Speeding up Huff Form of Elliptic Curves

Neriman Gamze Orhon and Huseyin Hisil
gamzeorhon@gmail.com
huseyin.hisil@yasar.edu.tr

Yasar University, Izmir, Turkey

Abstract. This paper presents faster inversion-free point addition formulas for the curve $y(1+ax^2) = cx(1+dy^2)$. The proposed formulas improve the point doubling operation count record[1] from $6\mathbf{M}+5\mathbf{S}$ to $8\mathbf{M}$ and mixed addition operation count record from $10\mathbf{M}$ to $8\mathbf{M}$. Both sets of formulas are shown to be 4-way parallel, leading to an effective cost of $2\mathbf{M}$ per either of the group operations.

1 Introduction

Huff form of elliptic curves [9] were introduced to the crypto community in [10] at ANTS-IX, where Joye, Tibouchi and Vergnaud investigated several interesting features of Huff form for cryptographic applications. Their investigation provided: a more general curve model than Huff's original model, the explicit derivation of fast group operations, as well as formulas for pairing computation and, an extension to the even characteristic case. Wu and Feng [11] further extended the coverage to all elliptic curves having three points of order two.

Despite all the developments, we weren't able to find any statement about Huff form being faster than the other widely known forms of elliptic curves. This work presents new and faster sets of formulas for the Huff form and shows that Huff form can be competitive. The summary of contributions of this paper are as follows:

- The group operations of Huff form can be significantly accelerated when the curve is embedded in $\mathbb{P}^1 \times \mathbb{P}^1$ rather than $\mathbb{P}^2$. Plus, unlike the previous studies suggest, there is no need to move the identity to a point at infinity in $\mathbb{P}^1 \times \mathbb{P}^1$ to obtain faster formulas. See also Section 5.
- There exist an efficient 2-isogeny from a Huff curve to another Huff curve which can be used to speed-up the doubling operation.
- The basic group operations mixed-addition and doubling can be computed in parallel 4-way with the $\mathbb{P}^1 \times \mathbb{P}^1$ embedding.
- This work provides an evidence that quartic projections can be of interest when studying efficient group arithmetic on affine cubic curves.
- Huff form becomes competitive in performance, see Section 5.

Unfortunately, there is no consensus on how to write the curve equation for Huff form. The original Huff form was introduced as $ax(y^2-1) = by(x^2-1)$ by Huff in [9] and a twisted version to cover more elliptic curves was given as $ax(y^2-d) = by(x^2-d)$ by Joye, Tibouchi and Vergnaud in [10]. In a later work, the curve equation that covers even more elliptic curves was given as $x(ay^2-1) = y(bx^2-1)$ by Wu and Feng in [11]. Finally, the curve equation that covers equally many elliptic curves as Wu and Feng's equation was given in the form $ax(y^2-c) = by(x^2-d)$ by Aziz and Sow in [5]. In this work, the curve equation is given by $y(1+ax^2) = cx(1+dy^2)$. The reasons of this way of writing is explained as follows:

This project is funded by Yasar University Scientific Research Project SRP-024.

[1] $\mathbf{I}$, $\mathbf{M}$, $\mathbf{S}$, $\mathbf{D}$, $\mathbf{a}$ represents the cost of various field operations. $\mathbf{I}$: inversion, $\mathbf{M}$: multiplication, $\mathbf{S}$: squaring, $\mathbf{D}$: multiplication by a curve constant, $\mathbf{a}$: addition/subtraction.

- Wu and Feng's equation $x(ay^2 - 1) = y(bx^2 - 1)$ is a good start due to its extended coverage. We simply obverse that all other curve forms are always written with "+" in their curve equations in the literature. E.g. $ax^2 + y^2 = 1 + dx^2y^2$ (twisted Edwards form) [1], $y^2 = dx^4 + 2ax^2 + 1$ (extended Jacobi quartic form) [4], $ax^3 + y^3 + 1 = dxy$ (twisted Hessian form) [2], $y^2 = x^3 + ax + b$ (short Weierstrass form), etc. So, as the first step, writing the equation in the form $x(1+ay^2) = y(1+bx^2)$ complies better with the literature of other curve forms. This tweak does not effect the coverage of Wu and Feng's equation.
- In all speed oriented formulas, both a and b do appear with respect to the equation $x(1 + ay^2) = y(1 + bx^2)$. Therefore, it is desirable to keep these constants as small as possible. But then, the number of elliptic curves over some suitable field, becomes very limited in the context of cryptographic applications. The constants outside the parentheses of Aziz and Sow's equation is helpful in this sense, however, both constants clash with a and b of Wu and Feng's equation. We can write $cx(1+ay^2) = dy(1+bx^2)$ to prevent the clashing. However, having just c suffices since a point satisfying this equation also satisfies $(c/d)x(1 + ay^2) = y(1 + bx^2)$. Renaming c/d as c we have $cx(1 + ay^2) = y(1 + bx^2)$.
- Finally, we keep a as is and replace b with d since most curve forms use the letters a and d. The curve equation now reads $y(1 + ax^2) = cx(1 + dy^2)$. The right and left hand sides of the equation is swapped in order to have the curve constant appear in alphabetic order. To prevent ambiguation, the name of this equation is referred to as extended Huff form in this work. Extended Huff form has the same coverage as Wu and Feng's equation, i.e. extended Huff form covers all elliptic curves having three points of order two.

This text is organized as follows. Section 1 provides fundamental properties of extended Huff form elliptic curves. Section 2 presents necessary tweaks for point addition and point doubling formulas for extended Huff form. Section 3 presents fast and inversion-free point addition and point doubling formulas in $\mathbb{P}^1 \times \mathbb{P}^1$. Section 3 also presents new isogeny maps that becomes very handy in speeding up the doubling formulas. Section 4 shows how to schedule point operations at low level to allow a full speed 4-way parallel implementation. Section 5 makes comparisons with the literature and derives conclusions.

2 Extended Huff Curves

Let $\mathbb{K}$ be a field with $\mathrm{char}(\mathbb{K}) \neq 2$ and $a, c, d \in \mathbb{K}$ such that $acd(a - c^2d) \neq 0$. An extended Huff curve over $\mathbb{K}$ is a non-singular curve of the form

$$H_{a,c,d} : y(1 + ax^2) = cx(1 + dy^2). \tag{1}$$

The curve $H_{a,c,d}$ has j-invariant $256(a^2 - ac^2d + c^4d^2)^3/(ac^2d(a - c^2d))^2$. The subscripts are dropped when clear from the context.

Theorem 1. *Every elliptic curve having three points of order two is isomorphic over $\mathbb{K}$ to an extended Huff curve.*

Proof. The proof is very similar to that of Wu and Feng [11]. The proof is included for self containment. Every elliptic curve having three points of order two is isomorphic over $\mathbb{K}$ to a Weierstrass curve of the form $W : y^2 = x(x - e')(x - f')$. The curve W with identity element at $(0 : 1 : 0)$ is isomorphic over $\mathbb{K}$ to an extended Huff curve with identity element at $(0, 0)$ under the birational maps

$$\begin{aligned} \phi : H \to W, \quad (x, y) &\mapsto \Big(cx\frac{a - c^2d}{cx - y}, c\frac{a - c^2d}{cx - y}\Big), \\ \phi^{-1} : W \to H, \quad (x, y) &\mapsto \Big(\frac{x}{y}, c\frac{x - (a - c^2d)}{y}\Big) \end{aligned} \tag{2}$$

where $e' = a - c^2 d$ and $f' = -c^2 d$ with c being a free variable satisfying $acd(a - c^2 d) \neq 0$. $\square$

The three points of order two $(0, 0)$, $(e', 0)$, $(f', 0)$ on W corresponds to the three points at infinity $(0 : 1 : 0)$, $(1 : 0 : 0)$, $(cd : a : 0)$ on H, respectively. The negative of a point (x, y) on H is given by $(-x, -y)$.

The birational maps in Theorem 1 could have been further simplified if the curve constant c were rescaled to 1 by substituting x/c to x. However, we intentionally keep the track of c because it is computationally more advantageous to rescale a and d to "small"[2] elements. Let a' and d' be such "small" elements so that $a = \alpha^2 a'$ and $d = \delta^2 d'$ for some $\alpha, \delta \in \mathbb{K}$. The rescaling isomorphism reads

$$H_{a,c,d} \to H_{a', \frac{\delta}{\alpha} c, d'}, \quad (x, y) \mapsto (\alpha x, \delta y). \tag{3}$$

Explicit derivation of the point addition formulas for H from the chord-and tangent rule leads to cumbersome expressions. Step by step derivations are given in [10, Section 2.1] and [11, Section 4.1]. Nevertheless, the derived formulas can be simplified using the curve equation. The simplified addition formulas in affine coordinates are given as

$$\begin{aligned} \big(x_1, y_1\big) + \big(x_2, y_2\big) &= \left(\frac{(x_1 + x_2)(1 - d y_1 y_2)}{(1 - a x_1 x_2)(1 + d y_1 y_2)}, \frac{(1 - a x_1 x_2)(y_1 + y_2)}{(1 + a x_1 x_2)(1 - d y_1 y_2)} \right) && (4) \\ &= \left(\frac{(x_1 - x_2)(y_1 + y_2)}{(1 + a x_1 x_2)(y_1 - y_2)}, \frac{(x_1 + x_2)(y_1 - y_2)}{(x_1 - x_2)(1 + d y_1 y_2)} \right). && (5) \end{aligned}$$

The first set of formulas is called the unified addition formulas while the second set is called the dedicated addition formulas. The simplified doubling formulas in affine coordinates are given as

$$[2](x_1, y_1) = \left(\frac{2 x_1 (1 - d y_1^2)}{(1 - a x_1^2)(1 + d y_1^2)}, \frac{2 y_1 (1 - a x_1^2)}{(1 + a x_1^2)(1 - d y_1^2)} \right). \tag{6}$$

All three sets of formulas (4), (5), and (6) are obtained from [10, 11, 5] with minor tweaks for the positioning of the curve constants and sign tweaks. Therefore, we do not claim credits on (4), (5), and (6). On the other hand, studying their properties in $\mathbb{P}^1 \times \mathbb{P}^1$ first appears in this work. Our results are given in the following section.

3 Embedding of H into $\mathbb{P}^1 \times \mathbb{P}^1$

The projective closure of H in $\mathbb{P}^1 \times \mathbb{P}^1$ is given by

$$\mathcal{H} = \big\{ \big((X : Z), (Y : T)\big) \in \mathbb{P}^1 \times \mathbb{P}^1 : YT(Z^2 + aX^2) = cXZ(T^2 + dY^2) \big\}. \tag{7}$$

A point (x, y) on H maps to $((x : 1), (y : 1))$ on $\mathcal{H}$. The point $((x : 1), (y : 1))$ can be represented by $((\lambda x : \lambda), (\delta y : \delta))$ for any nonzero $\lambda, \delta \in \mathbb{K}$ in the usual way. The identity element is $((0 : 1), (0 : 1))$. The negative of a point $((X : Z), (Y : T))$ on $\mathcal{H}$ is $((-X : Z), (-Y : T))$. All points $((X : Z), (Y : T))$ on $\mathcal{H}$ other than the points at infinity, corresponds to the point $(XT : YZ : TZ)$ in homogeneous projective coordinates. The three points at infinity $((0 : 1), (1 : 0))$, $((1 : 0), (0 : 1))$, $((1 : 0), (1 : 0))$ on $\mathcal{H}$ corresponds to the three points of order two $(0 : 1 : 0)$, $(1 : 0 : 0)$, $(cd : a : 0)$ in homogeneous projective coordinates as given in the proof of Theorem 1, respectively. The unified addition formulas correspond to the following in $\mathbb{P}^1 \times \mathbb{P}^1$ setting: $\big((X_1 : Z_1), (Y_1 : T_1)\big) + \big((X_2 : Z_2), (Y_2 : T_2)\big) =$

$$\begin{aligned} \Big(& \big((X_1 Z_2 + Z_1 X_2)(T_1 T_2 - d Y_1 Y_2) : (Z_1 Z_2 - a X_1 X_2)(T_1 T_2 + d Y_1 Y_2)\big), \\ & \big((Z_1 Z_2 - a X_1 X_2)(Y_1 T_2 + T_1 Y_2) : (Z_1 Z_2 + a X_1 X_2)(T_1 T_2 - d Y_1 Y_2)\big) \Big) \end{aligned} \tag{8}$$

[2] Here, "small" refers to any distinguished element for which multiplication with other field elements is significantly faster than the usual multiplication of two arbitrary elements in $\mathbb{K}$.

whereas the dedicated addition formulas correspond to

$$\Big(\big((X_1Z_2 - Z_1X_2)(Y_1T_2 + T_1Y_2) : (Z_1Z_2 + aX_1X_2)(Y_1T_2 - T_1Y_2)\big), \\ \big((X_1Z_2 + Z_1X_2)(Y_1T_2 - T_1Y_2) : (X_1Z_2 - Z_1X_2)(T_1T_2 + dY_1Y_2)\big) \Big). \tag{9}$$

Both sets of formulas are of lower total degrees when considered as polynomial expressions, in comparison with the corresponding formulas in [10, 11, 5]. Similar comments apply for the projective doubling formulas which correspond to the following in $\mathbb{P}^1 \times \mathbb{P}^1$ setting: $[2]((X_1 : Z_1), (Y_1 : T_1)) =$

$$\Big(\big(2X_1Z_1(T_1^2 - dY_1^2) : (Z_1^2 - aX_1^2)(T_1^2 + dY_1^2)\big), \\ \big(2Y_1T_1(Z_1^2 - aX_1^2) : (Z_1^2 + aX_1^2)(T_1^2 - dY_1^2)\big) \Big). \tag{10}$$

The unified addition and dedicated addition take $10\mathbf{M}+2\mathbf{D}+12\mathbf{a}$ and $10\mathbf{M}+4\mathbf{D}+13\mathbf{a}$, respectively. Both operation counts improve upon the previous best records $11\mathbf{M}+14\mathbf{a}$ in [10] and $11\mathbf{M}+3\mathbf{D}+14\mathbf{a}$ in [11]. The unified mixed-addition and dedicated mixed-addition (i.e. $Z_2 = T_2 = 1$) both take $8\mathbf{M}+2\mathbf{D}+6\mathbf{a}$. This operation count improves upon the previous best record $10\mathbf{M}+14\mathbf{a}$ in [10]. The doubling takes $4\mathbf{M}+6\mathbf{S}+2\mathbf{D}+10\mathbf{a}$ with a plain operation count which improves upon the previous best operation counts $10\mathbf{M}+1\mathbf{S}+14\mathbf{a}$ in [10] and $6\mathbf{M}+5\mathbf{S}+3\mathbf{D}+12\mathbf{a}$ in [11]. See Section 4 for justifications of our claimed operation counts.

We extend our investigation to 2-isogeny maps to find further improvements in the following subsection in order to investigate further speedup options for the point doubling operation.

3.1 Isogeny to an extended Huff curve

Moody and Shumov [6, Section 5] derived a 2-isogeny from a Huff form elliptic curve to another Huff form elliptic curve. We found in our investigation that more speed-oriented isogenies can be derived. The following theorem states this isogeny and its dual explicitly.

Theorem 2. *In accordance with Theorem 1, let $a, c, d, r \in \mathbb{K}$ satisfy*

$$acd(a - c^2d) \neq 0, \quad r^2 = ad.$$

Then, the curve

$$H : \; y(1 + ax^2) = cx(1 + dy^2)$$

is 2-isogenous over $\mathbb{K}$ to the extended Huff curve

$$G : y(1 - ax^2) = \left(\frac{a - cr}{a + cr}\right) x(1 - ay^2).$$

The 2-isogeny and its dual are given explicitly as follows:

$$\varphi : \; H \to G, \quad (x, y) \mapsto \left(\frac{x + \frac{r}{a}y}{1 + rxy}, \frac{x - \frac{r}{a}y}{1 - rxy}\right),$$
$$\hat{\varphi} : \; G \to H, \quad (x, y) \mapsto \left(\frac{x + y}{1 - axy}, \frac{x - y}{1 + axy} \cdot \frac{a}{r}\right).$$

Proof. We will first show that $\Delta_G \neq 0$, hence G is a Huff form elliptic curve, by using the inequality $\Delta_H = acd(a - c^2d) \neq 0$. We have $a - c^2d \neq 0$ and $c \neq 0$ since $\Delta_H \neq 0$. It follows that $(a/c)^2 \neq ad = r^2$. So, $r \neq \pm a/c$. We have, $a \pm cr \neq 0$. So, $(a - cr)/(a + cr) \neq 0$. We also

have, $(-a)((-a)-((a-cr)/(a+cr))^2(-a) = 2acr/(a+cr) \neq 0$ since $r \neq 0$ and $a+cr \neq 0$. Therefore,

$$\Delta_G = (-a)\left(\frac{a-cr}{a+cr}\right)(-a)\left((-a)-\left(\frac{a-cr}{a+cr}\right)^2(-a)\right) \neq 0.$$

Setting $u = \dfrac{x+\frac{r}{a}y}{1+rxy}$ and $v = \dfrac{x-\frac{r}{a}y}{1-rxy}$, we have $v(1-au^2) - \left(\dfrac{a-cr}{a+cr}\right)u(1-av^2) =$

$$\frac{-2r(1-ax^2)(a-r^2y^2)(ay(1+ax^2)-cx(a+r^2y^2))}{a^2(a+cr)(1-r^2x^2y^2)^2}.$$

By replacing r^2 with ad and organizing the terms, we get

$$\Big(y(1+ax^2)-cx(1+dy^2)\Big)\cdot\frac{-2r(1-ax^2)(1-dy^2)}{(a+cr)(1-adx^2y^2)^2}$$

which shows that φ is a rational map from H to G. In addition, $\varphi((0,0)) = (0,0)$. Therefore, φ is an isogeny from H to G.

Setting $u = \dfrac{x+y}{1-axy}$ and $v = \dfrac{a}{r}\cdot\dfrac{x-y}{1+axy}$, we have $v(1+au^2) - u(1+dv^2) =$

$$-\Big(c(x+y)(1-axy)(1+axy)^2/(r-ra^2x^2y^2)^2\Big)r^2+$$
$$\Big(a(x-y)(1+axy)(1+ax^2)(1+ay^2)/(r-ra^2x^2y^2)^2\Big)r-$$
$$\Big(a^2cd(x+y)(1-axy)(x-y)^2/(r-ra^2x^2y^2)^2\Big).$$

By replacing d with r^2/a and organizing the terms, we get

$$\left(y(1-ax^2)-\left(\frac{a-cr}{a+cr}\right)x(1-ay^2)\right)\cdot\frac{(a+cr)(1+ax^2)(1+ay^2)}{-r(1-a^2x^2y^2)^2}$$

which shows that $\hat{\varphi}$ is a rational map from G to H. In addition, $\hat{\varphi}((0,0)) = (0,0)$. Therefore, $\hat{\varphi}$ is an isogeny from G to H.

It remains to show that $\hat{\varphi}\circ\varphi = [2]_H \in \mathbb{K}(H)$ and $\varphi\circ\hat{\varphi} = [2]_G \in \mathbb{K}(G)$. We readily have, $\hat{\varphi}\circ\varphi =$

$$\left(\frac{-2xy(ax-cdy)(ax^2-2cdxy+dy^2)}{(ax^2-dy^2)(acx^2-2axy+cdy^2)}, \frac{-2xy(ax-cdy)(acx^2-2axy+cdy^2)}{c(ax^2-dy^2)(ax^2-2cdxy+dy^2)}\right).$$

To see that these formulas give $[2]_H \in \mathbb{K}(H)$, eliminate c using $c = y(1+ax^2)/(x(1+dy^2)$ which is obtained by the curve equation for H, and then, factor the remaining expression. Similarly, $\varphi\circ\hat{\varphi} = [2]_G \in \mathbb{K}(E)$ can be verified by eliminating c from the expanded $\varphi\circ\hat{\varphi}$, using the relation

$$c = -\frac{ay(1-ax^2)-ax(1-ay^2)}{ry(1-ax^2)+rx(1-ay^2)}$$

which is obtained by the curve equation for G, and then eliminating d using the relation $d = r^2/a$. See the full version of this work for a constructive proof to obtain φ and $\hat{\varphi}$. □

The maps φ and $\hat{\varphi}$ takes a particularly simple form for twisted Huff curves i.e. when $a = d$ and so, $r = \pm a$. We will use this setting in the remainder of the text for efficiency purposes. Taking $r = a$ and noting the $\mathbb{P}^1 \times \mathbb{P}^1$ embedding of G as

$$\mathcal{G}:\ YT(Z^2-aX^2) = \left(\frac{1-c}{1+c}\right)XZ(T^2-aY^2),$$

the projective 2-isogeny $\varphi_{\mathcal{H}} : \mathcal{H} \to \mathcal{G}$ and its dual $\hat{\varphi}_{\mathcal{G}} : \mathcal{G} \to \mathcal{H}$ are given as

$$\begin{aligned}\big((X:Z),(Y:T)\big) &\mapsto \big((XT+YZ:TZ+aXY),(XT-YZ:TZ-aXY)\big),\\ \big((X:Z),(Y:T)\big) &\mapsto \big((XT+YZ:TZ-aXY),(XT-YZ:TZ+aXY)\big),\end{aligned} \tag{11}$$

respectively. The kernel of $\varphi_{\mathcal{H}}$ is $\big\{\big((0:1),(0:1)\big),\big((1:0),(1:0)\big)\big\} \subseteq \mathcal{H}$. The kernel of $\hat{\varphi}_{\mathcal{G}}$ is $\big\{\big((0:1),(0:1)\big),\big((1:0),(1:0)\big)\big\} \subseteq \mathcal{G}$. An algorithm to evaluate $[2]_{\mathcal{H}} = \hat{\varphi}_{\mathcal{G}} \circ \varphi_{\mathcal{H}}$ at points of $\mathcal{H}$ is provided in the next section. See also Section 4 for further justifications.

We note that both $\mathcal{H}$ and $\mathcal{G}$ are non-singular. The maps $\varphi_{\mathcal{H}}$ and $\hat{\varphi}_{\mathcal{G}}$ can be used to accelerate the $4\mathbf{M}+6\mathbf{S}+2\mathbf{D}+12\mathbf{a}$ doubling formulas in Section 3. In particular, doubling on $\mathcal{H}$ now takes $8\mathbf{M}+2\mathbf{D}+8\mathbf{a}$ if $d = a$, with the new set of formulas. See Section 4 for further justifications.

4 Efficient computation

This section presents efficient algorithms and operation counts. The 4-way parallel algorithms are central to this work. The algorithms in this section are designed as to allow working destructively on registers X_1, Z_1, Y_1, and T_1. Specifically, X_3, Z_3, Y_3, and T_3 are allowed to be the same registers as X_1, Z_1, Y_1, and T_1, respectively. The registers R_i are temporary registers and C_i are cache registers.

In the following algorithms, a, c, d are arbitrary unless stated otherwise. The constants always satisfy $acd(a - c^2d) \neq 0$. We also assume that the inputs are carefully selected so that the output is defined. The output is always defined for doubling and unified addition formulas if the arithmetic is restricted to odd order subgroup. This result can be deduced from [10, Corollary 1]. The output is always defined for dedicated addition formulas if the arithmetic is restricted to odd order subgroup and the input operands do not represent the same point.

The first two algorithms make the assumption that $a = d = 1$. This assumption implies that the least possible cofactor is 8.

- Doubling with $a = d = 1$ via 2-isogeny decomposition (11) takes $8\mathbf{M}+8\mathbf{a}$. The following is a 4-way parallel algorithm with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot T_1,$	$R_1 := Y_1 \cdot Z_1,$	$R_2 := T_1 \cdot Z_1,$	$R_3 := X_1 \cdot Y_1,$
$X_3 := R_0 + R_1,$	$Y_3 := R_0 - R_1,$	$T_3 := R_2 - R_3,$	$Z_3 := R_2 + R_3,$
$R_0 := X_3 \cdot T_3,$	$R_1 := Y_3 \cdot Z_3,$	$R_2 := T_3 \cdot Z_3,$	$R_3 := X_3 \cdot Y_3,$
$X_3 := R_0 + R_1,$	$Y_3 := R_0 - R_1,$	$Z_3 := R_2 - R_3,$	$T_3 := R_2 + R_3.$

- Unified addition with $a = d = 1$ and $Z_2 = T_2 = 1$ takes $8\mathbf{M}+6\mathbf{a}$. The following is a 4-way parallel algorithm with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot X_2,$	$R_1 := Z_1 \cdot X_2,$	$R_2 := Y_1 \cdot Y_2,$	$R_3 := T_1 \cdot Y_2,$
$R_4 := Z_1 + R_0,$	$R_1 := X_1 + R_1,$	$R_5 := T_1 + R_2,$	$R_3 := Y_1 + R_3,$
$R_0 := Z_1 - R_0,$	*idle*	$R_2 := T_1 - R_2,$	*idle*
$Z_3 := R_0 \cdot R_5,$	$X_3 := R_1 \cdot R_2,$	$T_3 := R_2 \cdot R_4,$	$Y_3 := R_3 \cdot R_0.$

The following two algorithms make the assumption that $a = d = 2$. This assumption implies that the least possible cofactor is 4.

- Doubling with $a = d = 2$ via 2-isogeny decomposition (11) takes $8\mathbf{M}+10\mathbf{a}$. The following is a 4-way parallel algorithm with cost $4 \times (2\mathbf{M} + 4\mathbf{a})$:

idle	*idle*	*idle*	$R_4 := X_1 + X_1$
$R_0 := X_1 \cdot T_1,$	$R_1 := Y_1 \cdot Z_1,$	$R_2 := T_1 \cdot Z_1,$	$R_3 := R_4 \cdot Y_1,$
$X_3 := R_0 + R_1,$	$Y_3 := R_0 - R_1,$	$T_3 := R_2 - R_3,$	$Z_3 := R_2 + R_3,$
idle	*idle*	*idle*	$R_4 := X_3 + X_3$
$R_0 := X_3 \cdot T_3,$	$R_1 := Y_3 \cdot Z_3,$	$R_2 := T_3 \cdot Z_3,$	$R_3 := R_4 \cdot Y_3,$
$X_3 := R_0 + R_1,$	$Y_3 := R_0 - R_1,$	$Z_3 := R_2 - R_3,$	$T_3 := R_2 + R_3.$

– Unified addition with $a = d = 2$ and $Z_2 = T_2 = 1$ takes 8**M**+8**a**. The following is a 4-way parallel algorithm with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot X_2,$	$R_1 := Z_1 \cdot X_2,$	$R_2 := Y_1 \cdot Y_2,$	$R_3 := T_1 \cdot Y_2,$
$R_0 := R_0 + R_0,$	$R_1 := R_1 + X_1,$	$R_2 := R_2 + R_2,$	$R_3 := R_3 + Y_1,$
$R_4 := R_0 + Z_1,$	$R_7 := R_0 - Z_1,$	$R_6 := R_2 - T_1,$	$R_5 := R_2 + T_1,$
$T_3 := R_4 \cdot R_6,$	$Y_3 := R_7 \cdot R_3,$	$X_3 := R_6 \cdot R_1,$	$Z_3 := R_5 \cdot R_7.$

– Doubling (10) takes 4**M**+6**S**+2**D**+10**a**:

$R_0 := X_1^2,\ X_3 := X_1 + Z_1,\ X_3 := X_3^2,\ X_3 := X_3 - R_0,\ Z_3 := Z_1^2,\ X_3 := X_3 - Z_3,$
$R_0 := aR_0,\ R_1 := Z_3 + R_0,\ Z_3 := Z_3 - R_0,\ R_0 := Y_1^2,\ Y_3 := Y_1 + T_1,\ Y_3 := Y_3^2,$
$Y_3 := Y_3 - R_0,\ T_3 := T_1^2,\ Y_3 := Y_3 - T_3,\ R_0 := dR_0,\ R_2 := T_3 + R_0,\ T_3 := T_3 - R_0,$
$X_3 := X_3 \cdot T_3,\ Y_3 := Y_3 \cdot Z_3,\ Z_3 := Z_3 \cdot R_2,\ T_3 := T_3 \cdot R_1.$

– Doubling exploiting the 2-isogeny decomposition (11) with $a = d$, takes 8**M**+2**D**+8**a**:

$R_0 := aX_1,\ X_3 := X_1 \cdot T_1,\ R_0 := R_0 \cdot Y_1,\ T_3 := T_1 \cdot Z_1,\ Y_3 := Y_1 \cdot Z_1,\ Z_3 := T_3 + R_0,$
$T_3 := T_3 - R_0,\ R_0 := X_3 + Y_3,\ Y_3 := X_3 - Y_3,\ X_3 := aR_0,\ X_3 := X_3 \cdot Y_3,$
$R_0 := R_0 \cdot T_3,\ T_3 := T_3 \cdot Z_3,\ Y_3 := Y_3 \cdot Z_3,\ Z_3 := T_3 - X_3,\ T_3 := T_3 + X_3,$
$X_3 := R_0 + Y_3,\ Y_3 := R_0 - Y_3.$

– Unified readdition with $Z_2 = T_2 = 1$, takes 8**M**+6**a** if $C_0 := aX_2$ and $C_1 := dY_2$ is precomputed and cached. The following is a 4-way parallel algorithm when similar operations are grouped, with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot C_0,\ R_1 := Z_1 \cdot X_2,\ R_2 := Y_1 \cdot C_1,\ R_3 := T_1 \cdot Y_2,\ R_4 := Z_1 + R_0,$
$R_1 := X_1 + R_1,\ R_5 := T_1 + R_2,\ R_3 := Y_1 + R_3,\ R_0 := Z_1 - R_0,\ R_2 := T_1 - R_2,$
$Z_3 := R_0 \cdot R_5,\ X_3 := R_1 \cdot R_2,\ T_3 := R_2 \cdot R_4,\ Y_3 := R_3 \cdot R_0.$

– Unified readdition takes 10**M**+2**D**+10**a** if C_0 and C_1 are precomputed and reused in the below algorithm. Unified addition takes 10**M**+2**D**+12**a**:

$C_0 := X_2 + Z_2,\ C_1 := Y_2 + T_2,\ R_0 := X_1 \cdot X_2,\ R_1 := Y_1 \cdot Y_2,\ R_2 := Z_1 \cdot Z_2,$
$R_3 := T_1 \cdot T_2,\ X_3 := X_1 + Z_1,\ Y_3 := Y_1 + T_1,\ R_4 := R_0 + R_2,\ R_5 := R_1 + R_3,$
$X_3 := X_3 \cdot C_0,\ Y_3 := Y_3 \cdot C_1,\ Z_3 := aR_0,\ T_3 := dR_1,\ R_0 := R_3 + T_3,\ T_3 := R_3 - T_3,$
$R_1 := R_2 + Z_3,\ Z_3 := R_2 - Z_3,\ X_3 := X_3 - R_4,\ Y_3 := Y_3 - R_5,\ X_3 := X_3 \cdot T_3,$
$Y_3 := Y_3 \cdot Z_3,\ Z_3 := Z_3 \cdot R_0,\ T_3 := T_3 \cdot R_1.$

– Dedicated addition with $a = d = 1$ and $Z_2 = T_2 = 1$ takes 8**M**+6**a**. The following is a 4-way parallel algorithm when similar operations are grouped, with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot X_2,\ R_1 := Z_1 \cdot X_2,\ R_2 := Y_1 \cdot Y_2,\ R_3 := T_1 \cdot Y_2,\ R_0 := Z_1 + R_0,$
$R_4 := X_1 + R_1,\ R_2 := T_1 + R_2,\ R_5 := Y_1 + R_3,\ R_1 := X_1 - R_1,\ R_3 := Y_1 - R_3,$
$Z_3 := R_0 \cdot R_3,\ X_3 := R_1 \cdot R_5,\ T_3 := R_2 \cdot R_1,\ Y_3 := R_3 \cdot R_4.$

– Dedicated readdition with $Z_2 = T_2 = 1$ takes 8**M**+6**a** if $C_0 := aX_2$ and $C_1 := dY_2$ is precomputed and cached. The following is a 4-way parallel algorithm when similar operations are grouped, with cost $4 \times (2\mathbf{M} + 2\mathbf{a})$:

$R_0 := X_1 \cdot C_0,\ R_1 := Z_1 \cdot X_2,\ R_2 := Y_1 \cdot C_1,\ R_3 := T_1 \cdot Y_2,\ R_0 := Z_1 + R_0,$
$R_4 := X_1 + R_1,\ R_2 := T_1 + R_2,\ R_5 := Y_1 + R_3,\ R_1 := X_1 - R_1,\ R_3 := Y_1 - R_3,$
$Z_3 := R_0 \cdot R_3,\ X_3 := R_1 \cdot R_5,\ T_3 := R_2 \cdot R_1,\ Y_3 := R_3 \cdot R_4.$

– Dedicated readdition takes 10**M**+2**D**+11**a** if $C_0 = Z_2 - aX_2$ and $C_1 = T_2 - dY_2$ are precomputed and reused in the below algorithm. Dedicated addition takes 10**M**+4**D**+13**a**:

$C_0 := aX_2$, $C_0 := Z_2 - C_0$, $C_1 := dY_2$, $C_1 := T_2 - C_1$, $R_0 := Y_1 \cdot T_2$, $R_1 := T_1 \cdot Y_2$, $R_2 := dR_1$, $R_2 := R_2 + R_0$, $T_3 := T_1 - Y_1$, $T_3 := C_1 \cdot T_3$, $T_3 := T_3 + R_2$, $Y_3 := R_0 - R_1$, $R_2 := X_1 \cdot Z_2$, $R_0 := R_0 + R_1$, $R_1 := Z_1 \cdot X_2$, $Z_3 := Z_1 - X_1$, $Z_3 := C_0 \cdot Z_3$, $Z_3 := Z_3 + R_2$, $X_3 := R_2 - R_1$, $X_3 := X_3 \cdot R_0$, $R_0 := aR_1$, $R_0 := R_0 + Z_3$, $Z_3 := Y_3 \cdot R_0$, $R_0 := R_2 + R_1$, $Y_3 := Y_3 \cdot R_0$, $R_2 := R_2 - R_1$, $T_3 := T_3 \cdot R_2$.

5 Comparison and conclusion

This work showed how to speed up the extended Huff form elliptic curves by embedding the curve in projective space $\mathbb{P}^1 \times \mathbb{P}^1$. Table 1 compares the results of Section 4 with the literature. Our formulas set the new operation count record for each group operation in Huff form.

Table 1: Speed oriented operation counts for Huff form

Source & the curve equation	h	**DBL**	**muADD**	**uADD**
Wu, Feng [11] plus assuming $b = 1$, $X(aY^2 - Z^2) = Y(X^2 - Z^2)$	4	6**M**+5**S**+1**D**+12**a**	10**M**+1**D**+14**a**	11**M**+1**D**+14**a**
Joye, Tibouchi, Vergnaud [10], $aX(Y^2 - Z^2) = bY(X^2 - Z^2)$	8	6**M**+5**S**+13**a**	10**M**+14**a**	11**M**+14**a**
This work, $YT(Z^2 + 2X^2) = cXZ(T^2 + 2Y^2)$	4	8**M**+10**a** 4×(2**M**+4**a**)	8**M**+8**a** 4×(2**M**+2**a**)	10**M**+14**a**
This work, $YT(Z^2 + X^2) = cXZ(T^2 + Y^2)$	8	8**M**+8**a** 4×(2**M**+2**a**)	8**M**+8**a** 4×(2**M**+2**a**)	10**M**+12**a**

– The column h represents the least possible cofactor in the given curve model.
– **DBL**, **muADD**, and **uADD** refer to doubling, unified addition with $Z_2 = 1$, and unified addition operations, respectively. We stress the additional condition $T_2 = 1$ for **muADD** in the last two entries of Table 1.
– The counts for **a** do not appear in the reference works. We counted them without eliminating the common subexpressions.
– The best operation counts in the reference works are reported by selecting the identity element as $(0 : 1 : 0)$. In contrast, our formulas do not require moving the identity to a point at infinity.
– The operation count for the twisted Huff curve $aX(Y^2 - dZ^2) = bY(X^2 - dZ^2)$ is not provided in [10]. Therefore, this case is missing in Table 1.

In addition, the most common group operations, doubling and mixed addition, are shown to be 4-way parallelizable. The presented formulas are expected to be attractive for parallel processing (e.g. special hardware, SIMD, video card settings). There is a need for further investigation in this direction.

It is tempting to compare the performance of the new formulas with the fastest formulas developed for twisted Edwards curves for which efficient 4-way parallel algorithms are given in [8]. The presented formulas gets competitive with twisted Edwards curves in the 4-way parallel setting. In the fastest scenario, twisted Edwards doubling takes $4 \times (1\mathbf{M} + 1\mathbf{S})$ where extended Huff doubling takes $4 \times 2\mathbf{M}$. Both forms should give similar performance if

$\mathbf{M} = \mathbf{S}$. The fastest 4-way parallel mixed addition takes $4 \times 2\mathbf{M}$ in both forms. Therefore, if double-and-add scalar multiplication algorithm is used and $\mathbf{M} = \mathbf{S}$, then both curve models are expected to give similar performance. One advantage of twisted Edwards curves is that the conversion of a projective point $(X : Y : T : Z)$ to the affine point $(X/Z, Y/Z)$ takes $\mathbf{I} + 2\mathbf{M}$. The conversion of a projective point $((X : Z), (Y : T))$ on an extended Huff curve to the affine point $(X/Z, Y/T)$ takes $\mathbf{I} + 5\mathbf{M}$ using Montgomery's simultaneous inversion technique [7]. However, the performance difference should be minor since $\mathbf{I}$ is many times more costly than $\mathbf{M}$. Yet, we do not disguise the fact that twisted Edwards will always win in sequential implementations especially in windowed scalar multiplications.

$\mathbb{P}^1 \times \mathbb{P}^1$ embedding of twisted Edwards curves and explicit group law formulas are given in [3] in rather a different context, without operation counts. Therefore, it is not clear how the two compare with each other. We also leave further comparisons with other forms to the reader.

References

1. D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters. Twisted Edwards curves. In *AFRICACRYPT 2008 proceedings*, volume 5023 of *LNCS*, pages 389–405. Springer, 2008.
2. D. J. Bernstein, C. Chuengsatiansup, D. Kohel, and T. Lange. Twisted Hessian curves. In *Progress in Cryptology LATINCRYPT 2015 proceedings*, volume 9230, pages 269–294. Springer International Publishing, 2015.
3. D. J. Bernstein and T. Lange. A complete set of addition laws for incomplete Edwards curves. *Journal of Number Theory*, 131(5):858–872, 2011.
4. O. Billet and M. Joye. The Jacobi model of an elliptic curve and side-channel analysis. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 15th International Symposium, AAECC-15, Toulouse, France, May 12-16, 2003 proceedings*, volume 2643, pages 34–42. Springer Berlin Heidelberg, 2003.
5. A. A. Ciss and D. Sow. On a new generalization of Huff curves. Cryptology ePrint Archive, Report 2011/580, 2011. http://eprint.iacr.org/2011/580.
6. D. S. Dustin Moody. Analogues of Velu's formulas for isogenies on alternate models of elliptic curves. Cryptology ePrint Archive, Report 2011/430, 2001. http://eprint.iacr.org/2011/430.
7. D. Hankerson, A. J. Menezes, and S. A. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
8. H. Hisil, K. K.-H. Wong, G. Carter, and E. Dawson. Twisted Edwards curves revisited. In *ASIACRYPT 2008 proceedings*, volume 5350 of *LNCS*, pages 326–343. Springer, 2008.
9. G. B. Huff. Diophantine problems in geometry and elliptic ternary forms. *Duke Mathematical Journal*, 15(2):443–453, 06 1948.
10. M. Joye, M. Tibouchi, and D. Vergnaud. Huff's model for elliptic curves. In *Algorithmic Number Theory: 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010 proceedings*, volume 6197, pages 234–250. Springer Berlin Heidelberg, 2010.
11. H. Wu and R. Feng. Elliptic curves in Huff's model. *Wuhan University Journal of Natural Sciences*, 17(6):473–480, 2012.