International Journal of Hybrid Information Technology
Vol. 10, No. 9 (2017), pp.19-28
http//dx.doi.org/10.14257/ijhit.2017.10.9.02

Hiding Information as Main Content Use a Referential Method

Yuan Ren[1] and Xixu Fu[2]

[1]*Electronics and Information School, Shanghai Dianji University*
[2]*Institute of Information and Education Technology, Shanghai Ocean University*
reny@sdju.edu.cn

Abstract

Problems of detect ability and capacity are two major challenges of steganography. Traditional methods often hide secret message in least significant bits which can cause low capacity. The secret bits may be ignored by men but detectable for machines. This paper introduced a referential method which needs no cover image. In this method, secret message is ciphered using a cipher graph and written to an empty graph to form a figure. The method provides good concealment to both the secret message and the existence of the secret message. The capacity of the method is bigger than many traditional stego methods too.

Keywords*: steganography, referential encoding, location function, major bits information hiding, RGB contour space*

1. Introduction

Steganography [1] is a new branch of data protection which hides information in large media to avoid detection and extraction. Many methods have been advanced for information hiding. Secret message is usually hidden in cover media such as an image [2]. These methods protect secret messages in two aspects. Hackers who get a stegoimage may regard it as normal one and ignore the message embedded in it. Once hackers know the existence of secret message, they may face the problem of extracting the message. However, traditional steganography methods have many limits [4].

On one hand, traditional algorithms in steganography try to hide information into large media to avoid detecting. For example, LSB methods hide secret message into the least significant bits of images. The changes of cover image can hardly be detected by human eyes. However, the change of the carrier media can be always detected by de-tecting algorithms.

On the other hand, encryption and location of secret message are simple and isolated. Elements of secret messages can be easily extracted from stegomedia even by blind methods [13].

The motivation of this paper is to advance a method which hide information in different media as the main content rather than hidden bits. This paper advanced a novel method referential for information hiding. Secret bits are stored as a part or the whole content of carrier media. Secret message is encrypted by location as well as encoding. Although many location functions were discussed, the main idea of the paper is the referential method to create the content of stegomedia.

In this paper, different steganography and steganalysis methods were analyzed first. Based on the analysis, a new framework on referential stegnography was introduced. After that, a method for bitmap and RGB contour space was introduced. Then, experiments were carried out according the algorithms. At last, the results were evaluated ac-cording metrics of steganography.

Received (August 11, 2016), Review Result (November 15, 2016), Accepted (January 5, 2017)

ISSN: 1738-9968 IJHIT

2. Related Work and Motivation

2.1. Definition and Common Progress of Steganography

While classical cryptography is about concealing the content of messages, steganography is also about concealing their existence [4]. A popular definition of steganography and steganalysis [3][15] is described as Figure 1.

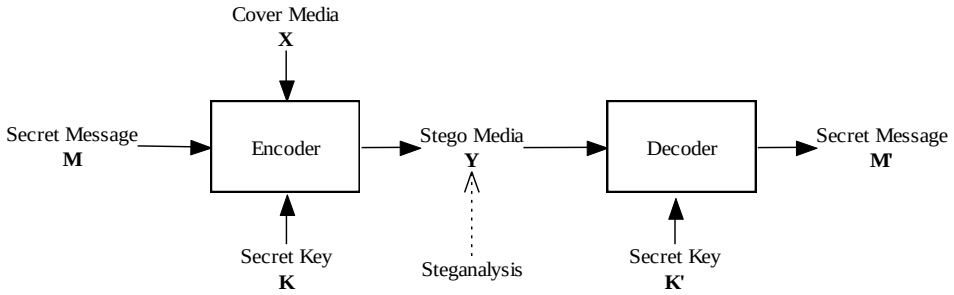


Figure 1. The Model of Steganography and Steganalysis

This model described how common stegnography systems work. Secret message (normally bits) is embedded into stegomedia (may be pictures, videos, texts or so on) use secret key (optional) to produce stegomedia. Receiver uses decoder and corresponding secret key to extract secret message. Attackers get stegomedia and try to extract secret message from it.

2.2. Steganography Methods and Evaluation

According to the methods and cover media, there are many categories of stegomethods. Images, including bitmaps and compressed images are the most popular cover media of stegomethods [1][2][6][8][9][10][15]. Chinese documents [5] and videos [11] are also used as cover media.

On the aspect of algorithms, least significant bits based methods [16][17] are popular. These methods try to hide secret message in the least significant bits so as to avoid the detection. However, these methods are vulnerable under detection [16].

There are many measures about steganography methods. Security (detect ability), robustness and secrecy (difficulty of extraction) [12] are most obvious measures according to the function of steganography. The most popular qualified measure is hiding capacity [7][12]. Capacity is also the measurement for robustness and limited by security [12]. Many stegosystems compress secret messages to enhance the utilization of capacity [7][9].

3. Hide Information in Bitmaps use A Referential Mode

3.1. Frame for Referential Data Hiding in bitmap

While traditional staganography methods change carrier media insignificantly to store secret information, referential method use secret information as the content of car-rier media.

A detailed framework of referential steganography can be described in a six ad [M, G_R, $f_L()$,$f_E()$,$f_D()$,G_S]. M is the secret information represented as bits. G_R is the bitmap used as a cipher dictionary. Function $f_L()$ is the location function of secret bits in cipher bitmap. Function $f_E()$ is an encryption function applied to the secret information. Both $f_L()$ and $f_E()$ are needed in decoder process as secret key. G_S Is the output stegograph contains the secret information. Function $f_D()$ is the decryption function. Corresponding to the model

in figure1, M, GR, $f_L()$,$f_E()$, are used in the encoding stage to generate stegoimage G_S. G_R, $f_L()$ and $f_E()$ are all secure for attackers. GR, $f_L()$, $f_D()$ can be used to extract secret message from stegoimage G_S.

The detailed method to generate stegoimage can be shown in Figure 2.

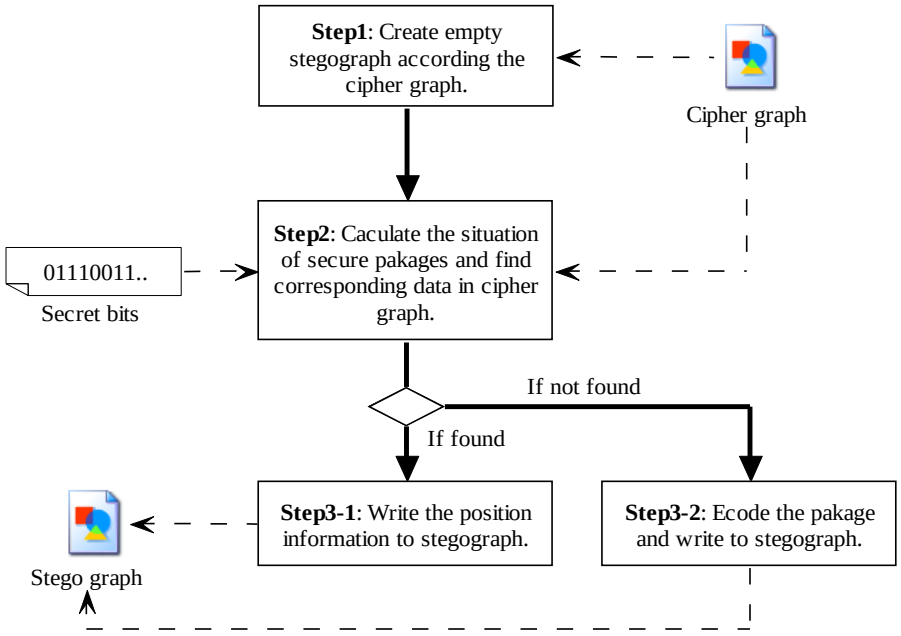


Figure 2. Hide Secret Information into Graphs use Referential Method

As shown in Figure 3, an empty graph is generated as stegograph first. Then, pack-ages are encoded according the position and the cipher graph. The representation of an encoded element is shown in Figure 3.

Referent Position / Content (Encoded)	Indicator

Figure 3. Embed Bits into Referential Mode

For example, a symbol 'A' (with ASCII code 65) is to be settled in position (100,100) of a bitmap according $F_L()$. If value 65 is find in the red contour of a pixel (103,101), the position (3,1) is recorded as referent position, the indicator is set to 1 which represent the contour red. If the value is not found, the code 65 is encoded and stored in the first segment and the indicator is set to 0.

The decoding method is similar to the encoding method. Secret bits are extracted from stegograph according $f_L()$ first. Then the content is decoded according the indicator and cipher graph and decode method $f_D()$.

3.2. Hide English Words in Pixels

A simple method is to hide one symbol in each contour of stegobitmap. For a 24 bit bitmap, a contour is represented in 8 bits. Representation of a symbol can be described as Figure 4.

Position X (3bits)	**Position Y (3bits)**	**Indicator (2 bits)**
Encoded content (6bits)		

Figure 4. Represent Symbols in a Contour

Since ASCII codes are represented in 8 bits, symbols should be encoded in fewer bits. The encoding of symbols can not be represented in referential position is shown in Table 1.

Table 1. Table Label

Symbol	*Value*	*Symbol*	*Value*	*Symbol*	*Value*	*Symbol*	*Value*
A/a	*1*	Q/q	17	7	33	/	49
B/b	*2*	R/r	18	8	34	[/]	50
C/c	*3*	S/s	19	9	35	\	51
D/d	*4*	T/t	20	0	36	^	52
E/e	*5*	U/u	21	blank	37	$	53
F/f	*6*	V/v	22	./!	38	&	54
G/g	*7*	W/w	23	‘/”	39	~	55
H/h	*8*	X/x	24	:	40	|	56
I/i	*9*	Y/y	25	?	41	>	57
J/j	*10*	Z/z	26	@	42	=	58
K/k	*11*	1	27	#	43	<	59
L/l	*12*	2	28	(/)	44	_	60
M/m	*13*	3	29	{/}	45	;	61
N/n	*14*	4	30	+	46	Others	62
O/o	*15*	5	31	-	47		
P/p	*16*	6	32	*	48		

Since ASCII codes are represented in 8 bits, symbols should be encoded in fewer bits. The encoding of symbols can not be represented in referential position is shown in Table 1.

The detailed algorithms of embedding and extraction are shown in Figure 5 and Figure 6 respectively.

```
Input: string txt, cipher bitmap cipherBMP
Output: stagano bitmap steganoBMP
Begin
  Create stegoBMP as a blank bitmap with same width and height of cipherBMP
  for all characters c in txt
     Compute the position x,y of c in stegoBMP
     Find a point p(x',y') in cipherBMP near (x,y) where p.Color.R=ASCII(c) or  p.Color.G=ASCII(c) or p.Color.B=ASCII(c)
          If p is found
               set indicate bits ind=1 if the color is R, ind=2 if the color is G,  ind=1 if the color is B
               set code(c)=(x'-x)×32+(y'-y)×4+ind
          Else
               code(c)=encode(c)*4+0
     put code(c) in queue q
  while q is not empty
     Get 3 codes r,g,b from q, if q is empty then the dimension would be 0
     Set a pixel into stegoBMP with color RGB(r,g,b) in computed position (x,y)
End
```

Figure 5. Algorithm used to Embed Text into a Stegobitmap

As shown in Figure 5, the stegobitmap is as a blank bitmap with the same size of the cipher bitmap created first. For every letter in the text, the beginning position should be first found in cipher bitmap. Then, the algorithm tries to find a contour meets the ASCII value of the letter. If the contour is found, then the indicator bits are set to corresponding contour, the referent positions are encoded as the higher bits (The structure of storage is shown in Figure 5). If no contour meets the value, the letter is encoded according to Table1 and the indicator bits are set to 0.


```
Input: cipher bitmap cipherBMP, stego bitmap stegoBMP
Output: secret message txt
Begin
 do
      Compute the position p(x,y) in stegoBMP according fL()
      For every contour R,G,B c
           if c=0 then return(txt)
           swich(last 2 bits b in c)
                case(1)
                     Find point p(x+first 3bits of c,x+second 3bits of c) in cipherBMP
                     m=char(p.R)
                case(2)
                     Find point p(x+first 3bits of c,x+second 3bits of c) in cipherBMP
                     m=char(p.G)
                case(3)
                     Find point p(x+first 3bits of c,x+second 3bits of c) in cipherBMP
                     m=char(p.B)
                case(0)
                   m=Decode(first 6bits of c)
           txt=txt+m
  while not the end of information
End
```

Figure 6. Extract Message from a Stegobitmap

The extracting of secret message is quite simple. Users need to know the cipher bitmap, the location function and the stegobitmap. In the extracting algorithm, positions of searching points are first calculated using location function. Then, encoded frames are extracted from the stegobitmap. By analyze the indicator bits and content, letters are found in cipher bitmap or calculated using decoding function until the end point is found.

3.3. Location Functions

Location functions decide how the encoded secret message distributed in the stego-graph. Take RGB bitmap as an example, a location is decided by the pixel position (X,Y) and contour location R,G or B. For the sake of accurate extracting, a location function must be a function which returns distinct positive integer values in different machines. There are some simple functions useful below.

Full bitmap distribution:

$f_{FULL}=\{(X, Y, C)|X\in N, Y\in N, C\in\{RGB\}, X< Width, Y<Height\}$

This location function means all contours of all pixels of a bitmap is used to store secret bits.

Sine distribution:

$f_{SIN}=\{(X, Y, C)|X\in N, X< Width, Y=fix(n\times sin(X)), C\in\{RGB\}\ \}$

This location function means all contours of pixels in a sine curve is used to store secret bits. For the sake of accuracy, values are turned into integers use fix function (The function must be same in both encryption and decryption process.). Locations in a curve needs use the fix function in normal.

Circle distribution:

$$f_{CIR}=\{(X, Y, C)\mid X\in N, X< Width, Y=fix(r+\sqrt{r^2-(X-r)^2}), C\in\{RGB\}, r=fix(Width/2)\}$$

This location function means all contours of pixels in a circle is used to store secret bits. Values are also turned into integers too.

For better capacity and visual effect, complex and multiple shapes can be used. For example:

Multi Circles distribution:

$$f_{MCIR} = \{(X, Y, C) \mid X \in N, X < \text{Width}, Y = \text{fix}(l + \sqrt{r^2 - (X - r)^2}), C \in \{RGB\}, l = \text{fix}(\text{Width}/2), r = 10, 20, ..\}$$

This location function stores secret bits into many circles so as to get better capacity and better appearance.

3.3. Notations of Methods

Using different location function and different encoding algorithms, many solutions and algorithms can be derived from the referential model. Main differences between these algorithms are location functions and the reference space to find the value. The example algorithms use 8 bits to representation a symbol. To improve the hit rate of reference, 12 bits can be used to encode one symbol. Two bits used as indicator and 10 bits as referent position or content.

In this paper, these methods are named by the location function and detailed structure of encryption package. Detailed nomination is shown as below:

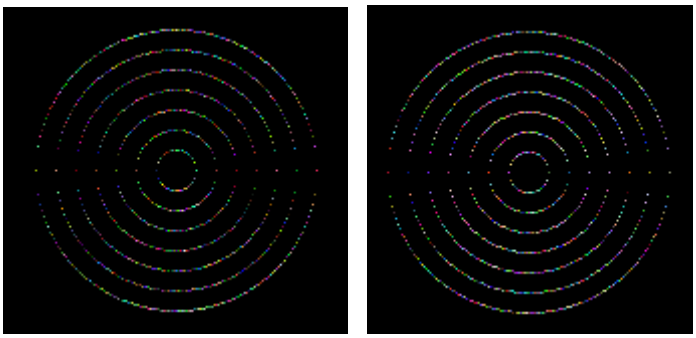
[Location Function][Reference bits X]- [Reference bits Y]-[Indicator bits]

For example, a method using package structure in figure 4 and using multi circles location function is named MCIR3-3-2.

4. Experiments and Evaluation

Experiments were carried out on methods MCIR3-3-2, MCIR4-4-2 and MCIR5-5-2 to evaluate the effectiveness and security of the algorithm. A bitmap with 600*600 pixels is used as cipher graph. Three categories of data were used as messages. The first category was composed of random letter sequence. The second category was composed of essays and articles. The third category was composed of XML documents, web pages and mails.

The stegograph of essay Three Days to See (about 2.7KB) using MCIR3-3-2 is shown in of Figure 7A. As a comparison, a bitmap consists of circles in random colors is shown in Figure 7B.



A B

Figure 7. Extract Message from a Stegobitmap

As shown in Figure 7, the stegograph just like randomly generated concentric circles. It is not easy to distinguish a stegograph from random colored bitmaps. Other distributions can show different stegographs but these stegographs are not easy to be distinguished from random colored bitmaps as well.

Because the letters can not be found are directly encoded use their ASCII code, the rate letter can be found (hit rate) decides the security of message. The length of reference bits decides the rate secret messages being found in cipher graph. More secret messages found,

harder to extract secret messages. Figure 8 shows the hit rate of symbols of various categories of data.

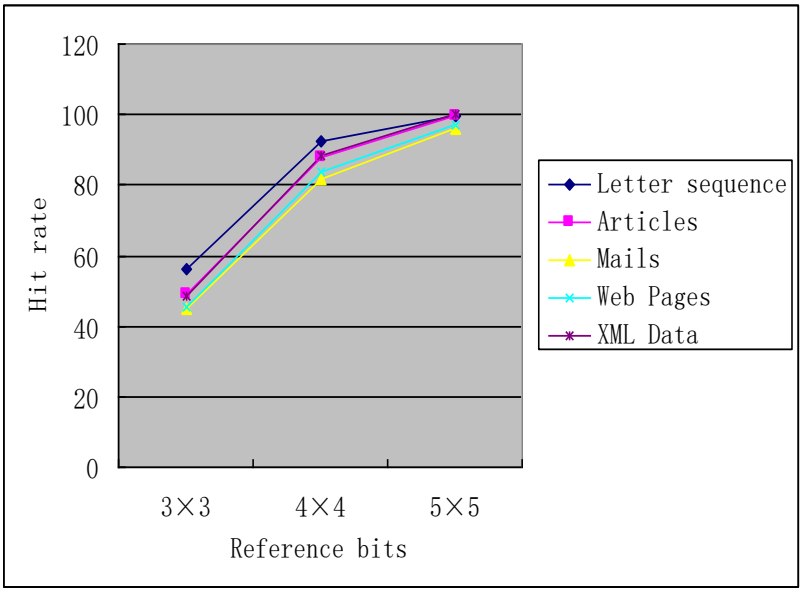


Figure 8. Hit Rate Improves while Searching Space Expands

As shown in Figure 10, searching space increases while reference bits increase. More pixels are used to find ASCII values. So the hit rate increases. It is shown that MCIR5-5-2 and MCIR4-4-2 get better performance than MCIR3-3-2.

The capacity of referential methods and other stegomethods are shown in Figure 9.

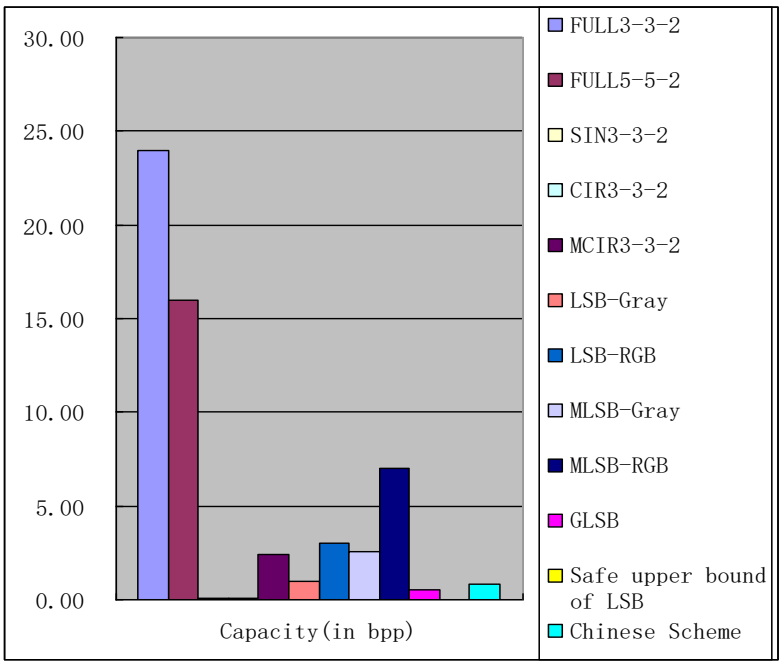


Figure 9. Capacity of Stegomethods

From the result, we can see the referential methods especially full bitmap location method gain bigger capacity than normal stegomethods.

5. Conclusion

This paper advanced a new stegomethod which use referential position to encode data and hide information as main content of a bitmap. From the experiments and analysis, it is concluded as below:

1. The referential method composes the stegograph using the encoded message. It provides good protection to the existence of secret message.

2. It is almost impossible to extract secret messages without knowing the de-tailed location function and the cipher graph.

3. The capacity of referential algorithms varies with the location function. The capacity of full distribution and compound distributions are better than LSB and many algorithms.

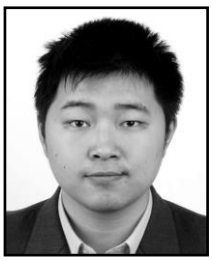
From the conclusions above, the method is secure and efficient for information hiding.

References

[1] K. Bailey and K. Curran, "An Evaluation of Image Based Steganography Methods Using Visual Inspection and Automated Detection Techniques", Multimedia Tools and Applications, vol. 30, no. 1, **(2006)**, pp. 55-58.
[2] A. Cheddad, J. Condell, K. Curran and P. Mc Kevitt, "Digital Image Steganography: Survey and Analysis of Current Methods", Signal Processing, vol. 90, no. 3, **(2010)**, pp. 727-752.
[3] P. Moulin and J. A. O Sullivan, "Information-Theoretic Analysis of Infor-mation Hiding", IEEE Transactions on Information Theory, vol. 49, no. 3, **(2003)**, pp. 563-593.
[4] R. J. Anderson and F. A. P. Petitcolas, "On the Limits of Steganography", IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, **(1998)**, pp. 474-481.
[5] Z.-H. Wang, C.-C. Chang, C.-C. Lin and M.-C. Li, "A Reversible Information Hiding Scheme Using Left–right and up–down Chinese Character Representation", The Journal of Systems and Software, vol. 82, **(2009)**, pp. 1362-1369.
[6] B. Ryabko and D. Ryabko, "Constructing Perfect Steganographic Systems", Information and Computation, vol. 209, **(2011)**, pp. 1223-1230.
[7] E. Satir and H. Isik, "A Compression-based Text Steganography Method", The Journal of Systems and Software, vol. 85, **(2012)**, pp. 2385-2394.
[8] N. Hamad, "Hiding Text Information in a Digital Image Based on Entropy Function", The International Arab Journal of Information Technology, vol. 7, no. 2, **(2010)**, pp. 146-151.
[9] R. Das and T. Tuithung, "A Novel Steganography Method for Image Based on Huffman Encoding", Proceedings of the 3rd National Conference on Emerging Trends and Applications in Computer Science, **(2012)**, pp. 14-18.
[10] M. Bilal, S. Imtiaz, W. Abdul, S. Ghouzali and S. Asif, "Chaos Based Zero-steganography Algorithm", Multimedia Tools and Applications, vol. 37, **(2013)**, pp. 1-20.
[11] J. P. Cruz, N. J. Libatique and G. Tangonan, "Steganogra-phy and Data Hiding in Flash Video (FLV)", Proceedings of IEEE Region 10 Conference: Sustainable Development Through Humanitarian Technology, **(2012)**, pp. 1-6.
[12] W.-M. Zhang and S.-Q. Li, "Security Measurements of Steganographic Systems", Proceedings of The Second International Conference of Applied Cryptography and Network Security, **(2004)**, pp. 194-204.
[13] X.-Y. Luo, D.-S. Wang, P. Wang and F.-L. Liu, "A Review on Blind Detection for Image Steganography", Signal Processing, vol. 88, **(2008)**, pp. 2138-2157.
[14] F. A. Stahl, "A Homophonic Cipher for Computational Cryptography", National Computer Conference, **(1973)**, pp. 565-568.
[15] B. Li, J.-H. He, J.-W. Huang and Y. Q. Shi, "A Survey on Image Steg-anography and Steganalysis", Journal of Information Hiding and Multimedia Signal Processing, vol. 2, **(2011)**, pp. 142-172.
[16] J. Fridrich, M. Goljan and R. Du, "Detecting LSB Steganography in Color and Gray-Scale Images", IEEE Multimedia, vol. 8, **(2001)**, pp. 22-28.
[17] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, "Lossless Generalized-LSB Data Embedding", IEEE Transactions on Image Processing, vol. 14, **(2005)**, pp. 253-266.

Authors

Yuan Ren, was born in Changchun city in 1984. He earned his doctorate of science from the School of Computer Science of Fudan University in 2013. He is now an instructor in Shanghai Dianji University. His research interests include computer vision and computer teaching.



Xixu Fu, was born in 1981. He got his master and ph. D degree in 2007 and 2015 at Fudan University respectively. Now he is an engineer in Shanghai Ocean University. His research interests include artificial intelligence, psychology, cloud computing and software engineering.

