記　事

ＩＲＩＳ－４Ｄで２次元情報を３次元空間へ展開する方法

～噴霧火炎中の燃料油滴の飛行速度計測結果の可視化～

大阪大学大型計算機センター研究開発部
出口　弘，小林一男
deguchi@center.osaka-u.ac.jp
大阪大学工学部機械工学科燃焼工学講座
赤松史光，田端誠司，中部主敬
akamatsu@combu.mech.eng.osaka-u.ac.jp

１．まえがき

研究や開発に携わる者にとって，現象が軸対称２次元領域で起こると仮定して数値解析あるいは実験計測を行なう機会は比較的多い．この場合，得られる情報は計算または測定対象場中の主方向対称軸を含む１枚の平面上にプロットされることになる．もちろん，この２次元図自体はそれだけで十分な説得力を有しており，プレゼンテーションによく用いられている．しかし，可視化ツールが豊富になってきた昨今において，デモンストレーション力の観点から見れば３次元カラーイメージに比して見劣りすることは否めない．そこで，得られた２次元データを，３次元空間に展開して視覚に訴える仮想的な３次元イメージに加工することを試みた．ここでは，その一例として，レーザードップラー流速計(ＬＤＶ)で計測された噴霧火炎中での油滴飛行速度の情報を*Power IRIS-4D/310VGX*に搭載されている*Explorer*を用いて可視化する方法を紹介する．

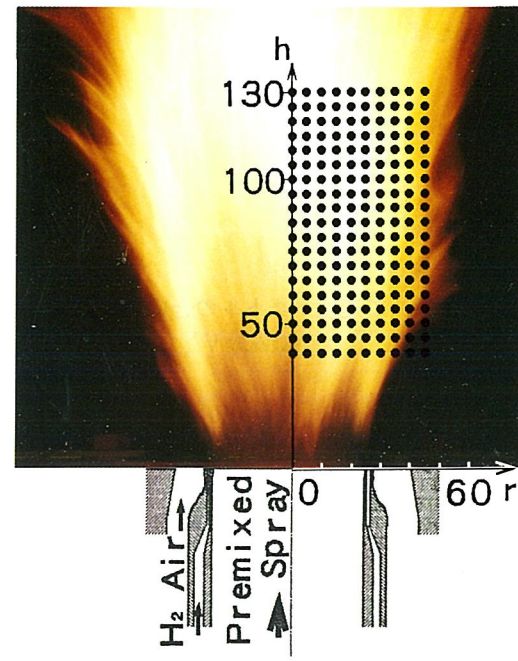

図 1　噴霧火炎バーナ

２．噴霧火炎中での油滴速度の計測

噴霧火炎は車両用ディーゼルエンジンのほかにも工業的に多用されており，その燃焼形態や火炎構造を明らかにすることは，燃焼の高効率化および自然環境保全の面から非常に重要な課題となっている．噴霧火炎は液体燃料を霧状に微粒化して燃焼させるため，燃料油滴と燃焼用空気との相互作用が燃焼形態に大きな影響を及ぼす．そこで，ＬＤＶを用いて，噴霧火炎中での燃料油滴の時間平均的な速度場を測定した．

図１に実験用バーナポート部の詳細を火炎の直接写真とともに示す．バーナは内径52.7mmの噴霧バーナポート，保炎用水素拡散パイロットバーナおよび火炎安定用空気ポートを有する３重円管構造になっている．燃料の白燈油はバーナポートの上流440mmに設置された二流体噴射弁で微粒化

される．実験条件は，非燃焼時の燃焼用空気流量1.2 L/s，燈油－空気質量流量比0.045kg/kg，霧化用空気圧力0.4MPaとした．

座標系はバーナポート中心を原点に，バーナ軸下流方向にh (mm)を，半径方向にr (mm)をとり，5mm等間隔でh＝40～130 mm，r＝0～45mmの範囲にある合計19×10＝190点で計測を行った．

2．1　1次元流れ場計測用ＬＤＶによる2次元流れ場の計測方法

計測対象となったレーザ用トレーサ粒子iについて，バーナ軸（h軸）を含む1枚の平面上で，h軸から右上方45°（α軸）方向の速度をa_i，左上方45°（β軸）方向の速度をb_iとし，それらをその時間平均値$\overline{a}$，$\overline{b}$と変動成分a'_i，b'_iとの和で表せば，それぞれ，

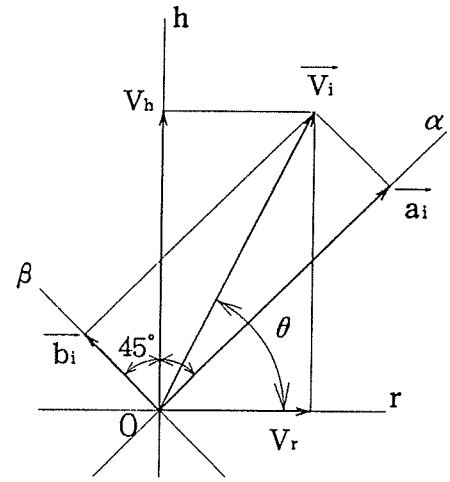

図 2　速度ベクトルの合成

$$a_i = \bar{a} + a_i' \qquad (1)$$
$$b_i = \bar{b} + b_i'$$

と書ける．

これを用いて速度ベクトルの合成を行えば，バーナ軸（h軸）方向および半径（r軸）方向の時間平均値$\overline{V}_h$，$\overline{V}_r$と変動成分$V'_{h,i}$，$V'_{r,i}$は，それぞれ，以下のように表せる（図2参照）．

$$\overline{V_h} = \frac{1}{\sqrt{2}}(\bar{a}+\bar{b}) \qquad V'_{h,i} = \frac{1}{\sqrt{2}}(a_i'+b_i') \qquad (2)$$
$$\overline{V_r} = \frac{1}{\sqrt{2}}(\bar{a}-\bar{b}) \qquad V'_{r,i} = \frac{1}{\sqrt{2}}(a_i'-b_i')$$

これから，速度変動成分の相互相関$V'_{h,i}V'_{r,i}$を求めると次式となる．

$$V'_{h,i}V'_{r,i} = \frac{1}{2}(a_i'+b_i')(a_i'-b_i') = \frac{1}{2}(a_i'^2-b_i'^2) \qquad (3)$$

よって，その時間平均値$\overline{V'_h V'_r}$は，

$$\overline{V_h'V_r'} = \overline{V_{h,i}'V_{r,i}'} = \overline{\frac{1}{2}(a_i'^2 - b_i'^2)} = \frac{1}{2}(\overline{a_i'^2} - \overline{b_i'^2}) \tag{4}$$

と表される．

また，油滴の速度変動の強さの目安として，乱れエネルギに類似した量 k_{di} を次式で定義した．ただし，トレーサ粒子の旋回方向速度成分は計測していないので，式(4)には含ませていない．したがって，k_{di} はｈ－ｒ平面内の変動速度ベクトルの絶対値の２乗平均値となっている．

$$k_{d_i} = \frac{1}{2}(V_{h,i}'^2 + V_{r,i}'^2) = \frac{1}{4}[(a_i'+b_i')^2 + (a_i'-b_i')^2] = \frac{1}{2}(a_i'^2 + b_i'^2) \tag{5}$$

このとき，その時間平均値 $\overline{k_d}$ は次式となる．

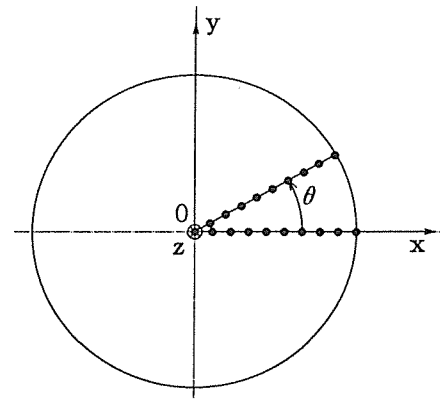

図 3　３次元座標系への展開

$$\overline{k_d} = \overline{k_{d_i}} = \overline{\frac{1}{2}(a_i'^2 + b_i'^2)} = \frac{1}{2}(\overline{a_i'^2} + \overline{b_i'^2}) \tag{6}$$

以上のことから，流れ場が定常である場合，α 方向と β 方向の速度を別々に１次元測定しても式(2)，(4)，(6)を用いて時間平均的なバーナ軸方向速度成分，半径方向速度成分，速度変動の相互相関，k_d 値を算出することができる．なお，ここでのＬＤＶ計測用トレーサ粒子は噴霧油滴そのものなので，上述の諸量は油滴粒径による分別を行わず，計測時間内にＬＤＶ検査体積を通過するＮ個の油滴にわたって計測された測定値を平均することにし，その値をもって時間平均値と見なした．それゆえ，式中に現われる $\overline{a}$，$\overline{b}$，$\overline{a'^2_i}$，$\overline{b'^2_i}$ は次式により求めた．

$$\overline{a} = \frac{\sum_{i=1}^{N} a_i}{N}, \qquad \overline{b} = \frac{\sum_{i=1}^{N} b_i}{N} \tag{7}$$

$$\overline{a'^2} = \frac{\sum_{i=1}^{N} a_i'^2}{N} = \frac{\sum_{i=1}^{N} (a_i - \overline{a})^2}{N}, \quad \overline{b'^2} = \frac{\sum_{i=1}^{N} b_i'^2}{N} = \frac{\sum_{i=1}^{N} (b_i - \overline{b})^2}{N}$$

ここで，Ｎは１測定点における測定データ個数であり，今回の測定では2000個とした．データは（ｒ座標，ｈ座標，$\overline{V_r}$，$\overline{V_h}$，$\overline{V'_h V'_r}$，$\overline{k_d}$）を１組としてファイルに格納される．

２．２　３次元データへの展開

ここでは，２．１節で得られたデータを３次元データに展開する方法を述べる．図３に示すようにバーナポート中心を原点としｘ，ｙ，ｚ座標をとる．ここで，ｚ座標はバーナ軸と一致しており，紙面に垂直で紙面の裏から表方向（バーナ下流方向）を正とする．２次元データ（ｒ座標，ｈ

座標，$\overline{V_r}$，$\overline{V_h}$，$\overline{V'_h V'_r}$，$\overline{k_d}$）を3次元データ（x座標，y座標，z座標，$\overline{u}$，$\overline{v}$，$\overline{w}$，$\overline{V_r}$，$\overline{V_h}$，$\overline{V'_h V'_{r\,3D}}$，$\overline{k_{3D}}$）に変換する際，軸対称2次元流を仮定しているため同一半径上のスカラー量は同じ値をとり，周方向速度成分がないことを考慮すると，角度θに対して以下の関係式が成り立つ．

$$
\begin{aligned}
&x = r\cos\theta,\quad y = r\sin\theta,\ z = h,\\
&\bar{u} = \overline{V_r}\cos\theta,\ \bar{v} = \overline{V_r}\sin\theta,\ \bar{w} = \overline{V_h}\ ,\\
&\overline{V'_h V'_{r,3D}} = \overline{V'_h V'_r}\ ,\quad \overline{k_{3D}} = \overline{k_d}
\end{aligned}
\tag{8}
$$

ここで，θは0°～360°の範囲で0°から10°ごとの値をとるようにしたため，合計37×19×10＝7030組のデータが出来ることになる．

3．*IRIS-4D* による可視化

前章で得られた3次元データをいよいよ可視化する．そのファイル形式は以下のようなフォーマットでアスキーコードの文字列として表現した．

θ方向の分割数(37)　h方向のデータ数(19)　r方向のデータ数(10)

x	y	z	$\overline{u}$	$\overline{v}$	$\overline{w}$	$\overline{V_r}$	$\overline{V_h}$	$\overline{V'_h V'_{r\,3D}}$	$\overline{k_{3D}}$
37	19	10							
0	0	40	-.21991	0	7.78171	-.21991	7.78171	-.434032	2.62147
0	0	40	-.216569	-.0381869	7.78171	-.21991	7.78171	-.434032	2.62147
					.				
					.				
					.				
					.				
5	0	100	.251023	0	8.67408	.251023	8.67408	-.13312	1.0528
4.92404	.86824	100	.247209	.0435897	8.67408	.251023	8.67408	-.13312	1.0528
4.69846	1.7101	100	.235884	.0858549	8.67408	.251023	8.67408	-.13312	1.0528
					.				
					.				
					.				
					.				
42.2861	-15.3911	130	2.34422	-.853238	8.29578	2.49467	8.29578	.122959	.859123
44.3163	-7.81439	130	2.45677	-.433207	8.29578	2.49467	8.29578	.122959	.859123
45	0	130	2.49467	0	8.29578	2.49467	8.29578	.122959	.859123

3．1　入力モジュールの作成

まず最初に /usr/explorer/bin/dscribe と入力して，DataScribe を立ち上げる．Template メニューから Template を選び，Direction を Input，Type を Ascii として入力テンプレートを作り，テンプレート名を filename とした．

次に，Template メニューから Template を選び，Direction を Output，TYPE を Explorer として出力テンプレートを5個作り，テンプレートの名前をそれぞれ UVW，U'V'，K，VR，VH とした．（本来 U'V' という名前は前述の物理量と対応するように VH'VR' とすべきものである．）

3．1．1　入力テンプレート

入力モジュールはファイルからデータの書いてある順に読み込んでいくので，入力テンプ

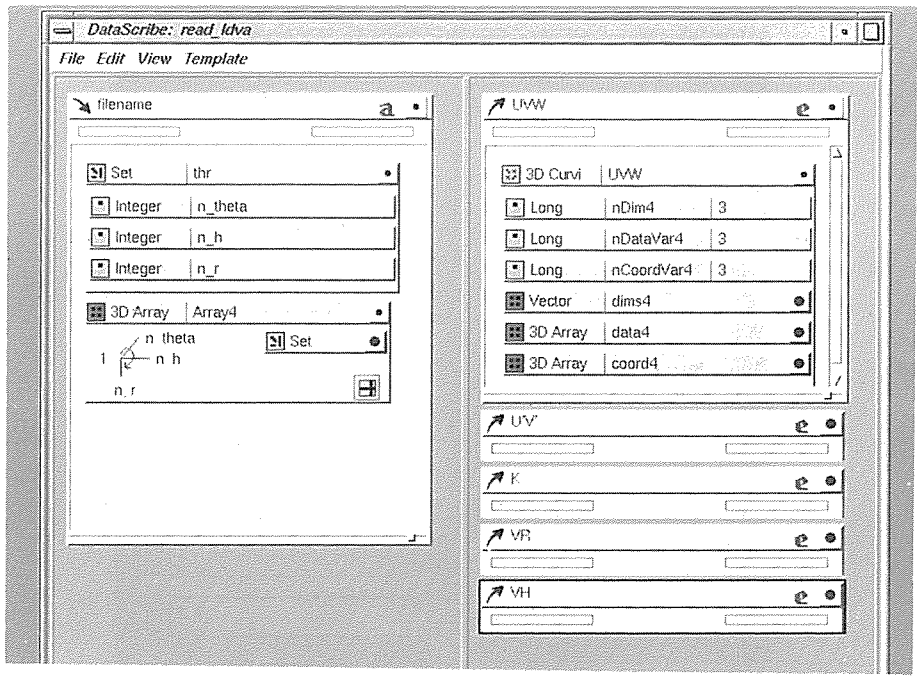

図 4(a)　データスクライブの編集（その１）

レートにデータの並びに合わせてそのデータ型のグリフ（アイコン）を順に並べて作っていく．

まず，ファイルの始めにデータの個数（37, 19, 10）を書いておいたので，最初にこれを読み込む部分を作る．パレットから Set をドラッグし，その名前を thr とする．このデータは整数で書かれているので，Set 右丸をクリックして開き，その中にパレットから Integer を３つドラッグする．データの並びに合わせて，一番上の Integer から，n_theta，n_h および n_r と名前を付け換える．

ファイルの次の行からは３次元の格子座標とその点での計測値が書かれているので，３次元の配列で読み込む．パレットから 3D Array を入力テンプレート中の Set の下にドラッグし，Array4 と名付ける．そして，3D Array Array4 の右丸をクリックして開き，座標系のインデックス N1，N2，N3 をデータ個数の情報を読み込んだ n_theta，n_h および n_r に書き換える．この順序が 3D Array Array4 のデータを読み込む順序となるのでデータファイルの並びに対応している必要がある．（すなわち，２．２節で述べた様に，実験データをまず θ を，次に h，最後に r を変化させて３次元格子データに変換したので，N1 を n_theta，N2 を n_h，N3 を n_r とするのである．）その後，Array4 のデフォルトの型を表している Integer 上にパレットから Set をドラッグし，配列の型を Set にする（図４(a)参照）．

その Set の右丸をクリックして開き，その中に座標（x，y，z）と速度（$\overline{u}$，$\overline{v}$，$\overline{w}$）を読み込むための Vector ２つと，$\overline{V}_r$，$\overline{V}_h$，$\overline{V'_h V'_r}_{3D}$および $\overline{k}_{3D}$ を読み込むための Float ４つをドラッグする．名前はそれぞれ，XYZ，UVW，VR，VH，U'V'，K とした．そして Vector XYZ および UVW の右丸をクリックして，ベクトルの次元を３（1-N を 1-3）に，パレットから Float をドラッグしてベクトルの要素の型を Float にする（図４(b)参照）．

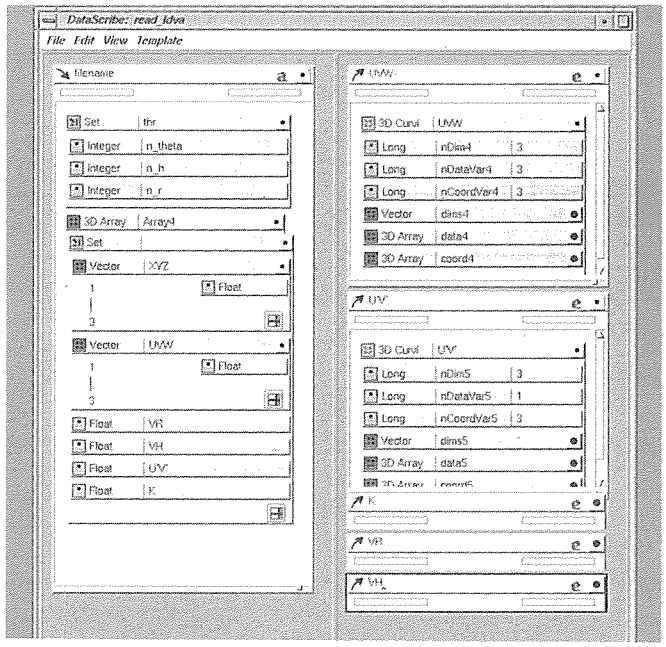

図 4(b)　データスクライブの編集（その２）

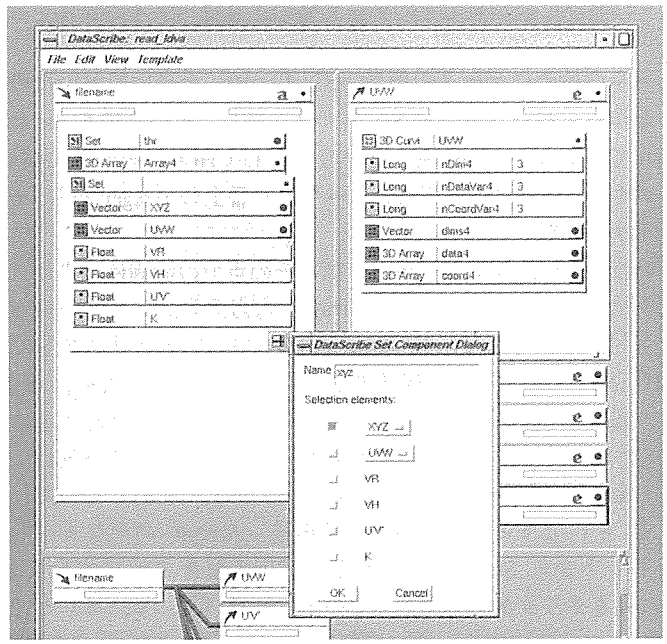

図 4(c)　データスクライブの編集（その３）

さらに，Set の中の緑色ボタン（正式名：Component Menu Button）上でマウスの右ボタンでドラッグして <New> を選択し，DataScribe Set Component Dialog を出す．その Name

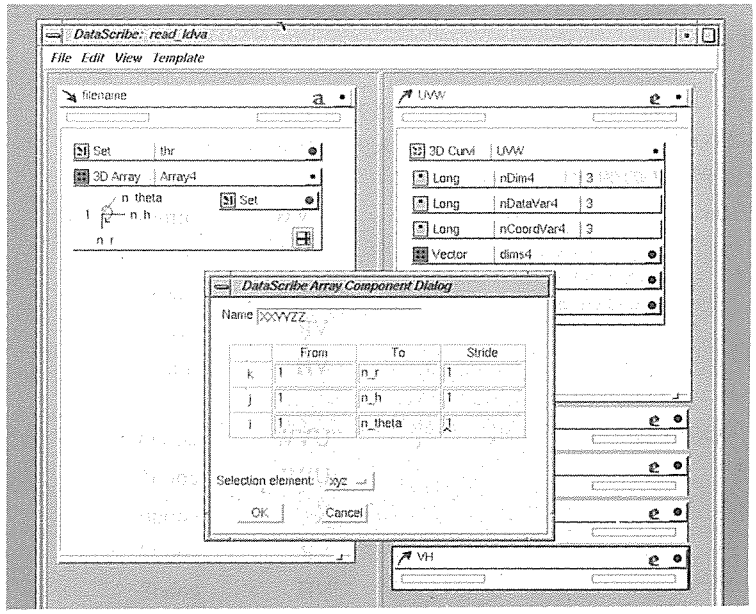

図 4(d)　データスクライブの編集（その４）

にそれぞれ xyz, uvw, vr, vh, u'v', k と名前を付け，対応する要素の所のみ緑色のスイッチを ON にする．例えば XYZ の場合，Name を xyz とし，一番上の XYZ の所のみ緑色にして，最後に OK をクリックする（図４(c)参照）．

Set を閉じた後，同様に 3D Array Array4 の中の緑色ボタン上でマウスの右ボタンでドラッグして <New> を選択し，DataScribe Array Component Dialog を出す．そして，xyz, uvw, vr, vh, u'v', k のそれぞれの要素に対して XXYYZZ, UUVVWW, VVRR, VVHH, U'U'V'V', KK と名付ける．例えば XYZ の場合，Name を XXYYZZ とし，Selection element のオプションスイッチを押し，エレメントの中から xyz を選び，最後に OK をクリックする（図４(d)参照）．この名前（XXYYZZ 等）でそれぞれの値を，入力テンプレートから出力テンプレートに引き渡すことになる．

Component Menu Button を使って行なっていることは，座標データ，速度ベクトルおよびその他の物理量を要素とする３次元配列から，座標データ，速度ベクトルおよびその他の物理量それぞれだけを要素とする３次元配列を作り出しているのである．

３．１．２　出力テンプレート

５つの出力テンプレート（UVW, U'V', K, VR, VH）は，それぞれ速度ベクトル（$\overline{u}$, $\overline{v}$, $\overline{w}$），変動速度の相互相関（$\overline{V'_h V'_r}_{3D}$），$\overline{k}_d$値（$\overline{k}_{3D}$），h 軸および r 軸方向の速度ベクトル（$\overline{V}_r$, $\overline{V}_h$）の３次元不定形格子データを出力する．パレットからそれぞれの出力テンプレートに 3D Curv をドラッグし，右丸をクリックして開き，各要素の次元を入力する．3D Curv UVW の nData Var には３，nCoordVar には３，3D Curv U'V', K, VR, VH の nData Var にはそれぞれ１，nCoordVar にはそれぞれ３を入力する（図４(b)参照）．

３．１．３　入出力テンプレートの接続と保存

入力テンプレートの出力ポートパッドから各要素を選択して，それらを，以下のように出力テンプレートの入力ポートパッドの各要素に接続する．これらの接続操作はマウスの右ボタンで行なう．

入力テンプレートの出力ポートパッド		出力テンプレートの入力ポートパッド	
次元（データの個数）			
thr	-----	UVW	dims4
		U'V'	dims5
		K	dims6
		VR	dims7
		VH	dims8
座標データ			
XXYYZZ	-----	UVW	cood4
		U'V'	cood5
		K	cood6
		VR	cood7
		VH	cood8
各データ			
UUVVWW	-----	UVW	data4
U'U'V'V'	-----	U'V'	data5
KK	-----	K	data6
VVRR	-----	VR	data7
VVHH	-----	VH	data8

完成したモジュールは File メニューの Save As で read_ldva と名付けて保存する．

3．2　マップの作成

/usr/explorer/bin/explorer と入力して，Explorer を立ち上げた後，File メニューの Open を使って DataScribe で作成したファイル入力モジュール read_ldva を呼び出す．ここでは$\overline{k}_{3D}$値の分布のカラー表示を例にとって，その縦断面内分布を表示させるための OrthoSlice と LatToGeom を２つずつ（バーナ中心軸を挟んで左右両側に展開するため２組必要であるため）と，MinMax および GenerateColormap を，３次元等値線（ワイアフレーム）を描くための Contour，カラーバーの値の表示のための Legendを，そして画像表示モジュール Render ２つをそれぞれモジュールライブラリからマップエディタ上にドラッグする．

これらのモジュールの入力および出力ポートパッドを，マウスの右ボタンを使って以下のように接続する．

モジュール名／出力ポートパッド	モジュール名／入力ポートパッド
read_ldva／K -- Lattice	OrthoSlice／Input --Lattice
	OrthoSlice<2>／Input --Lattice
	MinMax／Input -- Lattice
	Contour／Input -- Lattice

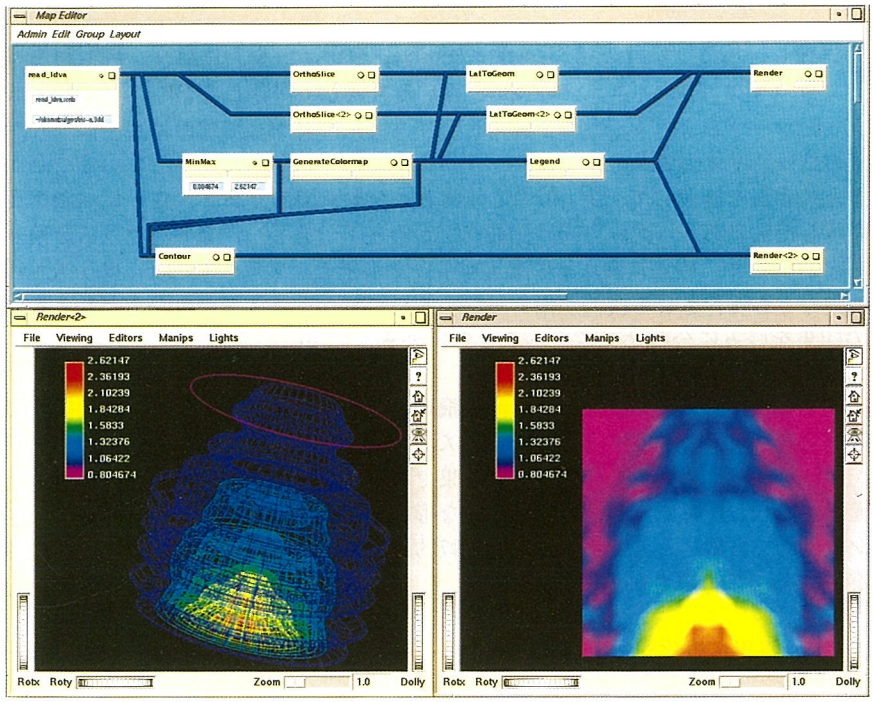

図 5　可視化マップと出力例

orthSlice／Slice -- Lattice
orthSlice<2>／Slice -- Lattice

MinMax／Minimum

MinMax／Maximum

GenerateColormap／Colormap -- Lattice

LatToGeom／Output -- Geometory
LatToGeom<2>／Output -- Geometory
Contour／Contours -- Geometory
Legend／Legend -- Geometory

LatToGeom／Input -- Lattice
LatToGeom<2>／Input -- Lattice

GenerateColormap／MinDomain
Contor／MinLevel
GenerateColormap／MaxDomain
Contour／MaxLevel

LatToGeom／colormap -- Lattice(Opt)
LatToGeom<2>／colormap -- Lattice(Opt)
Contour／colormap -- Lattice(Opt)
Legend／colormap -- Lattice(Opt)

Render／Input -- Geometory
Render／Input -- Geometory
Render<2>／Input -- Geometory
Render／Screen -- Geometory
Render<2>／Screen -- Geometory

ファイル入力モジュールにおいて加工した可視化したいデータファイルの名前を入力すれば，Render に図が出力されることになる．縦断面および横断面の位置は，それぞれの OrthoSlice のス

ライダーで調節し，色付けの仕方はGenerateColormapモジュールで行なう．再度利用出来るように，作成したマップはFileメニューのSave Allを選択して保存しておく．

今回作成したマップおよび$\overline{k}_{3D}$値の2次元および3次元に展開した出力結果を図5に示す．この図から油滴の持つ速度変動のエネルギがバーナポート出口から急激に減少していることが分かる．

図6は上述の手法を踏襲して，$\overline{k}_{3D}$値の等値面および縦断面内分布と，さらに油滴速度ベクトルも同時に表示した例である（流れ方向は画面右上から左下である）．なお，マウスを使って見せたい方向に図を回転させ，視覚に訴える図を得ることが出来る．また，Renderには3次元画像への光を当てる方向などを変化させる機能が備わっているのでコントラストや色調などもそれに伴って変わり，図の印象を好みに合わせて変えることが出来る．

4．まとめ

レーザードップラー流速計で計測された油滴飛行速度の2次元データを，3次元空間に展開して仮想的な3次元イメージに加工することを試みた．*Explorer*上で一度この画像処理マップを作成しておけば，データ配列は変わるものの，実験結果であれ，数値計算結果であれ，軸対称2次元データが3次元表示できる．特に，等値面は空間的な形状を直ちに視覚に訴えてくれるのでデモンストレーションに有効である．ここで扱った例は比較的単純な流れ場だったが，複雑な流れ場になる程に威力を発揮するものと思われる．

参考文献

1. 出口　弘，汎用可視化ツール*Explorer*の使い方，大阪大学大型計算機センターニュース，22-2 (1992), p.66.
2. 小林一男，蛋白質データベースを*IRIS-4D*で可視化する方法，大阪大学大型計算機センターニュース，22-4 (1993), p.87.

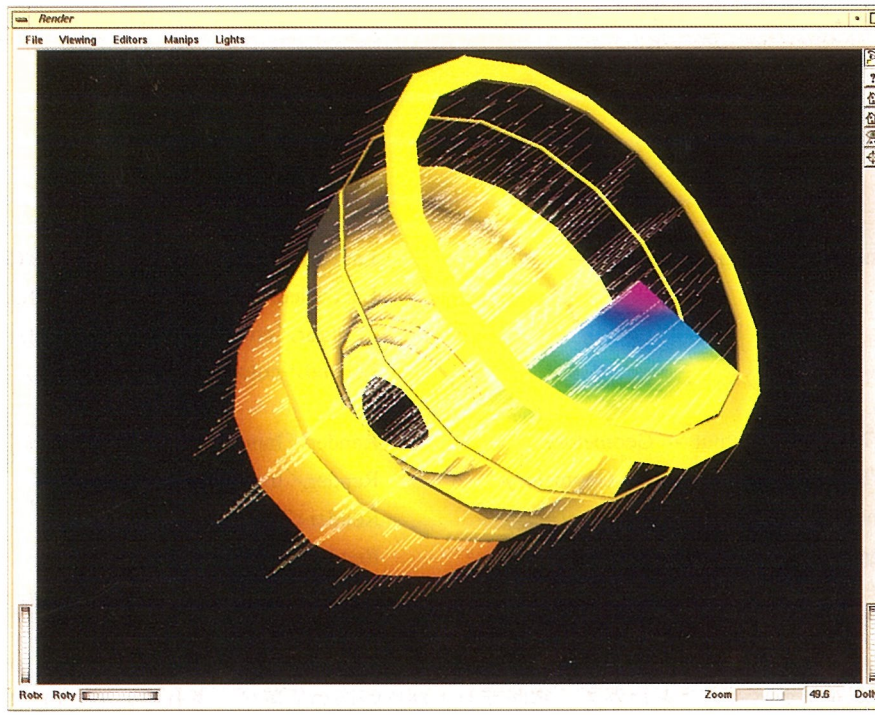

図 6　K_{3D}値の断面分布および等値面と油滴速度ベクトル