

# レーザー干渉実験用 画像データ収集システムの開発 ～新しい放射線検出技術を目指して～

指導教官 宮村 修 教授

広島大学理学部物理学科  
ハドロン物理学研究室  
0872039B 和田 康則

2000年2月10日

## Abstract

電場を伴う放射線は、物質中で分極を引き起こし原理的に屈折率の変化をもたらすはずである。一般に物質のわずかな屈折率の変化を観測するのに干渉計が応用されることが知られている。このことから干渉計を利用して放射線の影響を観測出来るのではないかと着想し、本研究ではそのための干渉計について考察を行なうことを目的とした。そこで、Michelson型をベースとした干渉計を設計・製作し、干渉画像を収集するシステムを構築した。このシステムはパルスレーザー光に同期して動作し、放射線が入射した場合にのみ CCD カメラで撮った干渉画像を確実にコンピュータに取り込むものである。実際に取り込んだ画像データを解析することで干渉計の性能の評価を行なった。

# 目次

1	序論	2
2	干渉計の設計・製作	4
3	データ収集システムの開発	7
3.1	レーザーとの同期	7
3.2	画像収集プログラム	8
3.3	画像キャプチャのタイミング	10
4	実験方法	14
5	データ解析	16
5.1	Kolmogorov テスト	16
5.2	光路上の静的な変化の影響	19
5.3	白色光によるシンチレータの静的な励起構造の変化	21
6	結果	26
7	考察	26

# 1 序論

電磁場を伴う粒子が物質に入射する場合、もしその電磁場の振動数  $\omega$  が物質中の全ての ( 或いは少なくとも大多数の ) 電子の振動数より大きければ、分極を考えると、物質中の電子同士の相互作用は無視でき、電子は自由と考えることが出来る。

物質中の原子内電子の速度  $v$  は光速に比べて小さいので、波 1 周期の間に通過する距離  $v/\omega$  は波長  $c/\omega$  に比べて小さい。したがって、電子から見れば電磁場は一様とみなすことが出来る。

原子内電子の運動方程式は

$$m \frac{dv'}{dt} = eE = eE_0 e^{-i\omega t} \quad (1)$$

である ( ここで、 $e$ 、 $m$  は電子の電荷及び質量 )。これより、 $v' = ieE/m\omega$  である。電場の影響による電子の変位  $r$  と  $v'$  の関係は  $\dot{r} = v'$  である。したがって、 $r = -eE/m\omega^2$  である。物質の分極  $P$  は単位体積当りの双極子モーメントであるから、全ての電子について加えあわせると

$$P = \sum er = -\frac{e^2}{m\omega^2} N E \quad (2)$$

となる (  $N$  は単位体積当りの全分子中の電子数 )。一方、電気誘導の定義から  $D = \epsilon E = E + 4\pi P$  である。これより以下の式を得る。

$$\epsilon(\omega) = 1 - \frac{4\pi N e^2}{m\omega^2} \quad (3)$$

一方、物質の屈折率  $n$  は

$$n = \frac{c}{v} = \frac{\sqrt{\epsilon_0 \cdot \mu_0}}{\sqrt{\epsilon \cdot \mu}} \quad (4)$$

から求められる。ここで  $v$  は物質中での光速、 $\epsilon$ 、 $\mu$  は物質の誘電率、透磁率である。以上のことから物質中に電磁場を伴った粒子が入射した場合、その物質の屈折率が変化することがわかる。

一般的に物質のわずかな屈折率の変化を観測するために干渉計が用いられる。放射線のなかには電磁場を伴った粒子であるものも存在する。このことから、干渉計を使ってわずかな屈折率の変化として現れる放射線を観測できるのではないかというのが、本研究が最終的に目指す新しい放射線検出技術の考え方である。

今回はそのための基礎的な研究として、放射線の入射による屈折率の変化のような微小な変化ではなく、シンチレータの励起構造の静的な変化のといった、より大きな変化が干渉計を使って観測できるかどうかを調べた。

干渉計には二光束干渉計を採用した。これは単一光源からの光を二つに分岐して、別々の光路をとらせた後、再び重ねて干渉を起こさせるというものである。この実験ではこれを応用して分岐した片方の光の光路上に変化を与えて、そのときの干渉模様を観測した。

## 2 干渉計の設計・製作

今回の実験で実際に使用した干渉計の全体図を図 1 に示す。この図において”M” はミラー、”BS” はビームスプリッター（半透明鏡、とくに透過率と反射率が 1:1 のものをハーフミラーという）を表している。

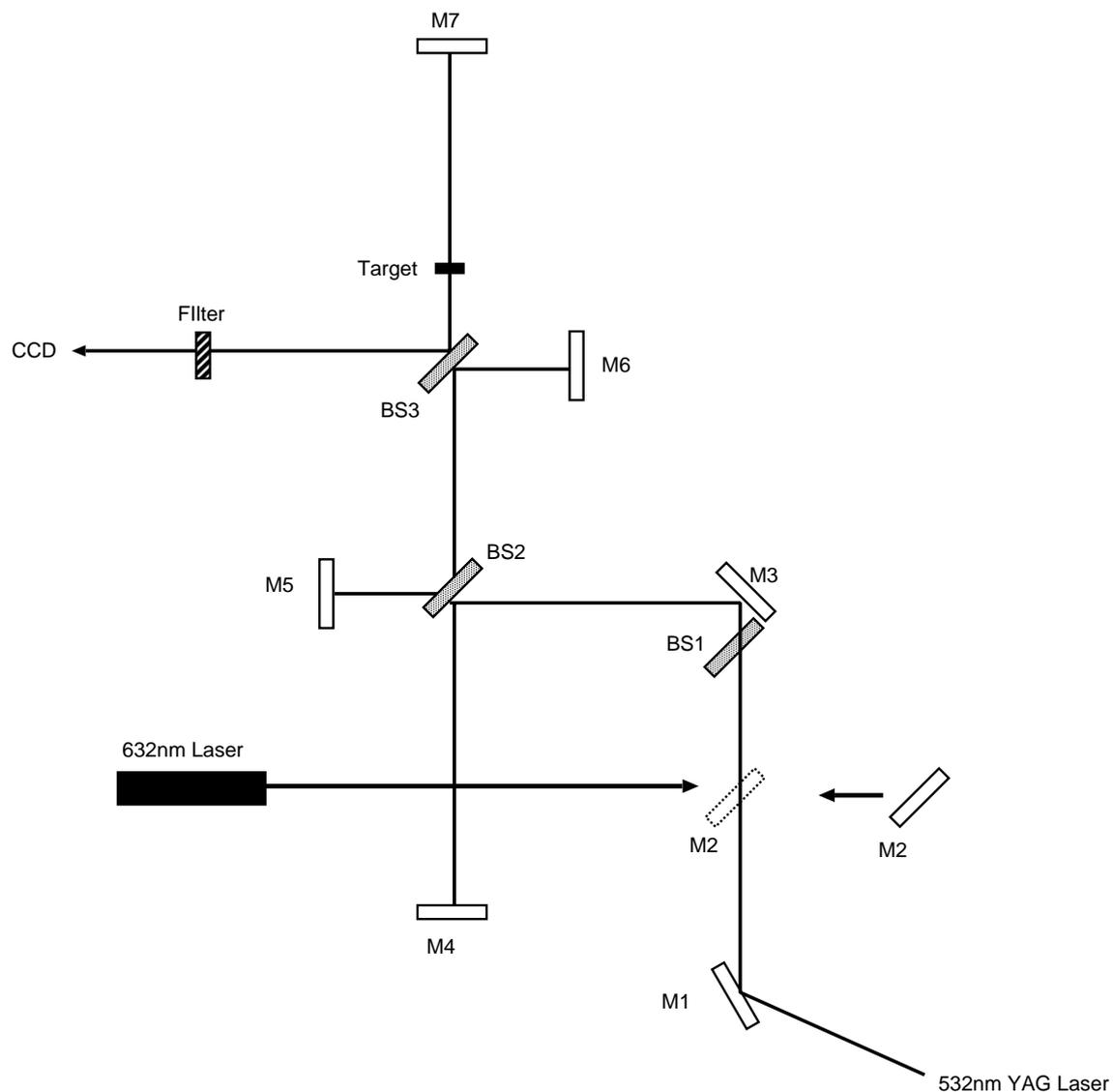


図 1: 干渉計の全体図

これは Michelson 型干渉計を二つ組み合わせたものである。Michelson 型干渉計は、平行光束をハーフミラーで二つに分岐し、それぞれの光路の先に平面ミラーをおいて、各々の光束をもとの方向に戻してハーフミラー上で重ね合わせ、そのときできる干渉光を観測するものである。図 1 において M1、M2 は実際には光軸の高さ、方向を変えるために 2 つのミラーを垂直方向に組み合わせたもので、光路が交差しているように見えるところでは実際には光軸の高さは異なっている。また、M2 は角度を保ったまま水平方向に移動できるステージの上に乗っており、これを動かすことで干渉計に入れる光源を波長 632nm の He-Ne ガスレーザーと波長 532nm の YAG パルスレーザーとに切替えることができる。

Target の場所には、干渉模様に変化が現れるかどうかを見るための障害物を置く。BS1 は透過光と反射光の強度比が 3:5 にしてあり、Target にシンチレーターを置いたときに使う。シンチレーターとは、これを構成する原子、分子が通過していく電場を伴った粒子のエネルギーを吸収して励起し、元のエネルギー準位に戻るときにそのエネルギー差を光として放出するものである。詳しくは後の節に譲るが、このシンチレーターを励起するために、BS1、M7、M8 を経由したレーザー光を使った。

図 2 からわかるように Michelson 干渉計を 2 つ組み合わせているので、最終的に CCD カメラに入るレーザー光は 4 つ ( 実際にはそのうちの 2 つが干渉するのでスポットは 3 つ : 図 4 参照 ) になる。CCD カメラで撮影する画像には干渉光のほかに直接光が映る。この直接光を使い画像データ間で生じるレーザー強度の揺らぎによる誤差をある程度補正することができる。

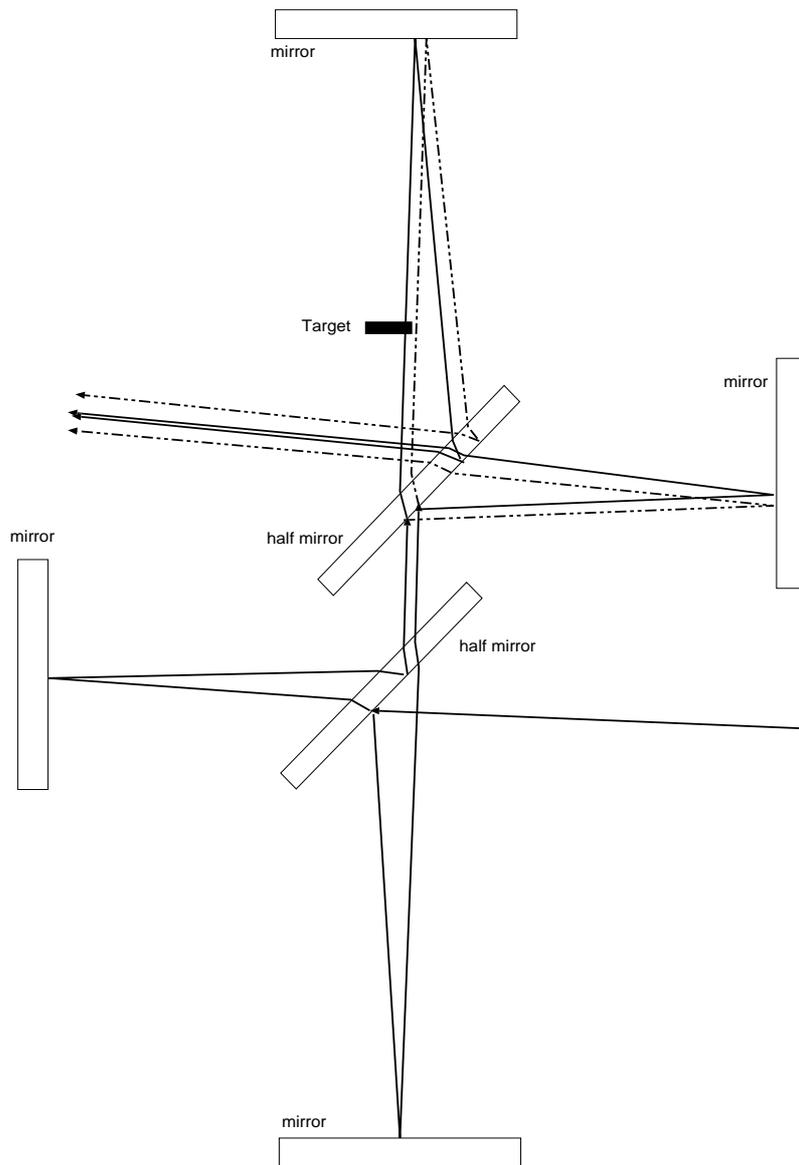


図 2: 光路の詳細

### 3 データ収集システムの開発

前節までで述べた干渉計によって実際に干渉模様を観測するために、コンピュータと、エレクトロニクス機器の CAMAC(Computer Automated Measurement and Control) を使用した。

#### 3.1 レーザーとの同期

画像データをパソコンに取り込むために、ビデオキャプチャカード (I・O DATA 社製 GV-VCP/PCI<sup>1</sup>) を使用した。ビデオキャプチャカードとは、ビデオ信号をデジタルデータ化してパソコンに取り込むためのものである。CCD カメラからのビデオ信号化された画像データは常にキャプチャカード上に流れてくるが、パルスレーザーを観測している場合データの殆どはレーザー光の映っていないものである。レーザー光が映っている画像データだけを取り込んでおきたいので、ソフトウェアを作って画像データの取り込み作業を制御することにした。

ソフトウェア側で CCD カメラにレーザー光が入ったことを知るために、CAMAC モジュールの LAM を用いた。これは、"Look At Me" の略で、モジュールに入力信号があるとそのことを知らせる信号である。CAMAC への入力信号としてレーザー本体からの同期信号を設定し、その LAM をソフトウェアの側で検知することで、画像データ取り込みのタイミングを知ることが出来る。レーザーからの同期信号と実際のレーザー光の放射の間には 15 ~ 500 マイクロ秒の時間間隔がある。レーザー光が放射されてから I.I.T. の感光面に達するまでは数ナノ秒であるので、同期信号と I.I.T. の時間間隔も 15 ~ 500 マイクロ秒である。この時間内に同期信号による LAM を検知し、ビデオキャプチャカードを動作させて画像データを取り込むことは不可能である。そのため同期信号による LAM を検知したら 1 周期あとに放射されるレーザー光を取り込むようにした。

---

<sup>1</sup>Brooktree 社製 Bt848 チップを使用しており、Bttv というドライバを用いて Linux で制御できる。

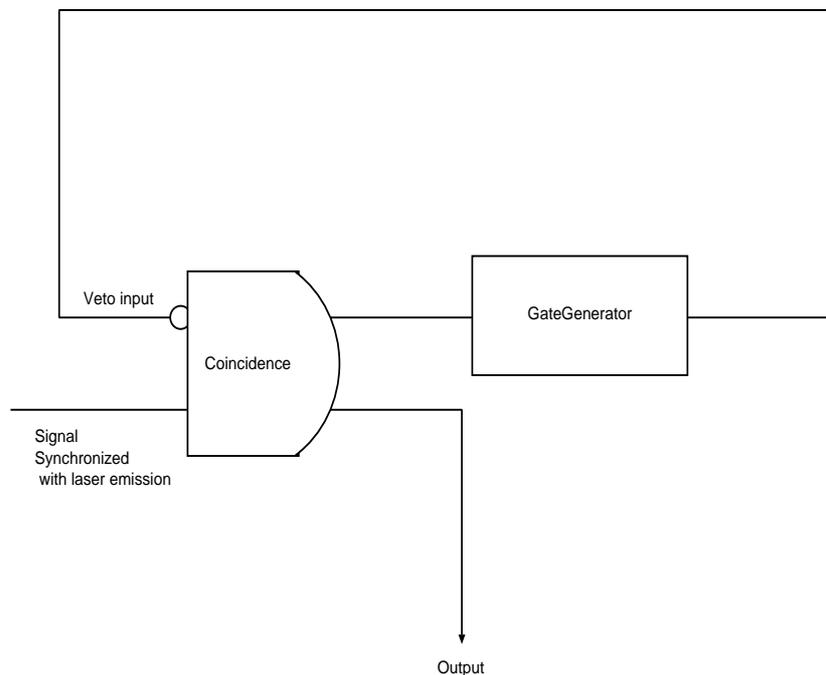


図 3: コンピュータの内部処理中に新たな信号が入るのを防ぐためのエレクトロニクス回路

LAM を立てる同期信号は 0.9 秒周期で出力される。1 イベントに対するコンピュータの内部処理時間は 0.6 ~ 1.5 秒程度である。つまりコンピュータが内部処理を行なっている間に次の同期信号が入ってきてしまう。通常こういった場合 interrupt register を使用して内部処理中は新たな信号を受け付けないようにするのだが、今回使用できない状態だったので、エレクトロニクス回路で代用した。回路図を図 3 に示す。仕組みは、まずレーザーからの同期信号を Coincidence 回路に入れ、Coincidence 回路からの出力信号を、一方は LAM を立てるべく CAMAC に入れ、他方は GateGenerator に入れる。GateGenerator は入力信号が入った直後に 1.5sec の幅のパルスを出力するよう設定しておく。この GateGenerator の信号を Coincidence 回路の Veto 入力に入れ、同期信号との論理積をとる。Veto 入力は信号が入ったとき偽、入っていないときは真になる。Coincidence 回路は最初の同期信号が入ったときはそのまま出力信号を出す。その直後から 1.5 秒間 GateGenerator を経由してきた信号の Veto 入力によって、同期信号が入ってきても出力信号を出さない。これによってコンピュータの内部処理中に LAM が立つのを防ぐことができる。

### 3.2 画像収集プログラム

LAM を検知して、画像データを取り込むまでのプログラムの動作は以下のようになっている。光源に連続レーザー光を使用する場合は Step2. は行なわない。(実際のソースコードは付録を参照)

Step 1. CAMAC に LAM を見に行く。LAM が立っていたら次のステップへ移る。立っていないときは LAM が立つまで繰り返し見に行く。

Step 2. LAM が立ってからレーザー光の画像がビデオキャプチャカードに流れてくるまでの 0.85 秒間、空のループを回して待つ。( CCD カメラにレーザー光が入るより少し前に次の動作に移る必要があるため )

Step 3. キャプチャカード上の画像データをメモリに書き出す。

Step 4. メモリ上の画像データをファイル化。

取り込んだ画像データはビットマップ形式でファイル化される。これは各ピクセル(画素)の明るさを数値データ(0~253)で表したものである。実際にとってきたバイナリファイルの画像データを ASCII ファイルに変換したものが図 4 である。

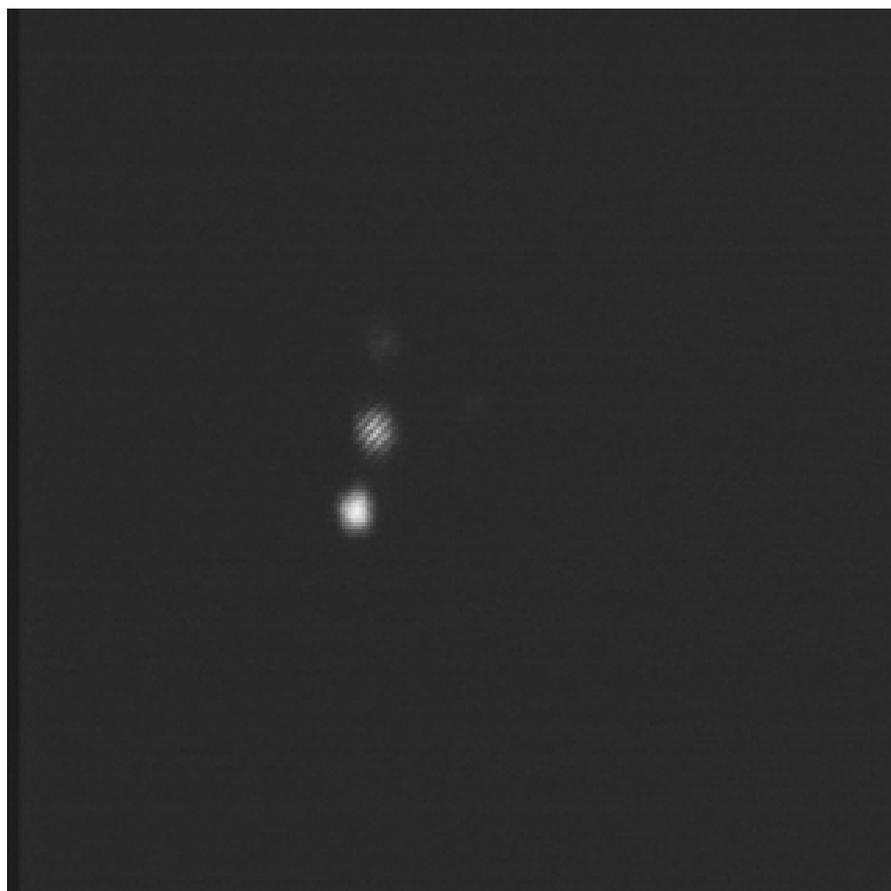


図 4: ビットマップ形式の画像データ

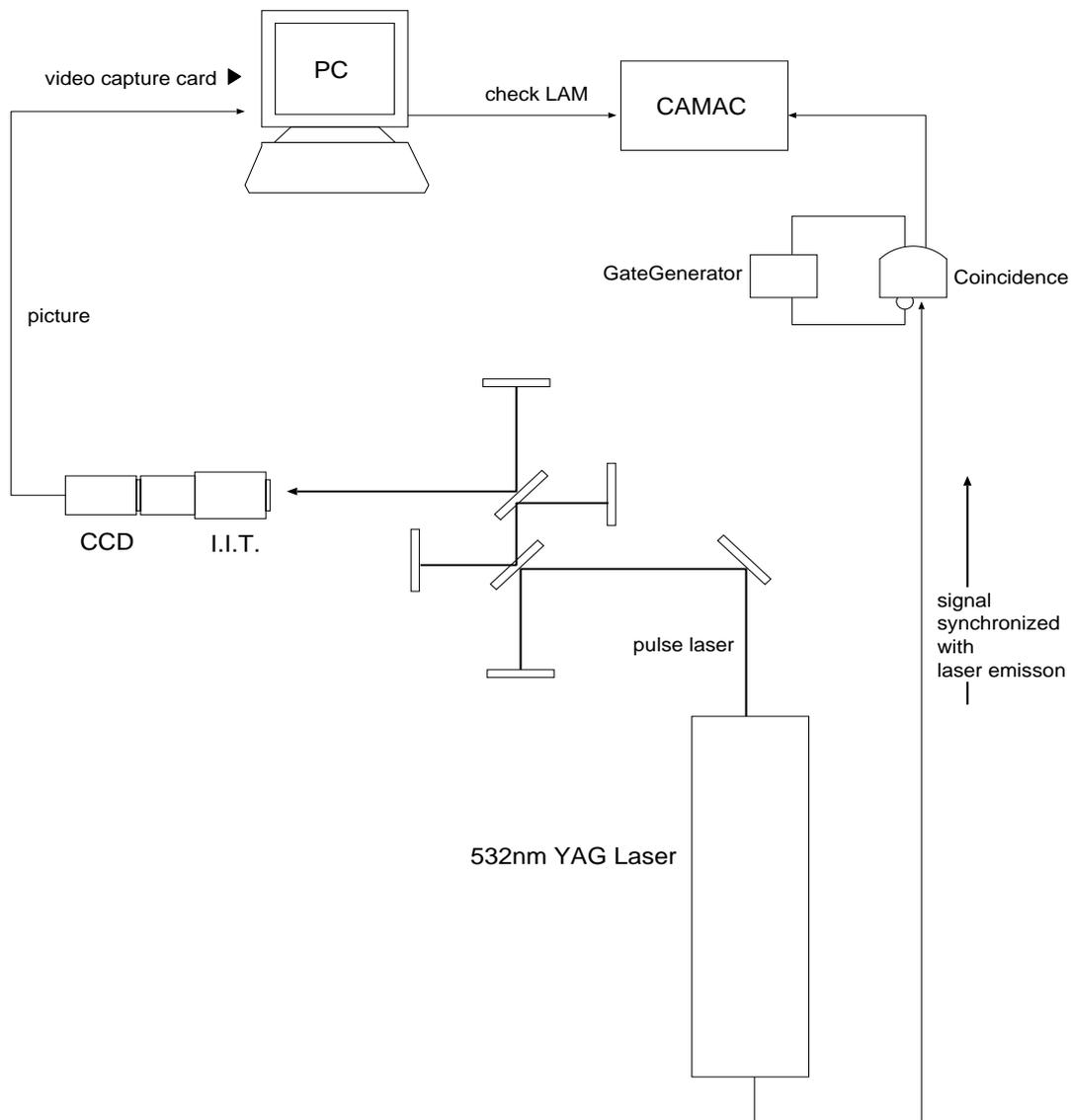


図 5: システム全体の概念図

### 3.3 画像キャプチャのタイミング

実際にこのシステムを使って画像データを集めるときにデータ収集効率を最も左右するのは、CCD カメラにレーザー光が入ってくるタイミングとビデオキャプチャカードを動作させられてくる画像データをキャプチャするタイミングである。最も効率の良いタイミングを調べるため LED (発光ダイオード) を使ってテストを行なった。各部分の動作するタイミングを図 6 に示す。

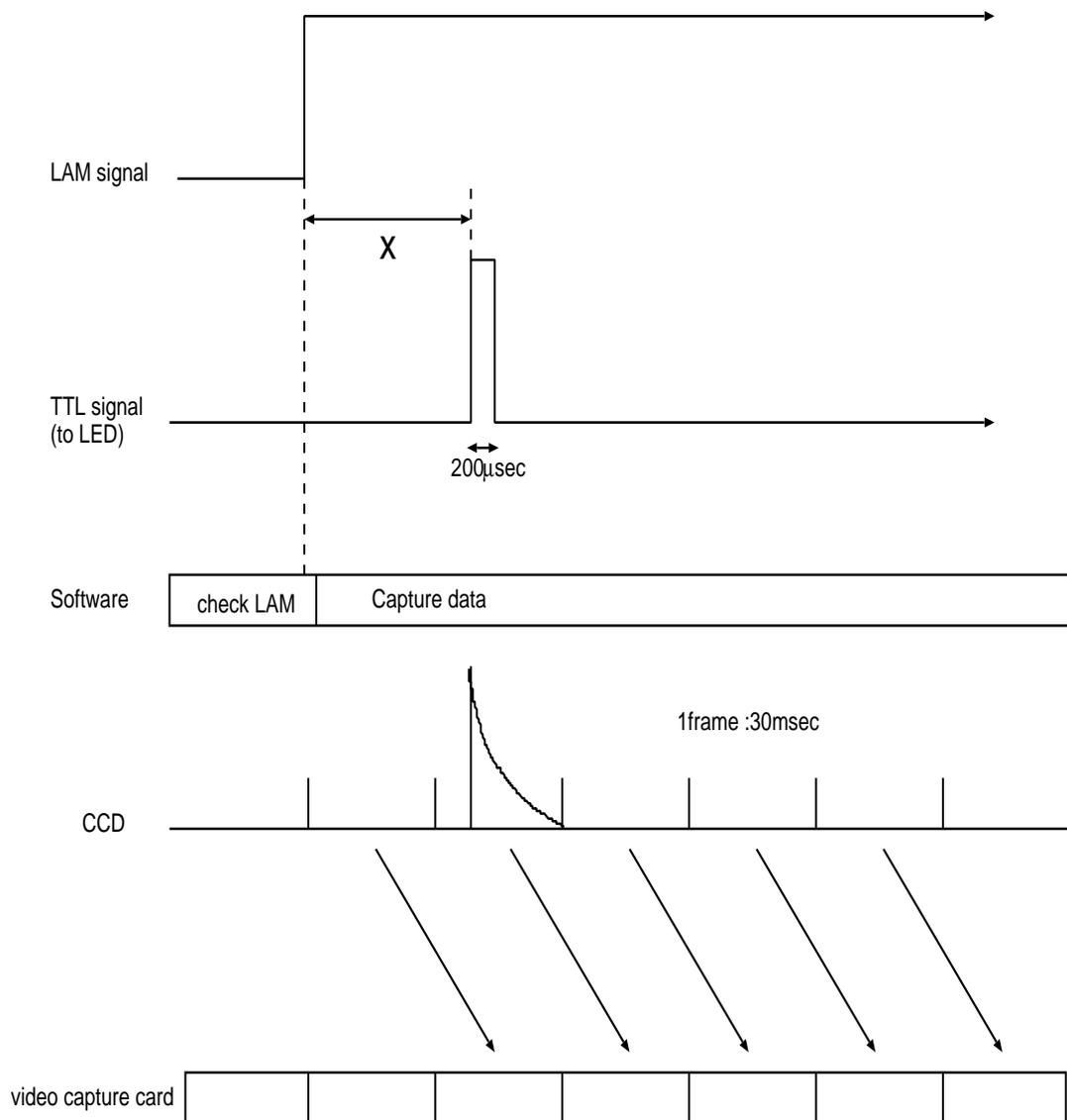


図 6: LAM シグナル、LED の発光、ソフトウェアの動作のタイミングチャート

図 6 の x を 10 ミリ秒刻みで変化させ、それぞれ 50 枚データをとったときの有効データの数を図 7 に示す。ここでは画像データ中の光の強度が最も高いピクセルの値が 170 ~ 253 の範囲内のものを有効データとしている。これから最も効率良く欲しい画像データを集めるには、その事象が CCD カメラに写る 30 ミリ秒前からビデオキャプチャカードの動作を開始すれば良いことがわかる。ビデオキャプチャカードの動作とは、キャプチャカード上に流れてきた画像データをメモリ上に書き出す作業とその同期をとる作業（付録ソースコード 199 行目、201 行目）である。60 ~ 70 ミリ秒かかるこの一連の作業が画像データを収集する際大きく効いている。しかし一番効率の良いタイミングでも 100 %データをとることができていない。そこで、x を 30 ミリ秒に固定し 100 枚データをとったときの各データ中

で最も光の強度の高かったピクセルの値の分布を調べた。そのプロットが図 8 である。図 8 から CCD カメラが受光するのとビデオキャプチャカードが動作するタイミングが一定であるにも関わらず、とれた画像データの光の強度は揺らいでいることがわかる。これは CCD カメラの性質によるものであると考えられる。今回の実験で使用している CCD カメラから流れてくる画像データは 30 ミリ秒間カメラに入って来た光が積分されて 1 フレーム ( 1 画面 ) 分のデータになったものである。LED 自体の発光時間は 200 マイクロ秒 ( 実際には LED に送っているパルスの幅 ) であるが CCD に組み合わせている I.I.T. の影響もあり一度入った光の強度が 10 % 以下になるには数ミリ秒を要する。したがって 1 フレームの 30 ミリ秒のうちどの時点で光が入って来るかによって出力されるデータに表れる光の強度は異なってくるのである。

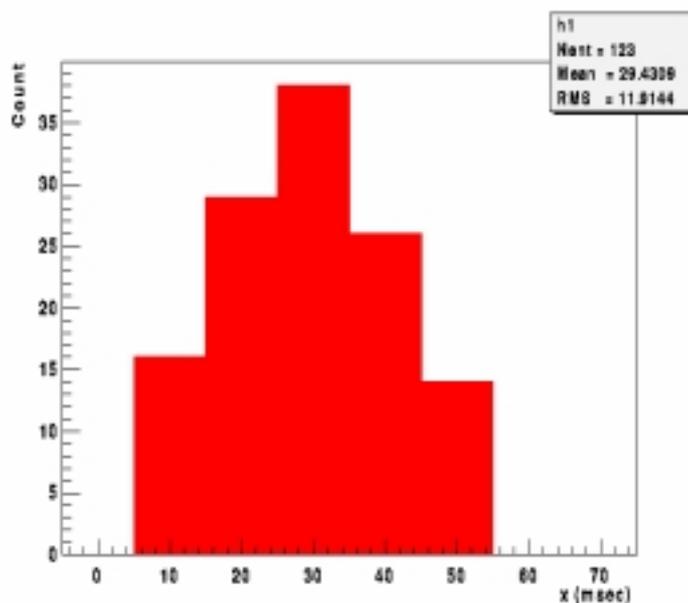


図 7:  $x$  10 ミリ秒刻みで変化させたときの画像データ 50 枚中の有効データ数

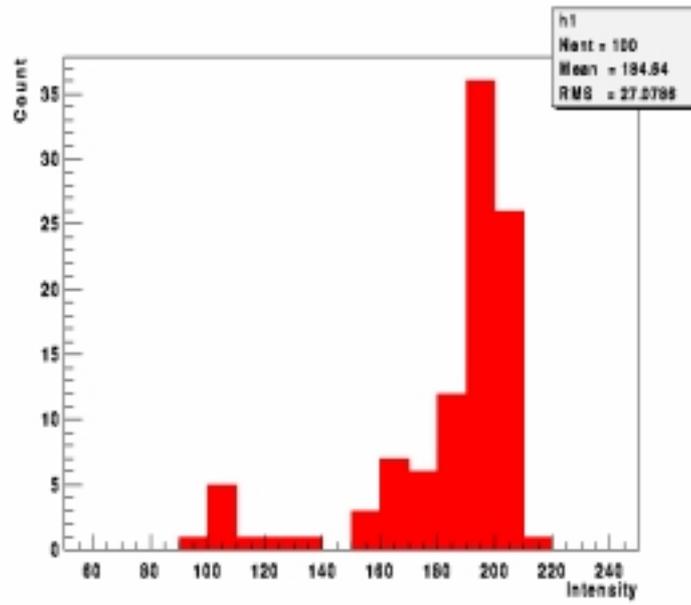


図 8:  $x = 30$  ミリ秒での画像データの光の強度の分布

## 4 実験方法

前節迄で述べたデータ収集システムを使って、光路上の変化が干渉模様及ぼす影響を調べた。

実験で使用したレーザーを表 1 にまとめた。

### 方法 1 連続レーザー光 - 障害物

まず、干渉計が期待するものとなっているかを調べるために、干渉模様明らかに変化を及ぼしそうなものとして図 1、図 2 の Target のところにビーム径 ( 約 2mm ) に比べ十分に細い糸状の障害物を置き、波長 632nm の He-Ne レーザー ( 連続レーザー光 ) を光源に使用して画像データをとった。具体的なデータのとり方は

条件 A : 光路上に障害物を置かない

条件 B : 光路上に障害物を置く

として、条件 A で 10 枚、次に条件 B で 10 枚、最後にもう一度条件 A で 10 枚画像データをとった。このとき PulseGenerator から 1.4 秒周期で信号を CAMAC に送って LAM を立て、画像キャプチャのタイミングとした。連続レーザー光であるので常に CCD には光は入って来ており、当然レーザーの放射と同期をとる必要はない。また、1 回の LAM について画像は 1 フレーム分とった。1.4 秒という周期は 1 フレーム分のデータをコンピュータ内で処理し終るのに十分な時間と考えて設定した。

He-Ne ガスレーザー (連続レーザー光)

波長 : 632.8nm

最大出力 : 5mW

ビーム径 : 0.65mm

ビーム拡がり角 : 約 1.23m rad

日本電気製

Nd:YAG レーザー (パルスレーザー)

波長 : 532nm

パルス幅 : 5nsec

周期 : 10Hz(10/n Hz ; n=1 ~ 9 で可変)

最大出力 : 600mW

( B.M.Industries 社製、広島大学ビーム物理学研究室所有 )

表 1: 実験で使用したレーザー機器

## 方法 2 連続レーザー光 - 白色光によるシンチレータの静的な励起

図 1、2の Target の位置にシンチレーターを置き白色光を当ててシンチレータを励起させ、連続レーザー光を使ってデータをとった。データのとり方は方法 1 と同じで

条件 A : シンチレータを励起させないとき

条件 B : 白色光で励起させたとき

として、A10 枚、B10 枚、A10 枚の順にとった。

## 方法 3 パルスレーザー - 白色光によるシンチレータの静的な励起

シンチレータの置き方、白色光の当て方は方法 2 と同じで、光源に波長 532nm のパルスレーザーを使用した。データのとり方は方法 2 の条件 A、B を画像データ 1 枚毎に交互に計 1000 枚とった。

尚、これらすべてのデータ収集において I.I.T. には常に一定の電圧 ( 連続レーザー光とパルスレーザー光では I.I.T. 入射時の強度が違うのでかける電圧の値も異なる ) をかけた状態にした。レーザー光が I.I.T. に入って来る度にその瞬間 ( 数百マイクロ秒 ) だけ電圧をかけると増幅率が安定せず、そのままデータの光の強度の揺らぎとなってしまう。そういった余計な揺らぎを防ぐためにこのような設定でデータをとった。

## 5 データ解析

### 5.1 Kolmogorov テスト

CCD カメラから流れてくるビットマップ形式の画像データを比較するために、 $320 \times 240$  ピクセルの画面を  $x - y$  平面に、光の強度を高さで表した三次元のプロットが図9である。

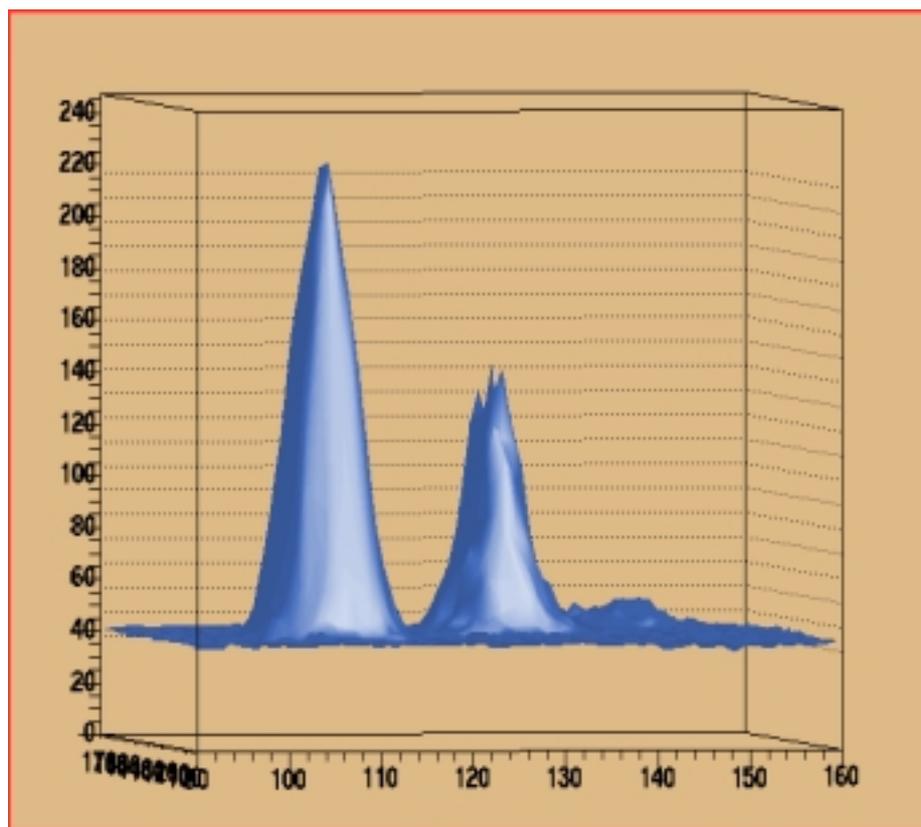


図9: 観測されるレーザー光の強度分布を3次元プロットにしたもの

この二次元ヒストグラムの形の違いを定量的に評価するため、ヒストグラムを比較する統計学的手法で Kolmogorov テスト [4,5] というものを用いた。このテストは元々、ある事象がそれを予言する仮説にどれほど従うかをテストするものである。今回の場合では、二つのプロットのうちの一方を基準とし、他方がどれほど違っているかを調べた。まず、二つのプロット各々について、各ピクセルの強度を全ピクセルの強度の合計で割って確率密度関数を求める。これは  $x - y$  平面上の確率密度関数となる。この確率密度関数にしてやったもの同士を Kolmogorov テストを行なうプログラムで比較した。以下、kolmogorov テストについて簡単に説明する。ここでは簡単のため変数は  $x$  のみとする。まず、二つの確率密度関数  $P_1(x)$ 、 $P_2(x)$  について、これらが0でない値を持つ領域  $x = x_0 \sim x$  について  $x$  の値を変化させながら積分した  $F_1(x)$ 、 $F_2(x)$  を求める。

$$\begin{aligned}
 F_1(x) &= \int_{x_0}^x P_1(x) dx \\
 F_2(x) &= \int_{x_0}^x P_2(x) dx
 \end{aligned}
 \tag{5}$$

この  $F_1(x)$ 、 $F_2(x)$  のずれ

$$d = \max|F_1(x) - F_2(x)| \tag{6}$$

を指標として二つの分布の類似性を評価する。 $P_1(x)$ 、 $P_2(x)$  は実際にはある  $x$  に対する規格化された光の強度分布なので、 $F_1(x)$ 、 $F_2(x)$  は図 10 の  $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$  のような階段関数になる。よって  $d$  は

$$d_{m,n} = \max|F_n^{(1)}(x) - F_m^{(2)}(x)| \tag{7}$$

となる。ここで、 $m$ 、 $n$  はそれぞれ、規格化された光の強度が 0 でない点の個数である。

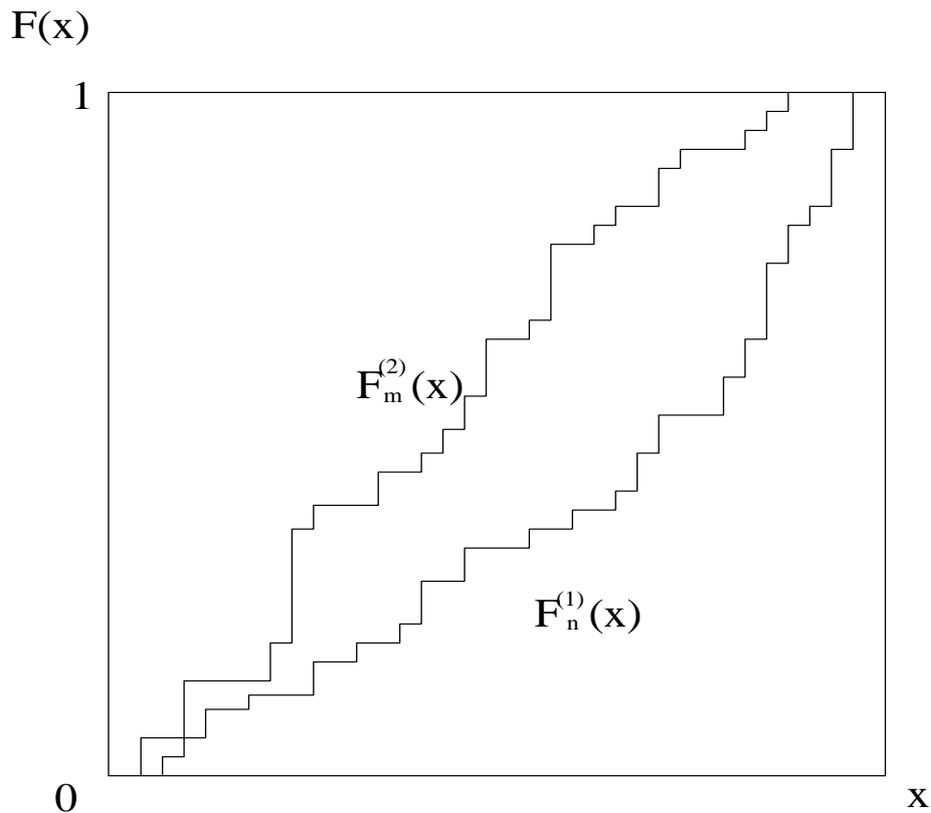


図 10: 階段関数になる  $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$

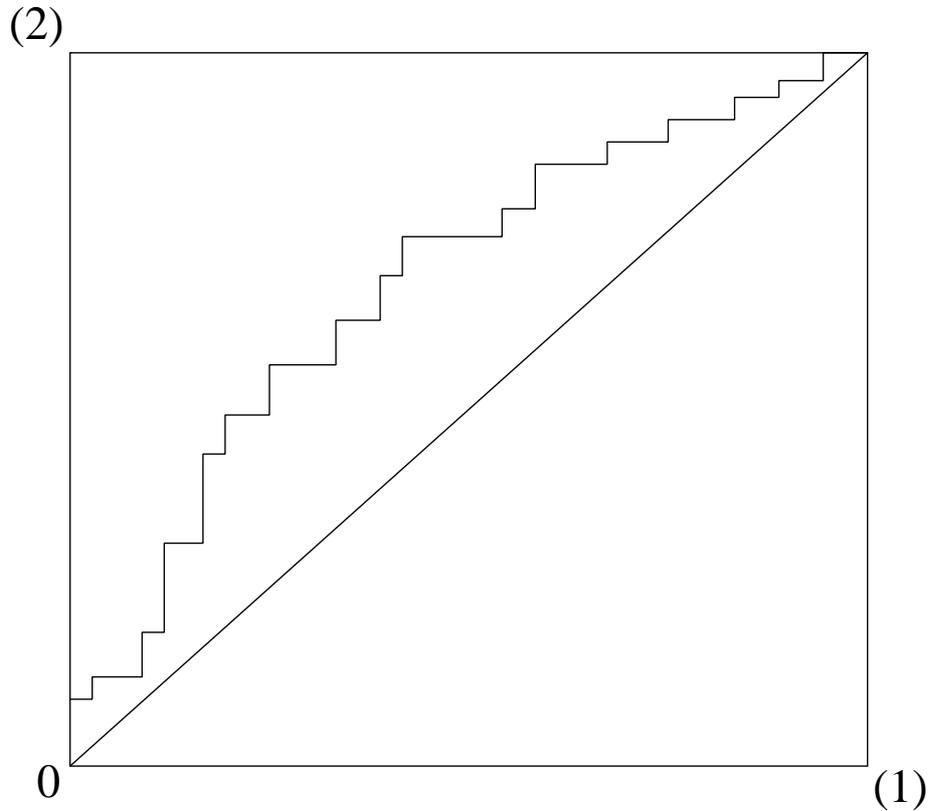


図 11:  $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$  の関係

図 10 を書き換えて、 $F_n^{(1)}(x)$  を横軸、 $F_m^{(2)}(x)$  を縦軸にとったのが図 11 である。これは  $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$  の各  $x$  に対応した  $F(x)$  の値をまぜて大きい順に並べ、 $(0,0)$  を出発し、 $F_n^{(1)}(x)$  からの値がきたら右へ 1 単位、 $F_m^{(2)}(x)$  からの値なら上へ 1 単位移動する。 $F_n^{(1)}(x)$  と  $F_m^{(2)}(x)$  で同じ値がきたら、ななめに進むとすると、 $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$  が全く同じものときは図 11 の直線と重なる。以下、簡単のため、 $m = n$  とする。

実際に類似の度合を  $0 \sim 1$  の数値であらわす probability  $p(k, n)$  は

$$p(k, n) = Pr[d_{n,n} \geq \frac{k}{n}] \quad (8)$$

で定義する。 $d_{n,n}$  は図 11 の折れ線上の点から横か縦にはかった直線までの距離を  $n$  で割ったものである。

$Pr[d_{n,n} \geq k/n]$  の意味するところは、図 11 において折れ線が直線から横、及び縦に  $k$  単位離れた範囲内に  $d_{n,n}$  が収まらない確率である。 $k$  を大きくすれば当然  $p(n, k)$  は小さくなり、 $k = n$  のところで 0 となる。 $k = n$  で  $d_{n,n} \geq k/n$  を満たす  $d_{n,n}$  は 1 以上であるが、確率密度関数  $P_1(x)$ 、 $P_2(x)$  の各  $x$  に対する積分値のプロットである  $F_n^{(1)}(x)$ 、 $F_m^{(2)}(x)$  の差  $d_{n,n}$  が 1 以上になることはない。 $k$  を 1 から増やしていき  $d_{n,n} \geq k/n$  が満たされたときの最小の  $k$  での  $p(k, n)$  の値が  $P_1(x)$ 、 $P_2(x)$  の類似の度合 (probability) である。

$Pr[d_{n,n} \geq k/n]$  はある  $k, n$  に対して図 11 の折れ線が  $k$  の範囲を越えずに通る全経路数に対する  $d_{n,n} \geq k/n$  を満たす経路の数の割合である。一般に  $k$  によって決まる範囲を越えることなく  $(0, 0)$  から  $(r, s)$  ( $n = m$  で考えている場合は  $r = s$ ) にいたる全経路数  $f(r, s)$  は、以下の漸化式

$$\begin{cases} f(r, s) = f(r-1, s) + f(r, s-1) & (r \neq 0, s \neq 0) \\ f(r, s) = 1 & r = 0 \text{ または } s = 0 \text{ のとき} \\ f(r, s) = 0 & |r - s| > k \text{ のとき} \end{cases} \quad (9)$$

より与えられる [5]。

画像データを比較する際、干渉模様ではなくレーザーの強度の違いによって probability の値が大きく変わってしまうのを防ぐため、1 画面のうちで最も明るいピクセルの強度が全 254 段階のうちの 160~253 だけのもをとり、更に干渉していないレーザー光部分のピクセルの強度の合計で干渉光部分の各ピクセルの強度を割って標準化したもので Kolmogorov テストによる比較を行なった。最も明るいピクセルの強度が 160 以上という条件は実際に画像データを何枚か見て、強度の標準化を行なっても明らかに干渉模様の違い以外のところで probability が大きく変わりそうな画像データを除ける適当な値として決めた。

## 5.2 光路上の静的な変化の影響

光路上に細い糸状の障害物をいれたときの干渉光の変化を調べるためのデータは、前節で述べたように

条件 A、C : 障害物を入れていないとき

条件 B : 障害物を入れたとき

で A、B、C、の順で各条件とも 10 枚ずつのデータをとった(ここで条件 B の前後での障害物を入れてないデータを区別するため条件 C を設定した)。異なる条件のデータそれぞれの中から任意に選んだ 2 つのデータ同士を比較し、Kolmogorov テストによる probability をプロットしたものが図 12、13である。

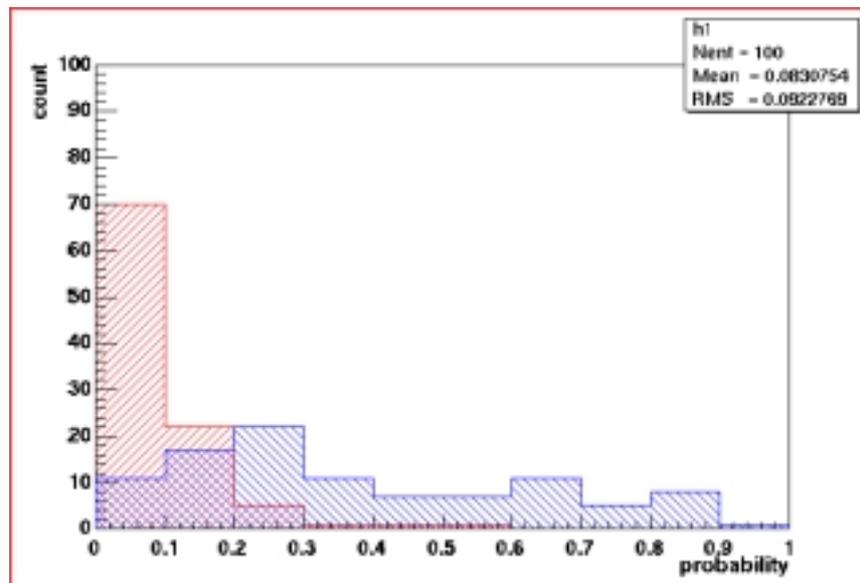


図 12: 条件 A、B の画像データを Kolmogorov テストで比較し、probability をプロットしたもの

青 : A - C の比較      赤 : A - B の比較

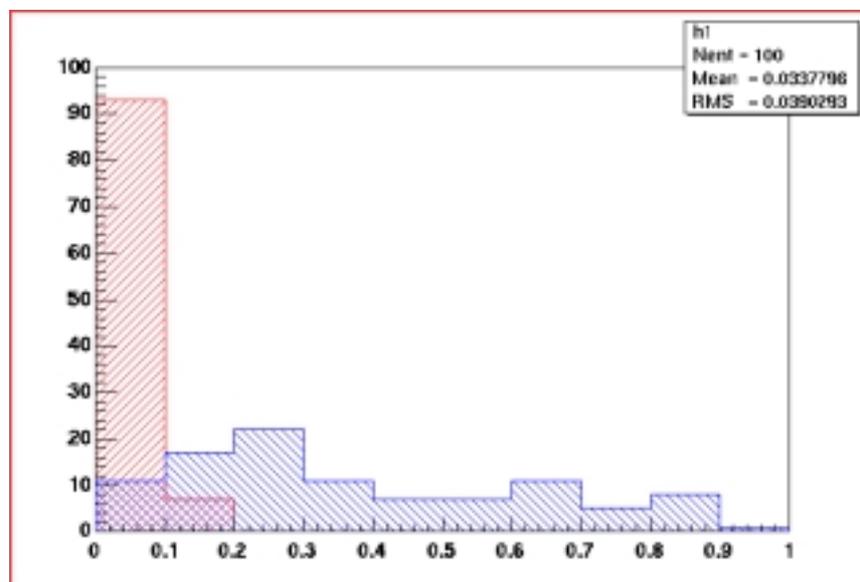


図 13: 条件 C、B の画像データを Kolmogorov テストで比較し、probability をプロットしたもの

青 : A - C の比較      赤 : C - B の比較

条件 A と C では障害物を置いていないという点では同じだが、あいだに条件 B でデータをとっているので数十秒の時間差がある。そのため様々な状況の変化があったことを考慮すると、A と C のデータを比較した probability を見るとある程度の類似性は認められる。一方、A と B、C と B のデータの組み合わせの probability はその殆どが 0.2 未満でこれらは明らかに違ったものであるといえる。

### 5.3 白色光によるシンチレータの静的な励起構造の変化

白色光によるシンチレータの励起構造の変化の観測は連続レーザー光、パルスレーザー光の両方で実験を行なった。

連続光レーザーを使った実験はデータのとり方、解析の方法は障害物を置いた変化を観測したときと同じである。probability のプロットを図 14、15 に示す。条件 A、C はシンチレータを励起していないとき、B は励起したときである。

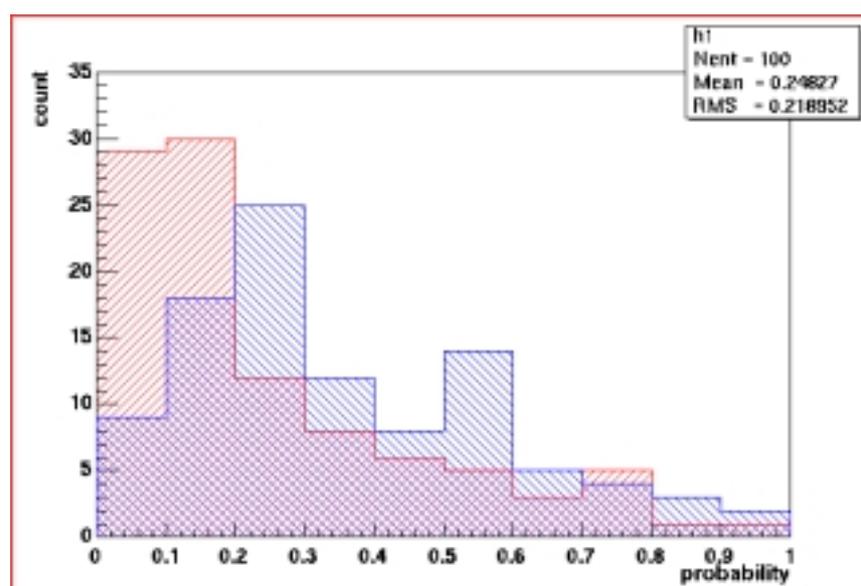


図 14: 条件 A、B の画像データを Kolmogorov テストで比較し、probability をプロットしたもの

青 : A - C の比較      赤 : A - B の比較

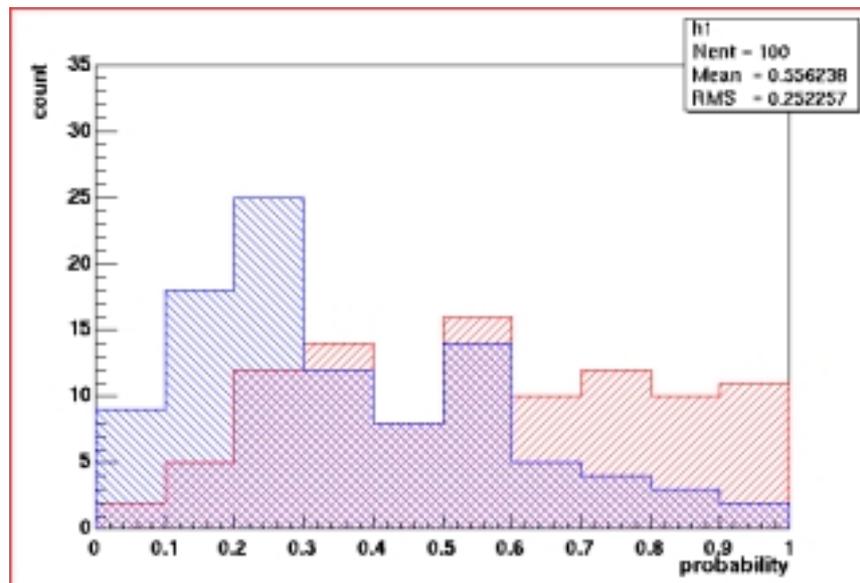


図 15: 条件 C、B の画像データを Kolmogorov テストで比較し、probability をプロットしたもの

青 : A - C の比較      赤 : C - B の比較

このプロットは障害物を置いたときと比べ、明らかに異なった条件のデータ間の probability が 0 にピークを持っていない。シンチレータの励起構造の変化が干渉模様には表れていないといえる。

一方、パルスレーザーを使用した実験で得たデータ同士を比べて probability をプロットしたものが図 16、17 である。前節でも述べたが、パルスレーザーを使用した実験では、

条件 A : シンチレータを励起させないとき

条件 B : 白色光で励起させたとき

として、データ 1 フレーム毎に条件 A と B を交互にとっている。図 16 は A のデータと、B を挟んだその次の A のデータとを比べたものである。図 17 は A のデータとその直後の B のデータの比較である。

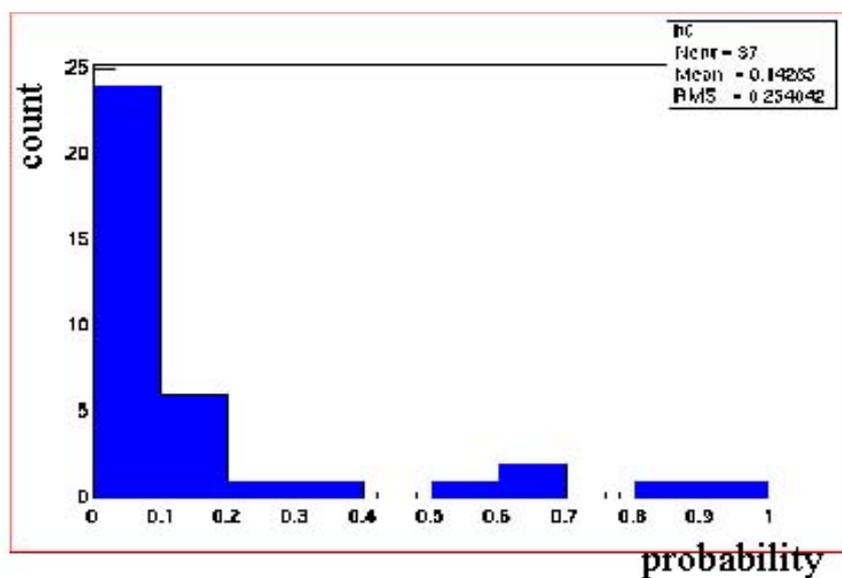


図 16: 条件 A 同士の画像データを Kolmogorov テストで比較し、プロットしたもの

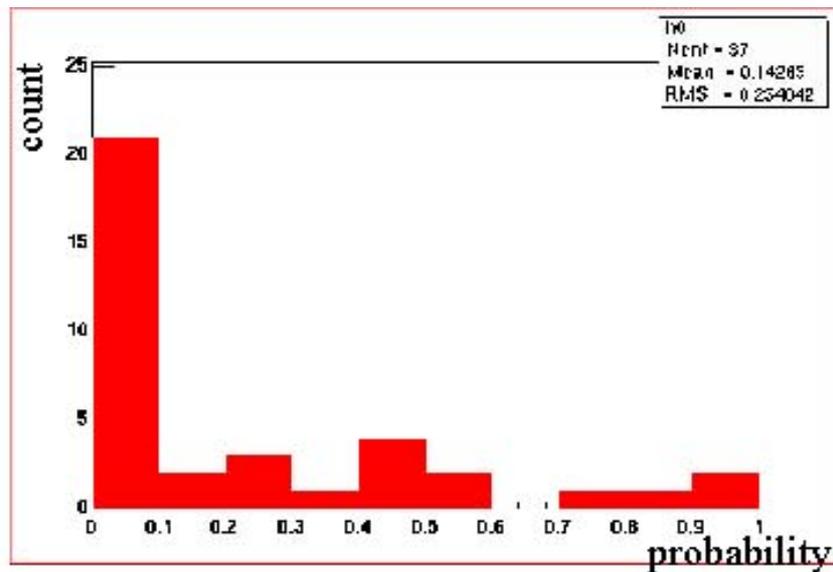


図 17: 条件 A、B の画像データを Kolmogorov テストで比較し、プロットしたもの

これら 2 つのプロットはほぼ同じような分布をしているように見える。これはパルスレーザーを使用していることに起因する。パルスレーザーを使用した場合、システムの性能から同じ条件でとっていても連続レーザー光を使用した場合に比べ、画像データにはある程度の系統誤差が乗ってしまう。そこで条件 A 同士の probability が高いところはその間 (1.8 秒間隔でデータを A、B、A と交互にとっているので、3.6 秒間) はシステムが安定していたと仮定し、そのときのデータのみをプロットしなおしたのが図 18、19 である。

統計量が少ないが、このとき A と B のデータを比較した probability は 0 のほうになっている。シンチレータの励起構造の変化が見えていると言えなくもない。

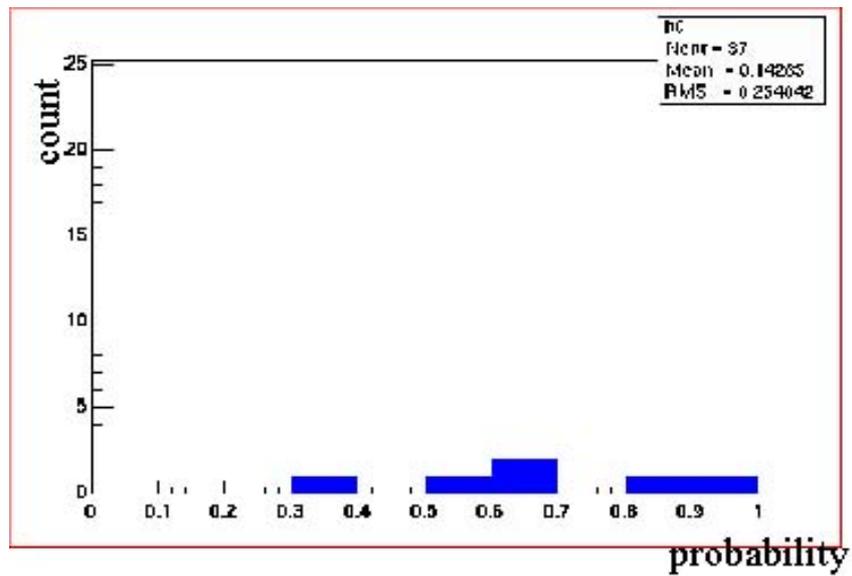


図 18: 条件 A のデータ同士の probability が 0.3 以上のときのプロット

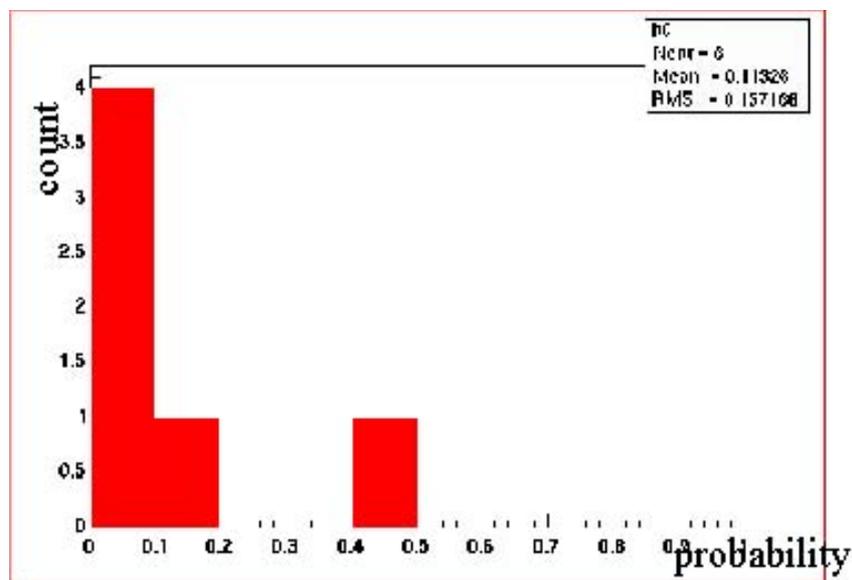


図 19: 条件 A のデータ同士の probability が 0.3 以上のとき A と B の比較のプロット

## 6 結果

まず光路上に障害物を置いたことによる干渉模様の変化は確認でき、製作した干渉計によって光の干渉が起こっていることが認められた。連続レーザー光では白色光を当てたことによるシンチレータの励起構造の変化は確認できなかった。これはその光路上に常に存在している連続レーザー光それ自身がすでにシンチレータを励起してしまっていたものと考えられる。パルスレーザーでは観測可能な兆しが見えている。

## 7 考察

今回の実験では、光路上に障害物を置いたという変化は明らかに干渉模様に見われ、シンチレータを静的に励起させた変化が何とか観測できつつあるというところまで進むことができた。

しかし、シンチレータの励起構造の変化については、まだ明確に違いが見えていないといえない。

まず考えられるのは、シンチレータの励起構造がどの程度変化しているのかということである。白色光を当てた場合は肉眼で見ても蛍光を発しているのがわかった。しかしレーザー光が透過したときどれほどの影響を受けるのかはよくわかっていない。より波長の短い光を当ててみるなどまだ試してみるべきことは残っている。

もう一つの要因として考えられるのは、システムの性能である。白色光によってシンチレータの励起構造の変化がある程度おこっていても、今回のシステムがそれを観測出来るだけの性能を備えていたのだろうか。シンチレータの励起構造を変化させるための光の強度や波長よりもこちらのほうの影響が大きいように思える。

読みだし部分の CCD カメラは入って来る光の強度を 254 階調でしか表せない。観測したい光の強度の変化が 1 階調の範囲内に入っている場合は使用できないことになる。今回の実験での光の強度の変化がどの程度であるかを見積り、それに適したものを新たに導入する必要があるのかもしれない。

システムの系統誤差も非常に大きな問題であった。3 節でも述べたが CCD カメラへのレーザー光の入射とプログラム、ビデオキャプチャカードの動作のタイミングを一定に出来なかったことによる画像データへの影響は考えられる誤差の中で最も大きかったと思われる。これを防ぐためには、CAMAC やパソコンを介さず直接レーザーの同期信号を CCD カメラに入れれば良いのである。それが不可能な場合は別時刻にとった二枚の画像データを比較するという解析方法を改めなければならない。先に述べた理由の他にもレーザーの強度の不安定さや光路上に起こる空気の揺れやホコリによる変化などデータ間には多くの誤差が介在している。観測したいものの影響はこれらに比べて大きいものであるかどうかはわからない。一枚の画像の中で干渉模様の変化が議論できればこれらの問題は解決する。シンチレータの一部にだけ励起させるための光をあて、画像データの干渉光のスポットのある部分だけがシンチレータ励起構造の変化しているところを通ったものになれば可能はずである。

また、実験開始時に分岐した光束を完全に一致させて光路上の変化を二次元で観測したいと考えていたが、現在の装置ではこの実現可能性は極めて低い。というのは、こうした二次元での変化を見るには光束断面内の各点の位置を分光後再び重ね合わせてやらねばならないからである。マイクロメーターを人間の手でまわして角度を調整するような精度のミラーでは調整することはできない。先に述べた1枚の画像データの中での比較もこの問題が解決しないと実現は不可能である。光束断面内の各点間での位相や強度のずれ、さらにはミラーなどの光学素子の表面の凹凸などもどの程度までが許容範囲かを吟味し、実験装置を構成しなければならない。

データ収集の効率の悪さも課題として残った。特に今回はレーザーにトリガーをかけて都合のよいタイミングでレーザー光を出すということができなかったため、レーザー側からの同期信号によってデータ収集プログラムを動かした。つまりシステム全体の動きがレーザーの動きに依ってしまっていた。そのためプログラム上でループを回してタイミングを合わせなければならなかった。コンピュータのOSとして採用したLinuxは常にバックグラウンドでいくつかのプロセスが動いており、それらのためにデータ収集プログラム中のタイミングを合わせるために待っている時間が10~30ミリ秒の範囲で揺らいでしまっていた。CCDからの画像データは30ミリ秒毎にビデオキャプチャカード上に送られてきており、この揺らぎはデータ収集効率を落す大きな要因の一つであった。しかし、この問題はトリガーをかけてタイミングをコントロールできるレーザー光源があれば解決できる。

## 謝辞

この実験は多くの人の指導、協力、助言によってここまで来ることができました。本当に心から感謝しています。ハドロン物理学研究室の本間先生、杉立先生にはこの実験全体に渡って熱心な指導をしていただきました。宮村先生からは貴重な助言をしていただきました。小浜さん、蜂谷さん、中村さん、中間・高エネルギー物理学研究室の平野さんには大変お世話になりました。特に平野さん、蜂谷さんには、実験を一部引き継いだこともあり、いろいろと教えていただきました。

またビーム物理学研究室の小方先生には実験のために多くの設備を貸していただきました。

共同実験者の宇津巻君にはいろいろと迷惑をかけることもあったと思います。どうもありがとうございます。

## 参考文献

- [1] 工藤 恵栄、分光の基礎と方法：オーム社
- [2] エリ・ランダウ、イエ・リフシッツ、( 訳 / 井上 健男、安河 内昂、佐々木 健 )、電磁気学：東京図書 (1965)
- [3] 谷田貝 豊彦、光とフーリエ変換：朝倉書店 (1992)
- [4] W.T.Eadie, D.Dryard, F.E.James, M.Roos, B.Sadoulet , STATISTICAL METHODS IN EXPERIMENTAL PHYSICS:NORTH-HOLLAND PUBLISHING COMPANY(1971)
- [5] 山内 二郎 編集、統計数値表 J S A : 日本規格協会 ( 1972 )

# 付録

## 画像データキャプチャプログラムソースコード

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <sys/times.h>
5 #include <sys/stat.h>
6 #include <fcntl.h>
7 #include <time.h>
8 #include <errno.h>
9 #include <sys/ioctl.h>
10 #include <sys/mman.h>
11 #include <linux/videodev.h>
12 #include "camlib.h"
13 #include "testlibdrv.h"
14
15 #define FRAME 1          /* capture frame number */
16
17 #define WIDTH 320
18 #define HEIGHT 240
19 #define V4L_DEVICE "/dev/video0"
20
21 #define AMASK 0x03ff    /* mask for ADC data */
22
23 struct video_capability vcap;
24 struct video_channel vc;
25 struct video_picture vp;
26 struct video_mmap mm;
27 struct video_mbuf mb;
28
29 #define NADC 14 /* station number of ADC */
30
31
32 void camac_init()
33 {
34     int status,q,x,naf;
35     unsigned short data=1;
36
```

```

37     /* initialize camac */
38     status=COPEN();
39     printf(" Open crate controller, status=%d\n",status);
40     status=CGENZ();
41     printf(" Initialize crate controller, status=%d\n",status);
42
43     /* clear crate */
44     status=CGENC();
45     printf(" Clear crate controller, status=%d\n",status);
46
47     status=CREMI();      /* ? */
48     printf(" Remove inhibit line, status=%d\n",status);
49     status=CENLAM(0);
50     printf(" Enable LAM interrupt, status=%d\n",status);
51 }
52
53 void close_camac()
54 {
55     int      status;
56
57     status=CCLOSE();
58     printf("\n Close crate controller, status=%d\n",status);
59 }
60
61 int waitLAM()
62 {
63     unsigned short  data;
64     int              naf,status,q,x;
65
66     naf = NAF( NADC, 0, 8 );      /* F=8(test LAM) : check ADC LAM
*/
67     status = CAMACW( naf, &data, &q, &x );
68     if( q == 0 )      return 0;      /* ADC's LAM = 0 */
69     if(status != 0 ) return 0;
70
71     return 1;
72 }
73
74 void clearLAM()
75 {

```

```

76         int      q,x,naf,status;
77         unsigned short  data=0;
78
79         naf = NAF( NADC, 0, 8 );          /* F=8(test LAM) : check ADC's L
M */
80         status = CAMACW( naf, &data, &q, &x );
81
82         /* F=9 : clear LAM */
83         do{
84             naf = NAF( NADC, 0, 9 );
85             status = CAMACW( naf, &data, &q, &x );
86         }
87         while( status != 0 );
88
89     }
90
91     int main()
92     {
93         unsigned short  tmp;
94         unsigned char   *buf;
95         int      i=0,j=0,n, count=0;
96         int      fl;
97         int      fd;
98         int      re;
99         FILE     *fp;    /* output file's filepointer */
100        char     filename[15]; /* output filename */
101
102
103        /* open output file(runX.XX.dat) */
104
105        camac_init();          /* initialize camac */
106
107        atexit(close_camac);
108
109
110        /*--- bttv information ---*/
111
112        /* open bttv driver */
113
114        fd = open(V4L_DEVICE,O_RDWR);

```

```

115         if( fd <= 0 )
116         {
117             perror("open");  exit(1);
118         }
119
120     /* search video-souce number */
121
122         if( ioctl(fd, VIDIOCGCAP, &vcap ) < 0 )
123         {
124             perror("VIDIOCGCAP");  exit;
125         }
126     //     printf("Video Capture Device Name : %s\n",vcap.name);
127
128     /* get video-souce's information */
129
130         for( i=0; i < vcap.channels; i++)
131         {
132             vc.channel = i;
133             if( ioctl(fd, VIDIOCGCHAN, &vc) < 0 )
134             {
135                 perror("VIDIOCGCHAN");  exit;
136             }
137             //fprintf(stderr, "Video Source (%d) Name : %s\n",vc.name);
138         }
139
140     /* get flame(mmap) infomation */
141
142         if( ioctl(fd, VIDIOCGMBUF, &mb) < 0 )
143         {
144             perror("VIDIOCGMBUF");  exit;
145         }
146
147     /* get memory */
148
149         buf = (unsigned char *)mmap(0, mb.size,
150             PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
151         if( (int)buf < 0 )
152         {
153             perror("mmap");  exit;
154         }

```

```

155
156         vc.channel = 0; /* input channel's No. */
157         vc.norm = 1;    /* NTSC */
158
159     /* write video-source information */
160
161         if( ioctl(fd, VIDIOCSCCHAN, &vc) < 0 )
162         {
163             perror("VIDIOCSCCHAN");  exit;
164         }
165
166         mm.frame = 0;
167         mm.height = HEIGHT;
168         mm.width  = WIDTH;
169         mm.format = VIDEO_PALETTE_GREY;
170
171
172         sleep(1);
173
174         clearLAM();
175         printf("\n    capture : running\n");
176
177     /*--- capture start ---*/
178
179     while(1)
180     {
181         /* capture loop */
182         while(1)
183         {
184             /*--- syncro capture to event ---*/
185
186             while(1)
187             {
188                 fl = waitLAM();
189                 if( fl == 1)
190                 {
191                     break;
192                 }
193             }
194

```

```

195         for(j=0;j<1;j++)
196             {
197             }
198
199         ioctl(fd, VIDIOCMCAPTURE, &mm );
200
201         ioctl(fd, VIDIOCSYNC, &mm.frame );
202         break;
203     }
204
205     /* write output file (runX.XX.dat) */
206
207     /* create picture file (binary pgm format) */
208
209         sprintf(filename,"p%d.p",count);
210
211         if( NULL == (fp=fopen(filename, "w")) )
212         {
213             perror("file open"); exit(1);
214         }
215
216         /**-- write binary file --**/
217         fwrite( buf, WIDTH, HEIGHT, fp );
218         fclose(fp);
219
220         count++;
221
222     /* clear LAM */
223         clearLAM();
224
225     } /* while */
226
227         munmap( buf, mb.size);
228         close(fd);
229 }

```