in *Proc. of the 7th Intl. Conf. on Development and Learning (ICDL)*, 2008.

[13] D. H. Ballard, "Animate vision," *Artificial Intelligence*, no. 48, pp. 57–86, 1991.

[14] A. Broggi and S. Cattani, "An agent based evolutionary approach to path detection for off-road vehicle guidance," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1164–1173, 2006.

[15] J. K. Rosenblatt, "DAMN: a distributed architecture for mobile navigation," in *Proc. of the AAAI Spring Symp. on Lessons Learned from Implemented Software Architectures for Physical Agents*, Stanford, CA, 1995.

[16] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

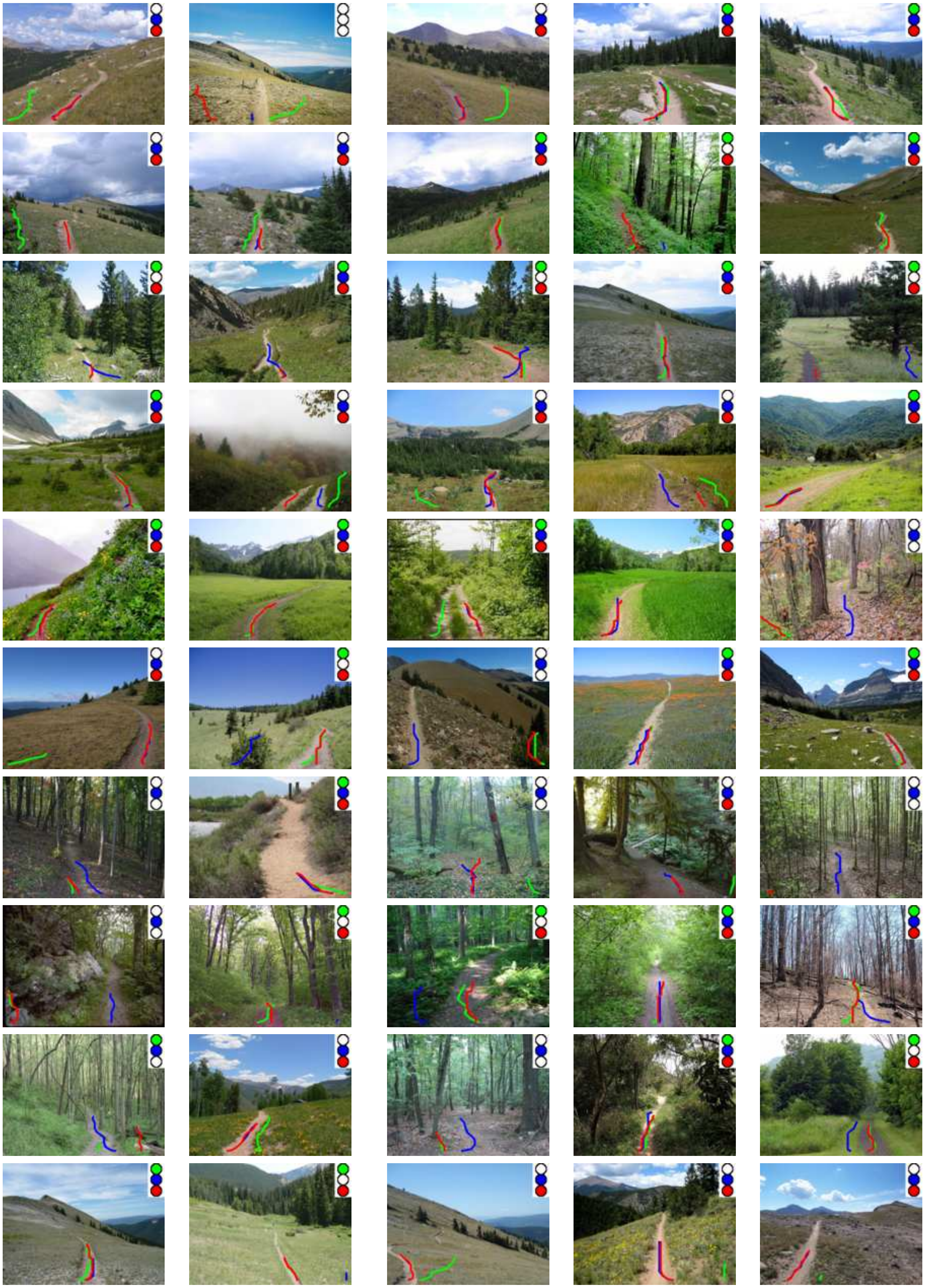

Fig. 4. Trail detection results. The best agent's path in each map is superposed on the corresponding input image. Path colour is green, blue and red for the colour, intensity and saliency maps, respectively. In the top-right corner of each image, the presence of a filled circle with a given maps' colour, indicates that the best agent's path of the corresponding map correctly represents the trail.