A Rover-based System for Searching Encrypted Targets in Unknown Environments

Danilo Avola[1], Luigi Cinque[2], Gian Luca Foresti[1], Marco Raoul Marini[2] and Daniele Pannone[2]
[1]*Department of Mathematics, Computer Science and Physics, University of Udine, Via delle Scienze 206, Udine, Italy*
[2]*Department of Computer Science, Sapienza University, Via Salaria 113, Rome, Italy*

Keywords: Visual Cryptography, Target Recognition, Rover, Client-Server System.

Abstract: In the last decade, there has been a widespread use of autonomous robots in several application fields, such as border controls, precision agriculture, and military operations. Usually, in the latter, there is the need to encrypt the acquired data, or to mark as relevant some positions or areas. In this paper, we present a client-server rover-based system able to search encrypted targets within an unknown environment. The system uses a rover to explore an unknown environment through a Simultaneous Localization And Mapping (SLAM) algorithm and acquires the scene with a standard RGB camera. Then, by using visual cryptography, it is possible to encrypt the acquired RGB data and to send it to a server, which decrypts the data and checks if it contains a target object. The experiments performed on several objects show the effectiveness of the proposed system.

1 INTRODUCTION

In recent years, autonomous robots such as rovers, Autonomous Underwater Vehicles (AUVs), and Unmanned Aerial Vehicles (UAVs) are used in a wide range of application fields, including Search and Rescue (SAR) operations (Cacace et al., 2016; Kiyani and Khan, 2016), environment monitoring (Avola et al., 2017b; Avola et al., 2017a), and military operations (Kaur and Kumar, 2015). In these missions, these robots can be used to substitute the human operators. For example, rovers can climb impervious grounds and grab potentially unreachable objects. Drones can provide strategical images from precise altitudes and angles of view. The underwater cave exploration case is even more explicative: usually there is no oxygen, light, and space for moving. These examples show the importance of using robots in specific conditions. Moreover, different sensors can be attached to these devices, thus allowing to collect sound, temperature, and other types of data. Due to this fact, sensible information can be obtained and it could be necessary to hide them for security or privacy reasons. For example, in military operations it is critical to protect data from enemies, civilians, or other potential intruders. Usually, in these operations, the most used type of data are images, which are dense source of information and difficult to crypt. The focus of this work is on this specific topic.

In this paper, a client-server rover-based system able to search an encrypted target in an unknown environment is presented. The system works in the following way. Firstly, by using visual cryptography technique two shares from a target image are generated and stored in the server. Then, the rover starts the environment exploration through a SLAM algorithm, and acquires the scene through an RGB camera. To hide the frames sent by the rover to the server, the frames are encrypted, again, by using visual cryptography. Then, on the server side, the shares received are used in conjunction with the shares extracted from the target image such that the decryption can be performed.

The paper is structured as follows. In Section 2, current state-of-the-art works regarding SLAM and visual cryptography are discussed. In Section 3, the system architecture and the proposed visual cryptography pipeline are shown. Section 4 presents the experiments performed and the results obtained. Finally, Section 5 concludes the paper.

2 RELATED WORK

Visual Cryptography is a technique used to encrypt an image by splitting it into *n* images called *shares*. These latter do not allow to distinguish any informa-

Avola, D., Cinque, L., Foresti, G., Marini, M. and Pannone, D.
A Rover-based System for Searching Encrypted Targets in Unknown Environments.
DOI: 10.5220/0006723402540261
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 254-261
ISBN: 978-989-758-276-9

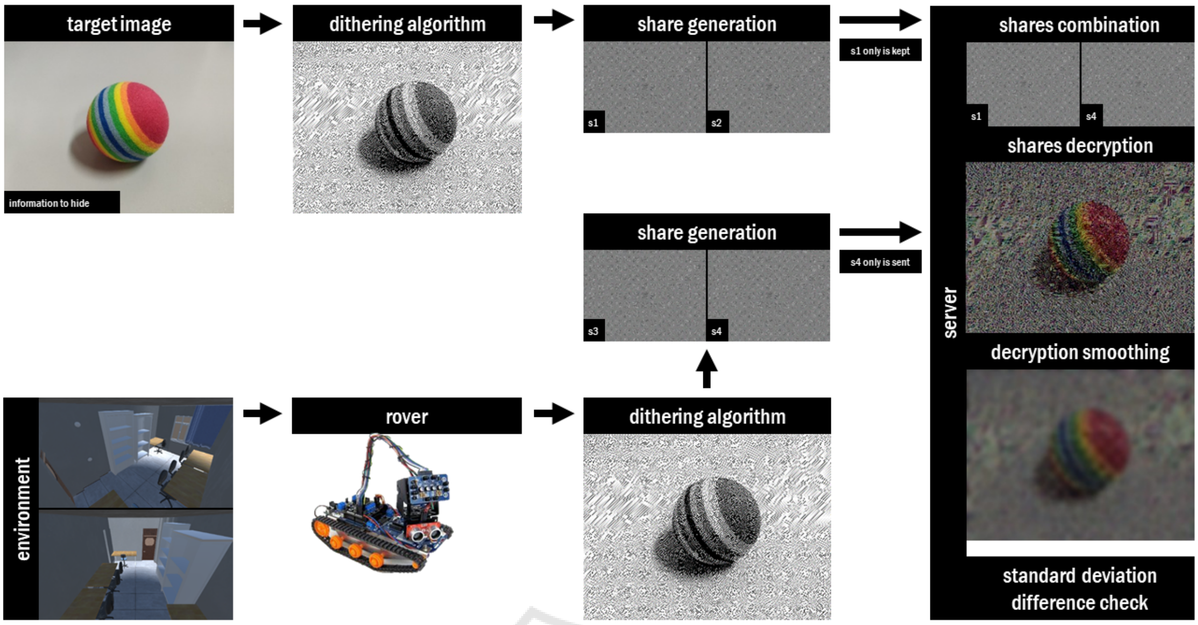

Figure 1: Architecture of the proposed system.

tion about the original image unless they are combined together. This means that if only one share is available, the source data is inaccessible. This cryptography approach has been introduced by Naor and Shamir (Naor and Shamir, 1995). Their method has never been heavily modified and still no one has proposed a completely different visual cryptography approach, but only variants and improvements (Liu et al., 2014). Some developments for grayscale (Lin and Tsai, 2003; Babu et al., 2013) and RGB images (Hou, 2003; Liu et al., 2008) have been proposed for providing an easier integration of visual cryptography in real applicative contexts. Moreover, also combinations of them have been considered for improving performances (Stinson, 1997; Shyu, 2006) in cyphering. Since in visual cryptography the image resulting from the algorithm is decrypted by the human eye, some techniques to enhance the quality of the decrypted image have been proposed. In particular, the most used technique is the dithering (Alex and Anbarasi, 2011; Pahuja and Kasana, 2017), which allows to create halftone images. In addition, visual cryptography is also used as enhancement of traditional protection schemes (Yang et al., 2017; Kadhim and Mohamed, 2016; Joseph and R, 2015).

Concerning the environment exploration, the current state-of-the-art is based on the SLAM (Leonard and Durrant-Whyte, 1991; Sim and Roy, 2005; Trivun et al., 2015) approach. The aim of the SLAM is to maximize the area coverage during the environment exploration and, at the same time, to make the robot conscious of its absolute position within it. The SLAM approach can be used with several sensors, such as depth/time of flight cameras (Li et al., 2016; Walas et al., 2016), thermal cameras (Chen et al., 2017), or a fusion of them (Mur-Artal and Tards, 2017; Camurri et al., 2015). In addition, it is used for different tasks, such as mosaicking generation (Bu et al., 2016), pipe rehabilitation (Kim et al., 2015), and environment mapping (Balclar et al., 2017).

3 SYSTEM ARCHITECTURE AND PROPOSED PIPELINE

In this section, both the system architecture and the proposed method are described. In particular, the client-server approach is explained and the visual cryptography pipeline is described.

3.1 System Architecture

In Figure 1, the architecture of the proposed system is shown. The system is composed by a client side, i.e., the rover, and the server side, i.e., a computer. The proposed system has been developed starting from the one presented in (Avola et al., 2016) by implementing new features.

In the server side, the first step consists in storing the information regarding the target image. By using the visual cryptography, two shares, i.e., S_1 and S_2, are extracted from the target image *I*. Using a public key cryptography approach we store only one share in the server, i.e., S_1, considering it as a private key, while S_2 is discarded.

In the client side, the SLAM algorithm is executed, through which the rover explores the environment in search of the target image used for generating the original shares. During the exploration, the rover acquires the scene with a standard RGB camera. On the images acquired with the latter, the visual cryptography algorithm is applied to the frames in order to generate two shares, i.e., S_3 and S_4. Of these two shares, only S_4 is sent to the server and it is used as a public key, so in conjunction with S_1 it is possible to decrypt the image.

The advantage of using shares instead of clear images is that if an intruder performs an attack, both physical (e.g., steals the rover) or digital (e.g., sniffing of the video stream) the attacker cannot retrieve to the original information (i.e., the target image). Moreover, with this approach it is possible to use objects of the environment as markers, thus allowing to distinguish, for example, an area or an object of interest. This is possible since the target image will be decrypted only with the correct shares, so if an image is correctly decrypted we known that we are facing the object of interest or we are in the correct position within the environment.

3.2 Proposed System Pipeline

In this section, the proposed method for generating shares and reconstructing the original information is explained. As in the pipeline described in (Hou, 2003), the first step in creating the shares consists in applying the dithering algorithm to the target image *I*. The dither is a form of noise intentionally applied in order to reduce the quantization error. With the dithering step, the image is converted into an approximate binary image so that the encryption and decryption processes are easier. There are several dithering algorithms:

- Average dithering (Boiangiu et al., 2012) is one of the simplest techniques. It consists in calculating the middle tone of each area and assigning this value to that portion of image;
- Floyd-Steinberg (Knuth, 1987) is still the most used, it consists in diffusing the quantization error to the near pixels of each pixel of the image;
- Average ordered dithering (Bayer, 1973) is similar to average dithering but generates cross-hatch patterns;

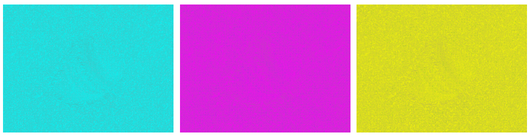

Figure 2: Example of halftones obtained by applying dithering algorithm on the three image channels.

- Halftone dithering (Knuth, 1987) looks similar to newspaper halftone prints and produces clusters of pixel areas;
- Jarvis dithering (Jarvis et al., 1976) is similar to Floyd-Steinberg but distributes the quantization error farther than it, increasing computational cost and time.

Due to its easiness of implementation and its good quality results, the used dithering algorithm is the Floyd-Steinberg. As mentioned, the algorithm diffuses the quantization error of a pixel to the neighbour pixels, and it is done in the following way:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & p & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{bmatrix} \tag{1}$$

where the pixel p is the current pixel scanned during the execution of the algorithm. Since *I* is scanned from left to right and from top to bottom, the pixels are quantized only once. Considering that we are using color images, the dithering algorithm is applied on the three channels of the image, obtaining three halftone images. In Figure 2, the result of this step is shown.

The second step is the generation of the shares from the halftone image. As starting point, we show how shares are generated from a grayscale image since the same algorithm will be applied on the color image. The algorithm is the following:

1. Image I is tranformed into a black and white halftone image H.
2. For each pixel in the halftone image, a random combination is chosen among those depicted in Figure 3.
3. Repeat step 2 until every pixel in H is decomposed.

The pseudocode is reported in Algorithm 1. For generating the shares from a color image, we have applied the third method proposed in (Hou, 2003). This method has been chosen since despite a color image is used, it requires only two shares to reconstruct the original image, and it does not sacrifice too much image contrast for color visual cryptography. The method works in the following way. The image *I* is transformed in three halftone images, one for

Algorithm 1: Algorithm generating shares from a grayscale image.

```
procedure SharesFromGray(grayImage)
    Transform grayImage to halftone image H
    for each pixel in H do
        Choose randomly a share among those in
            Figure 3
    end for
end procedure
```

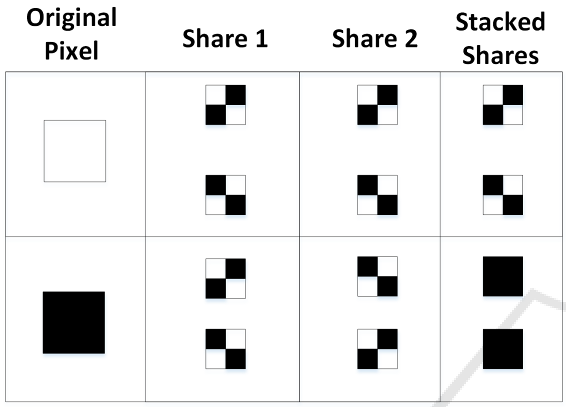

Figure 3: Sharing and stacking combination in grayscale images visual cryptography.

Cyan (C), one for *Magenta* (M) and one for *Yellow* (Y). Subsequently, for each halftone image the Algorithm 1 is used to generate six 2×2 sharing images, called *C1, C2, M1, M2, Y1* and *Y2*. Each of these images have two white pixels and two color pixels. Then, *C1, M1* and *Y1* are combined to generate the colored Share 1, while *C2, M2* and *Y2* are combined to generate the colored Share 2. The color intensity of the share blocks is a value between 0 and 1, where 0 means the absence of that color and 1 means full intensity. So, for a pixel $p_{i,j}$ the color intensity for each channel is defined as (I_C, I_M, I_Y). For each block generated with this method, we have that the color intensity is $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, while after stacking Shares 1 and Shares 2 the range of color intensity is between $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and $(1,1,1)$. The decryption step simply consists in overlapping the two shares, obtaining the decrypted image I_{dec}. In Figure 4 a representation of the algorithm is depicted, while in Algorithm 2 the pseudocode of the method is shown.

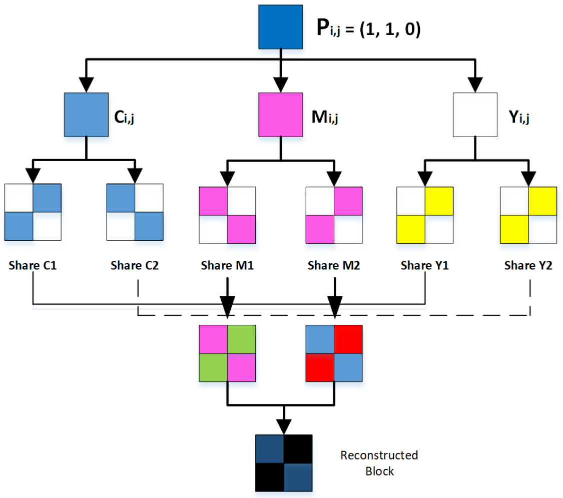

Figure 4: Decomposition and reconstruction of color pixel.

Algorithm 2: Algorithm generating shares from a color image.

```
procedure SharesFromColor(colorImage)
    Transform colorImage in three halftone images C, M and Y
    for each pixel $p_{i,j}$ in C,M and Y do
        According to Algorithm 1, create
            $C1_{i,j}, C2_{i,j}, M1_{i,j}, M2_{i,j}, Y1_{i,j}, Y2_{i,j}$
        Combine $C1_{i,j}$, $M1_{i,j}$ and $Y1_{i,j}$ for the
            corresponding block of Share 1
        Combine $C2_{i,j}$, $M2_{i,j}$ and $Y2_{i,j}$ for the
            corresponding block of Share 2
    end for
    After stacking the two Shares,
    the original image can be decrypted.
end procedure
```

Since the image acquired by rover could not be acquired at the same distance, position and angulation of the original target image I, and the pixels composing the two shares must be almost perfectly aligned in order to perform the decryption, a morphing procedure is applied on S_4. This allows to optimize, in some cases, the alignment of the two shares. Since with the shares a feature-based (e.g., by using keypoints and homography) alignment cannot be performed due to their random pixel arrangement, we have defined 8 standard transformations to apply. In Figure 5, these transformations are shown.

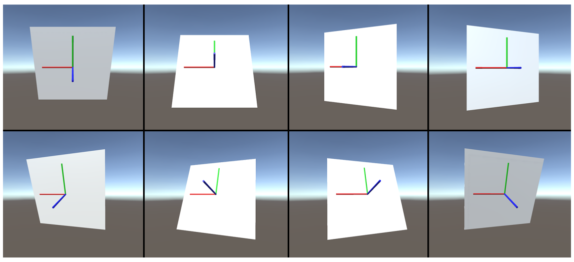

Figure 5: Transformations applied to S_4 for optimizing the shares overlap.

After the generation of I_{dec}, it must be checked that it is a valid decrypted image. In order to do so,

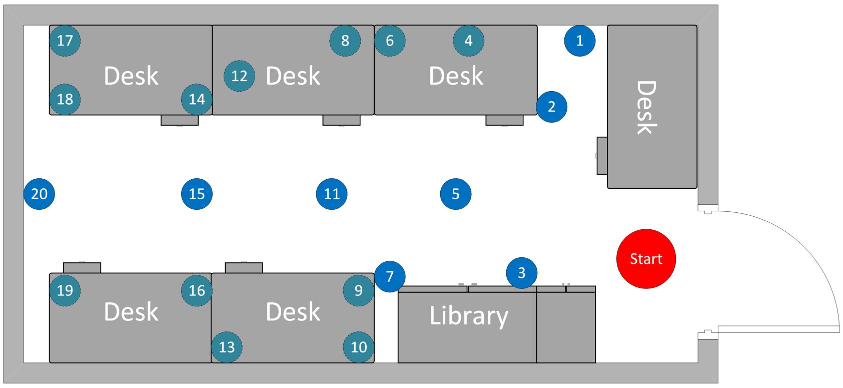

Figure 6: The testing environment. The red circle is the rover starting point, the dark blue circles are uncovered the target objects, while the dashed light green circles are the target underneath the desks.

we computed the difference between the standard deviation of I_{dec} and its smoothed copy. The smoothing operation has been performed by using a median filter with kernel size of 3×3. If I_{dec} is a valid decrypted image, we have that the standard deviation between it and its smoothed copy is low (e.g., less than 10), otherwise higher values are obtained.

The generation of the shares is performed once for I and once per second on the video acquired by the rover. The last assumption is due two main factors: a) the dithering operation cannot be performed in real time, and b) the used rover has not enough computational power to perform the dithering at higher rate.

4 EXPERIMENTS AND DISCUSSION

In this section, the performed experiments are discussed. In detail, we report the correctness in decrypting shares and the time needed to find the target objects within the testing environment. Since there is no dataset for visual cryptography, all the experiments have been performed with our acquisitions. Moreover, the experiments have been performed in controlled conditions, without changes of light (which can affect the share generation) and moving objects (which can affect the performances of SLAM algorithm). In Figure 6, the environment used for the experiments is depicted, which has dimensions of about 2.5×6 meters. In the performed tests, the rover starts the recognition always from the same starting point. To test the reliability of the system, we used both clear (i.e., just placed on the ground) and covered (i.e., underneath the desks) objects. A total of 20 objects have been used, of which 8 uncovered and 12 covered. In Table 1, the objects used in the experiments are reported. To test exhaustively the system, objects with high variability of colours and sizes have been chosen.

Table 1: List of objects used during the experiments.

Object Number	Object Type
Object 1	Ball
Object 2	Toy Gun
Object 3	USB Keyboard
Object 4	Pen
Object 5	Calculator
Object 6	Pencil
Object 7	Paperweight
Object 8	Credit card
Object 9	Sponge
Object 10	USB mouse
Object 11	Cup
Object 12	Coffe Machine
Object 13	Wallet
Object 14	Plastic Bottle
Object 15	Monitor
Object 16	Toy Robot
Object 17	DVD Case
Object 18	Small Box
Object 19	Big Box
Object 20	Book

In Figure 7, the rover used for the experiments is shown. The rover is composed by an Arduino UNO microcontroller, which handles both the servomotors and the ultrasonic sensor used by the SLAM algorithm, and by a Raspberry Pi 2 model B, which handles the camera and the video stream. Despite the used rover is able to perform all the tasks required (i.e., SLAM algorithm and sending the video stream to the server), its computational power affects the time needed to explore the environment. The first experiments have been done by testing the entire system on the rover. After capturing the frame, the Raspberry

Table 2: Standard deviations confusion matrix of the shares experiments.

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7	Object 8	Object 9	Object 10	Object 11	Object 12	Object 13	Object 14	Object 15	Object 16	Object 17	Object 18	Object 19	Object 20
Object 1	4.9792	24.2623	21.1189	21.6967	24.7812	19.2538	24.8378	20.1352	23.0027	22.5186	23.0507	21.1661	22.7217	23.8669	19.1155	19.5032	24.8488	22.9081	20.3874	21.4209
Object 2	19.7321	4.5369	20.5471	20.9900	19.9134	21.0880	19.7300	24.3049	19.5657	24.5802	21.3941	19.2844	21.0542	23.4158	23.7681	22.2694	23.1173	24.3618	19.3288	20.8220
Object 3	19.2771	20.1729	5.4403	23.3305	24.2668	22.4946	19.4241	24.5365	23.8022	20.7157	22.2620	24.9087	23.2941	24.0338	21.5996	21.8237	22.3643	20.6145	23.4941	22.0233
Object 4	22.8809	20.8465	19.8323	4.9511	21.1748	23.7287	23.6818	23.0111	19.8010	19.1293	22.3590	20.8049	24.6365	24.8854	20.7197	23.8049	24.3767	22.5852	24.3041	24.6624
Object 5	22.2949	23.3703	22.4605	19.1551	4.8931	22.8778	22.1272	21.2339	24.6228	23.9772	24.0945	21.2352	22.5591	24.2353	24.6010	23.0108	20.2407	22.9231	19.4323	21.4404
Object 6	23.0016	24.6024	23.8657	21.9073	23.5405	4.8341	24.8307	24.9278	24.1849	21.3333	21.7285	20.4801	23.7065	24.2970	24.4823	22.3497	22.5932	19.8933	24.3983	21.7024
Object 7	20.2340	24.3979	23.5755	24.2949	20.7097	23.0394	5.3286	19.7369	21.4439	20.6517	23.3000	20.7003	24.3772	23.9595	21.3402	21.9874	23.1688	24.0062	22.6578	22.4484
Object 8	20.9563	21.7385	23.2828	24.3064	23.3251	19.1117	23.0487	18.7589	21.6269	19.7022	23.8881	20.9491	20.4774	21.0563	21.2542	22.2793	22.3715	21.3749	21.3888	22.0922
Object 9	22.9452	24.7055	23.3341	21.4005	23.9912	19.8060	19.3628	19.5055	19.2645	20.9453	20.8104	19.0701	22.2394	19.5722	19.8791	22.7868	24.1559	24.8453	22.4250	24.9811
Object 10	22.3212	22.0928	20.9841	21.5800	21.9508	19.4262	24.3264	19.3878	21.6171	5.6533	21.3672	22.6808	23.9118	24.3174	24.5867	20.1447	20.5515	24.3872	22.5602	22.0230
Object 11	22.6769	23.9165	22.1913	20.2125	21.7234	21.5675	24.7963	22.7203	23.1723	23.3210	4.6938	22.1019	22.3402	19.9390	22.3723	23.1688	21.5587	24.0176	23.3883	21.1602
Object 12	21.7253	21.3183	23.6533	23.4056	21.5817	23.1625	24.6713	23.7054	23.2334	19.6560	21.3396	5.1818	21.7563	19.3020	20.3721	24.0051	19.0939	24.1823	19.4684	23.0143
Object 13	22.0013	20.3080	22.4297	19.7331	23.0270	22.5975	19.3359	19.3381	19.9150	19.1177	21.6111	23.9933	5.2348	22.1208	24.1832	19.5862	24.4483	19.6481	22.1020	19.8589
Object 14	22.3562	19.0275	23.6001	24.0923	24.5009	24.9218	22.0308	20.6285	19.6045	22.0471	22.5137	23.5773	19.4978	5.3232	22.1019	20.0263	24.6313	22.5429	21.6438	24.6515
Object 15	22.9355	21.7117	24.0382	22.1957	22.3233	23.0804	21.2031	20.4357	22.4735	24.2013	21.4407	19.6757	21.6631	20.8011	4.8028	24.0002	21.4218	21.3411	21.1627	19.8415
Object 16	20.5608	19.5209	21.5764	20.5437	20.7853	21.5492	19.7152	21.9704	23.2384	20.4614	23.7104	19.4445	21.3633	19.0204	20.3241	4.0026	20.1351	19.8549	20.6085	20.0494
Object 171	19.8319	22.5933	24.4063	24.6363	20.3271	21.8960	21.2561	22.1427	20.5892	19.4101	21.6180	20.0431	19.1566	24.7281	21.5836	24.7694	5.5248	19.0441	23.0802	23.2357
Object 18	22.8708	22.3139	20.3087	23.6342	20.3682	21.2252	24.3456	24.1383	21.4146	20.9081	22.6518	24.4612	24.4546	22.5496	20.9954	24.1184	21.6544	5.8087	19.1991	22.1946
Object 19	23.2990	20.0758	21.0192	20.1263	20.9316	21.4231	22.2914	19.2924	22.3164	20.6489	20.4490	20.4589	19.9250	24.7385	24.6140	23.9123	23.3696	20.0549	4.7207	20.1327
Object 20	19.0072	20.8985	23.1977	22.7515	22.2584	21.6342	20.7246	22.0100	23.5693	23.5744	22.4563	23.4860	22.8732	19.7393	22.0264	21.0836	19.5529	19.8871	20.1890	19.8832

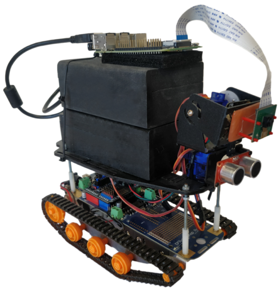

Figure 7: The rover used for the experiments.

Pi 2 proceeds with the pipeline described in Section 3. Due to the low power of the embedded processor, it takes about 20 seconds for each frame for completing the entire pipeline. To overcome this problem, the client-server approach has been adopted for improving the performance. In fact, the embedded system has only to compute the share of the frame and to send it with POST request over Wi-Fi to a directly-connected (ad-hoc network) machine, the server. The rest of computation is given to this latter. The total time has been reduced to around 5 seconds per frame. Concerning the accuracy of shares decryption, Table 2 shows the confusion matrix of the standard deviations obtained from the experiments. In our experiments, we have found that a correct decrypted image has a standard deviation value between 4 and 6, while a wrong decrypted image has a standard deviation value between 19 and 25. It is possible to observe that the decryption step works generally well, but due to their characteristics the decryption fails for the credit card (Figure 8(a)), the sponge (Figure 8(b)) and the cup (Figure 8(c)). In detail, the decryption fails because even if we apply the transformation depicted in Figure 5, it may be not sufficient to correctly align the shares.

Regarding the execution time, the rover took about 30 to 45 minutes to explore the entire environment. This is due to the random approach of the used SLAM algorithm, and also due to the fact that for this kind rover the exploration under the desks may be tricky.

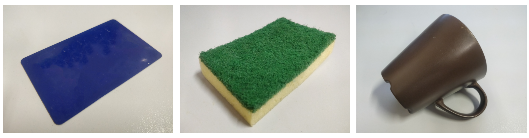

Figure 8: Object with which the decryption process fails: a) a credit card, b) a sponge and c) a cup.

Since at the current state-of-the-art there are no works exploiting visual cryptography and SLAM algorithms to search targets in unknown environments, the only comparisons that can be made with those works are with the single functionalities of the proposed system. In Table 3, these comparisons are shown.

Since the proposed system makes use of both visual cryptography and SLAM algorithms, it takes the benefits from both. Of course, there are some improvements that can be done. A first improvement could be the design of a more robust share alignment method instead of using fixed transformation. Another improvement could be the use of a more powerful (but more expensive) rover, in order to integrate sensors such as time of flight cameras allowing the use of a faster SLAM algorithm.

5 CONCLUSIONS

In recent years, autonomous robots are used in several fields due to their capacity to scour dangerous and difficult to reach areas of interest. Usually, these robots send the acquired data to a ground station in order to perform elaborations on it, and in some cases there is the need to hide the information sent. In this paper, a rover-based system able to search encrypted targets in an unknown environment is presented. The system exploits two well-known approaches for achieving the goal. The first approach is the SLAM, which is used to explore the unknown environment. The second approach is the visual cryptography, which is used to decrypt the target object and to send encrypted video frames. The experimental results shown that the proposed pipeline works quite well in a controlled exper-

Table 3: Functionality comparison between the proposed system and the state-of-the-art-works.

	Type of Algorithm	Information Encryption/Decryption	Target Search	Sensors	Moving System
Pahuja et al. (Pahuja and Kasana, 2017)	Visual Cryptography	Yes	No	N/A	No
Alex et al. (Alex and Anbarasi, 2011)	Visual Cryptography	Yes	No	N/A	No
Balclar et al. (Balclar et al., 2017)	SLAM	No	No	RGBD Camera + laser range finder	Yes
Li et al. (Li et al., 2016)	SLAM	No	No	RGBD Camera	Yes
Walas et al. (Walas et al., 2016)	SLAM	No	No	RGBD Camera + time of flight camera	Yes
Proposed System	Visual Cryptography + SLAM	Yes	Yes	RGB Camera + Ultrasonic Sensors	Yes

imental environment.

REFERENCES

Alex, N. S. and Anbarasi, L. J. (2011). Enhanced image secret sharing via error diffusion in halftone visual cryptography. In *2011 3rd International Conference on Electronics Computer Technology*, volume 2, pages 393–397.

Avola, D., Foresti, G. L., Cinque, L., Massaroni, C., Vitale, G., and Lombardi, L. (2016). A multipurpose autonomous robot for target recognition in unknown environments. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pages 766–771.

Avola, D., Foresti, G. L., Martinel, N., Micheloni, C., Pannone, D., and Piciarelli, C. (2017a). Aerial video surveillance system for small-scale uav environment monitoring. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Avola, D., Foresti, G. L., Martinel, N., Micheloni, C., Pannone, D., and Piciarelli, C. (2017b). *Real-Time Incremental and Geo-Referenced Mosaicking by Small-Scale UAVs*, pages 694–705. Springer International Publishing, Cham.

Babu, R., Sridhar, M., and Babu, B. R. (2013). Information hiding in gray scale images using pseudo-randomized visual cryptography algorithm for visual information security. In *2013 International Conference on Information Systems and Computer Networks*, pages 195–199.

Balclar, M., Yavuz, S., Amasyal, M. F., Uslu, E., and akmak, F. (2017). R-slam: Resilient localization and mapping in challenging environments. *Robotics and Autonomous Systems*, 87(Supplement C):66 – 80.

Bayer, B. E. (1973). An optimum method for two-level rendition of continuous-tone pictures. In *IEEE Int. Conf. on Communications*, volume 26, pages 11–15.

Boiangiu, C.-A., Bucur, I., and Tigora, A. (2012). The image binarization problem revisited: Perspectives and approaches. *Journal of Information Systems & Operations Management*, 6(2):1.

Bu, S., Zhao, Y., Wan, G., and Liu, Z. (2016). Map2dfusion: Real-time incremental uav image mosaicing based on monocular slam. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4564–4571.

Cacace, J., Finzi, A., Lippiello, V., Furci, M., Mimmo, N., and Marconi, L. (2016). A control architecture for multiple drones operated via multimodal interaction in search rescue mission. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 233–239.

Camurri, M., Bazeille, S., Caldwell, D. G., and Semini, C. (2015). Real-time depth and inertial fusion for local slam on dynamic legged robots. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 259–264.

Chen, L., Sun, L., Yang, T., Fan, L., Huang, K., and Xuanyuan, Z. (2017). Rgb-t slam: A flexible slam framework by combining appearance and thermal information. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5682–5687.

Hou, Y.-C. (2003). Visual cryptography for color images. *Pattern Recognition*, 36(7):1619 – 1629.

Jarvis, J. F., Judice, C. N., and Ninke, W. (1976). A survey of techniques for the display of continuous tone pictures on bilevel displays. *Computer Graphics and Image Processing*, 5(1):13–40.

Joseph, S. K. and R, R. (2015). Random grid based visual cryptography using a common share. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 656–662.

Kadhim, A. and Mohamed, R. M. (2016). Visual cryptography for image depend on rsa algamal algorithms. In *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, pages 1–6.

Kaur, T. and Kumar, D. (2015). Wireless multifunctional robot for military applications. In *2015 2nd International Conference on Recent Advances in Engineering Computational Sciences (RAECS)*, pages 1–5.

Kim, D. Y., Kim, J., Kim, I., and Jun, S. (2015). Artificial landmark for vision-based slam of water pipe rehabilitation robot. In *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 444–446.

Kiyani, M. N. and Khan, M. U. M. (2016). A prototype of search and rescue robot. In *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pages 208–213.

Knuth, D. E. (1987). Digital halftones by dot diffusion. *ACM Transactions on Graphics (TOG)*, 6(4):245–273.

Leonard, J. J. and Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3.

Li, C., Wei, H., and Lan, T. (2016). Research and implementation of 3d slam algorithm based on kinect depth sensor. In *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1070–1074.

Lin, C.-C. and Tsai, W.-H. (2003). Visual cryptography for gray-level images by dithering techniques. *Pattern Recognition Letters*, 24(1):349 – 358.

Liu, F., Wu, C. K., and Lin, X. J. (2008). Colour visual cryptography schemes. *IET Information Security*, 2(4):151–165.

Liu, S., Fujiyoshi, M., and Kiya, H. (2014). A cheat preventing method with efficient pixel expansion for naor-shamir's visual cryptography. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5527–5531.

Mur-Artal, R. and Tards, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Naor, M. and Shamir, A. (1995). *Visual cryptography*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg.

Pahuja, S. and Kasana, S. S. (2017). Halftone visual cryptography for color images. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 281–285.

Shyu, S. J. (2006). Efficient visual secret sharing scheme for color images. *Pattern Recognition*, 39(5):866–880.

Sim, R. and Roy, N. (2005). Global a-optimal robot exploration in slam. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 661–666.

Stinson, D. (1997). An introduction to visual cryptography. *Public Key Solutions*, 97:28–30.

Trivun, D., alaka, E., Osmankovi, D., Velagi, J., and Osmi, N. (2015). Active slam-based algorithm for autonomous exploration with mobile robot. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 74–79.

Walas, K., Nowicki, M., Ferstl, D., and Skrzypczyski, P. (2016). Depth data fusion for simultaneous localization and mapping - rgb-dd slam. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 9–14.

Yang, D., Doh, I., and Chae, K. (2017). Enhanced password processing scheme based on visual cryptography and ocr. In *2017 International Conference on Information Networking (ICOIN)*, pages 254–258.