Journal of Convex Analysis
Volume 14 (2007), No. 3, 657–666

Parametric Computation of the Legendre-Fenchel Conjugate with Application to the Computation of the Moreau Envelope*

Jean-Baptiste Hiriart-Urruty
Laboratoire de Mathématiques pour l'Industrie et la Physique,
Institut de Mathématiques, Université Paul Sabatier,
118 route de Narbonne, 31 062 Toulouse Cedex, France
jbhu@cict.fr

Yves Lucet
Computer Science, I. K. Barber School of Arts and Sciences,
University of British Columbia Okanagan, 3333 University Way,
Kelowna BC V1V 1V7, Canada
yves.lucet@ubc.ca

Received: April 28, 2006

A new algorithm, named the Parametric Legendre Transform (PLT) algorithm, to compute the Legendre-Fenchel conjugate of a convex function of one variable is presented. It returns a parameterization of the graph of the conjugate except for some affine parts corresponding to nondifferentiable points of the function. The approach is extended to the computation of the Moreau envelope, resulting in a simple yet efficient algorithm.

Theoretical results, the description (and extension) of the algorithm, its approximation error and the convergence, as well as the comparison with known algorithms are included.

Keywords: Legendre-Fenchel transform, Fenchel conjugate, Moreau envelope, Moreau-Yosida regularization, Fast algorithm, Computational convex analysis

2000 Mathematics Subject Classification: 65Y20, 52A41

1. Introduction

In convex duality the Legendre-Fenchel transform, which associates with a convex extended-valued function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ its Legendre-Fenchel conjugate

$$f^*(s) = \sup_{x \in \mathbb{R}^d} \langle s, x \rangle - f(x)$$

allows to relate primal optimization problems of the type

$$p = \inf_{x \in \mathbb{R}^d} f(x) + g(Ax)$$

where $A \in \mathbb{R}^{n \times d}$ and f, g are extended-valued proper lower semi-continuous (lsc) convex functions, with the dual problem

$$d = \sup_{x \in \mathbb{R}^m} -f^*(-A^T z) - g^*(-z),$$

*This work was partly supported by the second author NSERC Discovery grant.

ISSN 0944-6532 / $ 2.50

(where A^T denotes the transpose of A) by Fenchel's duality theorem [5, Theorem 3.3.5]. Convex duality encompasses linear programming duality, the Min-Max Theorem from game theory, and also allows to state duality results for semi-definite optimization problems. Earlier work on duality used the notion of Legendre transform as detailed in [21, Section 26].

Recent work has focused on the numerical computation of the Legendre-Fenchel conjugate. Symbolic computation was explored in [3, 2, 10] where a specific class of functions was identified. It relies on inverting the gradient mapping with recent advances focusing on extension to multidimensional functions. When symbolic computation fails, or when the function is only known through a black box or at specific points, numerical algorithms still allow the computation of an approximation of the conjugate. Work in that area started with the Fast Legendre Algorithm (FLT) [6, 16], and was later improved with the Linear-time Legendre Transform (LLT) algorithm [17]. Recent work [18] noticed the close relationship between the Fenchel conjugate and the Moreau envelope

$$M_\lambda(s) = \inf_{x\in\mathbb{R}^d} f(x) + \frac{\|s-x\|^2}{2\lambda},$$

resulting in several linear-time algorithms for both transforms.

While the FLT and LLT algorithms initially focused on solving numerically partial differential equations, namely the Hamilton-Jacobi equation, more specialized algorithms have been developed in image processing to compute the so-called square Euclidean distance transform

$$DT^2(p) = \|p\|^2 - g^*(p)$$

of a binary image $p \mapsto f(p) \in \{0,1\}$, where $g(q) = \|q\|^2/2 + I(q)$ and I is the indicator function of the background: $I(q) = 0$ if $f(q) = 1$, $+\infty$ otherwise. Algorithms from image processing take additional advantage of the simplified grid $\{1,\ldots,n_1\} \times \{1,\ldots n_2\}$ on which the function is defined, and of the integer value of f to speed up the computation by using integer arithmetic instead of floating point operations. See [14, 15, 8, 9, 7, 1, 11, 20, 23, 18] for the latest Euclidean distance transform algorithms.

The framework of the present work is slightly different. We assume $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is a one-variable function with f and f' available through a black box. In that context, we investigate a parameterization of the Fenchel conjugate in Section 2, which allows us to define the PLT algorithms in Section 3. Section 4 focuses on the approximation error and the convergence, while Section 5 extends the approach to the computation of the Moreau envelope. Numerical comparison is performed in Section 6, and Section 7 concludes the paper.

2. Theoretical preliminaries

Note $\Gamma(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\} |\ f$ is convex, lower-semicontinuous, proper $\}$, where f proper means that $\operatorname{dom} f = \{x \in \mathbb{R} | f(x) < +\infty\}$ is nonempty. Additionally, we note $\tilde{\Gamma}(\mathbb{R})$ the set of functions $f \in \Gamma(\mathbb{R})$ whose domain is not a singleton. Among the set of all lsc convex functions, Γ disregards the special case $f \equiv +\infty$, and $\tilde{\Gamma}$ the case f is a needle function *i.e.* $f = I_{\{x_0\}}$ for some $x_0 \in \mathbb{R}$. Define

$$G = \{(s,z) \in \mathbb{R}^2 \mid \exists x \in \mathbb{R},\ f \text{ differentiable at } x,\ s = f'(x), \text{ and } z = sx - f(x)\}.$$

We first prove that the parameterization that describes G allows us to recover the graph of the conjugate.

Lemma 2.1. *If $f \in \tilde{\Gamma}(\mathbb{R})$, then $G \neq \emptyset$.*

Proof. By definition of $\tilde{\Gamma}$, there are $x_1, x_2 \in \operatorname{dom} f$ with $x_1 \neq x_2$. Since f is convex, $\operatorname{dom} f$ is also convex, so the segment $[x_1, x_2]$ is contained in $\operatorname{dom} f$. Applying [21, Theorem 25.3], f is differentiable almost everywhere in the open interval (x_1, x_2). So there is $x \in (x_1, x_2)$ with f differentiable at x, and G is nonempty since it contains the point $(f'(x), f'(x)x - f(x))$. □

Remark 2.2. The assumption that there are two distinct points in $\operatorname{dom} f$ is needed since $G = \emptyset$ when $f = I_{\{x_0\}}$ is a needle function. It is equivalent in our context to the function f^* being lower-bounded by two affine functions with different slopes. See the so-called class of epi-pointed functions introduced in [4] for a generalization to higher dimension.

Proposition 2.3. *For any $f \in \tilde{\Gamma}(\mathbb{R})$, $\overline{co} G = \operatorname{epi} f^*$, where $\overline{co} G$ denotes the closed convex hull of G.*

Proof. We invoke [21, Corollary 25.12] which states that G is the set of exposed points of $\operatorname{epi} f^*$. Since the lemma above ensures $G \neq \emptyset$, taking the closed convex hull gives the result. □

Hence the graph of the conjugate is described by the set G up to some affine parts.

To better understand the parameterization, we now recall the relationship between the conjugate and the closed perspective function $\tilde{f}$ (also called epi-multiplication, see [22, Section 1.H] or dilation) defined by [13, Section 2.2 and Example IV.3.2.4]:

$$\tilde{f}(x,t) = \begin{cases} t f(\frac{x}{t}) & \text{for } t > 0, \\ f'_\infty(x) & \text{for } t = 0, \\ +\infty & \text{otherwise;} \end{cases} \tag{1}$$

where f'_∞ denotes the asymptotic function, or recession function [13, Section IV.3.2] (also called the horizon function [22, Section 3.C]) of f, and is defined by:

$$f'_\infty(d) = \sup_{t>0} \frac{f(x_0 + td) - f(x_0)}{t} = \lim_{t \to +\infty} \frac{f(x_0 + td) - f(x_0)}{t},$$

where x_0 is any point in $\operatorname{dom} f$.

Since $\tilde{f}$ is a positively homogeneous lower-semicontinuous (lsc) convex function, it is the support function of a closed convex set in $\mathbb{R}^2$. We note $\partial f(x)$ the (convex) subdifferential of a function f at a point $x \in \mathbb{R}$:

$$\partial f(x) = \{s \in \mathbb{R} \mid \forall y \in \mathbb{R},\ f(y) \geq f(x) + \langle s, y - x \rangle\}.$$

For simplicity we state the next proposition for functions defined on $\mathbb{R}$, although it is valid on $\mathbb{R}^d$.

Proposition 2.4. *Assume f is a proper lsc convex function on $\mathbb{R}$. Then* (*i*) *the closed perspective function* $\tilde{f}$ *is the support function of the set*

$$C_f = \{(x, -r) | (x, r) \in \operatorname{epi} f^*\}$$

the symmetric of epi f^* *with respect to the* x*-axis.*

(*ii*) *the set* C_f *is the subdifferential of* $\tilde{f}$ *at the origin and*

$$C_f = \partial \tilde{f}(0,0) = \overline{co} \overrightarrow{\nabla} \tilde{f}(0,0) + N_{\operatorname{dom} \tilde{f}}(0,0)$$

where $N_{\operatorname{dom} \tilde{f}}(0,0)$ *is the normal cone at the origin to* dom $\tilde{f}$*, the domain of the function* $\tilde{f}$*, and* $\overline{co} \overrightarrow{\nabla} \tilde{f}(0,0)$ *is the closed convex hull of the upper limit set, i.e. it is the set of all the limits of sequences* $\nabla \tilde{f}(x_n, t_n)$*, where* $\tilde{f}$ *is differentiable at* (x_n, t_n) *and* $(x_n, t_n) \to (0,0)$*.*

Proof. (*i*) See [21, Corollary 13.5.1].

(*ii*) Since $\tilde{f}$ is the support function of C_f, we have $C_f = \partial \tilde{f}(0,0)$. The remaining part is [21, Theorem 25.6]. □

While $\overline{\text{co}} \overrightarrow{\nabla} \tilde{f}(0,0)$ is difficult to compute as a general rule, in our case it is constant along the half-lines (x,t) with $t > 0$ (since $\tilde{f}$ is positively homogeneous). We deduce

$$\nabla \tilde{f}(x,t) = (f'(\frac{x}{t}), f(\frac{x}{t}) - \frac{x}{t} f'(\frac{x}{t}))$$

as soon as $\tilde{f}$ is differentiable at (x,t) with $t > 0$. Consequently, a parameterization of the border of C_f, except for some affine parts, is given by

$$\begin{cases} x = f'(s), \\ y = f(s) - s f'(s). \end{cases} \tag{2}$$

For functions of one variable, the subdifferential of f, a lsc proper convex function, is either empty, a single point, or a segment where the endpoints are the left- and right-derivative (see Section 4 below). So handling nondifferentiable points is not too difficult and the parameterization can be fully captured (at differentiable and nondifferentiable points) by the formula:

$$\begin{cases} s \in \partial f(x), \\ f^*(s) = sx - f(x). \end{cases}$$

3. The Parametric Legendre Transform algorithm

Assume $f \in \tilde{\Gamma}(\mathbb{R})$. Given $a \in \operatorname{dom} f$, a simple binary search allows the computation of dom f, so we can always assume our input consists of $a, b \in \operatorname{dom} f$, $a < b$, and a black box returning f, and f' (wherever it is defined).

The PLT algorithm returns a set S of slopes in dom f^* and the values of f^* on S. It can be stated as follows.

(i) Build an uniform grid $x_i = a + i(b-a)/(n-1)$ between a and b.

(ii) Compute $s_i = f'(x_i)$ and $f^*(s_i) = x_i s_i - f(x_i)$. Remove duplicate values to obtain a (non-uniform) grid s_i. Note that since f is convex, $(s_i)_i$ is a nondecreasing sequence.
(iii) Interpolate as needed between $[s_i, s_{i+1}]$ (See Section 4).
(iv) Extrapolate outside $[s_1, s_m]$.

The extrapolation can be performed either by

- assuming that $\operatorname{dom} f$ is bounded; in other words, we only consider our input to be valid on a compact set, which amounts to replacing the function f with $f + I_{[a,b]}$ *i.e.* f is $+\infty$ outside $[a, b]$. The resulting computation returns $f^* \Box \sigma_{[a,b]}$: the inf-convolution of f^* with the support function of $[a, b]$. Then we use [12] to obtain a very strong convergence result when $a \to +\infty$ and $b \to +\infty$. This is the approach used by fast algorithms [16, 17].
- computing $f'_\infty(-1)$ (resp. $f'_\infty(1)$). If it is finite, extrapolate by $+\infty$ at slopes $s \in (-\infty, f'_\infty(-1)] \cup [f'_\infty(1), +\infty)$, otherwise perform a linear extrapolation. We can use the estimates

$$f'_\infty(-1) \simeq \frac{f(a-t) - f(a)}{t} \quad \text{and} \quad f'_\infty(1) \simeq \frac{f(b+t) - f(b)}{t},$$

  for a large value of $t > 0$. Then we approximate by assuming that the supremum is attained at the boundary of $\operatorname{dom} f$ *i.e.* at b when $s > f'_\infty(1)$, and at a when $s < f'_\infty(-1)$.

4. Approximation error and convergence results

We will say that a numerical approximation of f^* at a slope s is *exact* if it equals $f^*(s)$ up to the usual errors occurring when using floating point operations. These errors are much smaller than the errors due to sampling and extrapolation, so we will ignore them.

The PLT algorithm returns a set of slopes $S = \{s_i | i = 1, \dots n\}$ and (an approximation of) the value of f^* at these slopes $f_n^*(s_i)$. Assume $s \in \mathbb{R}$, and consider the following four possible cases.

(i) The slope s is outside $f'([a, b])$ *i.e.* $s > f'_\infty(1)$ or $s < f'_\infty(-1)$. Then we approximate $f^*(s)$ by performing an extrapolation using one of the two possibilities explained previously.
(ii) The slope s equals s_i for some index i. Since $(s_i, f^*(s_i)) \in G$, the computation is exact.
(iii) The slope s is between two given slopes: $s_i \leq s < s_{i+1}$ for some index i. We consider two subcases:
  (a) $(s, f^*(s)) \in G$. We need to interpolate between the point $(s_i, f^*(s_i))$ with derivative x_i, and $(s_{i+1}, f^*(s_{i+1}))$ with derivative x_{i+1}. There are several possibilities: Using the line segment and disregarding the first-order information, building a polyhedral function using the first-order information, using Hermite interpolation polynomials (but we must ensure the resulting polynomial is convex between s_i and s_{i+1}), or using other shape-preserving interpolation scheme. We choose to build a polyhedral function using the first-order information, *i.e.*

$$f_n^*(s) = \max(x_i(s - s_i) + f^*(s_i), x_{i+1}(s - s_{i+1}) + f^*(s_{i+1})).$$

(b) $(s, f^*(s)) \notin G$. Proposition 2.3 ensures that f^* is affine between s_i and s_{i+1}. So we interpolate linearly to obtain an exact computation.

In practice, we may have a black box to compute f at any point x, but the computation of f' may not be available. However, using finite differences and [21, Theorem 24.1], we can compute

$$f'_+(x) = \lim_{t\to 0+} \frac{f(x+t)-f(x)}{t} \quad \text{and} \quad f'_-(x) = \lim_{t\to 0+} \frac{f(x-t)-f(x)}{t}$$

and detect points $x \in \operatorname{int}\operatorname{dom} f$ where f is not differentiable. They are the points were $\partial f(x) = [f'_-(x), f'_+(x)]$ is not a singleton. So we can easily distinguish between case (iii)(a) and case (iii)(b) above.

The convergence of the PLT algorithm results directly from convergence results in [17] for both the extrapolation (when $a \to -\infty$ and $b \to +\infty$) and the interpolation parts (when $n \to +\infty$ *i.e.* a thinner grid is used).

5. Parametric Moreau Envelope

By expanding the squared norm in the definition of the Moreau envelope, we can relate it to the conjugacy operation

$$M_\lambda(s) = \frac{\|s\|^2}{2\lambda} - \frac{1}{\lambda} g^*_\lambda(s), \text{ with } g_\lambda(y) = \lambda f(y) + \frac{\|y\|^2}{2}. \tag{3}$$

Hence we obtain immediately a parameterization of the Moreau envelope.

Proposition 5.1. *Assume f is a proper lsc convex function on $\mathbb{R}^d$. Then the Moreau envelope can be parameterized by*

$$\begin{cases} z = x + \lambda \nabla f(x), \\ M_\lambda(z) = f(x) + \frac{\lambda}{2}\|\nabla f(x)\|^2, \end{cases}$$

where $x \in \mathbb{R}^d$ is a point where f is differentiable. Missing parts of the graph of M_λ are recovered by piecewise quadratic interpolation (knowing that M_λ is C^1).

Proof. Apply Formula (3) with the conjugate parameterization of Formula (2).

Alternatively, apply [22, Example 10.2] to evaluate the proximal mapping

$$x = P_\lambda(z) = (I + \lambda\partial f)^{-1}(z)$$

(the proximal mapping is the set of points where the infimum in the definition of the Moreau envelope is attained; when f is lsc proper convex, the infimum is always attained). So $z = x + \lambda\partial f(x)$, and when f is differentiable at x we obtain the first part of the parameterization. The second part follows from substituting $x = P_\lambda(z)$ into the definition of $M_\lambda(z)$.

Finally, to recover the full graph of M_λ let us see how to operate at any point x. If f is differentiable at x, the corresponding point on the graph of M_λ is recovered by the parameterization process. Assume f is not differentiable at x. Then g_λ is not differentiable at x either. Hence the point $(s, g_\lambda^*(s))$ with $s = \nabla g_\lambda(x)$ belongs to the boundary of $\operatorname{epi} g_\lambda^*$ but is not an exposed point (by Proposition 2.3 and its proof). So it is a convex combination of exposed points, which means that g_λ^* is affine around $s = \nabla g_\lambda(x)$. Then by Formula (3) M_λ is quadratic around x. So we can recover the graph of M_λ by performing a piecewise quadratic interpolation. □

Example 5.2. Consider the function

$$f(x) = \begin{cases} -x-1 & \text{when } x \leq -1, \\ 0 & \text{when } -1 \leq x \leq 1, \\ x-1 & \text{when } 1 < x. \end{cases}$$

The function f is the convex envelope of $x \mapsto ||x|-1|$. Its Moreau envelope is

$$M_\lambda(x) = \begin{cases} -x-1-\frac{\lambda}{2} & \text{when } x \leq -\lambda-1, \\ \frac{(x+1)^2}{2\lambda} & \text{when } -\lambda-1 \leq x \leq -1, \\ 0 & \text{when } -1 \leq x \leq 1, \\ \frac{(x-1)^2}{2\lambda} & \text{when } 1 \leq x \leq \lambda+1, \\ x-1-\frac{\lambda}{2} & \text{when } 1+\lambda < x. \end{cases}$$

The parameterization formula gives

$$\begin{cases} z = x-\lambda \text{ and } M_\lambda(z) = -x-1+\frac{\lambda}{2} & \text{when } x < -1, \\ z = x \text{ and } M_\lambda(z) = 0 & \text{when } -1 < x < 1, \\ z = x+\lambda \text{ and } M_\lambda(z) = x-1+\frac{\lambda}{2} & \text{when } 1 < x. \end{cases}$$

So we deduce

$$M_\lambda(z) = \begin{cases} -z-\frac{\lambda}{2}-1 & \text{when } z < -\lambda-1, \\ 0 & \text{when } -1 < z < 1, \\ z-\frac{\lambda}{2}-1 & \text{when } \lambda+1 < z. \end{cases}$$

The two remaining segments, $[-\lambda-1, -1]$ and $[1, \lambda+1]$, are recovered by quadratic interpolation.

Like for the conjugate, if one wants to deal with nondifferentiable points of f, there is no need to recover quadratic parts. The complete parameterization formula is then

$$\begin{cases} z \in x + \lambda \partial f(x), \\ M_\lambda(z) = f(x) + \frac{\|z-x\|^2}{2\lambda}. \end{cases}$$

6. Numerical comparisons

While the PLT algorithm has clearly a linear-time complexity, we compare it with similar algorithms [19, 18, 8] for computing the Moreau envelope. All the algorithms have a linear worst-case time complexity. The Parabolic Envelope (PE) algorithm computes the lower envelope of quadratic functions. It relies on the fact that computing the intersection of two quadratic functions can be done in constant time $O(1)$. The Linear-time

Legendre Transform (LLT) algorithm reduces the computation of the Moreau envelope to the computation of the Legendre-Fenchel transform, and then uses the LLT algorithm, which takes advantage of convexity to run in linear time. Finally, the Non-Expansive Prox (NEP) algorithm uses the non-expansiveness of the proximal mapping to reduce its complexity. Note that the proximal mapping is nonexpansive for convex functions (and is Lipschitz for slightly more general functions like prox-regular functions [22]).

n	PE	LLT	NEP	PLT
1,000	0.3	0.43	0.16	0.01
3,000	0.99	1.37	0.54	0.
5,000	1.75	2.41	0.98	0.
7,000	2.59	3.57	1.5	0.01
9,000	3.51	4.84	2.11	0.
11,000	4.51	6.21	2.8	0.
13,000	5.57	7.69	3.55	0.01
15,000	6.73	9.28	4.42	0.01
17,000	7.97	10.99	5.35	0.01
19,000	9.3	12.82	6.33	0.02
21,000	10.67	14.72	7.44	0.02
23,000	12.16	16.72	8.58	0.01
25,000	13.82	18.92	9.87	0.02
27,000	15.35	21.33	11.25	0.01
29,000	17.39	23.72	12.77	0.01

Table 6.1: Numerical comparison of the PLT, NEP, LLT, and PE algorithms for the function $f(x) = x^2/2$ on the interval $[-n/2, n/2]$.

Table 6.1 shows the result. The computation was run on a Pentium IV 2.8 GHz processor using Scilab v3.1.1 under Linux Mandriva 2005 LE.

The PLT algorithm is the simplest to code and takes naturally advantage of the optimized vectorized operations of Scilab to achieve a speed improvement by a factor of more than 400. The typical slopes of the least-squares approximation are shown in Table 6.2. They correspond to the multiplicative constant factor associated with each algorithm (each algorithm runs in $Cn + o(n)$ and Table 6.2 shows an approximation of the constant C).

While the NEP and PLT algorithms are restricted to convex functions, the PLT algorithm additionally only computes a parameterization of the graph of the Moreau envelope (instead of an evaluation of the Moreau envelope on a given grid). As such, it relies on more restrictive assumptions to achieve a much faster computation.

If the function is convex but a parameterization is not suitable, the NEP algorithm is

Algorithm	Least-squares slope
PLT	0.000000
NEP	0.000447
LLT	0.000603
PE	0.000835

Table 6.2: Slopes of the least-squares approximation for the data on Table 6.1.

the fastest. When the function is not convex, the PE and LLT algorithms are applicable. While they both share a linear-time complexity, the asymptotic multiplicative constant in the $O(n)$ notation depends mostly on the convexity of the input data.

7. Conclusion

We have presented and proven the correctness of a new algorithm to compute the Legendre-Fenchel conjugate and the Moreau envelope. The algorithm is the simplest to implement and naturally takes advantage of optimized vector operations occurring in Scilab to achieve a more than 400 fold speed improvement. However, the algorithm does not output the value of the transform on a grid but only a parameterization. It is also restricted to convex data.

The PLT algorithm could be extended to nonconvex functions by first computing the convex envelope as the LLT algorithm does, and then computing the parameterization of the conjugate of the piecewise linear function going through the vertices of the convex envelope. However, the computation would not be significantly faster than the LLT algorithm, since most of the computation cost occurs in computing the vertices of the convex hull.

Acknowledgements. The algorithms were implemented on Scilab v3.1.1. Some examples were investigated using the Computational Convex Analysis Toolkit (SCAT) [10].

References

[1] D. G. Bailey: An efficient Euclidean distance transform, in: Combinatorial Image Analysis (Auckland, 2004), R. Klette, J. Žunić (eds.), Lecture Notes in Computer Science 3322, Springer, Berlin (2004) 394–408.

[2] H. Bauschke, M. von Mohrenschildt: Symbolic computation of Fenchel conjugates, ACM SIGSAM Bulletin, to appear.

[3] H. H. Bauschke, M. von Mohrenschildt: Fenchel conjugates and subdifferentials in Maple, Tech. Rep. CORR 97-23, Department of Combinatorics and Optimization, University of Waterloo, Waterloo (1997).

[4] J. Benoist, J.-B. Hiriart-Urruty: What is the subdifferential of the closed convex hull of a function?, SIAM J. Math. Anal. 27 (1996) 1661–1679.

[5] J. Borwein, A. S. Lewis: Convex Analysis and Nonlinear Optimization. Theory and Examples, CMS Books in Mathematics 3, Springer, New York (2000).

[6] L. Corrias: Fast Legendre-Fenchel transform and applications to Hamilton Jacobi equations and conservation laws, SIAM J. Numer. Anal. 33 (1996) 1534–1558.

[7] O. Cuisenaire, B. Macq: Fast and exact signed Euclidean distance transformation with linear complexity, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Phoenix, 1999), IEEE Computer Society Press (1999) 3293–3296.

[8] P. F. Felzenszwalb, D. P. Huttenlocher: Distance transforms of sampled functions, Tech. Rep. TR2004-1963, Cornell Computing and Information Science, Cornell University, Cornell (2004).

[9] M. Gavrilova, M. H. Alsuwaiyel: Two algorithms for computing the euclidean distance transform, Tech. Rep. 2000-661-13, Computer Science Technical Reports, University of Calgary, Calgary (2000).

[10] C. H. Hamilton: Symbolic Convex Analysis, M.Sc. Thesis, Department of Mathematics, Simon Fraser University, Burnaby (2005).

[11] W. Hesselink: A linear-time algorithm for euclidean feature transform sets, available at http://www.cs.rug.nl/∼wim/pub/mans.html (2005).

[12] J.-B. Hiriart-Urruty: Lipschitz r-continuity of the approximate subdifferential of a convex function, Math. Scand. 47 (1980) 123–134.

[13] J.-B. Hiriart-Urruty, C. Lemaréchal: Convex Analysis and Minimization Algorithms, I: Fundamentals; II: Advanced Theory and Bundle Methods, Comprehensive Studies in Mathematics 305, 306, Springer, Berlin (1993).

[14] J. Toriwaki, K. Mori: Distance transformation and skeletonization of 3d pictures and their applications to medical images, in: Digital and Image Geometry: Advanced Lectures, G. Bertrand et al. (ed.), Lecture Notes in Computer Science 2243, Springer, Berlin (2001) 412–429.

[15] C. R. Maurer Jr., R. Qi, V. V. Raghavan: A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 265–270.

[16] Y. Lucet: A fast computational algorithm for the Legendre-Fenchel transform, Comput. Optim. Appl. 6 (1996) 27–57.

[17] Y. Lucet: Faster than the fast Legendre transform, the linear-time Legendre transform, Numer. Algorithms 16 (1997) 171–185.

[18] Y. Lucet: Fast Moreau envelope computation I: Numerical algorithms, Tech. Rep., University of British Columbia Okanagan, Kelowna (2005).

[19] Y. Lucet: A linear euclidean distance transform algorithm based on the linear-time Legendre transform, in: Proc. Second Canadian Conference on Computer and Robot Vision (Victoria, 2005), IEEE Computer Society Press (2005) 262–267.

[20] S. Mauch: A fast algorithm for computing the closest point and distance transform, Tech. Rep. 077, ASC Center for Simulation of Dynamic Response of Materials, California Institute of Technology, Pasadena (2000).

[21] R. T. Rockafellar: Convex Analysis, Princeton University Press, Princeton, New York (1970).

[22] R. T. Rockafellar, R. J.-B. Wets: Variational Analysis, Springer, Berlin (1998).

[23] C. Sigg, R. Peikert, M. Gross: Signed distance transform using graphics hardware, in: Proceedings of IEEE Visualization (Zürich, 2003), IEEE Computer Society Press (2003) 83–90.