基于 Luffa 杂凑函数的旋转攻击

李云强[a]，赵土华[a]，曹进克[b]
（信息工程大学 a. 密码工程学院；b. 科研部，郑州 450004）

摘　要：针对 Luffa 杂凑函数抗旋转攻击的能力进行了研究，通过分析 Luffa 置换的特点，定义了针对 Luffa 置换的旋转对，给出了基本字变换对旋转关系的影响，证明了“与”和“或”运算对旋转关系影响的等效性，提出了缩减轮 Luffa 置换与随机置换的区分算法。理论分析和实验结果均表明，3 轮 Luffa 置换难以抵抗旋转攻击，攻击的计算复杂度仅为 2^{16}。

关键词：Luffa 杂凑函数；SHA3 候选算法；旋转攻击；区分攻击
中图分类号：TP309　　**文献标志码**：A　　**文章编号**：1001-3695(2013)12-3807-03
doi:10.3969/j.issn.1001-3695.2013.12.075

Rotational attack on Luffa hash function

LI Yun-qiang[a], ZHAO Tu-hua[a], CAO Jin-ke[b]
(*a. Institute of Cryptographic Engineering, b. Scientific Research Department, Information Engineering University, Zhengzhou* 450004, *China*)

Abstract: This paper studied the capacity of resisting rotational attack of Luffa hash function. Through analyzing characteristics of Luffa permutation, it defined the rotational pair of Luffa permutation, gave the influence of basic word transformations to rotational relation, proved that the AND operation and the OR operation had the same influence on rotational relation, and presented the distinguisher algorithm between round-reduced Luffa permutation and a random permutation. Theoretical analysis and experiment results show that 3 round Luffa permutation can't resist rotational attack and the attack complexity is only 2^{16}.
Key words: Luffa hash function; SHA3 candidate algorithms; rotational attack; distinguisher attack

随着 Internet 网络的快速发展，网络信息安全已成为人们研究的一个重点课题。在网络通信环境中，攻击者可以利用窃听、重放、修改消息内容顺序等方法来达到攻击的目的，因此就必须采取十分可靠的安全技术来保证信息的机密性、完整性和不可伪造性。密码学中的杂凑函数是实现上述性能不可缺少的工具，从 1990 年 Rivest 提出 MD4 算法开始，有关 hash 函数的设计和安全性研究在密码学中始终是一个重要方向。

随着对杂凑函数攻击技术的研究进展，人们开始质疑经典杂凑函数 MDx、SHAx 的安全性。美国国家标准技术研究所(NIST)从 2007 年开始向全世界征集新一代杂凑算法标准 SHA3，共征集到 64 个杂凑算法。这些算法从不同侧面代表了杂凑函数的最新设计理念。Luffa 杂凑函数[1,2]是进入 NIST SHA3 竞赛第二轮的 14 个候选算法之一，尽管没有进入最终候选算法的争夺，但也显示出良好的安全性和较高的实现效率。本文利用旋转攻击技术进一步分析 Luffa 杂凑函数的安全性。

旋转攻击[3]是 2010 年提出的一类新的密码分析技术，人们证明了 ARX(addition，rotation，XOR)模型抗击旋转攻击的性能仅与所使用的 addition 数相关[3]，利用旋转攻击能够区分 53 轮 Skein-256 和 57 轮 Skein-512 的杂凑函数[4]，能够区分 CubeHash 15 轮压缩函数[5]，提出了针对 KECCAK 杂凑算法 4 轮的原象攻击和 5 轮区分攻击[6]。这些结果是目前对这些杂凑算法最好的分析结果，由此可见旋转攻击的攻击效率。

对 Luffa 杂凑函数的已知攻击可以分为两类：a)针对所使用的压缩函数和置换的非随机性研究(如高阶区分、旋转区分等)；b)针对整个杂凑函数的攻击(如碰撞攻击、原象攻击等)。目前对 Luffa 杂凑函数攻击效果较好的攻击方法是区分攻击，文献[7]利用高阶差分分析方法区分第一版 7 轮压缩函数的时间复杂度为 2^{216}，并迫使作者在提交第二轮时对压缩函数进行了局部调整；文献[8]利用差分分析的方法区分第二版 8 轮置换的时间复杂度为 2^{116}，利用反弹攻击区分第二版 8 轮压缩函数的时间复杂度为 2^{116}，存储复杂度为 2^{102}。

1　Luffa 杂凑函数介绍

Luffa 杂凑函数采用变形的 Sponge 迭代结构(图 1)，输出比特长度为 224、256、384、512 的杂凑函数对应的 w 值分别为 3、3、4、5。杂凑函数由三部分构成，分别是消息填充、迭代函数和输出函数。消息填充是在消息的后面附加一个 1，然后填充最少比特的 0，使得填充之后的比特数为 256 的倍数，每 256 bit 为一个消息块，把消息分块 $M = M_0 \| \cdots \| M_{m-1}$。迭代函数是由一个消息注入函数 MI 和置换 P 组成，其中置换 P 是由 w 个具有相同规模的置换 $Q_0, Q_1, \cdots, Q_{w-1}$ 构成。输出函数由空消息注入 MI 和置换 P 组成，最多经过两次空消息注入输出所需长度的比特杂凑值。

这里重点介绍 $Q_j(j = 0, 1, \cdots, w-1)$ 的构成，其他环节可参看文献[2]。非线性置换 Q_j 的输入和输出数据的长度都是 256 bit，它是由输入的一次 tweak 和一个步函数的 8 次迭代所构成的，其中步函数是由三个变换 subCrumb、mixWord 和 addConstant 构成(图 2)。

收稿日期：2013-01-27；**修回日期**：2013-04-02
作者简介：李云强(1968-)，男，河南尉氏人，教授，博士，主要研究方向为密码理论与技术(lyq203@126.com)；赵土华(1987-)，男，山西阳泉人，硕士研究生，主要研究方向为密码理论与技术；曹进克(1964-)，男，河南偃师人，副教授，硕士，主要研究方向为信息安全理论与技术.

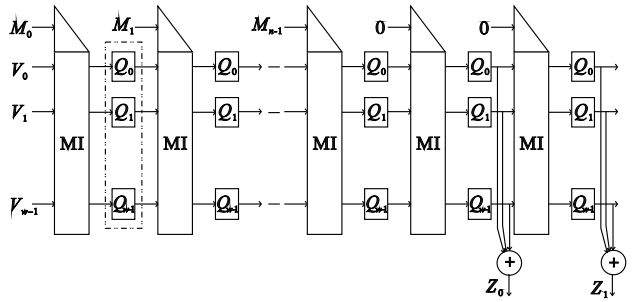

图1 Luffa杂凑函数的迭代结构

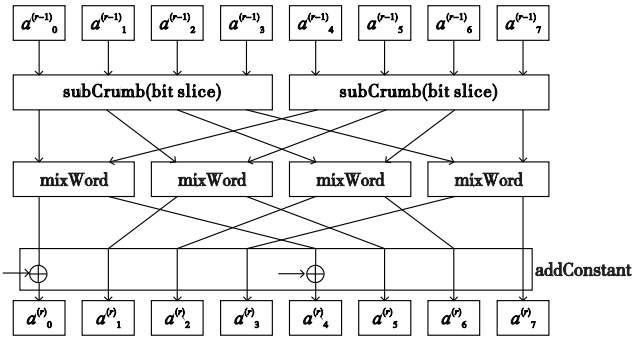

图2 步函数的结构

Q_j 的伪代码表示如下：

```
permute(a[8], j)  //permutation Q_j
{
  tweak(a);
  for (r=0;r<8; r++)
    {
      subCrumb(a[0],a[1],a[2],a[3]);
      subCrumb(a[5],a[6],a[7],a[4]);
      for (k=0; k<4; k++)
        mixWord(a[k],a[k+4]);
        addConstant(a, j, r);
    }
}
```

通过分析可知，除了循环移位变换外，Q_j 的所有环节都可以通过字之间的 AND、XOR、OR、NOT 实现，而循环移位变换只改变旋转关系的位置不改变旋转关系的量化。因此，只要确定这些基本字变换对旋转关系的影响，就能确定 Q_j 抗旋转攻击的性能。

2 针对 Luffa 杂凑函数的旋转攻击

旋转攻击的基本思想是：首先寻找满足 $F(\overleftarrow{X})=\overleftarrow{F(X)}$ 的旋转对 $(X,\overleftarrow{X})$，然后利用旋转对的性质进行区分攻击、密钥恢复攻击和原象攻击。其中，$\overleftarrow{X}$ 是 X 的某种移位变换，$\overleftarrow{F(X)}$ 是 $F(X)$ 的某种移位变换。对于映射到 Z^n（n 为正整数）上的函数 $Y=F(X)$ 而言，如果函数 F 是随机的，$F(\overleftarrow{X})=\overleftarrow{F(X)}$ 的概率为 2^{-n}；反过来，如果 $F(\overleftarrow{X})=\overleftarrow{F(X)}$ 的概率大于 2^{-n}，则说明 $Y=F(X)$ 不具有随机性，能够与随机函数进行区分。

旋转攻击成功与否主要决定于函数对旋转关系的保持性，也即对于给定函数的一个输入旋转对，经过函数变换后其相应的输出是否仍为旋转对。一般用变量之间的循环移位定义旋转关系，但不同的函数变换可能需要不同的定义方法。下面针对 Q_j 置换的特点给出旋转对的定义和概率刻画。

2.1 旋转对及概率刻画

定义 1 设 $A=(A_0,A_1,\cdots,A_7)$，$\overleftarrow{A}=(A_0<<<n,A_1<<<n,\cdots,A_7<<<n)\in(Z_2^{32})^8$，则称 $(A,\overleftarrow{A})$ 为旋转数为 n 的旋转对。其中 n 为正整数，$<<<$ 为字循环移位变换。

对于满足旋转对关系的一对输入，经过 Q_j 置换中的几步变换会发现，只有少量的比特之间满足旋转关系，为此需要考察比特对之间旋转关系的演变规律。为了研究的方便，本文用 $A_{(x,y)}$ 表示 32 bit 字 A_x 的第 y bit，A_x 的 bit 从右到左依次称为第 0，1，…，31 bit，并且约定 $y+n=(y+n)\bmod 32$，$y-n=(y-n)\bmod 32$。

定义 2 对于 Q_j 置换，定义

$$p_{(x,y)}=Pr(Q_j(A)_{(x,y)}\neq Q_j(\overleftarrow{A})_{(x,y+n)})$$

表示当输入为旋转对时对应输出两比特不相等的概率。其中 $x=0,1,\cdots,7$；$y=0,1,\cdots,31$。

显然，当 $p_{(x,y)}=1$ 时说明对应的输出两比特一定不相等，当 $p_{(x,y)}=0$ 时说明对应的输出两比特一定相等，当 $p_{(x,y)}=0.5$ 时说明对应的输出两比特是互相独立的。对于随机置换而言，对于所有的 (x,y)，$p_{(x,y)}$ 都应等于 0.5。如果对于某些 (x,y)，能够确定 Q_j 置换对应的 $p_{(x,y)}$ 不等于 0.5，则以此能够区分随机置换和 Q_j 置换。

2.2 基本字变换对旋转概率的影响

为了计算连续步骤中 $p_{(x,y)}$ 的变化概率，需要分析 Q_j 置换中所用基本字变换 AND、XOR、OR、NOT 对 $p_{(x,y)}$ 变化的影响。

引理 1 在输入相同的情况下，OR 和 AND 两种操作的输出满足：

$$p_{\text{out}}^{\text{or}}=p_{\text{out}}^{\text{and}}$$

其中，$p_{\text{out}}^{\text{or}}$ 和 $p_{\text{out}}^{\text{and}}$ 根据定义 2 定义。

证明 不妨设 OR 和 AND 对应的输入字均为 A、B，$y\in\{0,1,\cdots,31\}$，根据定义 2，令

$$p_a=Pr(A_{(a,y)}\neq A'_{(a,y+n)})$$
$$p_b=Pr(B_{(b,y)}\neq B'_{(b,y+n)})$$
$$p_{\text{out}}^{\text{and}}=Pr(A_{(a,y)}\text{ AND }B_{(b,y)}\neq A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)})$$
$$p_{\text{out}}^{\text{or}}=Pr(A_{(a,y)}\text{ OR }B_{(b,y)}\neq A'_{(a,y+n)}\text{ OR }B'_{(b,y+n)})$$

在下面四种条件情况求相应的条件概率。

a) 当 $A_{(a,y)}=A'_{(a,y+n)}$ 且 $B_{(b,y)}=B'_{(b,y+n)}$ 时，

$$A_{(a,y)}\text{ AND }B_{(b,y)}=A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)}$$
$$A_{(a,y)}\text{ OR }B_{(b,y)}=A'_{(a,y+n)}\text{ OR }B'_{(b,y+n)}$$

此时条件概率 $p_{\text{out}}^{\text{or}}=p_{\text{out}}^{\text{and}}=0$。

b) 当 $A_{(a,y)}\neq A'_{(a,y+n)}$ 且 $B_{(b,y)}\neq B'_{(b,y+n)}$ 时，如果 $A_{(a,y)}\neq B_{(b,y)}$，则

$$A_{(a,y)}\text{ AND }B_{(b,y)}=0=A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)}$$
$$A_{(a,y)}\text{ OR }B_{(b,y)}=1=A'_{(a,y+n)}\text{ OR }B'_{(b,y+n)}$$

如果 $A_{(a,y)}=B_{(b,y)}$，则

$$A_{(a,y)}\text{ AND }B_{(b,y)}\neq A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)}$$
$$A_{(a,y)}\text{ OR }B_{(b,y)}\neq A'_{(a,y+n)}\text{ OR }B'_{(b,y+n)}$$

而 $A_{(a,y)}=B_{(b,y)}$ 的概率为 $\frac{1}{2}$，所以此时条件概率 $p_{\text{out}}^{\text{or}}=p_{\text{out}}^{\text{and}}=\frac{1}{2}$。

c) 当 $A_{(a,y)}=A'_{(a,y+n)}$ 且 $B_{(b,y)}\neq B'_{(b,y+n)}$ 时，如果 $A_{(a,y)}=1$，则

$$A_{(a,y)}\text{ AND }B_{(b,y)}\neq A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)}$$
$$A_{(a,y)}\text{ OR }B_{(b,y)}=1=A'_{(a,y+n)}\text{ OR }B'_{(b,y+n)}$$

如果 $A_{(a,y)}=0$，则

$$A_{(a,y)}\text{ AND }B_{(b,y)}=0=A'_{(a,y+n)}\text{ AND }B'_{(b,y+n)}$$

$$A_{(a,y)}\ \mathrm{OR}\ \mathrm{B}_{(b,y)} \neq A'_{(a,y+n)}\ \mathrm{OR}\ \mathrm{B}'_{(b,y+n)}$$

而 $A_{(a,y)}=1$ 和 $A_{(a,y)}=0$ 的概率是相等的,都为 1/2,所以此时条件概率 $p_{\mathrm{out}}^{\mathrm{or}}=p_{\mathrm{out}}^{\mathrm{and}}=1/2$。

d)当 $A_{(a,y)}\neq A'_{(a,y+n)}$ 且 $B_{(b,y)}=B'_{(b,y+n)}$ 时,与 c)同理可得条件概率 $p_{\mathrm{out}}^{\mathrm{or}}=p_{\mathrm{out}}^{\mathrm{and}}=1/2$。

综上所述,$p_{\mathrm{out}}^{\mathrm{or}}=p_{\mathrm{out}}^{\mathrm{and}}$,定理得证。

从引理 1 的证明过程可知引理 2 和 3。

引理 2　对于 OR 操作,若输入比特为 a、b,输出比特为 out,则

$$p_{\mathrm{out}}^{\mathrm{or}}=\frac{1}{2}((1-p_a)p_b+p_a(1-p_b)+p_ap_b)$$

其中,p_a、p_b、$p_{\mathrm{out}}^{\mathrm{or}}$根据定义 2 定义。

引理 3[6]　对于 AND 操作,若输入比特为 a、b,输出比特为 out,则

$$p_{\mathrm{out}}^{\mathrm{and}}=\frac{1}{2}((1-p_a)p_b+p_a(1-p_b)+p_ap_b)$$

其中,p_a、p_b、$p_{\mathrm{out}}^{\mathrm{and}}$根据定义 2 定义。

引理 4[6]　对于 XOR 操作,若输入比特为 a、b,输出比特为 out,则

$$p_{\mathrm{out}}^{\mathrm{xor}}=(1-p_a)p_b+p_a(1-p_b)$$

其中,p_a、p_b、$p_{\mathrm{out}}^{\mathrm{xor}}$根据定义 2 定义。

引理 5[6]　对于 NOT 操作,若输入比特为 a,输出比特为 out,则

$$p_{\mathrm{out}}^{\mathrm{not}}=p_a$$

其中,p_a、$p_{\mathrm{out}}^{\mathrm{not}}$根据定义 2 定义。

通过上述引理可以处理 subCrumb 变换对于旋转关系的影响,mixColumn 只用到两种字变换 XOR 和 <<<,利用引理 4 可计算 XOR 操作对于旋转关系的影响,<<<操作并不改变旋转关系的概率值,只是改变旋转关系的位置。假如字 A 中满足旋转关系的概率为 $p_{(a,y)}$,则 $A<<<\sigma$ 中满足旋转关系的概率为 $p_{(a,y+\sigma)}^{<<<}=p_{(a,y)}$,其中 $y=0,1,\cdots,31$。

关于 addConstant 对旋转关系影响,可以按照如下方式确定:假如字 A 中满足旋转关系的概率为 $p_{(a,y)}$,$y=0,1,\cdots,31$。如果常数 C 的第 y bit 不为 0,则令

$$p_{(a,y)}:=1-p_{(a,y)}\quad p_{(a,y-n)}:=1-p_{(a,y-n)}$$

至此,已经解决了 Q_j 置换的所有变换环节对旋转概率 $p(x,y)$ 的影响,由此可以计算经过 Q_j 置换任意轮后 $p(x,y)$ 的值,根据是否存在 $p(x,y)\neq 0.5$ 确定是否能把某轮 Q_j 置换和随机置换进行区分。

2.3　Q_j 置换的 3-轮区分

对于 Q_j 置换,开始时对所有 $x\in\{0,1,\cdots,7\}$,$y\in\{0,1,\cdots,31\}$,令 $p(x,y)=0$,经过加常数变换环节一些 $p(x,y)$ 的值开始变化,并引起后续 $p(x,y)$ 值的变化。对于任意的旋转数 n 而言,经过 3 轮的变换,仍有一部分 $p(x,y)$ 的值偏离 0.5,从而存在 3 轮 Q_j 置换和随机置换的旋转区分;但经过 4 轮变换,所有的 $p(x,y)=0.5$,则不能对 4 轮 Q_j 置换和随机置换进行旋转区分。图 3 给出了旋转数 $n=1$ 时 3 轮 Q_0 置换的相应概率 $p(x,y)$ 的变化情况,其中每一轮变换前后 $p(x,y)$ 用 8×32 个小方块表示,第 x 行 y 列的方块颜色用于刻画 $p(x,31-y)$ 的值所在的相应区间,见电子版。

本文利用实验验证 3 轮 Q_j 置换与随机置换的区分是否与理论预测一致。例如,对于旋转数 $n=5$ 经过 3 轮 Q_0 的变换 $p(6,0)=0.495\ 223$,首先通过 Chernoff 界计算所需旋转对的最小个数 m,即

$$m\geqslant\frac{1}{(p-0.5)^2}\ln\frac{1}{\sqrt{\varepsilon}}$$

其中:ε 为检验出现误判的概率。从该不等式可得 $m=65\ 640$,实验中本文选取 $m=70\ 000$ 个旋转对进行实验。然后按照如下步骤进行区分:

a)计数器设为 0,随机生成 70 000 个旋转对;

b)对于每个旋转对 $(A,\overleftarrow{A})$,分别计算 3 轮 Q_0 的变换,如果 $Q_0(A)_{(6,0)}\neq Q_0(\overleftarrow{A})_{(6,5)}$,则计数器加 1。

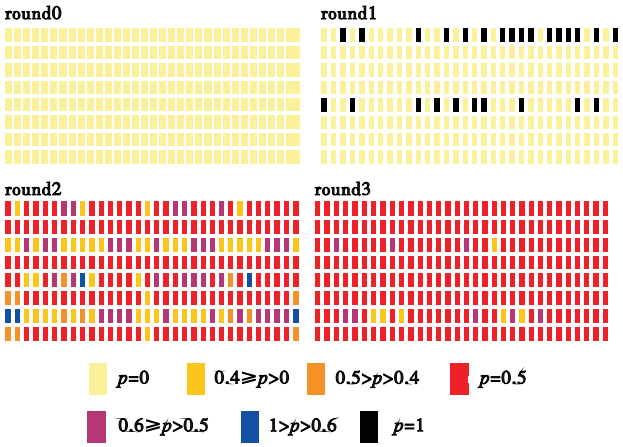

图3　3轮Q_0置换中概率$p(x,y)$的演变，其中旋转数$n=1$

实验结果符合本文的理论预测,计数器的值为 34668,非常接近理论值 $p(6,0)\times 70000=34665.61$;而对于随机置换而言,其计数器的值应非常接近 35000。同样本文对于其他 $p(x,y)$ 和旋转数 n 进行了大量的实验,实验结果符合本文的理论预期值,能够把 3 轮 Q_j 置换和随机置换进行有效区分。

3　结束语

本文提出了针对 Luffa 杂凑函数缩减轮置换的旋转攻击方法,能以约 2^{16} 的计算复杂度把 3 轮 Q_j 置换和随机置换进行有效区分,但对 8 轮 Luffa 杂凑函数的安全性尚未构成直接威胁。下一步的研究可以从两方面展开[9~12]:a)把旋转攻击与其他攻击方法相结合攻击更多轮 Luffa 置换;b)把对 Luffa 置换的攻击扩展到对整个 Luffa 杂凑函数的攻击。

参考文献:

[1] De CANNIERE C, SATO H, WATANABE D. Hash function Luffa: specification submission to NIST (round 1) [EB/OL]. 2008 (2008-10-31). http://ehash.iaik.tugraz.at/wiki/Luffa.

[2] De CANNIERE C, SATO H, WATANABE D. Hash function Luffa: specification submission to NIST (round 2) [EB/OL]. 2009 (2009-09-28). http://ehash.iaik.tugraz.at/wiki/Luffa.

[3] KHOVRATOVICH D, NIKOLIC I. Rotational cryptanalysis of ARX [C]//Proc of the 17th International Workshop on Fast Software Encryption. Berlin: Springer-Verlag, 2010: 333-346.

[4] KHOVRATOVICH D, NIKOLIC I, RCHBERGER C. Rotational rebound attacks on reduced Skein [C]//Proc of the 16th International Conference on Theory and Application of Cryptology and Information Security. 2010: 1-19.

(下转第 3869 页)

条件 4　夜间自然条件,待测者上方有两处白炽灯源,实验过程光照强度较弱。测量结果如表 2 所示。

表 2　四种不同条件测量准确率对比

测量条件	规定距离测量准确率 ρ/%			有效量程/cm($\rho \geq 95\%$)	单次平均耗时/ms
	30 cm	55 cm	70 cm		
条件 1	97.54	99.31	97.32	18 ~ 86	233
条件 2	97.11	98.92	97.04	21 ~ 81	241
条件 3	95.12	96.45	93.23	27 ~ 66	272
条件 4	96.29	97.94	95.21	24 ~ 73	253

通过实验 2 可以发现,系统的测量性能整体表现稳定,在各种测试条件下都完成了测距任务,且获得较高的测量准确率,随外界光照条件的改变而发生较大变化。在光照充足且无其他光源干扰的条件下,测量性能最好;当系统受外界光源干扰时,测量精度和量程都受较大影响,测量实时性也降低。

实验 3　为检验测量系统对不同待测者的适应性,选择六名待测者,分别标记为待测者#1 ~ #6,检验在 25 ~ 80 cm 范围内测量的精度,以验证测量系统对不同待测者的适应性和使用的普遍性。六名待测者如图 10 所示。

图10　六名待测者人脸图像

为保证测量的可靠性,本文对测量环境进行了约束,测量实验都规定在实验 2 中条件 1 下进行。其中前三个待测者没有佩戴眼镜,后三个待测者佩戴有框眼镜。测量结果如图 11 所示。

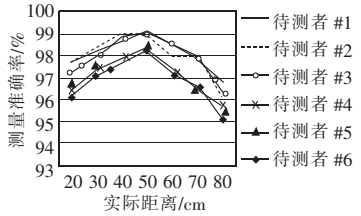

图 11　不同待测者测量准确率对比图

测量系统准确地捕捉了不同待测者的人脸特征信息,完成测距任务,使用普遍性较好。分析测量结果可知,佩戴眼镜的待测者测量准确率略低于未佩戴眼镜的测量者。但对于所有待测者来说,测量精度都保持在 95% 以上。

5　结束语

本文提出一种基于人脸特征区域像素面积的单目人机测距新方法。该方法结构简单,测量时不需要附加设备,测量具有较好的精度且能够满足实时性。系统首先通过图像预处理,将输入图像处理为满足系统要求的图像,其次通过精简特征数量和样本扩张策略改进传统的 AdaBoost 算法快速检测并定位特征区域,构建特征三角形。最后利用系统约束条件、摄像机标定原理和面积映射关系推导像素面积与人机距离的数学关系,建立像距测量方程,完成距离测量。通过规定距离实验、复杂环境实验和系统适应性实验,验证了测量系统的可行性、稳定性和普遍适用性。实验结果表明,系统测量精度较高,处理速度约为 5 fps,满足精度与实时性要求。

参考文献:

[1] 黄桂平,李广云,王保丰. 单目视觉测量技术研究[J]. 计量学报,2004,25(4):314-317.

[2] HSU Chen-chien, LU Ming-chil, WANG Wei-yen. Distance measurement based on pixel variation of CCD images[J]. ISA Trans, 2009, 48(1):389-395.

[3] LU Ming-chih, WANG Wei-yen, CHU Chun-yen. Image-based distance and area measuring systems[J]. IEEE Sensors Journal, 2006, 6(2):495-503.

[4] 王玮,黄非非,李见为. 采用 LBP 金字塔的人脸描述与识别[J]. 计算机辅助设计与图形学学报,2009,21(1):94-100.

[5] 陈文飞,廖斌,许雪峰,等. 基于 Piecewise 直方图均衡化的图像增强方法[J]. 通信学报,2011,32(9):153-158.

[6] VIOLA P, JONES M. Robust real-time face detection[J]. International Journal of Computer Vision, 2004, 57(2):137-154.

[7] 张彦峰,何佩琨. 一种改进的 AdaBoost 算法——M-Asy AdaBoost[J]. 北京理工大学学报,2011,31(1):64-68.

[8] 甘玲,朱江,苗东. 扩展 Haar 特征检测人眼的方法[J]. 电子科技大学学报,2010,39(2):247-250.

[9] 杜志军,王阳生. 正面人脸图像中眼睛的定位算法[J]. 计算机辅助设计与图形学学报,2009,21(6):763-769.

[10] 艾娟,姚丹,郭跃飞. 基于块的眼睛定位方法[J]. 中国图象图形学报,2007,12(10):1841-1844.

[11] 郝明刚,董秀成,黄亚勤. 一种精确的人眼瞳孔定位算法[J]. 计算机工程,2012,38(8):141-143.

[12] 敬泽,薛方正,李祖枢. 基于单目视觉的空间目标位置测量[J]. 传感器与微系统,2011,30(3):125-127.

[13] 朝廷祥,张志胜,戴敏. 用于目标测距的单目视觉测量方法[J]. 光学精密工程,2011,19(5):1111-1117.

[14] ZHANG Zheng-you. A flexible new technique for camera calibration[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2000, 22(11):1330-1334.

(上接第 3809 页)

[5] ALIZADEH J, MIRGHADRI A. A new distinguisher for CubeHash-8/b and CubeHash-15/b compression functions[J]. IJCSI International Journal of Computer Science Issues, 2011, 8(5):184-192.

[6] MORAWIECKI P, PIEPRZYK J, SREBRNY R. Rotational cryptanalysis of round-reduced KECCAK[EB/OB]. (2012-12-18). http://eprint.iacr.org/2012/546.

[7] WATANABE D, HATANO Y. Higher order differential attack on reduced round Luffa[R/OB]. (2010-11-19). http://eprint.iacr.org/2010/589.

[8] KHOVRATOVICH D, NAYA-PLASENCIA M, ROCK A, *et al*. Cryptanalysis of Luffa v2 components[C]//Proc of the 17th International Conference on Selected Areas in Cryptography. Berlin: Springer-Verlag, 2011:388-409.

[9] JIA Ke-ting, DESMEDT Y, HAN Li-dong, *et al*. Pseudo-cryptanalysis of Luffa [R/OB]. (2009-05-19). http://eprint.iacr.org/2009/224.

[10] PRENEEL B, YOSHIDA H, WATANABE D. Finding collisions for reduced Luffa-256 v2[C]//Proc of the 16th Australasian Conference on Information Security and Privacy. 2011:423-427.

[11] AUMASSON J P, MEIER W. Zero-sum distinguishers for reduced Keccak-*f* and for the core functions of Luffa and Hamsi[EB/OL]. (2009-09-09). http://131002.net/data/papers/AM09.pdf.

[12] KIRCANSKI A, YOUSSEF A M. Boomerang and slide-rotational analysis of the SM3 hash function [R/OB]. (2012-05-15). http://eprint.iacr.org/2012/274.