# On a Polynomial Vector Field Model for Shape Representation

Mickael Chekroun[1], Jérôme Darbon[2], and Igor Ciril[1]

[1] Institut Polytechnique des Sciences Avancées
24 rue Pasteur F-94270, Le Kremlin Bicêtre, France.

[2] EPITA Research and Development Laboratory (LRDE)
14-16 rue Voltaire F-94276 Le Kremlin-Bicêtre, France

**Abstract.** In this paper we propose an efficient algorithm to perform a polynomial approximation of the vector field derived from the usual distance mapping method. The main ingredients consist of minimizing a quadratic functional and transforming this problem in an appropriate setting for implementation. With this approach, we reduce the problem of obtaining an approximating polynomial vector field to the resolution of a not expansive linear algebraic system. By this procedure, we obtain an analytical shape representation that relies only on some coefficients. Fidelity and numerical efficiency of our approach are presented on illustrative examples.

## 1 Introduction and Motivations

In many computer vision applications, recognition of an object is performed from its shape. The latter is generally obtained thanks to a thresholding or a segmentation [14] of the observed image. Then a shape representation and/or attributes are computed from these shapes in order to get feature vectors. The latter are used for the recognition process. Thus the shape representation is of capital importance. In this paper we present a novel approach to represent a shape.

From a qualitative point of view, a shape representation should enjoy the following properties:

- **Fidelity.** The represented shape should be close to the original (observed) shape.
- **Discrimination.** The representation should allow for a good separation between shapes. This property is mandatory for shape recognition purposes,
- **Robustness.** The representation should be as less sensitive as possible to small deformations which could alter the shape.
- **Compactness.** The representation should require few bits in order to allow for its archiving.

Many shape representations and descriptions have been proposed in the literature to address one or many of the above issues. In this paper we consider only the two dimensional case. Most probably, the most known representation is the one based on the characteristic function. This representation yields a binary image where one value encode a point of the object whereas the other one corresponds to a point of the background. A natural extension of this approach to the case of two or more objects is based on a connected component labeling approach. A label is attributed to every connected component where each of them is assumed to be an object.

Another class of shape representation is the shape descriptor approach. It is dedicated to shape recognition purposes and has generally a poor fidelity behavior and consists on computing attributes associated to a shape. Many descriptors can be computed such as the iso-perimetric ratio, compactness ratio. . . Specific descriptors which are invariant to translation, rotation and scaling are known as the moment invariants [7]. In order to assure robustness minimal box and bounding box can be considered. Due to the discrete nature of an image, computations and properties can be violated.

Moreover these descriptors are not reliable for small objects. The choice of a descriptor generally depends on the application.

Contour based representation is another class of approach. Chain codes are described in the seminal work of Freeman in [4, 5]. It consists of encoding the direction of the contours of a shape. Such a representation is widely used both for pattern recognition [9] and shape transmission in video, as in MPEG-4 for instance. Assuming a shape lives in the complex plane, Fourier descriptors and complex representation [2] can be used. Polygonal approximation of the contour of a shape consists of representing the shape via piecewise affine functions. Extension of this approach using splines instead of affine functions is presented in [**?**]. For the two latter approaches, the Hough transform can be used, as described in [1], in order to determine the number of segments or breaking points.

In [11], Osher *et al.* propose to represent the contour of a shape as the zero level set of an implicit function. This variational representation of the contour is used to make evolve the contour. The implicit function is generally taken as the signed distance function to the contour. In order to help to contour to be closed to the boundary of the shape, Xu *et al.* propose in [15], to consider not only the distance function but also its direction: it yields a vector field where every vector points toward the closest point of the shape. The latter is particularly useful to move the contour to points of the shape which have high curvature.

In this paper we follow in part the work of Xu *et al.* [15]. Contrary to their work our goal is not to use the vector field associated to a shape to move a curve close to this shape but to *represent* the shape itself by its *associated* vector field. More precisely, we consider the polynomial representation of this vector field. The main contributions of this paper are the following. We propose a new shape representation based on the polynomial representation of a vector field associated to a shape. Since this vector field is observed on discrete points only, we propose a convex minimization problem to get the polynomial representation. We show that this problem is well-posed. In order to get a fast algorithm we recast the problem into an equivalent one such that the minimization is reduced to perform a simple matrix multiplications. To our knowledge, all these results are new even if they are related in some aspects to the work of [10].

The rest of this paper is as follows. Sect. 2 is devoted to the theoretical setting of the minimization problem along with its restatement such that the solution is given as a matrix multiplication. This matrix is studied in Sect. 3. Practical implementation details and some experiments are presented in Sect. 4. Finally in Sect. 5, we draw some conclusion and sketch future prospects.

## 2 Obtaining a polynomial vector field for shape representation

In this section we describe the problem of obtaining a polynomial vector field for shape representation as a problem of optimization. The framework retained here allows us to obtain rigorous results on existence and uniqueness for this problem. We recast the problem in order to design a fast algorithm.

### 2.1 Setting of the problem ; existence and uniqueness theorem

First of all we introduce some standard notations: $\|\cdot\|_{2,n}$ stands for the Euclidean norm in $\mathbb{R}^n$ and $\langle\cdot|\cdot\rangle_n$ denotes the usual Euclidean inner product in $\mathbb{R}^n$. Note that in $\|\cdot\|_{2,n}$, the dependance on the dimension will be useful for subsequent computations (cf. for e.g. Eqs. (10) and (15)).

We denote by $\Omega \subset \mathbb{R}^2$ the continuous support of the image under consideration, which we assume to be non empty. For any positive integer $q$, we denote by $\Omega_q$ the discrete set $\{(x_s, y_s) \in \mathbb{Z}^2 \,|\, s \in \{1, ..., q\}\}$ as the associated discrete support of $\Omega$, related to a given discretization.

For any integer $d$, we consider $\mathfrak{X}_d$ the set of polynomial real vector fields on $\Omega$ of degree less than or equal to $d$. An element $W \in \mathfrak{X}_d$ is defined basically via $N = (d+1)(d+2)$ real coefficients $(a^1_{ij}, a^2_{ij})$ which can be written in a coordinates system for each $(x, y) \in \Omega$ as:

$$W(x,y) = \left[\sum_{0\le i+j\le d} a^1_{ij} x^i y^j, \sum_{0\le i+j\le d} a^2_{ij} x^i y^j\right]^T := [f_W(x,y), g_W(x,y)]^T \quad , \quad (1)$$

or in a more detailed fashion

$$W(x,y) = \left[\sum_{k=0}^{d}\sum_{l=0}^{k} a^{k,1}_{l,k-l} x^l y^{k-l}, \sum_{k=0}^{d}\sum_{l=0}^{k} a^{k,2}_{l,k-l} x^l y^{k-l}\right]^T \quad . \quad (2)$$

Note that Eq. (2) will be used in the sequel for the explicit description and clarity of our computations. Besides, note that we do not prescribe conditions on the boundary $\partial\Omega$ of the support.

A binary image being given and an integer $q$ chosen, let $V$ be the observed vector field computed from $\Omega_q$ as described in [15]. More precisely, we compute $V$ as the gradient vector field of the distance mapping. Roughly speaking, it corresponds to a vector field where each vector is *unitary* and points toward the *closest* point of the *border* of the shape. In the sequel, for each point $(x_s, y_s) \in \Omega_q$ this observed vector field $V$ in site $s$ will be given by the following notation:

$$V(x_s, y_s) := [\alpha_s, \beta_s]^T \quad . \tag{3}$$

Now we describe our process to obtain a polynomial vector field approximation of $V$. It relies on a minimization problem of a standard $l_2$-norm, and is slightly related to the work of [10], even if in this last work the accuracy of the approximation seems to be more central than its effectiveness in CPU time for instance. Our proposed approach gives satisfactory results on both aspects (cf. Sect. 4 for illustrative examples). We propose to compute a polynomial vector field approximation of $V$ denoted by $W_{d,V}$ as the *argmin* of the following optimization problem:

$$(\mathbf{P}_{d,V}) \;\; \text{Find} \;\; W^* \in \mathfrak{X}_d \text{ such that } E_{d,V}(W^*) = \inf_{W \in \mathfrak{X}_d} E_{d,V}(W) \;\; ,$$

where $E_{d,V}$ is the following elementary functional:

$$E_{d,V} : \begin{cases} \mathfrak{X}_d \to \mathrm{I\!R}^+ \\ W \mapsto E_{d,V}(W) = \sum_{s=1}^{s=q} ||V(x_s, y_s) - W(x_s, y_s)||_{2,2}^2 \;\; . \end{cases} \tag{4}$$

For any $W \in \mathfrak{X}_d$, let us define $N_q(W)$ as follows:

$$N_q(W) := \left( \sum_{s=1}^{q} \|W(x_s, y_s)\|_{2,2}^2 \right)^{\frac{1}{2}} \quad . \tag{5}$$

In order to deal with a well-posed problem we need to show that $N_q(\cdot)$ is a norm. Note that the only axiom that need some works for making $N_q(\cdot)$ a norm, is the axiom of separation (i.e. $N_q(W) = 0 \Rightarrow W = 0$). We give here some elements to discuss this last important point, which is linked with algebraic geometry. The complete proof can be found in [3]. First of all, consider a vector field $X \in \mathfrak{X}_d$ and recall that a *singular* point $\omega \in \Omega$ of a vector field $X$ is a point such that $X(\omega) = 0$. The singular points of $X$ are given as the common zeros of $f_W$ and $g_W$ defined in Eq. (1). If the two polynomials $f_W$ and $g_W$ are relatively prime, the intersection of the algebraic curves $f_W = 0$ and $g_W = 0$ consists of isolated points whose number (i.e., number of isolated points) is less than or equal to $d^2$ thanks to the classical Bezout theorem [8]. Thus if $q > d^2$, we have that the axiom of separation holds for the subset of vector fields $X \in \mathfrak{X}_d$ whose components are relatively prime. If $f_W$ and $g_W$ have common factors, let us denote by $P$ their greatest common divisor (GCD). The singular points are given as the zeros of $P$. For this last case, curves constituted of singular points may exist but the number of such curves is still bounded in function of the degree $d$ (cf. [8] and references therein). In order to conclude in this last case, we introduce the degree $d^*(q)$ which corresponds to the minimal degree such that there exists a non-zero polynomial in $\mathfrak{X}_d$ which gives zero for all points in the discrete support $\Omega_q$, i.e.,

$$d^*(q) = \inf\{d \in \mathbb{N} \mid \exists X \in \mathfrak{X}_d \backslash \{0\}, X(x_s, y_s) = 0, \forall (x_s, y_s) \in \Omega_q\} \ . \qquad (6)$$

Using the previous considerations we can show the following theorem as proved in [3]:

**Theorem 1.** *For each $q \in \mathbb{N}^*$, there exist $d^*(q) \in \mathbb{N}^*$ such that for any $d \leq d^*(q)$, $N_q(\cdot)$ is a norm on $\mathfrak{X}_d$ .*

Then we get the following existence and uniqueness result by using standard arguments from convex analysis:

**Corollary 1.** *Let $q$ be an integer. Let $V$ be the observed vector field computed from $\Omega_q$ via the standard distance map. Then there exist $d^*(q) \in \mathbb{N}^*$ such that for any integer $d \leq d^*(q)$ the problem of minimization $(\mathbf{P}_{d,V})$ admits a unique solution $W^*_{d,V} \in \mathfrak{X}_d$ .*

*Proof.* Let $q$ and $V$ given as stated in Corollary 1. Then by Theorem 1 we know that there exist $d^*(q) \in \mathbb{N}^*$ such that $N_q(\cdot)$ is a norm on $\mathfrak{X}_d$, for each $d \leq d^*(q)$. Take such a degree $d$, then it is obvious that the functional $E_{d,V}$ defined in Eq. (4) is strictly convex and continuous with respect to the norm $N_q(\cdot)$. We have also the following inequality for each $W \in \mathfrak{X}_d$:

$$E_{d,V}(W) \geq |N_q(V) - N_q(W)| \ ,$$

which leads to

$$E_{d,V}(W) \to +\infty, \text{ as } N_q(W) \to +\infty \ .$$

Thus, classical theorem 1.9 of [13] allows us to conclude. □

We now describe how to perform effectively the optimization.

### 2.2 Rewriting the problem $(\mathbf{P}_{d,V})$

In this section we show that the problem $(\mathbf{P}_{d,V})$ can be restated as an optimization one with a functional of the basic form $J(\xi) = a(\xi, \xi) - l(\xi) + c$ where $a$ (resp. $l$) is a bilinear symmetric form (resp. a linear form) on a finite dimensional vector space $\mathcal{E}$, and where $c$ is a real constant.

Recall that for any point $(x_s, y_s) \in \Omega_q$ the real vector $V(x_s, y_s)$ is given by Eq. (3). Note that a polynomial vector field $W \in \mathfrak{X}_d$ given in Eq. (2) is isomorph to the following vector $\boldsymbol{w}$ of $\mathbb{R}^N$ with $N = (d+1)(d+2)$:

$$\boldsymbol{w} = \left[(\boldsymbol{w}^T_{k,1})_{0 \leq k \leq d}, (\boldsymbol{w}^T_{k,2})_{0 \leq k \leq d}\right]^T \in \mathbb{R}^N \ , \qquad (7)$$

where for each $k \in \{0, ..., d\}$,

$$\boldsymbol{w}_{k,1} = \left[\left(a^{k,1}_{l,k-l}\right)_{0 \leq l \leq k}\right]^T \in \mathbb{R}^{k+1}, \ \boldsymbol{w}_{k,2} = \left[\left(a^{k,2}_{l,k-l}\right)_{0 \leq l \leq k}\right]^T \in \mathbb{R}^{k+1} \ , \qquad (8)$$

with for $i \in \{1,2\}$, $\left(a^{k,i}_{l,k-l}\right)_{0\le l\le k}$ which denotes the row:

$$\left(a^{k,i}_{l,k-l}\right)_{0\le l\le k} = (a^{k,i}_{0,k}, a^{k,i}_{1,k-1}, \cdots, a^{k,i}_{k,0}) \ . \tag{9}$$

Starting from Eq. (4) with notations given by Eq. (2), some elementary algebraic computations permit to show that the problem $(\mathbf{P}_{d,V})$ in $\mathfrak{X}_d$ is equivalent to the following problem restated in $\mathbb{R}^N$:

$$(\widetilde{\mathbf{P}}_{d,V}) \ \ \text{Find } \boldsymbol{w}^* \in \mathbb{R}^N \text{ such that } J_{d,V}(\boldsymbol{w}^*) = \inf_{\boldsymbol{w}\in\mathbb{R}^N} J_{d,V}(\boldsymbol{w}) \ ,$$

where we define $J_{d,V}(\boldsymbol{w})$ for any $\boldsymbol{w} \in \mathbb{R}^N$ as,

$$J_{d,V}(\boldsymbol{w}) = \langle H\boldsymbol{w}|\boldsymbol{w}\rangle_N - 2\langle \boldsymbol{h}|\boldsymbol{w}\rangle_N + c \ . \tag{10}$$

We now detail the formulae describing the matrix $H$ and the vector $\boldsymbol{h}$ on their direct expressions. We also rewrite them in order to get more manageable expressions in terms of numerical computations. The direct formulae are only given for pedagogical purposes and can be viewed as the starting step for obtaining the manageable ones. The complete description along with proofs can be found in [3].

• *Direct expressions*: Using once more Eq. (4) and Eq. (2), by developing, it is easy to show that:

$$\langle H\boldsymbol{w}|\boldsymbol{w}\rangle_N = \sum_{s=1}^{q}\left(\left(\sum_{\substack{k=0\\0\le l\le k}}^{d} a^{k,1}_{l,k-l}x_s^l y_s^{k-l}\right)^2 + \left(\sum_{\substack{k=0\\0\le l\le k}}^{d} a^{k,2}_{l,k-l}x_s^l y_s^{k-l}\right)^2\right) \ , \tag{11}$$

and to show that $\boldsymbol{h}$ is a vector depending on the data $V$ and the discrete support $\Omega_q$ through the following expression:

$$\langle \boldsymbol{h}|\boldsymbol{w}\rangle_N = \sum_{s=1}^{q}\left(\alpha_s\left(\sum_{\substack{k=0\\0\le l\le k}}^{d} a^{k,1}_{l,k-l}x_s^l y_s^{k-l}\right) + \beta_s\left(\sum_{\substack{k=0\\0\le l\le k}}^{d} a^{k,2}_{l,k-l}x_s^l y_s^{k-l}\right)\right) \ . \tag{12}$$

•*Manageable expressions*: In order to obtain implementable expressions, we introduce the following matrix block notations for each $(k,p) \in \{0,...,d\}^2$:

$$[XY]_{k,p} = \sum_{s=1}^{q}[X_sY_s]_k[X_sY_s]_p^T \in \mathcal{M}_{(k+1),(p+1)}(\mathbb{R}) \ , \tag{13}$$

where $[X_sY_s]_k$ (with one index) denotes the following column vector:

$$[X_sY_s]_k = \left[\left(x_s^l y_s^{k-l}\right)_{0\le l\le k}\right]^T \in \mathbb{R}^{k+1} \ . \tag{14}$$

One can show that the matrix $H$ is a symmetric matrix which can be implicitly defined through the following expression valid for any $\boldsymbol{w} \in \mathbb{R}^N$:

$$\langle H\boldsymbol{w}|\boldsymbol{w}\rangle_N = \sum_{\substack{0\leq k+p\leq 2d \\ 0\leq k,p\leq d}} \left(\langle [XY]_{k,p}\boldsymbol{w}_{p,1} \,|\, \boldsymbol{w}_{k,1}\rangle_{k+1} + \langle [XY]_{k,p}\boldsymbol{w}_{p,2} \,|\, \boldsymbol{w}_{k,2}\rangle_{k+1}\right) \ . \tag{15}$$

Using this formalism we get the following expression of $\mathbf{h}$:

$$\boldsymbol{h} = \sum_{s=1}^{q} \boldsymbol{h}_{((x_s,y_s);(\alpha_s,\beta_s))} \in \mathbb{R}^N \ , \tag{16}$$

with

$$\boldsymbol{h}_{((x_s,y_s);(\alpha_s,\beta_s))} = \left[\alpha_s \left[X_s Y_s\right]^T, \beta_s \left[X_s Y_s\right]^T\right]^T \ , \tag{17}$$

and

$$[X_s Y_s] = \left[\left([X_s Y_s]_k^T\right)_{0\leq k\leq d}\right]^T \in \mathbb{R}^{\frac{N}{2}} \ , \tag{18}$$

where $[X_s Y_s]_k$ is given by Eq.(14).

Finally, still by using Eqs. (2), (3) and (4), it is easy to show that the constant $c$ in Eq. (10) is given by:

$$c = \sum_{s=1}^{q} \left((\alpha_s)^2 + (\beta_s)^2\right) \ . \tag{19}$$

These preceding manageable expressions allow us to restate the problem $(\mathbf{P}_{d,V})$ as the following equivalent one in $\mathbb{R}^N$

$$(\widetilde{\mathbf{P}}_{d,V}) \ \text{ Find } \boldsymbol{w}^* \in \mathbb{R}^N \text{ such that } J_{d,V}(\boldsymbol{w}^*) = \inf_{\boldsymbol{w}\in\mathbb{R}^N} J_{d,V}(\boldsymbol{w}) \ ,$$

which is suitable for an efficient implementation as the following sections will show. For the sake of completeness, we expose here the obvious simplifications brought by our approach, permitting to obtain the solution as the one of a linear algebraic system. Indeed, since $J_{d,V}$ is convex and $\nabla J_{d,V}(\boldsymbol{w}) = H\boldsymbol{w} - \boldsymbol{h}$ , then according to the theorem 27.1 of [12], we have that $\boldsymbol{w}^*$ is the solution of the following equivalent problem $(\widetilde{S}_{d,V})$:

$$(\widetilde{S}_{d,V}) \ \text{ Find } \boldsymbol{w}^* \in \mathbb{R}^N \text{ such that } H\boldsymbol{w}^* = \boldsymbol{h} \ .$$

According to Theorem 1, we know that the matrix $H$ is symmetric invertible, and is thus definite positive. Consequently, the problem $(\widetilde{\mathbf{P}}_{d,V})$ admits the single following solution:

$$\boldsymbol{w}^*_{d,V} = H^{-1}\boldsymbol{h} \ . \tag{20}$$

It is then sufficient to exhibit the explicit structure of $\boldsymbol{H}$ in order to compute the solution $\boldsymbol{w}^*_{d,V}$, by a classical Cholesky method as the symmetric structure of $H$ permits. This explicit structure of $H$ is described in the next section.

## 3 Properties of the matrix H

In this section we describe some properties of the matrix $H$ introduced in Eq. (11) and in Eq. (15). The latter will be used for the computations (see Sect. 4). The main objective is to show that the matrix $H$ is sparse in some meaning. First of all, note that without loss of generality, we can assume that our discrete support $\Omega_q$ is a cartesian product of an odd number of points which are labeled with zero as center of symmetry, that is:

$$\Omega_q = \{-2\delta - 1, \cdots, 2\delta + 1\}^2 \ , \tag{21}$$

with $\delta \in \mathbb{Z}_+$.

¿From Eq. (15), some attentive look shows that $H$ can be written by block as follows:

$$H = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \tag{22}$$

where $B = (B_{i,j})_{1 \le i,j \le \frac{N}{2}} \in \mathcal{M}_{\frac{N}{2}}(\mathbb{R})$ is a matrix described below and $B_{i,j}$ are its constitutive real coefficients. More precisely, the matrix $B$ is decomposed by block $[B]_{k,p}$ as follows for any $(k,p)$ in $\{0, ..., d\}^2$ :

$$[B]_{k,p} = [XY]_{k,p} = \left(B_{i_k+r, j_p+g}\right)_{\substack{0 \le r \le k \\ 0 \le g \le p}} \ , \tag{23}$$

with $i_k = 1 + \frac{k(k+1)}{2}$, and $j_p = 1 + \frac{p(p+1)}{2}$.

For each $(r,g)$ in $\{0, ..., k\} \times \{0, ..., p\}$ we have

$$B_{i_k+r, j_p+g} = Y^{(k+p)-(r+g)} X^{(r+g)} \ , \tag{24}$$

where for each $(m,n)$ in $\{0, ...d\}^2$, $Y^n X^m$ denotes the real

$$Y^n X^m := \sum_{s=1}^{q} y_s^n x_s^m \ . \tag{25}$$

Note that one can also show the identity $Y^n X^m = Y^m X^n$ since we assume a symmetric support. Therefore in order to construct, $H$ it is sufficient to construct $B$ which is defined by the blocks $([XY]_{k,p})_{0 \le k,p \le d}$. Thanks to the symmetry of our support defined by Eq. (21), we get the two following propositions. The Proposition 1 gives us the null blocks, whereas the Proposition 2 gives us the repartition of null coefficients in the non-null blocks. These propositions are proved [3] and rely on the Cholesky decomposition of $H$:

**Proposition 1.** *Let us suppose that $\Omega_q$ checks (21), then for any $(k,p) \in \{0, ..., d\}^2$ such that $k+p$ is odd,*

$$[XY]_{k,p} = [0]_{k,p} \in \mathcal{M}_{(k+1),(p+1)}(\mathbb{R}), \ \textit{and} \ [XY]_{p,k} = [0]_{p,k} \in \mathcal{M}_{(p+1),(k+1)}(\mathbb{R}) \ .$$

**Proposition 2.** *Let us suppose that $\Omega_q$ checks (21), then for any $(k,p) \in \{0, ..., d\}^2$ such that $(k+p)$ is even, the following properties holds:*

*(i)* For each $(r,t) \in \{0,...,k\} \times \{0,...,p\})$ such that $r+t$ is odd, we have $B_{i_k+r,j_p+t} = 0$.

(ii) For each $m \in \{0\} \cup \{n \leq r+t \,|\, n \text{ is even }\}$, and for each $(r,t) \in \{0,...,k\} \times \{0,...,p\})$ such that $(r+t=m)$ or $(r+t=(k+p)-m)$, then

$$B_{i_k+r,j_p+t} = Y^{(k+p)-m}X^m = Y^m X^{(k+p)-m} > 0 \ .$$

The latter propositions allows us to build efficiently the matrix $H$. We emphasize that this matrix $H$ does not depend on the observed vector field $V$ but only on the discrete support $\Omega_q$.

# 4 Experimental results

We first give some notes on the technical implementation before presenting some results.

## 4.1 Notes on our implementation

Let us give a simple example of the matrix $H$. Assume that the degree $d = 4$. The block $B \in \mathcal{M}_{15}(\mathbb{R})$ whose $H$ is made of which, have the following form:

$$B = \begin{pmatrix} [0]_{1,0} & [B]_{1,1} & [0]_{1,2} & [B]_{1,3} & [0]_{1,4} \\ [B]_{2,0} & [0]_{2,1} & [B]_{2,2} & [0]_{2,3} & [B]_{2,4} \\ [0]_{3,0} & [B]_{3,1} & [0]_{3,2} & [B]_{3,3} & [0]_{3,4} \\ [B]_{4,0} & [0]_{4,1} & [B]_{4,2} & [0]_{4,3} & [B]_{4,4} \end{pmatrix}$$

where $[0]_{\cdot,\cdot}$ denotes a block filled of zeros. There are 12 zero-block and 13 non-zero-block. We can further express a non-zero block. We have a non empty block for any $(k,p) \in \{0,...,4\}^2$ such that $k+p$ even with $k \geq p$. For instance for $k=4$ and $p=2$, we have $[B]_{4,2} \in \mathcal{M}_{5,3}(\mathbb{R})$ which can be computed with the two terms $Y^6$ and $Y^4X^2$:

$$[B]_{4,2} = \begin{pmatrix} Y^6 & 0 & Y^4X^2 \\ 0 & Y^4X^2 & 0 \\ Y^4X^2 & 0 & Y^4X^2 \\ 0 & Y^4X^2 & 0 \\ Y^4X^2 & 0 & Y^6 \end{pmatrix}$$

As one can see, the exponent necessary for the numerical computations, grows rapidly as the degree gets larger. In practice, it means that classical types (such as `int,float,double`) used in computer program to store numbers do not have enough bits to encode these numbers. This leads to wrong computations. Therefore special libraries which are dedicated to encode large numbers has to be used. For our implementation, we have used the GNU Multiple Precision library [6].

(a) (b)

**Fig. 1.** Original shapes: a circle in (a) and a U in (b).

### 4.2 Experiments

Figure 1 depicts two original shapes. The first one is a circle while the second one is has the shape of the letter U. Size of these two images is $109 \times 109$. The associated vector field of these two shapes are shown in Figure 2-(a). Reconstruction of these vector fields with degrees 13 and 25 are respectively shown on Figure 2-(b) and Figure 2-(c). As one can see, the more the degree is high the better the reconstruction is.

We have used a precision of 256 bits to encode numbers. For a degree 25 the minimization takes 11.68 seconds while it takes 2.56 seconds for a degree 13. Computations are performed on a Centrino 2.1 GHz. Note that in these time results, most of the time is devoted to the construction of the vector $\mathbf{h}$ defined in Eq. (12). Indeed, the matrix $H$ does not depend on the considered shape, this is a key point for concluding of the time efficiency of our algorithm for more general shapes, but also justify the need of other experiments.

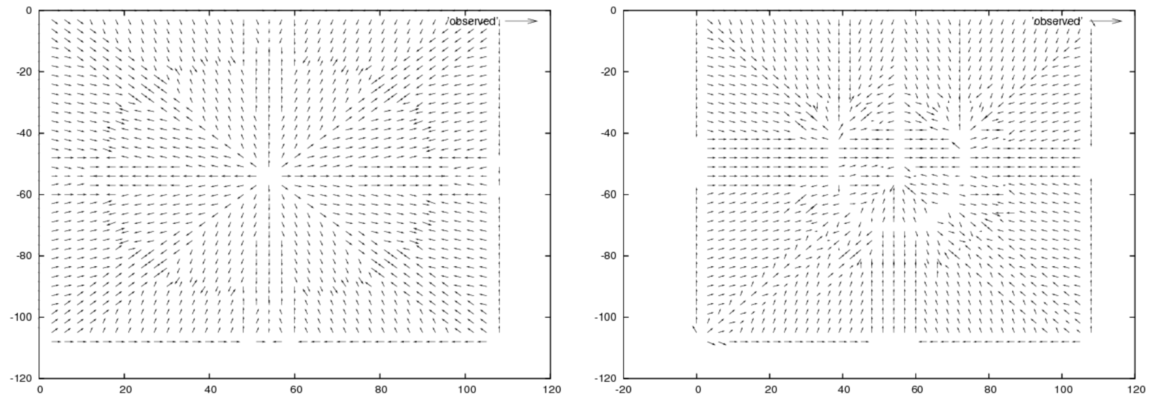
## 5 Conclusion

We have shown in this paper, how to attach a polynomial vector field to a shape in order to represent it through an optimization procedure that can be recast into an algebraic system in some appropriate vector space. By this procedure, the representation thus achieved permits good fidelity to the initial shape and can be performed with reasonable time. The quality of this fidelity seems to increase with the degree of the polynomial vector field considered for the approximation. Of course these comments rely on our experiments that must be completed in the experimental side as the theoretical one.

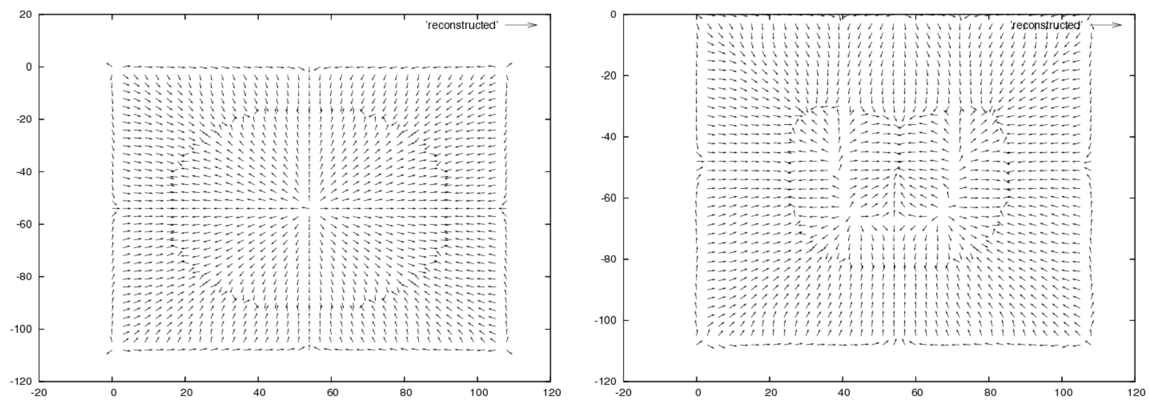
Many future works are under investigation. First of all, given a shape, we need to derive a condition for estimating the degree of the polynomial vector field which is able to approximate the vector field associated to the shape, with good accuracy and not expensive time. Besides, since we have a polynomial representation of the shape we can use algebraic tools. In particular, this framework could be convenient to study shape deformations and/or diffeomorphisms via tools such as Lie transforms. However many efforts remain to be done. Finally, the use of the approach developed in this paper for performing vector field compression attached to the shape, will be presented in a forthcoming paper.

## References

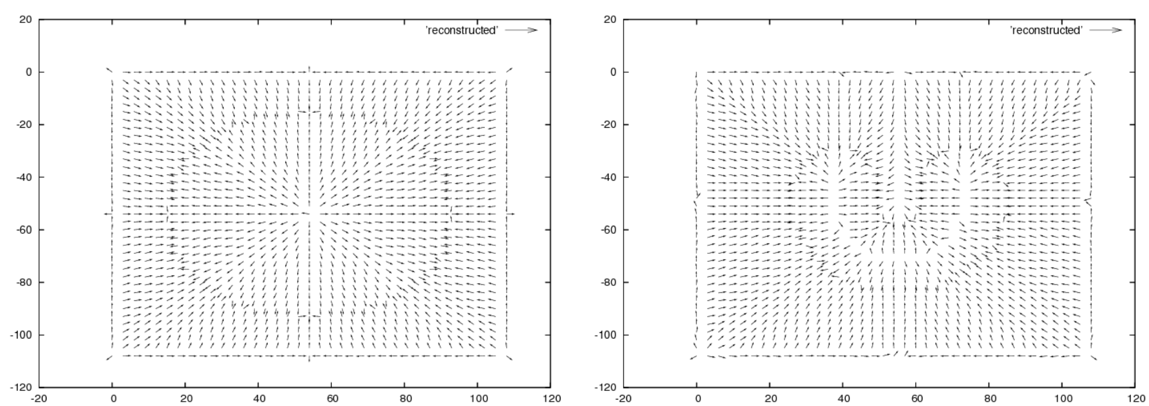
1. Ballard, D.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition **13**(2) (1978) 111–122
2. Bertrand, G., Queval, R., Maître, H.: Shape interpolation by fourier descriptors with application to animation graphics. Signal Processing **4** (1982) 53–58
3. Chekroun, M., Darbon, J., Ciril, I.: On a polynomial vector field model for shape representation: mathematical foundation. In preparation.
4. Freeman, H.: On the encoding of arbitrary geometric configuration. IEEE Transactions on Electronic Computing **10** (1961) 260–268
5. Freeman, H.: Computer processing of lin drawing images. ACM Computer Survey **6** (1977) 57–97
6. GNU Multiple Precision library ftp://ftp.gnu.org/gnu/gmp/.
7. Hu., M., K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, **IT-8**(2) (1962) 179–187
8. Kirwan, F.: Complex Algebraic Curves. Cambridge University Press (1992)
9. Lowrance, R., Wagner, R.: An extension of the string to string correction problem. J. Ass. Comp. Mach. **22** (2) (1975) 177–183
10. Mussa-Ivaldi, F., A.: From basis functions to basis fields: vector field approximation from sparse data. Biol. Cybernetics **67** (1992) 479–489
11. Osher, S., Sethian, J.: Front propagating with curvature dependent speed. Journal of Computational Physics **79** (1988) 12–49
12. Rockafellar, R., T.: Convex Analysis. Princeton University Press (1970)
13. Rockafellar, R., T., Wets, R., J., B.: Variational Analysis. Springer-Verlag, New York (1998)
14. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. PWS (1999)
15. Xu, C., Prince, J., L.: Snakes, Shapes, and Gradient Vector Flow. IEEE Transactions on Image Processing **7**(3) (1998).




(a) Original vector fields




(b) Reconstructed vector fields with degree 13




(c) Reconstructed vector fields with degree 25

**Fig. 2.** Vector fields associated with a circle (left column) and a U (right column). The original vector fields are depicted in (a), while the reconstructed ones are shown in (b) for degree 13 and in (c) for a degree 25.