

平成 14 年度

修士論文

非線形光学効果を用いたマイクロ光造形

Micro-Photofabrication by Using Nonlinear Optical Effect

高知工科大学大学院

工学研究科 基盤工学専攻（博士課程前期） 知能機械システムコース

1055058 日野聡一

目次

第1章	緒言	1
第2章	マイクロ光造形法の原理	3
2.1	光の回折限界	3
2.2	非線形光学効果	4
2.2.1	2光子吸収	4
2.2.2	第2高調波発生の原理	5
2.3	光重合反応	8
2.4	光硬化性樹脂	11
2.5	開口数と硬化分解能	13
第3章	実験方法	15
3.1	使用機器	15
3.2	機器の配置	19
3.3	光学系	21
3.4	実験手順	22
3.5	収縮率の改善	23
3.5.1	加温時間による収縮率の変化	23
3.5.2	走査回数による収縮率の変化	25
第4章	実験結果	27
4.1	分解能の調査	27
4.1.1	XY平面内の硬化分解能	27
4.1.2	Z軸方向の硬化分解能	28
4.2	ピエゾアクチュエーターの校正	31
4.2.1	校正方法	31
4.2.2	立方体の製作	31
4.2.3	計測	33
4.2.4	測定結果	34
4.2.5	検証	38
4.3	プログラム製作	39
4.4	3次元マイクロ構造物の作製	43

第5章 結言	45
付録	46
参考文献	66
謝辞	67

1章 緒言

ミクロンオーダーの機能要素で構成されたマイクロマシンの実現には、エネルギー供給手段、計測技術、微細加工、精密組み立て技術、通信制御技術、材料開発などの広い分野での総合的な要素技術開発が必要となる。中でも光技術は、計測、表示、通信、記録、加工などの分野で大きな貢献をしてきた。さらに、マイクロマシンの技術では、より複雑で立体的な微小構造物を作製する精密組み立て技術や、電気的なアクセスが困難なため、従来の電気配線に代わる非接触エネルギー供給技術が要望され、光技術は組み立て、アSEMBリ、そして駆動手段としても、その新しい応用が期待されている^{(1) (2) (3)}。その中で最近注目されているのが、マイクロ光造形である。光を力やエネルギー源だけでなく、光造形によって作製した構造物をマイクロマシンの部品として使用する試みがなされている⁽⁴⁾。

現在、一般的に行われている光造形法は、紫外光を直接光硬化性樹脂に照射し、1光子吸収によって硬化させる。この方法では光の回折限界で構造物の空間分解能が決定されてしまい、それを越えた分解能での光造形は原理的に不可能である。また、紫外光を直接照射しなければならないため、構造物の形状や作製手順に制限が生まれてしまう。それに対して最近、2光子吸収を用いたマイクロ光造形が注目されている^{(5) (6) (7)}。

3次元空間を光が伝播するということは3次元光学ではなく、単に光は波動であるといっているにすぎない。3次元空間の任意の一点に自在に光反応を発生させることが、本当に3次元光学である。そのためには、光軸上で前後の3次元空間に影響を与えずに、特定の1点だけに光反応を発生させることが必要である。これを実現させるためには非線形応答が必要である。

光源波長は、近赤外域を用いる。物質内の3次元空間を伝播する際に光吸収がなく、物質が透明であるからである。近赤外の中でも700~1100 nmあたりの、より短い波長側を用いる。この波長域は、官能基の分子振動の倍音、結合音から比較的遠く、また電子遷移冷機波長の程度に相当するからである。

さらに、非線形光学現象は、2次、3次...と高次なものになるにつれて、

現象の発生率がレーザー強度の2乗，3乗...に比例する．よって，高い非線形効果によって反応を局在化するために高い光強度が必要であり，パルスレーザーが用いられる．パルスレーザーは瞬間的に高い強度も持ちながら，平均パワーは低いので都合が良い⁸ ⁹．なぜなら，照射時間が長くなると非線形光学効果に加えて熱効果が加わり，多光子分光過程以外の破壊現象が生じてしまうからである．熱効果が発生する前にパルス発生を止めるにはパルス幅は例えば200 fs以下である．

フェムト秒レーザーの大衆化とナノスケールの科学技術への興味の高まりによって，このようなマイクロ・ナノ空間スケールの非線形分光法が最近特に注目されるようになり，応用技術が著しく進歩している⁹ ¹⁰．このような非線形光学効果を用いた光造形は，将来的に，より高次の光学過程である3光子吸収，4光子吸収などによって，空間分解能をさらに高くできる可能性を秘めている¹¹．

本研究では，1光子吸収型光造形法の分解能を超えた「2光子吸収を用いた光造形法」の分解能を調査し高分解能の光造形法を確立させ，3次元マイクロ部品を試作することが目的である．

2章 マイクロ光造形法の原理

2.1 光の回折限界

物体が散乱や放出する光を使用し、遠くからその位置や形状を測定する時、利用する光学機器と光の波長に応じて、測定精度の限界が存在する。これは光の波の性質により、光が伝播する際に波をさえぎる物体の陰にも回りこむという、回折の性質の現れで、広い意味で回折限界と呼ばれる⁽¹²⁾。物体から出た光は、さまざまな経路を通過して観測面に到達し干渉効果によって強度分布のパターンを作る。物体の形状をフーリエ変換した波のうち、光波長よりも短い成分は遠方に運ばれずに欠落し、対応する細かい物体形状を再現できない。このような光の回折限界により、1光子吸収を用いた光造形では、光の回折限界を超えた分解能で造形を行うことが不可能なのである(図.1)。

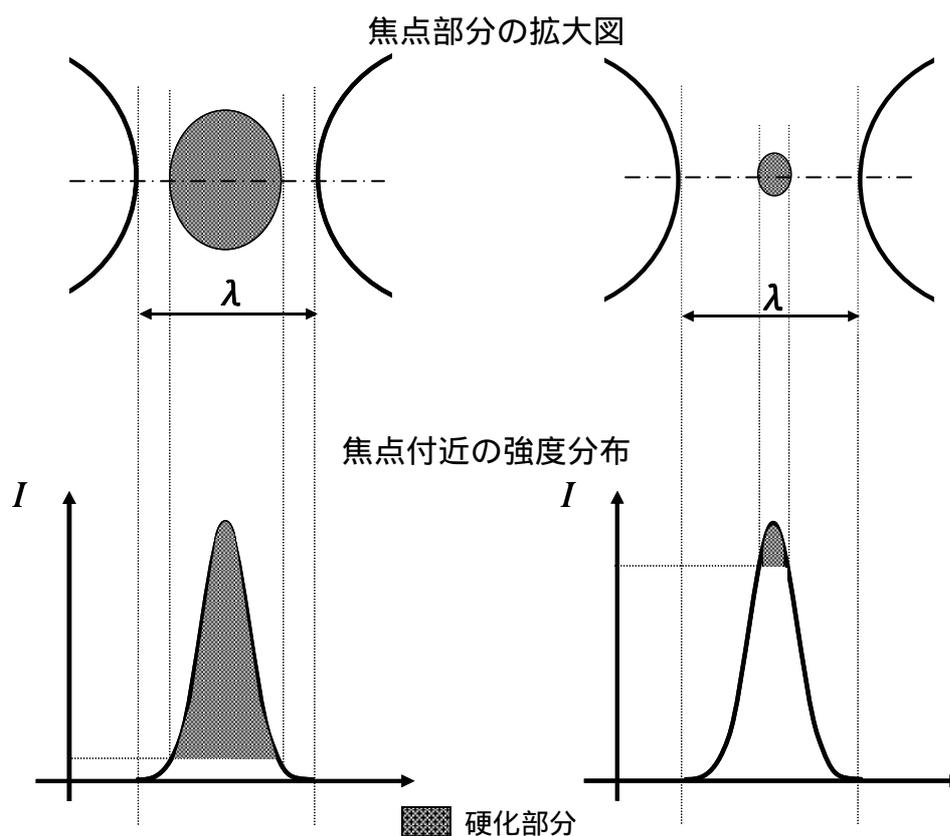


図.1 硬化分解能の違い
1光子吸収(左), 2光子吸収(右)

2.2 非線形光学効果

非線形光学効果は媒質に入射した光の電界が大きい時、媒質中に誘起させる分極密度が比例関係でなくなるために起こる現象である。一般に反射、吸収、発光量が光強度に比例しない現象を非線形光学現象と呼び、これは光のエネルギーを受け取った同士が互いに影響を及ぼしあう結果、生じる現象である⁽¹³⁾。緒言でも触れたように、フェムト秒レーザーはピーク強度が高いので非線形光学効果を起こしやすい。この非線形な分極は自己位相変調や高調波発生、入射光強度の吸収飽和変化、和周波、差周波の発生などの様々な現象を引き起こす。本実験ではその中の「2光子吸収」、「第2高調波発生」を利用している。

2.2.1 2光子吸収

2光子吸収とは非線形光学効果のひとつであり、2つの光子を同時に吸収することによって、照射された光子の2倍のエネルギーに相当する吸収が生じる現象である(図.2)⁽¹¹⁾。この現象は、2つのフォトンと同時に吸収することにより焦点面で照射光の半波長の光が得られることを利用している。

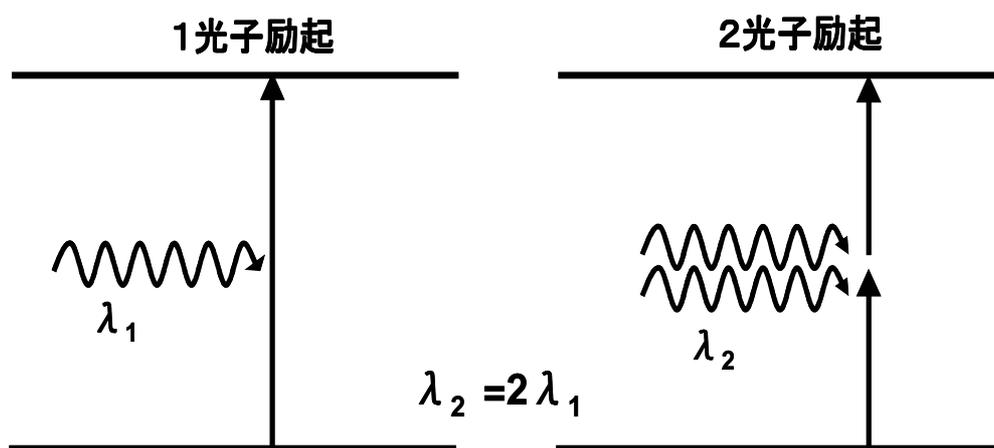


図.2 エネルギー準位と励起の様子。

1光子吸収(左), 2光子吸収(右)

2.2.2 第2高調波発生の原理

どんな誘電体でも電磁場の強度が強いと光に対して非線形を示す．その理由は印可された場によって起こる束縛電子の非調和運動を関係したもので、その結果、電気双極子に起因する誘導分極 P は電界 E に比例しなくなる．

ここで、 ε_0 を真空の誘電率、 $x^{(1)}$ 、 $x^{(2)}$ 、 $x^{(3)}$ をそれぞれ1次、2次、3次の感受率で一般にテンソルで表される．そして、ここでは簡単のためにその感受率をそれぞれ

$$x^{(1)} = \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} x^{(2)} = \begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{pmatrix} x^{(3)} = \begin{pmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & 0 & c_{33} \end{pmatrix}$$

とする．ここで電界を E とすると誘導分極 P は

$$\begin{aligned} P &= \varepsilon_0 \left(x^{(1)} \cdot E + x^{(2)} \cdot E \cdot E + x^{(3)} \cdot E \cdot E \cdot E + \dots \right) \\ &= \varepsilon_0 \left\{ \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} + \begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{pmatrix} \begin{pmatrix} E_x^2 \\ E_y^2 \\ E_z^2 \end{pmatrix} + \begin{pmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & 0 & c_{33} \end{pmatrix} \begin{pmatrix} E_x^3 \\ E_y^3 \\ E_z^3 \end{pmatrix} + \dots \right\} \\ &= \varepsilon_0 \left\{ \begin{pmatrix} a_{11} E_x \\ a_{22} E_y \\ a_{33} E_z \end{pmatrix} + \begin{pmatrix} b_{11} E_x^2 \\ b_{22} E_y^2 \\ b_{33} E_z^2 \end{pmatrix} + \begin{pmatrix} c_{11} E_x^3 \\ c_{22} E_y^3 \\ c_{33} E_z^3 \end{pmatrix} + \dots \right\} \quad (1-1) \end{aligned}$$

となり、 x 方向の電界を考えると、

$$P_{(x)} = \varepsilon_0 \left(a_{11} E_x + b_{11} E_x^2 + c_{11} E_x^3 + \dots \right) \quad (1-2)$$

となる。

簡単のため電界ベクトルが1つの座標成分のみの時を考えると、

$$P = \epsilon_0 \left(x^{(1)} E + x^{(2)} E^2 + x^{(3)} E^3 + \dots \right) \quad (1-3)$$

と表される (x は x 方向の感受率) この式のそれぞれの項は2次, 3次... の非線形現象を引き起こす。また, 式(1-3)から2次, 3次の非線形現象はそれぞれ E^2 , E^3 に比例していることがわかる。式(1-3)は入射光の強度の低い時には1次以外の項は無視することができ, そのときには入射光の強度と分極の関係は線形で表される。

高調波発生とは, ある非線形媒質に光を入射すると入射光の2倍, 3倍... の周波数の光が発生する現象である。2次の非線形効果で起こるものを第2高調波と呼ぶ。

(1-3)の2項目から2次の非線形効果による分極 P_2 は次の式で表される。

$$P_2 = \epsilon_0 x^{(2)} E^2 \quad (1-4)$$

(ただし, ϵ_0 は真空の誘電率, $x^{(2)}$ は2次の感受率とする。)

ここで, 入射光の周波数を ω とし, 電解 E を

$$E = A \sin(\omega t) \quad (1-5)$$

とすると, 式(1-4)は

$$P_2 = \epsilon_0 x^{(2)} A^2 \sin^2(\omega t)$$

$$= \frac{1}{2} \epsilon_0 x^{(2)} A^2 (1 - \cos 2\omega t) \quad (1-6)$$

となることから，入射光の2倍の周波数 2ω の光が発生することがわかる．本実験で使用した光源は近赤外光（波長 710 nm）である．このレーザー光をレンズで集光することにより，焦点近傍で非線形光学効果が誘起され，紫外光が発生する．つまり，光硬化樹脂硬化用の光源として2倍波長のレーザー光が使えるため，「近赤外の光源で紫外線硬化樹脂の造形」が可能となるのである．

2.3 光重合反応

光硬化性樹脂に光（レーザー）を照射すると、光硬化性樹脂が光を吸収し、樹脂の中の光重合開始剤が解離し、生成した成分が核となり、その周囲のプラスチック分子が互いに結合して巨大分子となり固化する。このことを光重合反応という。

光造形に用いる光硬化性樹脂は、光重合開始剤、モノマー、オリゴマー、各種添加剤で構成されている⁽¹⁴⁾。モノマーとは、重合して分子となりプラスチックを形成する材料で、オリゴマーとはモノマーをあらかじめいくつか反応させてあるものである。オリゴマーもモノマー同様、重合してプラスチックを形成する材料である。光重合開始剤は光を吸収して活性化し、開裂反応、水素引き抜き、電子移動などの反応を起こし、光重合反応のトリガーの役目を果たすものである。

光硬化性樹脂は次に示す段階を経て硬化する。オリゴマー（M）、モノマー（M'）、光重合開始剤（I）とする。

光重合開始剤が紫外線を吸収する。

開始反応：

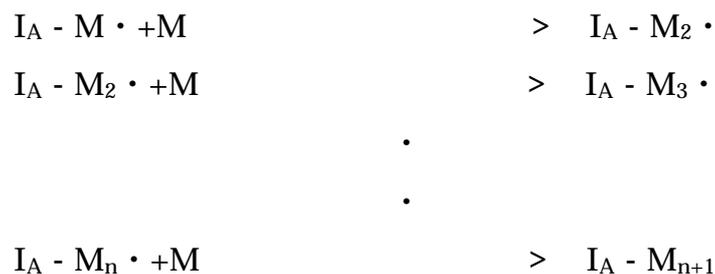
紫外線を吸収した光重合開始剤が活性化する。



成長反応：

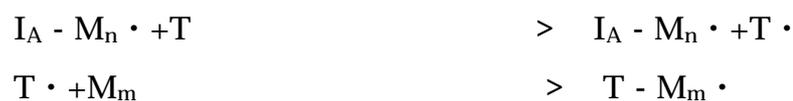
活性化した光重合開始剤は分解などを経て、モノマーやオリゴマーなどの樹脂成分に反応する。





連鎖反応：

この反応生成物は、他の樹脂成分にも反応し、連鎖的に反応が進行する。生成されたラジカルが他の分子(T)に移動して反応を開始する。



停止反応：

3次元的に架橋反応が進み、分子量が増大する。ラジカル同士が結合するか、オリゴマー以外の分子と反応してラジカルが消滅し、重合反応が終了する。



これらの開始、成長、連鎖、停止の4過程により、光硬化性樹脂は液体から固体に変化する(図.3)。樹脂内に含まれているモノマーやオリゴマーなどの光吸収度は紫外領域でそれほど大きくないため、組成物全体の光吸収性は光重合開始剤に大きく依存する。光硬化の長所として次のものが挙げられる。

速硬化性：

熱加工型や溶剤完走型樹脂では長大な過熱乾燥炉が必要であるが、光硬化型は非常に短時間の光照射ですむので、硬化設備が小形化、省スペース化できる。また、生産性の工場や不良率の低下が可能となる。

低温硬化：

光エネルギーで重合反応や架橋反応が開始するため、比較的低温で硬化することが可能となり、熱に弱い基材にも用意に適用できる。

硬化特性：

硬化性樹脂の選択により、高硬度、耐摩耗性、耐薬品性、耐汚染性、耐ブロッキング性、高光沢など用途に応じた優れた硬化特性が得られる。

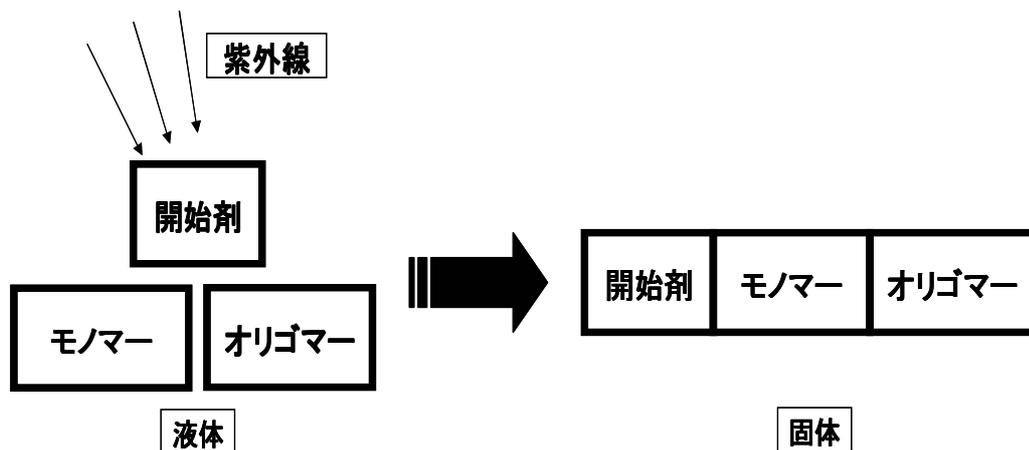


図.3 光重合反応簡略図

2.4 光硬化性樹脂

反応形態から光硬化性樹脂を分類すると、ラジカル重合型とカチオン重合型に大別される。ラジカル重合型は光硬化塗料、インキ用として本格的な研究開発が始まってすでに20年以上が経過しており、その中でもアクリル系樹脂を主体とした応用技術が最も進展している⁽¹⁰⁾。一方、カチオン重合型光硬化性樹脂はプレポリマーとしてエポキシ樹脂を主成分とするもので、カチオン系光重合開始剤に特徴がある光硬化システムである。ラジカル重合型と比べて応用開発が遅れているが、空気中での速硬化性、保存安定性、硬化物の低収縮性、基材への付着性、塗膜物性などの点で優れた特性を有している。しかし、厚膜硬化性、塩基性物質との相性が欠点である。

ラジカル重合型光硬化性樹脂

ラジカル重合型光硬化性樹脂には不飽和ポリエステル系、アクリル系、エン・チオール系などがある。アクリル系重合性プレポリマー（オリゴマー）は官能基としてのアクリロイル基（ $\text{CH}_2=\text{CH}-\text{CO}-$ ）を有するベースレンジである。酸素による硬化阻害を受けやすいという特徴を持つ。

カチオン重合型光硬化性樹脂

カチオン重合型光硬化性樹脂システムはラジカル型とは基本的な反応メカニズムが異なり、プレポリマーや反応性モノマーおよび光重合開始剤はラジカル系とは全く異なったものを使用する。カチオン型光硬化性樹脂には官能基を有するエポキシ樹脂、ビニルエーテル系化合物、環状エーテル系化合物などがあり、最も汎用しているのはエポキシ系樹脂である。エポキシ系樹脂は官能基として炭素原子2個と酸素原子1個から作られた三員環（オキシラン環）を有したベースレンジである。常温での重合反応性が低いという特徴を持つ。

本実験で使用した光硬化性樹脂は、SCR701という種類であり、光造形に用いられる樹脂としては最も一般的なものである。この光硬化性樹脂は先に述べた「ラジカル重合系」と「カチオン重合系」の混合物である。こ

の樹脂の一般的な特徴としては、速硬化性、硬化精密性などが挙げられる。SCR701の光吸収特性は図4のようになっており、波長が350~360 nm付近の光を特に吸収することがわかる。

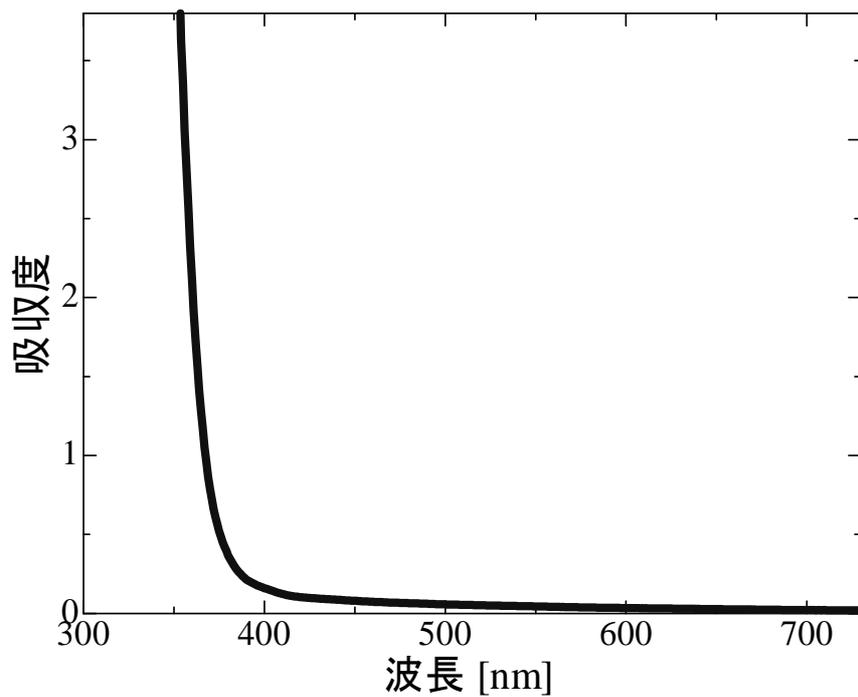


図.4 SCR701の光吸収特性

2.5 開口数と硬化分解能

開口数とは一般に NA で表され、NA とは「Numerical Aperture」の略称である。開口数は対物レンズの性能としては倍率より重要なものである。一般に開口数は

$$NA = n \sin \theta \quad \text{又は} \quad NA = n \sin \left(\frac{\theta}{2} \right)$$

と定義される⁽¹⁵⁾⁽¹⁶⁾。ここで θ は図.5 に示すように対物レンズの先端のレンズの最外周部から物体を見込む角度で、 n はレンズと物体の間の媒質の屈折率である。媒質が空気なら n は 1 なので、NA は最大で 1 以下の値となる。しかし、対物レンズと物質の間を適当な屈折率の物体で充填すれば NA は 1 より大きくなる⁽¹⁷⁾。具体的には、レンズと物質の間に屈折率 1.5 程度の油で浸漬する油浸レンズでは NA が 1.4 程度のものがある。

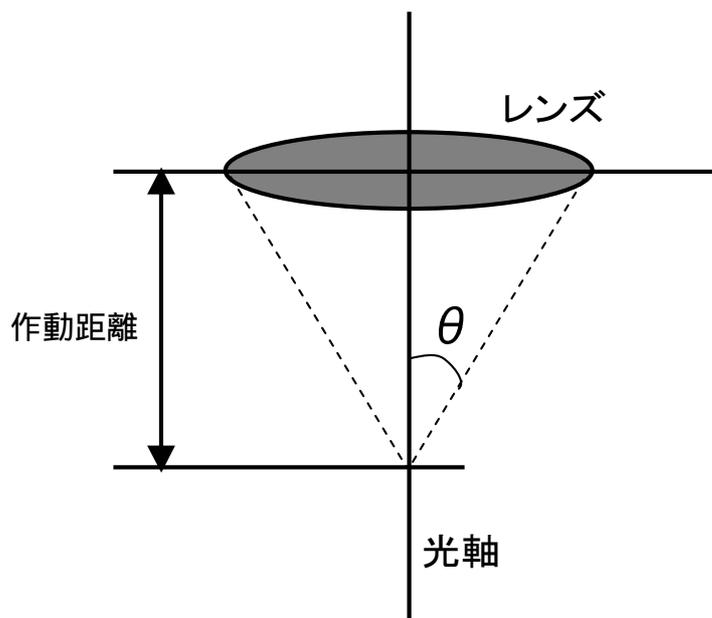


図.5 レンズと開口数の関係

今回、レーザー光を集光させるのに用いた対物レンズは倍率 100 倍で浸漬した時の N.A. が 1.3 である。図.6 に示すように、NA の高いレンズを使用することにより、Z 軸方向の分解能をより高くすることができる。

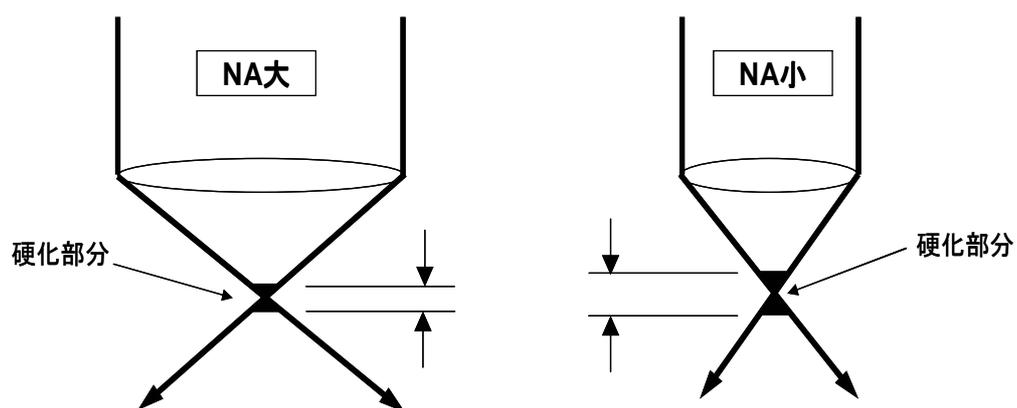


図.6 開口数と Z 軸方向の硬化分解能の関係

3章 実験方法

3.1 使用機器

本実験を行うにあたって次の機器を使用した。

- ・ Ti:サファイアフェムト秒レーザー
- ・ ガルバノミラー
- ・ ピエゾアクチュエーター
- ・ 光学顕微鏡
- ・ プログラミング電源 , 高電圧電源
- ・ CCD カメラ , モニター
- ・ パソコン

) Ti:サファイアフェムト秒レーザー

本実験で使用した Ti:サファイアフェムト秒レーザーは Tsunami system(Ti:サファイアレーザー , MilleniaVs , モードロッカー , 波長 710 nm , パルス幅 130 fs , 繰り返し周波数 80MHz) で Spectra-Physics 社製のものである。レーザー本体は図.7 のような装置の構成で成り立っている。Ti:サファイアフェムト秒レーザーは MilleniaVs というレーザーから励起用レーザー光を得て発振される。

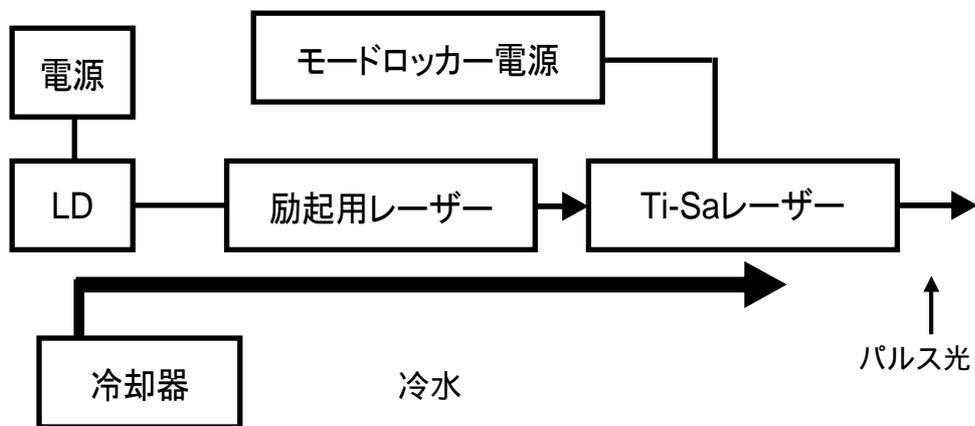


図.7 Ti:サファイアフェムト秒レーザー構造図

) ガルバノミラー

ガルバノミラーとはミラーに軸を付け、電気信号に応じてミラーの回転角を変えられるようにした偏向器である。非共振型の偏向器なので、のこぎり波形型の電気信号やランダムな信号で駆動できる。ミラーを利用した偏向器ではポリゴンミラーが多用されるが、ポリゴンミラーは複数のミラーを回転させるため、再現性よくスキャンさせるにはいわゆる「面倒れ補正」が必要になる。一方、ガルバノミラーは単一のミラーを振動させるため「面倒れ補正」をしなくても再現性のよいスキャンができるのが特長である。本実験では2つ1組のガルバノミラーを使用し、レーザーの集光点をXY平面内で走査した。また、ガルバノミラーの制御は、GSI Lumonics社製のSC2000というコントローラーを使用した。

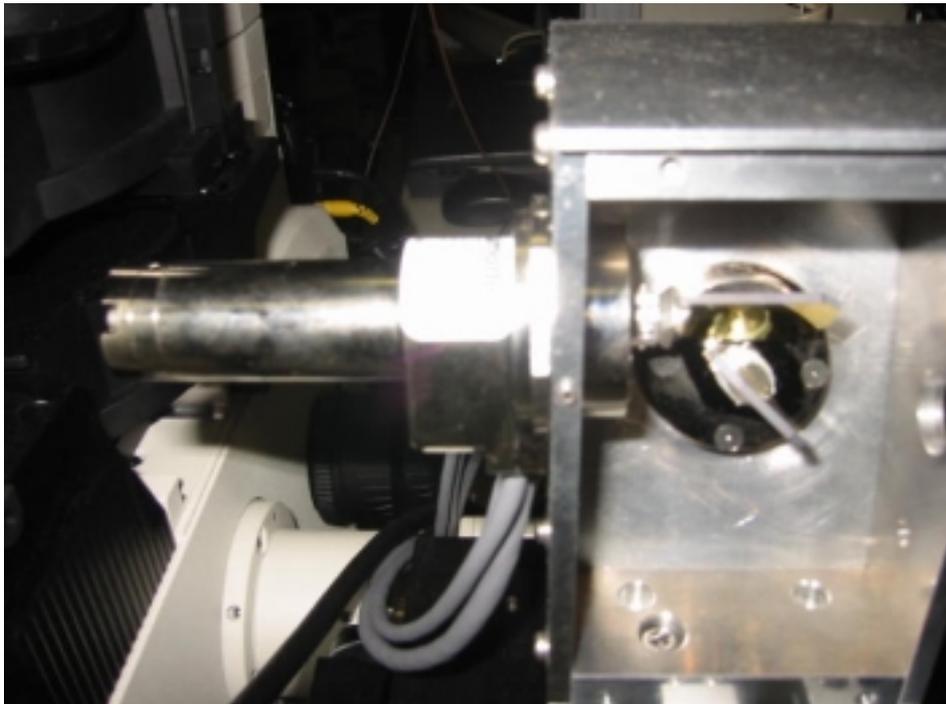


図.8 2つのガルバノミラーとミラーホルダー

) ピエゾアクチュエーター

ピエゾ素子とは、電圧を加えることにより変形する物質である。その変位は加えた電圧にほぼ比例する。本実験ではピエゾ素子を内蔵したアクチュエーターを使用しサンプルをZ軸方向に移動させ、樹脂内でレーザー集光点をZ軸方向に走査させた。仕様書には+1000で約25 μm 変異すると記載されている。

) 顕微鏡

使用した顕微鏡はNikon社製、ECLIPSE TE2000-Uである。ステージ部分にピエゾアクチュエーターを設置し、そこに光硬化性樹脂を塗布したスライドガラスをセットした。実験に使用した対物レンズは4個で、光造形用1個(100倍, NA1.3), 観測用3個(60倍, 40倍, 20倍)である。



図.9 顕微鏡外観

) プログラミング電源，高電圧電源

本実験で使用したピエゾアクチュエーターの稼動のため，高電圧電源（MATSUSADA 社製，PL-650-0.1）を使用した．この電源は+650 V まで出力できる．またその+650 V の出力を制御するため，プログラミング電源（Agilent 社製，E3640A）と接続し，リモートコントロール機能によってピエゾアクチュエーターに加える電圧を制御した．

リモートコントロールしたときの高電圧電源の出力電圧 E_0 と，プログラミング電圧からの入力電圧 E_1 には次の関係が成り立つ．

$$E_0 = E_{\max} \times \frac{E_1}{10} \quad (E_{\max} \text{ は高電圧電源の最大定格電圧 } 650 \text{ V})$$

プログラミング電圧の最大定格電圧が 8 V であるため，高電圧電源の出力電圧の上限は 520 V となる．よって，ピエゾアクチュエーターを約 13 μm 動作させることができることがわかる．

) CCD カメラ，モニター

光造形をするにあたって，レーザー集光点の走査をモニターで確認するために顕微鏡に CCD カメラを設置した．また，サンプルを設置した後，対物レンズの焦点位置や硬化確認をする際や，サンプルの洗浄時の様子を確認する際にも使用した．

) パソコン

4 章で述べるプログラムを使用して，ガルバノミラーとピエゾアクチュエーターを制御するためにパソコンを使用した．

3.2 機器の配置

実験で使用した機器の簡単な配置図を図.10 に示す。図のような光学系を用いてマイクロ光造形を行った。まず、フェムト秒レーザーから発射されたレーザー光をスペイシャルフィルターに通し、造形に不必要な光の成分を取り除く。いくつかのレンズとミラーを経て、ガルバノミラーに反射し、顕微鏡内の対物レンズに入射させる。顕微鏡にはピエゾアクチュエーターを取り付けられており、ガルバノミラー（GM）によってレーザー光をXY平面内で走査させ、ピエゾアクチュエーターによってZ軸方向に走査させた。ガルバノミラーを制御するコントローラーとピエゾアクチュエーターに電圧を加える電源はパソコン（PC）に接続されており、それぞれを制御できる。実際にはこの光学系に加え、顕微鏡に CCD カメラを接続し、その映像をモニターで観察しながら光造形を行った。

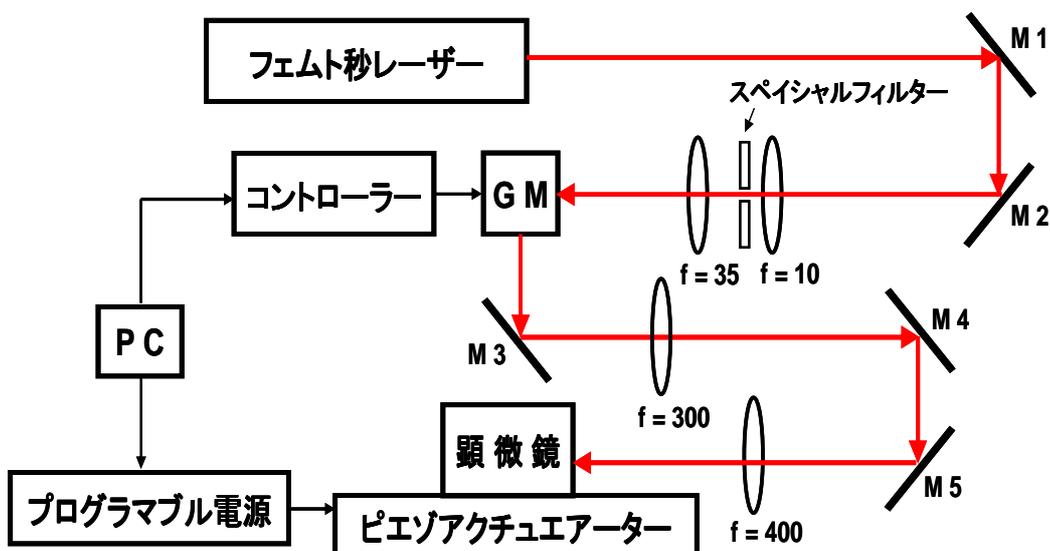


図.10 実験装置配置図.

(GM は 2 枚 1 組のガルバノミラーを指す)

実験装置の全体写真を図.11 に示す .

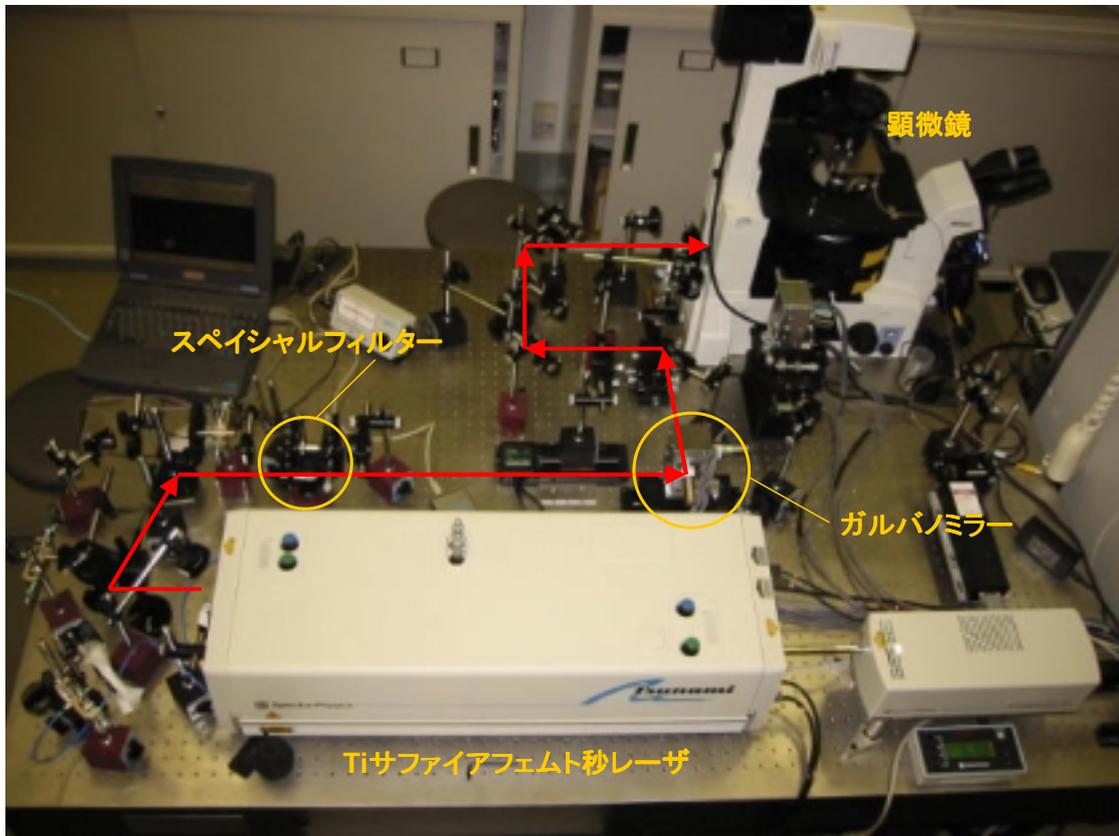


図.11 実験装置の写真

3.3 光学系

配置図にはレンズが多数配置されているが、これには意味がある。図.10の配置図の主な光学系は図.12のようになっている。これらのレンズを図のように配置してやることにより、光源からのレーザー光の幅を広げて NA（開口数）の高い対物レンズに入射させることができる。さらに構造物の Z 軸方向の分解能を上げ、レーザー光を走査する際、ガルバノミラーに反射したレーザー光が対物レンズの瞳から外れなくなる構造にもなっている。

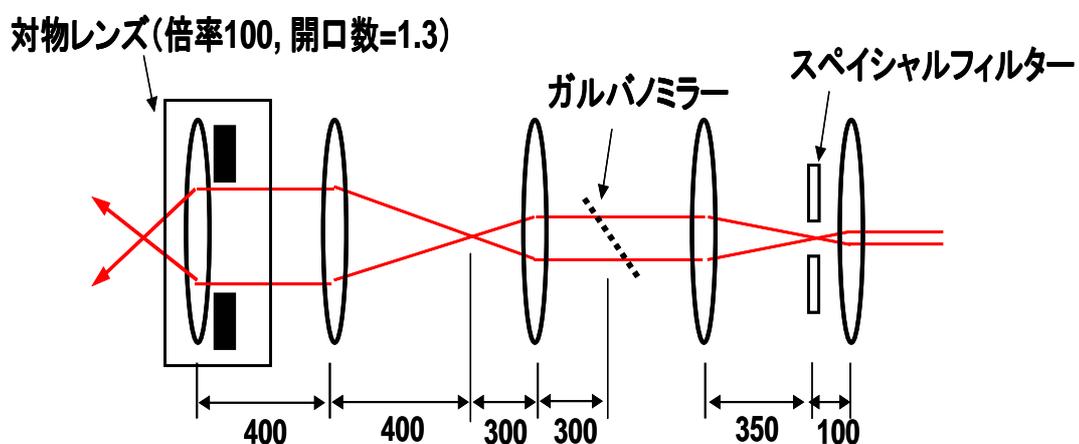


図.12 主な光学系配置図

3.4 実験手順

マイクロ光造形をするにあたって、以下の手順で実験を行った。

光硬化性樹脂が蛍光灯から出る紫外光成分で硬化しないように部屋を暗くする。

スライドガラス（円形）に光硬化性樹脂をごく少量塗布する。

顕微鏡にスライドガラスを設置する。

対物レンズのオイルを塗布し、スライドガラスとレンズの間を屈折率の高い媒質で満たし、レンズの NA を適正化する。

ガルバノミラーを制御してレーザーの集光点を樹脂内で走査する。

顕微鏡に設置し、モニターで観察しながらエタノールで未硬化部分の樹脂を洗浄する。

十分に乾燥させる。

SEM（電子顕微鏡）で観察する。

3.5 収縮率の改善

本実験で使用した光硬化性樹脂は SCR701 という種類であり，ラジカル重合系とカチオン重合系の混合物である．光造形で硬化させた構造物を，作製してすぐに洗浄すると，乾燥する際に収縮してしまうという現象が起こった．その収縮率は 20%以上にもなっており，精密なマイクロ光造形を行うには，この収縮率の問題を解決しなければならないと判断した．作製した構造物の収縮要因を推測してみた．

2.4 でも述べたように，ラジカル重合系は空気中の酸素によって禁止効果を受ける．本実験のように樹脂が少量であれば酸素の供給がたやすくなるため，硬化しづらくなる．このことは照射するレーザー強度を上げてやることや，同じ場所での走査回数を増やすことにより，ある程度硬化の進行を促進させることができるのではないかと推測できる．

また，カチオン重合系はラジカル重合系に比べ，非常に反応速度が遅く，常温では完全硬化に 1 時間程度かかる．さらに重合反応は加温によって促進できるので，洗浄後に 1 時間ほど放置し，その後加温して洗浄すればいいのではないかと推測できる．よって，加温と走査回数の 2 つの視点から収縮率改善を試みた．

3.5.1 加温時間による収縮率の変化

まず，加温時間を変化させて収縮率の変化を測定した．そのために，以下の 5 つのプロセス (A~E) でサンプルを作製し，収縮率の変化を測定した．

A	．加温 (3 時間)	造形	加温 (3 時間)	洗浄
B	．加温 (3 時間)	造形	加温 (5 時間)	洗浄
C	．加温 (3 時間)	造形		洗浄
D	．	造形	加温 (3 時間)	洗浄
E	．	造形	加温 (5 時間)	洗浄

それぞれのプロセスごとに収縮率を測定し、グラフにすると図.13 のようになった。グラフを見ると、プロセス E が最も収縮率が小さいことがわかる。

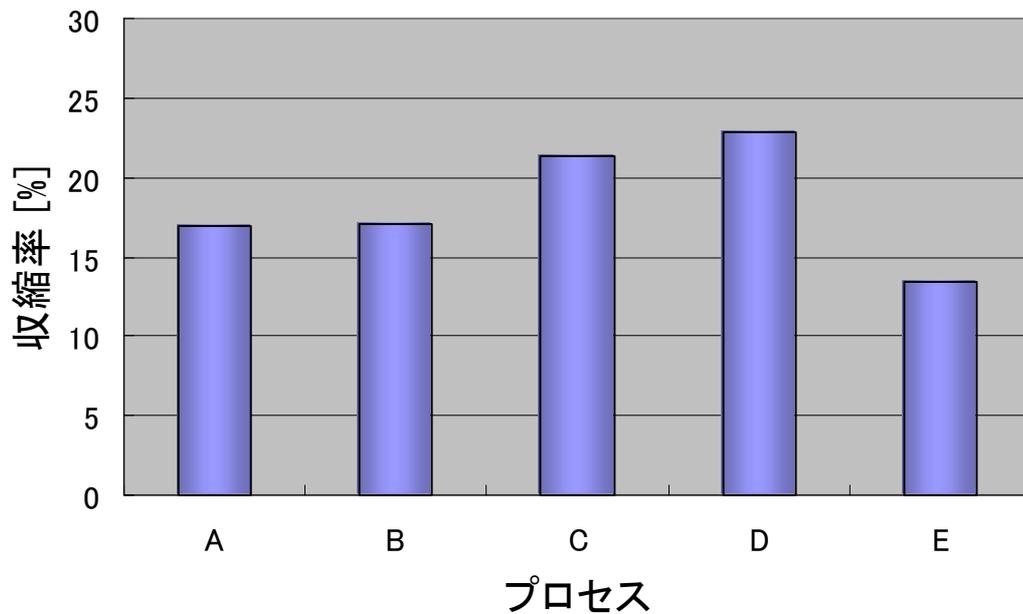


図.13 加温時間と収縮率の関係

(作製条件)

レーザー波長 710 nm

走査回数 1 回

レーザーパワー 9 mW

洗浄液 エタノール (2.5 ml)

走査速度 42 μ m/s

測定装置 レーザー顕微鏡 (楠川研)

対物レンズ 100 倍 , NA=1.3

3.5.2 走査回数による収縮率の変化

次に、同じ場所での走査回数を変化させて収縮率を測定した。走査回数は1回、3回、5回、7回と4つの条件で行った。条件（走査回数）ごとに収縮率を測定し、グラフにすると図.14 のようになった。グラフを見ると、同じ場所で5回以上走査する方法が最も収縮率が小さいことがわかる。さらに同じ場所での走査回数を15回にまで増やして収縮率の変化を測定したが、同じ場所で5回以上走査させても収縮率に変化はなかった。

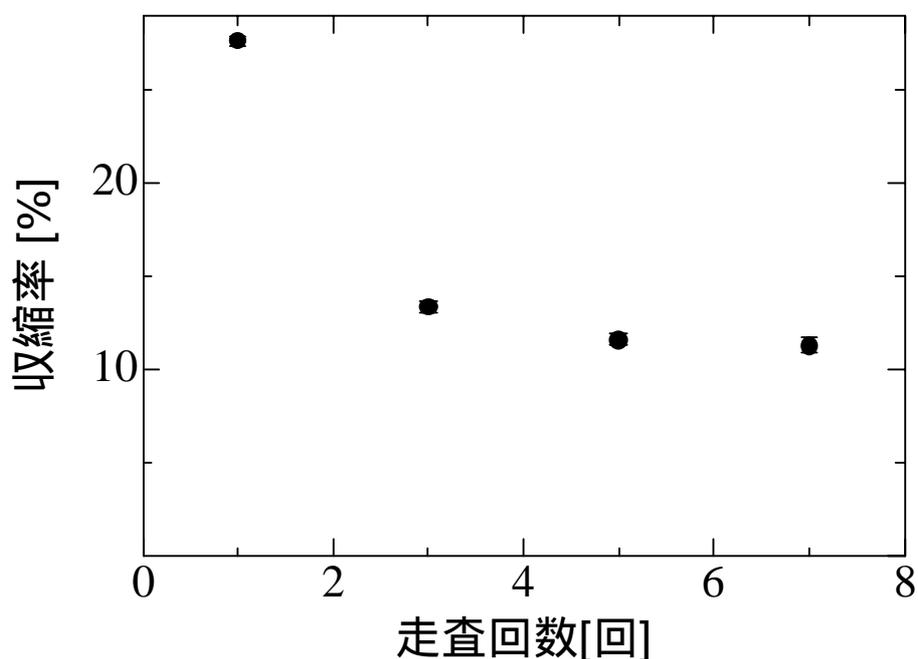


図.14 走査回数と収縮率の関係

(作製条件)

レーザー波長 710 nm

走査回数 1回

レーザーパワー 9 mW

洗浄液 エタノール (2.5 ml)

走査速度 42 $\mu\text{m/s}$

測定装置 レーザー顕微鏡 (楠川研)

対物レンズ 100倍 , NA=1.3

これらのことを踏まえて、走査回数と加温時間の最適化を行ったところ、

5 回走査で作成後 1 時間放置 80 で 5 時間加温 洗浄

というプロセスが収縮率を最も抑えることができ、最終的に収縮率は 6%
まで改善できた。

4章 実験結果

4.1 分解能の調査

硬化した樹脂の線幅や強度は，レーザーパワー，走査速度，走査回数により，大きく左右される．3次元マイクロ構造物を作製するためには，硬化分解能を調査する必要がある．そこで，XY平面内，Z軸方向の硬化分解能のレーザーパワー依存性を調査した．

硬化分解能を調査するために，コンピューターによる制御可能なガルバノミラーを用いてレーザー光の集光点を高速走査し，2次元構造物を作製した

4.1.1 XY平面内の硬化分解能

レーザーパワーとXY平面内の硬化分解能の関係を調査した結果を図.15に示す．見るとレーザーパワーが5 mW付近に境界線があり，これ以下のレーザーパワーだと硬化しないことがわかる．

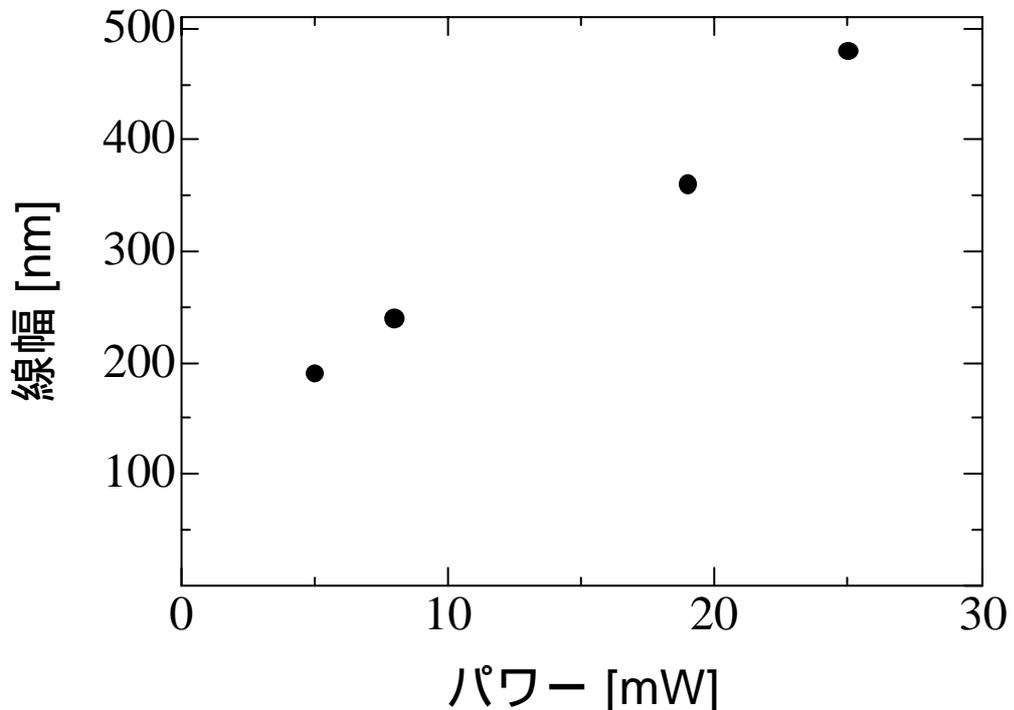


図.15 XY平面内硬化分解能のレーザーパワー依存性

4.1.2 Z 軸方向の硬化分解能

レーザーパワーとZ軸方向の硬化分解能の関係を調査した結果を図.17に示す。Z軸方向の硬化分解能は2.5でも述べたように、レーザーパワーの他に硬化に用いる対物レンズの開口数（NA）にも大きく左右される。

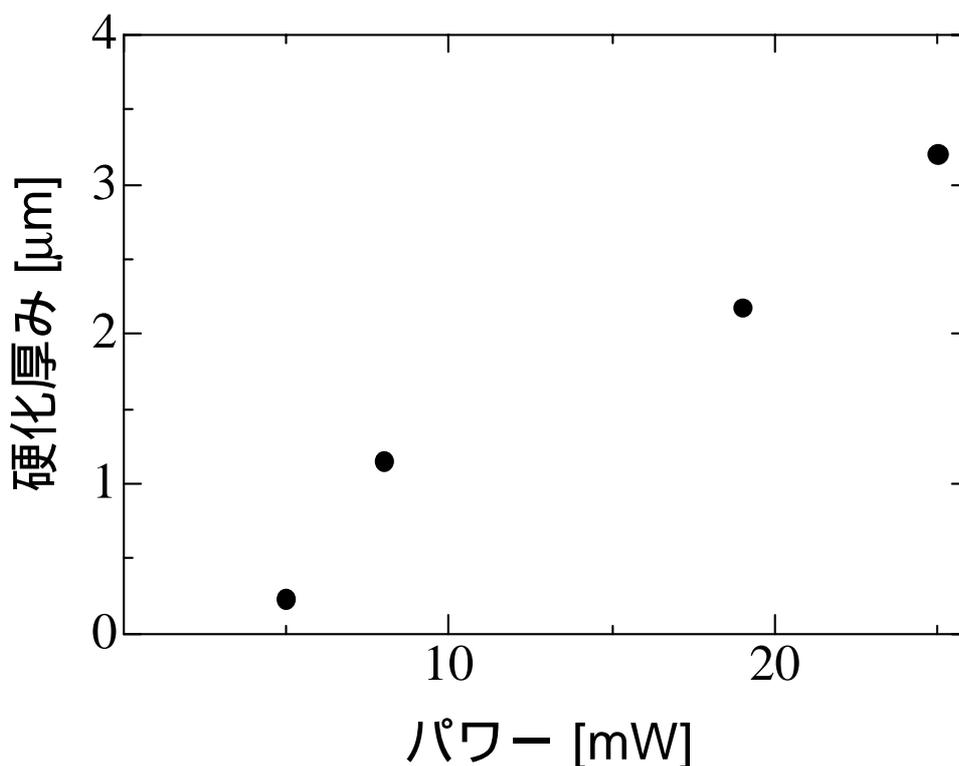


図.15 Z軸方向硬化分解能のレーザーパワー依存性

レーザーパワー、走査速度、走査回数の最適化により、格子状の構造物を作製した。作製した構造物のSEM（電子顕微鏡）像を図.16、図.17に示す。この構造物から、本実験でのXY平面内とZ軸方向の硬化分解能はそれぞれ190 nm、230 nmとわかった。

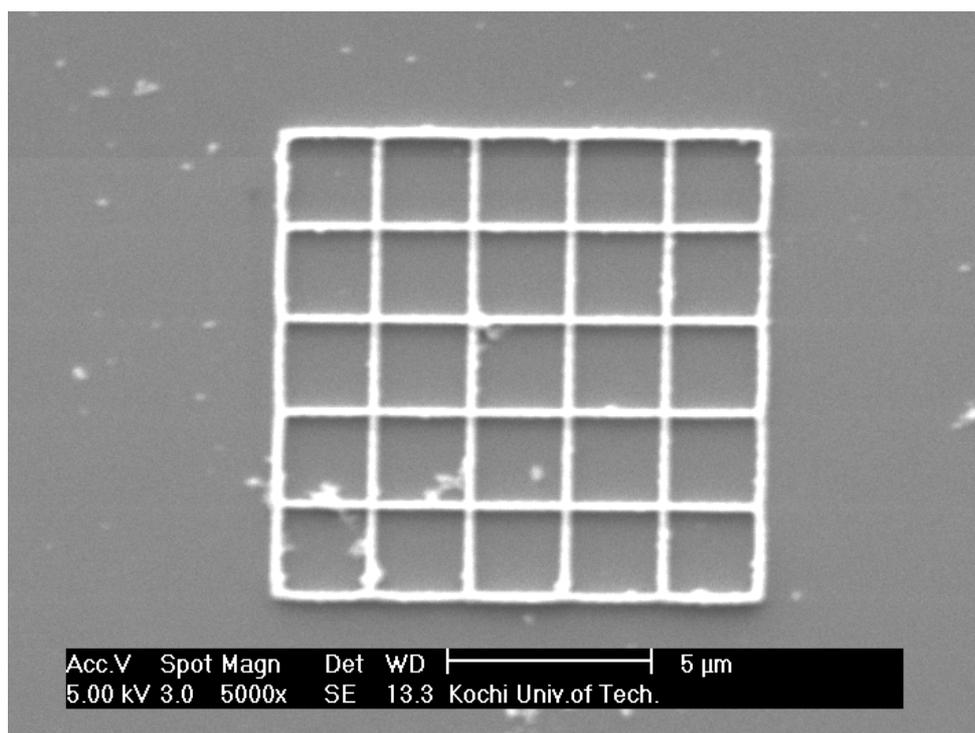


図.16 光造形で作製した格子状の構造物

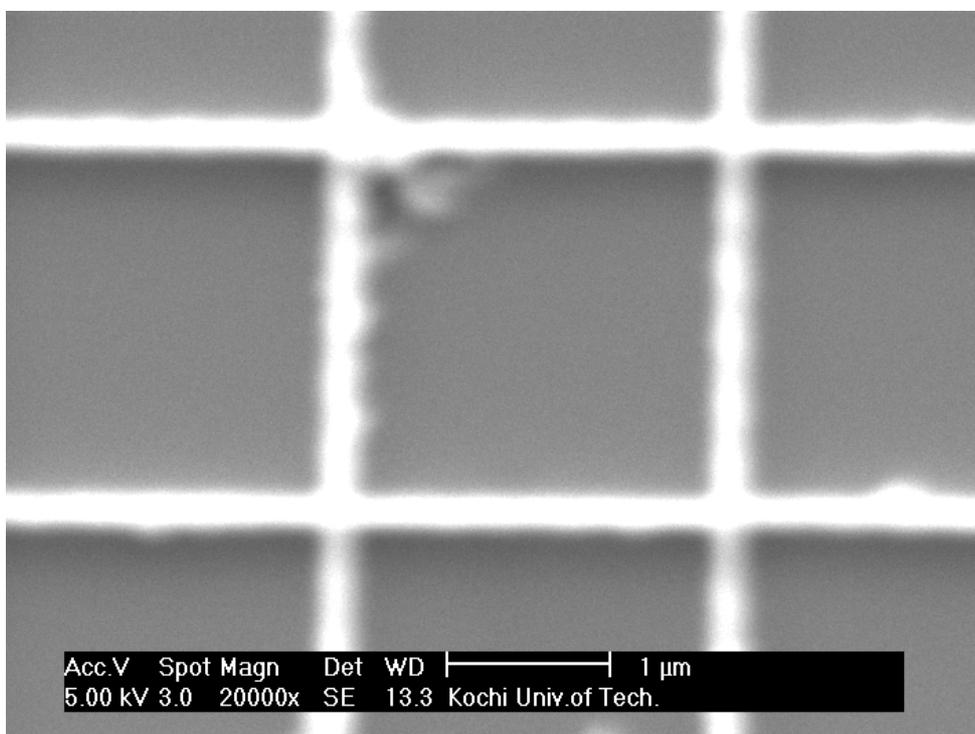


図.17 光造形で作製した格子状の構造物（拡大図）

(作製条件)

レーザー波長 710 nm

走査回数 1 回

レーザーパワー 9 mW

洗浄液 エタノール (2.5 ml)

走査速度 42 μ m/s

測定装置 レーザー顕微鏡 (楠川研)

対物レンズ 100 倍 , NA=1.3

4.2 ピエゾアクチュエーターの校正

4.2.1 校正方法

本実験ではミクロン精度でピエゾアクチュエーターを制御しなければいけないため、正確な移動量を調べておく必要があった。作製した3次元構造物のZ軸方向の高さと、重ね合わせる段数、1段当たりのピエゾアクチュエーターの移動量などの関係を知るために、立方体の3次元構造物を作製した。

4.2.2 立方体の製作

作製した立方体は図.18のような構造物である。縦、横、高さ、共に5 μm の立方体である。

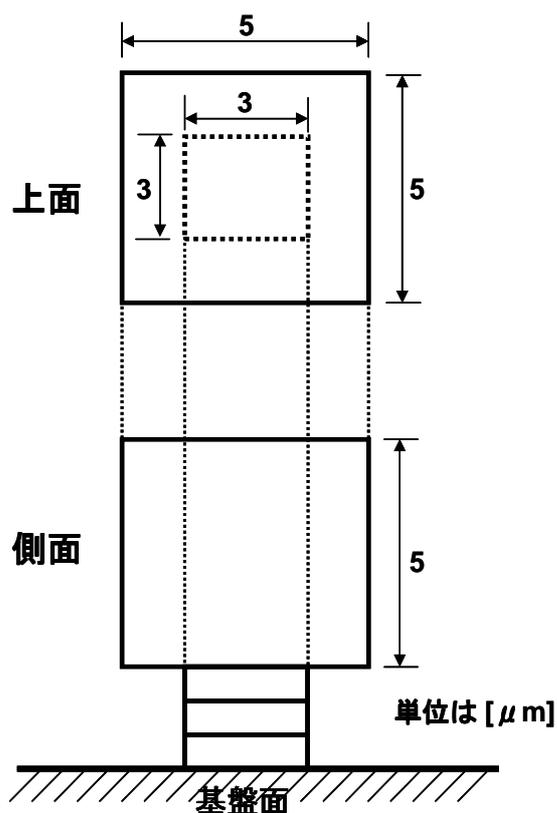


図.18 立方体見取り図

サンプルの作製条件を以下に示す．

- ・レーザー　：近赤外フェムト秒レーザー（波長 710 nm , パワー 9 mW）
- ・対物レンズ：倍率 100 倍 , NA1.3
- ・走査速度　：20 nm/s
- ・洗浄液　　：エタノール
- ・測定装置　：電子顕微鏡（河田研）

立方体を作製するにあたって，1 段あたりの Z 軸方向の硬化厚みを 1 μm となるように走査方法などを設定し，5 段重ねることで立方体の作製を試みた．この方法で立方体を作製する時には構造物（立方体）の構造は次の図.19 のようになっていると推測できる．

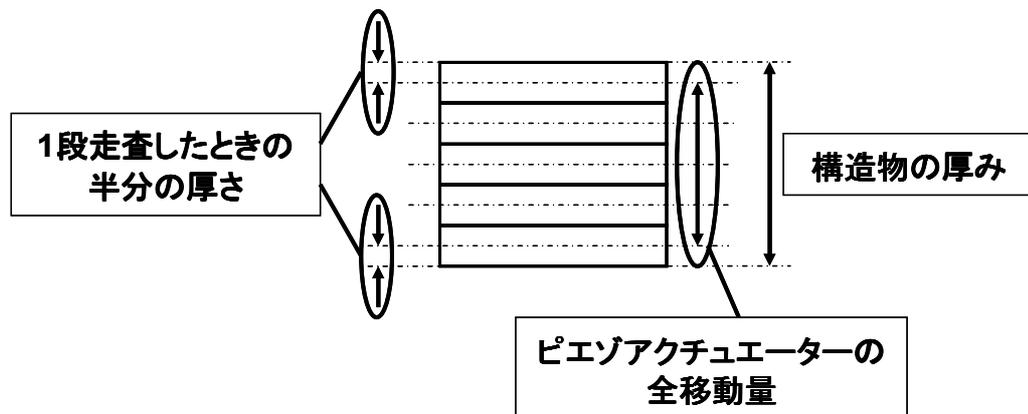


図.19 立方体の構造

4.2.3 計測

1 段当たりの piezoアクチュエーターの移動量を 3 種類設定し SEM (電子顕微鏡) 像を観察し, 作成した構造物の Z 軸方向の厚みを計測した。その時, 図.20 のような関係から, SEM 像で計測した数値実際の数値には次のような関係が成り立つ

$$(\text{構造物厚さ}) = (\text{計測値}) \times \frac{1}{\sin 45^\circ} \quad (2-1)$$

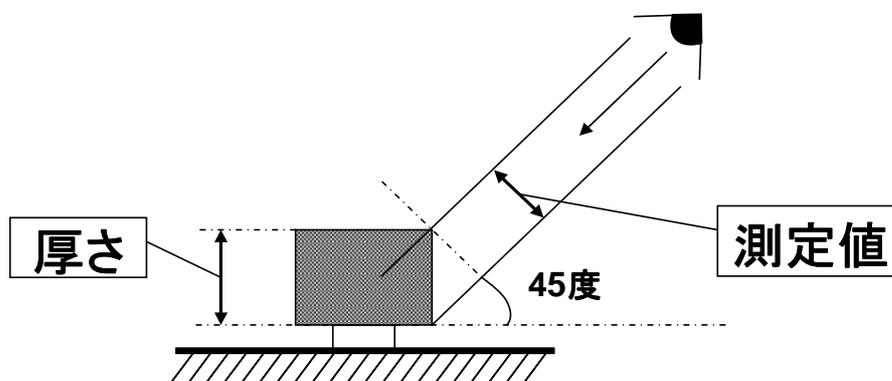


図.20 構造物厚さと計測値の関係

作製した 5 つのサンプルの計測値をこの式に代入し, 条件ごとにまとめたものを次に示す。

4.2.4 測定結果

1 段のみ走査した場合

図.21 はレーザーの集光点を正方形の形に走査したものである。この構造物が 1 段分の構造物となり、これを 5 段重ねることによって立方体を作製する。表を見ると、平均は $1.61 \mu\text{m}$ となっており、立方体作製時には 1 段につき $1.61 \mu\text{m}$ の厚さで硬化させていたことがわかる。

サンプル	厚み
1	1.65
2	1.65
3	1.49
4	1.63
5	1.64
平均	1.61

単位[μm]

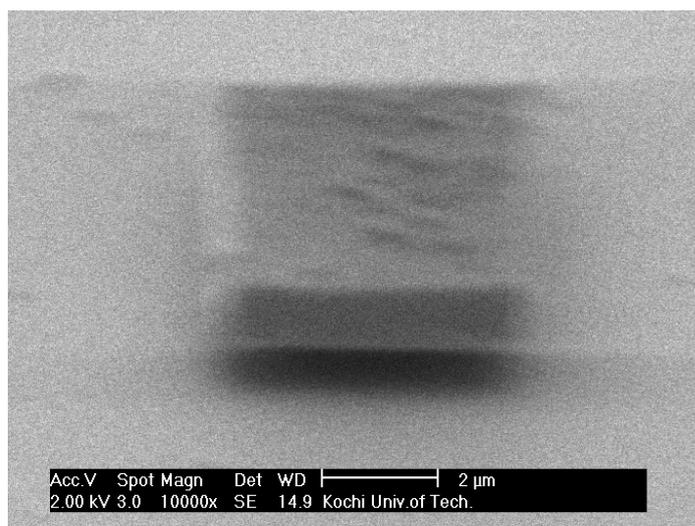


図.21 1 段のみ走査した時の構造物の SEM 像

1 段につき，20 V 加えた場合

図.22 は piezoアクチュエーターに 1 段につき 20 V 加えた時に作製された構造物の様子である．表からもわかるように，構造物の厚みはどのサンプルも 5 μm をかなり下回っており，立方体とは言えない．このことから 1 段に 20 V では，1 段当たりの移動量が少ないことがわかる．

サンプル	厚み
1	3.22
2	3.34
3	3.22
4	3.22
5	3.55
平均	3.31

単位[μm]

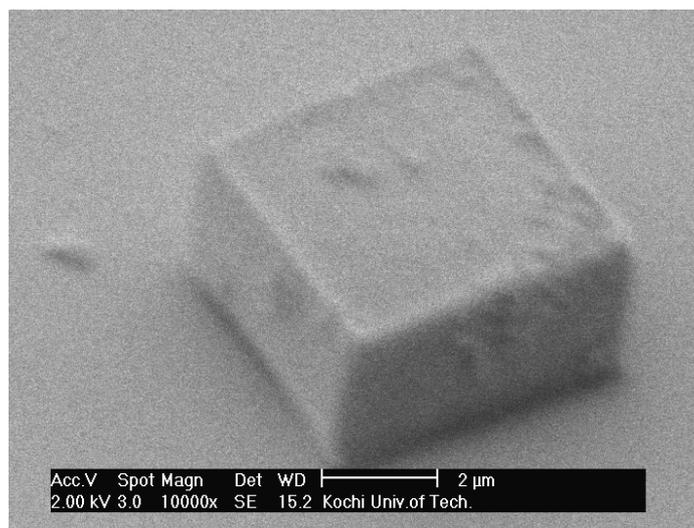


図.22 1 段につき 20V 加えた時の構造物の SEM 像

1 段につき，30 V 加えた場合

図.23 は piezoアクチュエーターに 1 段につき 30 V 加えた時に作製された構造物の様子である．20 V の時と同様に，構造物の厚みはどのサンプルも 5 μm を下回っており，立方体とは言えない．このことから 1 段に 30 V でも，1 段当たりの移動量が少ないことがわかる．

サンプル	厚み
1	4.59
2	4.64
3	4.57
4	4.56
5	4.63
平均	4.60

単位[μm]

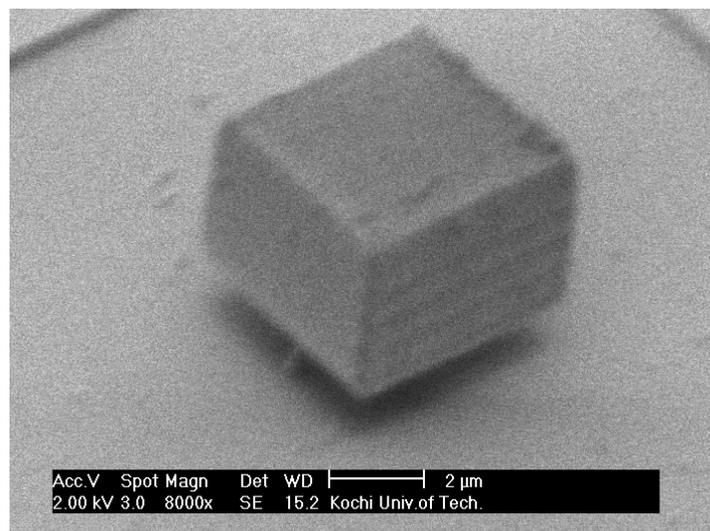


図.23 1 段につき 30V 加えた時の構造物の SEM 像

1 段につき，40 V 加えた場合

図.24 は piezoアクチュエーターに 1 段につき 40 V 加えた時に作製された構造物の様子である．構造物の厚みはどのサンプルも 5 μm をかなり上回っている．このことから 1 段に 40 V では，1 段当たりの移動量が大きすぎるのがわかる．

サンプル	厚み
1	5.80
2	5.86
3	5.73
4	5.88
5	5.82
平均	5.82

単位[μm]

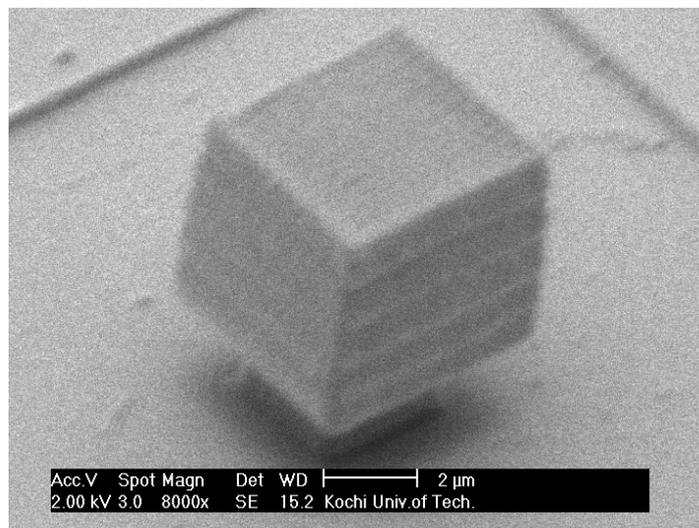


図.24 1 段につき 40V 加えた時の構造物の SEM 像

4.2.5 検証

以上の結果を元に、 piezoアクチュエーターに1段あたり加える電圧と、作製された構造物のZ軸方向の厚みをグラフにすると、図.25 のようになる。このグラフから本実験に使用した piezoアクチュエーターは、1 V あたり、31 nm 変化するという結果が得られた。

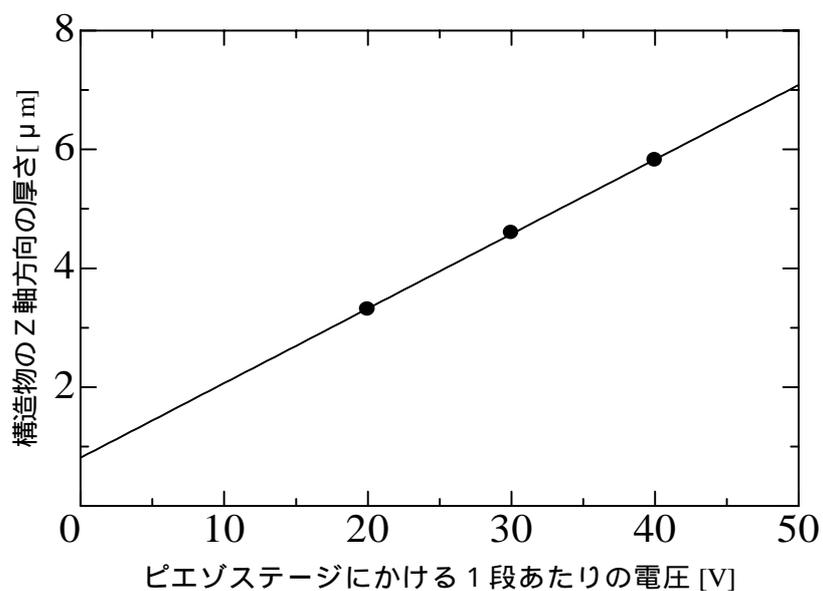


図.25 piezoアクチュエーターに加えた電圧と作製された構造物のZ軸方向の厚さの関係

4.3 プログラム製作

次に、このデータを用いて3次元構造物を作製するために、XY平面とZ軸方向の制御を同時に行うプログラムを作成した。このプログラムは、ガルバノミラーとピエゾステージをパソコンで用いて同時に制御でき、次の3つのプロセスを繰り返し行う。

スライスされた立体構造物の座標データを取り込む。

座標データをガルバノミラーのコントローラーに送信し走査を実行させる。

走査完了を確認し、ピエゾステージに電圧をかけてZ軸方向の制御を行う。

作製したプログラムの簡略化したフローチャートを図.26に示す。

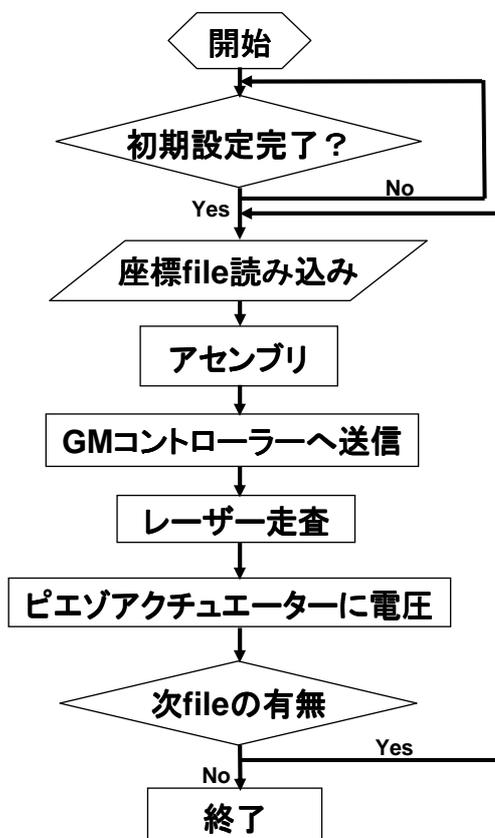


図.26 作製したプログラムのフローチャート

フローチャートの詳細は次のようになる。

初期設定：

プログラミング電圧，ガルバノミラーコントローラーとのデータ通信方式は，それぞれ GP-IB，RS-232C インターフェースである．そのインターフェースを利用可能にするため，それぞれに対して初期化，及び，初期設定を行う．

座標ファイルの読み込みと変換，送信：

1つのフォルダに格納されている最初のファイルを手動で読み込ませる．ファイルが読み込まれると，そのデータを 16 進数に変換し，ガルバノミラーコントローラーに送信される．またその際，読み込んだファイルの走査時間を計算し，変数に格納する．

走査実行：

データ送信が終了すると，走査が開始する．その走査中に次の座標ファイルの有無を判別し，ファイルが存在すれば読み込み，16 進数に変換し，ガルバノミラーコントローラーへ送信しておく．

電圧制御：

走査が完了すると，指定された電圧量だけピエゾアクチュエーターに電圧を加える．電圧が加え終わると，あらかじめ送信しておいた次の座標ファイルのデータにしたがって走査を開始させる．

このプログラムを用いることによって，自動的に 3 次元構造物を作製することが可能となった．図.27 はプログラムの外観図である．

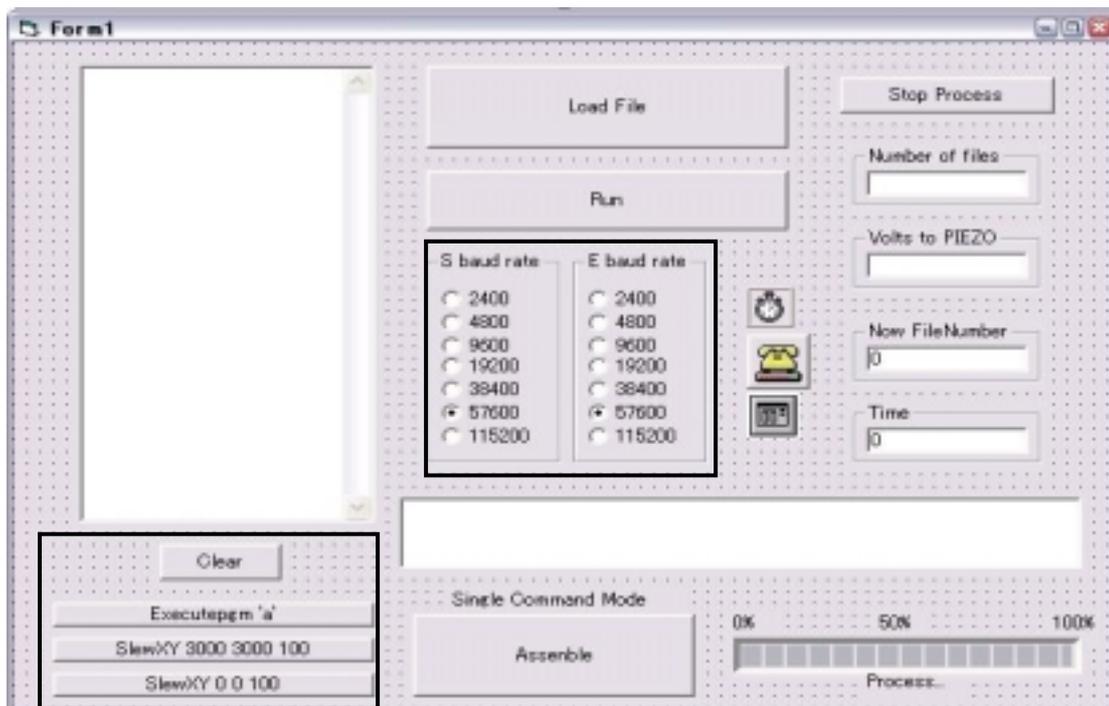


図.27 プログラム外観図

各部分の機能は以下のようにになっている。

読み込んだ座標ファイルの内容を表示するテキストボックス

最初の座標ファイルを読み込むボタン

3次元構造物を作製するプログラムの実行ボタン

ガルバノミラーコントローラーとの通信設定タブ

プログラムの停止ボタン

3次元構造物を作製するための座標ファイルの数を入力するテキストボックス

ピエゾアクチュエーターに一段あたりに加える電圧を入力するテキストボックス

プログラム実行中に、このテキストボックスに今実行している走査は何番目の座標ファイルのものを表示するテキストボックス

現在行われている走査の所要時間を表示するテキストボックス

エラーメッセージ，走査完了メッセージなどを表示する情報テキストボックス

ガルバノミラーコントローラーに独立した命令を直接送るためのボタン

ガルバノミラーコントローラーに手入力で直接命令を送信するためのボタン

3次元構造物作製の進行状況を示すプログレスバー

4.4 3次元マイクロ構造物の作製

以上のデータを元にして，3次元マイクロ部品を作製した．
作製は次の条件で行った．

(作製条件)

- ・レーザー：近赤外フェムト秒レーザー(波長 710 nm, パワー 9 mW)
- ・対物レンズ：倍率 100 倍, NA1.3
- ・走査速度：20 nm/s
- ・洗浄液：エタノール
- ・測定装置：光学顕微鏡

作製した構造物の設計図を図.28 に示す．

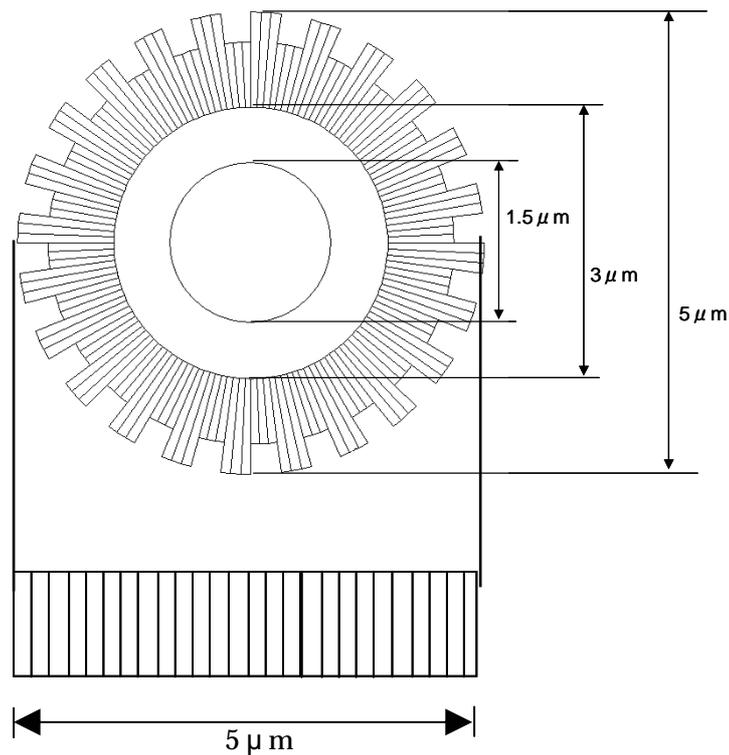


図.28 作製した3次元マイクロ構造物の設計図

作製した3次元マイクロ部品品のSEM像を以下に示す。写真の四角の部分
は作製した構造物が軸から外れないようにふたの役割を果たしている。

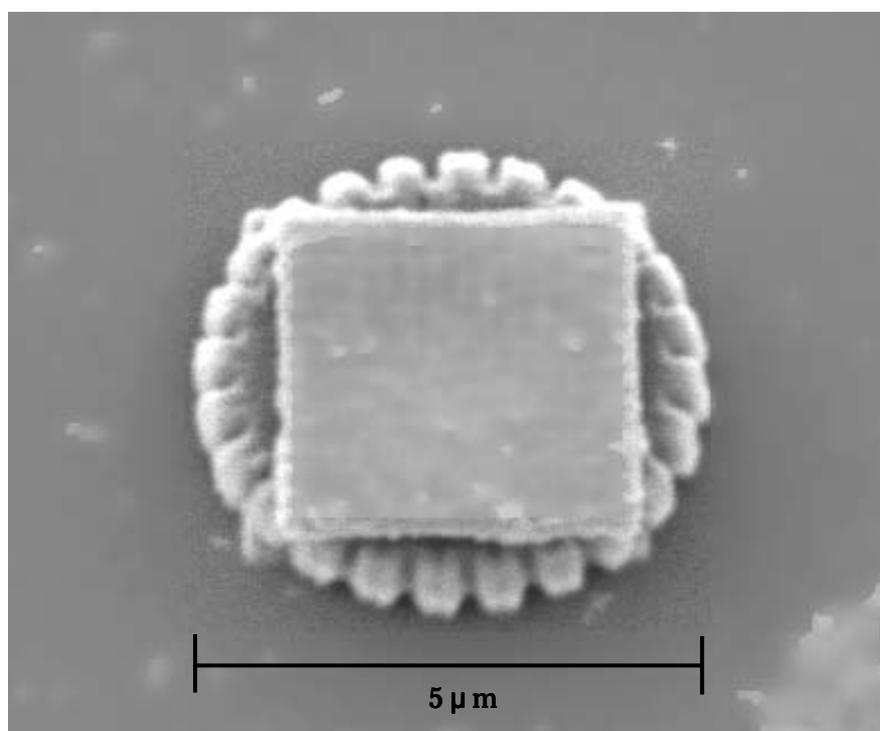


図.29 3次元マイクロ構造物のSEM像(上面)

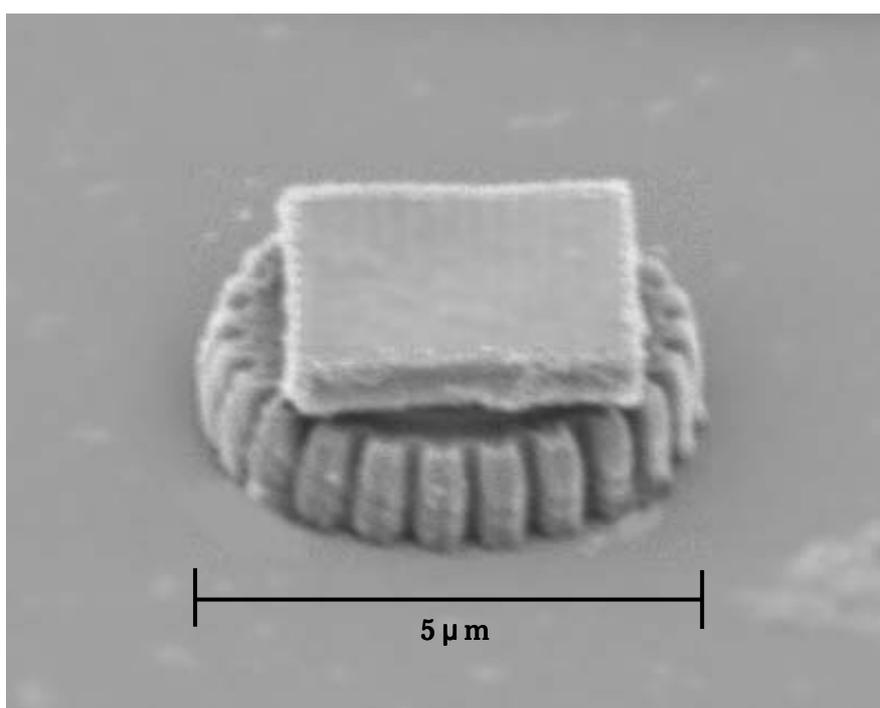


図.30 3次元マイクロ構造物のSEM像(正面45度)

5章 結言

本実験では、非線形光学効果のひとつである2光子吸収を用いて、光造形を行うためにレーザー強度、走査速度、走査回数の最適化を行った。光造形の光源として近赤外 Ti:サファイアフェムト秒レーザーを使用し、レーザーの集光点をガルバノミラーで走査してやることにより、2次元構造物を作製した。その際、レーザー強度、走査速度、走査回数の最適化によって、XY平面内、Z軸方向の硬化分解能を調査し、それぞれ190 nm、230 nmという結果を得た。そして、顕微鏡にピエゾアクチュエーターを設置し、XY平面内とZ軸方向の制御が同時できるプログラムを作製した。さらに、それらのことを利用して3次元マイクロ部品を試作した。

今後は本実験で得られたことを応用して硬化精度を向上させ、複雑な形状の立体構造物を作製したい。また、CADで作製した立体物を、マイクロサイズで作製できるシステムを確立していきたいと考えている。

資料 ピエゾアクチュエーター，ガルバノミラー同時制御プログラム

Option Explicit

Dim File As String
Dim filename As String
Dim Filee As String
Dim Filen As Integer
Dim Fname As String

Dim strRecBuff As String
Dim strValue As String
Dim strFileName As String
Dim intFileNo As Long
Dim strArray() As String
Dim intCounter As Long
Dim BinaryString As String

Dim T As Long
Dim TT As Long
Dim TTA As Long
Dim TA As Long
Dim TI As Long

Dim PIEZO As String
Dim Mode As Long
Dim SendBuf As String
Dim Pstb(15) As Long
Dim ErrText As String
Dim Srlen As Long
Dim Delim As Long
Dim SendPIEZO As String

'モードの設定
'送信バッファ
'ステータスバイトの配列
'テキスト表示用
'文字列の長さ

Dim Number As String
Dim eNumber As Long

Dim I As Long
Dim X As Long

'シリアルポートの設定
Private Function GetSbaudRate() As Integer

 If (S2400.Value = True) Then
 GetSbaudRate = 1
 ElseIf (S4800.Value = True) Then
 GetSbaudRate = 2
 ElseIf (s9600.Value = True) Then
 GetSbaudRate = 3
 ElseIf (s19200.Value = True) Then
 GetSbaudRate = 4
 ElseIf (s38400.Value = True) Then
 GetSbaudRate = 5
 ElseIf (s57600.Value = True) Then
 GetSbaudRate = 6
 ElseIf (s115200.Value = True) Then
 GetSbaudRate = 7
 Else: GetSbaudRate = 0
 End If

End Function

'シリアルポートの設定

Private Function GetEbaudRate() As Integer

```
If (e2400.Value = True) Then
    GetEbaudRate = 1
    ElseIf (e4800.Value = True) Then
        GetEbaudRate = 2
    ElseIf (e9600.Value = True) Then
        GetEbaudRate = 3
    ElseIf (e19200.Value = True) Then
        GetEbaudRate = 4
    ElseIf (e38400.Value = True) Then
        GetEbaudRate = 5
    ElseIf (e57600.Value = True) Then
        GetEbaudRate = 6
    ElseIf (e115200.Value = True) Then
        GetEbaudRate = 7
    Else: GetEbaudRate = 0
End If
```

End Function

'シリアルポートの設定

Private Sub JustifyBaudRate()

```
S2400.Value = e2400.Value
S4800.Value = e4800.Value
s9600.Value = e9600.Value
s19200.Value = e19200.Value
s38400.Value = e38400.Value
s57600.Value = e57600.Value
s115200.Value = e115200.Value
```

End Sub

'シリアルポートの設定

Private Function MakeVBCon String(BaudNumber As Integer) As String

Select Case BaudNumber

```
Case 1
    MakeVBCon  String = "2400,N,8,1"
Case 2
    MakeVBCon  String = "4800,N,8,1"
Case 3
    MakeVBCon  String = "9600,N,8,1"
Case 4
    MakeVBCon  String = "19200,N,8,1"
Case 5
    MakeVBCon  String = "38400,N,8,1"
Case 6
    MakeVBCon  String = "57600,N,8,1"
Case 7
    MakeVBCon  String = "115200,N,8,1"
```

Case Else

```
    MakeVBCon  String = ""
```

End Select

End Function

シリアルポートの設定

```
Private Function MakeSC2000Con (String(BaudNumber As Integer) As String
```

```
Select Case BaudNumber
```

```
Case 1
```

```
MakeSC2000Con (String = "comcon ( 1 8 1 0 232"
```

```
Case 2
```

```
MakeSC2000Con (String = "comcon ( 2 8 1 0 232"
```

```
Case 3
```

```
MakeSC2000Con (String = "comcon ( 3 8 1 0 232"
```

```
Case 4
```

```
MakeSC2000Con (String = "comcon ( 4 8 1 0 232"
```

```
Case 5
```

```
MakeSC2000Con (String = "comcon ( 5 8 1 0 232"
```

```
Case 6
```

```
MakeSC2000Con (String = "comcon ( 6 8 1 0 232"
```

```
Case 7
```

```
MakeSC2000Con (String = "comcon ( 7 8 1 0 232"
```

```
Case Else
```

```
MakeSC2000Con (String = ""
```

```
End Select
```

```
End Function
```

```
Private Sub ChangeBaudRate()
```

```
Dim StartBaud As Integer
```

```
Dim EndBaud As Integer
```

```
StartBaud = GetSbaudRate()
```

```
EndBaud = GetEbaudRate()
```

```
If (StartBaud = 0) Then Exit Sub
```

```
If (EndBaud = 0) Then Exit Sub
```

```
If (StartBaud = EndBaud) Then Exit Sub
```

```
Dim Ass As GSIDIass
```

```
Dim BinaryString As String
```

```
Set Ass = New GSIDIass
```

```
Ass.InitializeAssembler
```

```
Let BinaryString = Ass.AssembleString(MakeSC2000Con (String(EndBaud))
```

```
MSComm1.Settings = MakeVBCon (String(StartBaud)
```

```
Dim Fred() As Byte
```

```
ReDim Fred(LenB(BinaryString))
```

```
Fred = BinaryString
```

```
MSComm1.Output = Fred
```

```
Do
```

```
DoEvents
```

```
Loop Until 0 = MSComm1.OutBufferCount
```

```
MSComm1.Settings = MakeVBCon (String(EndBaud)
```

```
JustifyBaudRate
```

```
End Sub
```

```

Private Sub Assemble_Click()

    'タイマー起動
    Timer1.Interval = 10
    Timer1.Enabled = True

End Sub

Private Sub Command1_Click()

    'シングルコマンドモード
    Dim Ass As GSIDIass
    Dim BinaryString As String

    Set Ass = New GSIDIass

    Ass.InitializeAssembler

    Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

    Dim TestItem As Variant
    Dim TestArray(50) As Variant
    Dim Count As Integer

    Count = LenB(BinaryString) ¥ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

    Count = Count + 1

    Dim TempCount As Integer
    TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

    'TestItem = Fred

    MSComm1.CommPort = 2

    'MSComm1.Settings = "2400,N,8,1"
    MSComm1.Handshaking = comRTS
    MSComm1.OutBufferSize = LenB(BinaryString)

    progress.Text = "Open COMM port"
    MSComm1.PortOpen = True

    ChangeBaudRate

    progress.Text = "Send Program"

    Dim Fred() As Byte
    Dim Thing As String

    Print LenB(BinaryString)

    Dim lines As Integer

    lines = 1

```

```

Private Sub Assemble_Click()
    Do
        Thing = TestArray(lines)
        ReDim Fred(LenB(Thing))

        Fred = Thing

        MSComm1.Output = Fred

        Do
            DoEvents
            progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
        Loop Until MSComm1.OutBufferCount = 0

        lines = lines + 1
    Loop Until lines = Count

MSComm1.PortOpen = False

End Sub

Private Sub Command2_Click()
src.Text = ""

End Sub

Private Sub Command3_Click()
'プログラム停止
Timer1.Enabled = False
Assemble.Enabled = False

'SC2000 に中断命令
Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

src.Text = "ExitPgm"

Ass.InitializeAssembler

Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

Dim TestItem As Variant
Dim TestArray(50) As Variant

Dim Count As Integer

Count = LenB(BinaryString) ¥ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

```

```

Do
    Thing = TestArray(lines)
    ReDim Fred(LenB(Thing))

    Fred = Thing

    MSComm1.Output = Fred

    Do
        DoEvents
        progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
    Loop Until MSComm1.OutBufferCount = 0

    lines = lines + 1
Loop Until lines = Count

MSComm1.PortOpen = False

End Sub

Private Sub Command2_Click()
src.Text = ""
End Sub

Private Sub Command3_Click()
'プログラム停止

Timer1.Enabled = False
Assemble.Enabled = False

'SC2000 に中断命令

Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

src.Text = "ExitPgm"

Ass.InitializeAssembler

Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

Dim TestItem As Variant
Dim TestArray(50) As Variant

Dim Count As Integer

Count = LenB(BinaryString) ¥ 512

If (LenB(BinaryString) Mod 512) <> 0 Then
    Count = Count + 1
End If

```

```

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

'TestItem = Fred

MSComm1.CommPort = 2

'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS

MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Program"

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1

    Do
        Thing = TestArray(lines)
        ReDim Fred(LenB(Thing))

        Fred = Thing

        MSComm1.Output = Fred

        Do
            DoEvents
            progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
        Loop Until MSComm1.OutBufferCount = 0

        lines = lines + 1
    Loop Until lines = Count

progress.Text = "走査を中断しました . 今の走査が終了するまでお待ちください . "

MSComm1.PortOpen = False

End Sub

```

```

Private Sub Command4_Click()

'Executepgm 'a'
Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

Ass.InitializeAssembler

Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

Dim TestItem As Variant
Dim TestArray(50) As Variant

Dim Count As Integer

src.Text = "Executepgm 'a'"

Count = LenB(BinaryString) \ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

'TestItem = Fred

MSComm1.CommPort = 2

'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS

MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Program"

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1

```

```

Do
    Thing = TestArray(lines)
    ReDim Fred(LenB(Thing))

    Fred = Thing

    MSComm1.Output = Fred

    Do
        DoEvents
        progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
        Loop Until MSComm1.OutBufferCount = 0

    lines = lines + 1
    Loop Until lines = Count

MSComm1.PortOpen = False

End Sub

Private Sub Command5_Click()

'SlewXY 3000 3000 100

Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

Ass.InitializeAssembler

Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

Dim TestItem As Variant
Dim TestArray(50) As Variant

Dim Count As Integer

src.Text = "SlewXY 3000 3000 100"

Count = LenB(BinaryString) \ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
        Loop Until TempCount = Count

TestItem = Fred

MSComm1.CommPort = 2

```

```

'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS

MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Program"

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1

    Do
        Thing = TestArray(lines)
        ReDim Fred(LenB(Thing))

        Fred = Thing

        MSComm1.Output = Fred

        Do
            DoEvents
            progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
            Loop Until MSComm1.OutBufferCount = 0

        lines = lines + 1
    Loop Until lines = Count

MSComm1.PortOpen = False

End Sub

Private Sub Command6_Click()

'SlewXY 0 0 100
Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

Ass.InitializeAssembler

Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

Dim TestItem As Variant
Dim TestArray(50) As Variant

Dim Count As Integer

```

```

src.Text = "SlewXY 0 0 100"

Count = LenB(BinaryString) \ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

"TestItem = Fred

MSComm1.CommPort = 2

'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS

MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Program"

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1

    Do
        Thing = TestArray(lines)
        ReDim Fred(LenB(Thing))

        Fred = Thing

        MSComm1.Output = Fred

        Do
            DoEvents
            progress.Text = "Send Program. Bytes to go: " & MSComm1.OutBufferCount
        Loop Until MSComm1.OutBufferCount = 0

        lines = lines + 1
    Loop Until lines = Count

MSComm1.PortOpen = False

End Sub

```

```

Private Sub Form_Load()

'GP-IB の初期設定
Ret = GpibInit(ErrText)

    If (Ret = 1) Then GoTo Err
        Ret = GpBoardsts(&HA, Mode)

    If Mode = 1 Then
        ErrText = "この機器はマスタではありません．設定を確認してください．"
        GoTo Err
    End If

Ret = GpCrst(5)
Ret = GpCcls(5)

Assemble.Enabled = False

Text1.Text = "0"
Text2.Text = "0"
Text4.Text = "0"

Err:
    progress.Text = ErrText

End Sub

Private Sub LoadFile_Click()

Dim Ass As GSIDIass
Dim BinaryString As String

'最初のファイル読み込み
CommonDialog1.ShowOpen
Let FName = CommonDialog1.filename

'時間の計算と表示
strFileName = CommonDialog1.filename

' 空いているファイル番号を取得
intFileNo = FreeFile

' ファイルを開く
Open strFileName For Input As intFileNo

' ファイルの最後に達するまでループ
T = 0
    Do Until EOF(intFileNo)
        Line Input #intFileNo, strValue
        strArray = Split(strValue, " ")
        strRecBuff = strArray(3)
        TT = TT + Val(strRecBuff)
    Loop

Let fMainForm.src.Text = "CreatePGM 1 'a" & vbCrLf & "Setsync 1" & vbCrLf &
ReadAllFileBytes(Fname) & "end"

TTA = TT * 23 * (10 ^ -6)
Text1.Text = TTA

Filee = Val(Text4.Text)

```

```

Assemble.Enabled = True
progress.Text = "First file was Readed... Ready..OK..."

File = 1
Number = 1

'PIEZO にかける電圧の設定
PIEZO = Text2.Text
SendPIEZO = "VOLT:STEP " & PIEZO
SendBuf = SendPIEZO           '送信データをバッファに入力します。
Srlen = Len(SendBuf)         'バッファの大きさを取得します。
Cmd(0) = 2                   '総コマンド数
Cmd(1) = MyAddr              '送信側アドレスを決定
Cmd(2) = 5                   '相手機器のアドレスを決定

    If Srlen = 0 Then         '送信データが0の時の処理
        progress.Text = "送信データがありません。"
        Exit Sub
    End If

Ret = GpTalk(Cmd(0), Srlen, SendBuf) '送信を実行

'OUTPUT ON

SendPIEZO = "OUTPUT ON"
SendBuf = SendPIEZO           '送信データをバッファに入力します。
Srlen = Len(SendBuf)         'バッファの大きさを取得します。
Cmd(0) = 2                   '総コマンド数
Cmd(1) = MyAddr              '送信側アドレスを決定
Cmd(2) = 5                   '相手機器のアドレスを決定

    If Srlen = 0 Then         '送信データが0の時の処理
        progress.Text = "送信データがありません。"
        Exit Sub
    End If

Ret = GpTalk(Cmd(0), Srlen, SendBuf) '送信を実行

'アセンブリ開始
Set Ass = New GSIDIass

progress.Text = "Assemble start..."

Ass.InitializeAssembler
Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

'Text2.Text = ConvertBSToAsciiHex(BinaryString)
'

Dim TestItem As Variant
Dim TestArray(50) As Variant
Dim Count As Integer

Count = LenB(BinaryString) ¥ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

```

```

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

'TestItem = Fred

MSComm1.CommPort = 2
'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS
MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Ready..."

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1

    Do
        Thing = TestArray(lines)
        ReDim Fred(LenB(Thing))
        Fred = Thing
        MSComm1.Output = Fred

        Do
            DoEvents
            progress.Text = "Sending Program... Bytes to go: " &
                MSComm1.OutBufferCount
            Loop Until MSComm1.OutBufferCount = 0

        lines = lines + 1
    Loop Until lines = Count

MSComm1.PortOpen = False

'プログレスバーの設定

ProgressBar1.Max = Filee

End Sub

```

```

' input a binary string and convert to ascii hex string, add a line feed to the end
Public Function ConvertBSToAsciiHex(BString As String) As String

Dim MyHexResult As String
Dim X As Integer

Let MyHexResult = ""

    For X = 1 To LenB(BString)
        Let MyHexResult = MyHexResult & AddHexPadding(AscB(MidB(BString, X, 1)), 2)
    Next X

Let ConvertBSToAsciiHex = MyHexResult

End Function

'input a log integer and the print width of the Hex number output.
Public Function AddHexPadding(MyNumber As Long, Width As Integer) As String

Dim MyHexNumber As String

On Error GoTo ErrorHandler
Let MyHexNumber = Hex(MyNumber)

    If (MyNumber < 0) Then
        Let AddHexPadding = Right(MyHexNumber, Width)
    Else
        Let AddHexPadding = String(Width - Len(MyHexNumber), "0") & MyHexNumber
    End If

Exit Function

ErrorHandler:
SystemErrorHandler "AddHexPadding"

End Function

' Description: Read all of the file bytes into a string.
' Errors:      all errors are fatal
'             All errors are system errors
'             System errors are trapped and moved into error global
' Parameters:
'             name = The full path of the file to read.
' Returns: A string, the full contents of the file.

Public Function ReadAllFileBytes(name As String) As String

Dim NumBytesToGet
Dim FileNumber As Integer

On Error GoTo ErrorHandler
Let NumBytesToGet = FileLen(name)
Let FileNumber = FreeFile(0)

    Open name For Input As FileNumber

        ReadAllFileBytes = Input(NumBytesToGet, FileNumber)

    Close FileNumber

Exit Function

```

```

'TODO: perhaps close the open file here
SystemErrorHandler ("ReadAllFileBytes")

End Function

'write out a binary file given a string
Public Sub WriteFile(Fname As String, TheData As String)

Dim FNumber As Integer
Dim X As Integer
Dim TmpStr As String

On Error GoTo ErrorHandler

Let FNumber = FreeFile(0)

    Open Fname For Binary Access Write As #FNumber
        Put #FNumber, , TheData
    Close FNumber

Exit Sub
ErrorHandler:

End Sub

Private Sub MSComm1_OnComm()

Print 1

progress.Text = Str(MSComm1.CommEvent)

End Sub

Private Sub Timer1_Timer()

Dim Ass As GSIDIass
Dim BinaryString As String

Set Ass = New GSIDIass

Text3.Text = Number
ProgressBar1.Value = Number

'ファイル番号の判別

Number = Val(Number)
eNumber = Number - 1

    If Number = Filee Then
        Assemble.Enabled = False
        Timer1.Enabled = False
        progress.Text = "終了"
    End If

'PIEZO に電圧

    If Number >= 2 Then
        PIEZO = Text2.Text
        SendPIEZO = "VOLT UP"
        SendBuf = SendPIEZO
    
```

```

        送信データをバッファに入力
    
```

```

    Srlen = Len(SendBuf)           'バッファの大きさを取得します。
    Cmd(0) = 2                     '総コマンド数
    Cmd(1) = MyAddr                '送信側アドレスを決定
    Cmd(2) = 5                     '相手機器のアドレスを決定

    If Srlen = 0 Then              '送信データが0の時の処理
        progress.Text = "送信データがありません。"
        Exit Sub
    End If

    Ret = GpTalk(Cmd(0), Srlen, SendBuf) '送信を実行
End If

'ウェイト
Dim X As Long
Dim Y As Long
Dim I As Long
Dim machi As Long

X = 0
machi = 1000000

    For I = 0 To machi
        Y = Exp(I / machi)
    Next I

'アセンブリ開始

progress.Text = "Assemble start..."
src.Text = "Executepgm 'a'"

Ass.InitializeAssembler

On Error GoTo ErrorHandler
Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

'''
"Text2.Text = ConvertBSToAsciiHex(BinaryString)
'''

Dim TestItem As Variant
Dim TestArray(50) As Variant
Dim Count As Integer

Count = LenB(BinaryString) \ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

Dim TempCount As Integer

TempCount = 1

    Do
        TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
        TempCount = TempCount + 1
    Loop Until TempCount = Count

```

```

"TestItem = Fred

MSComm1.CommPort = 2
'MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS
MSComm1.OutBufferSize = LenB(BinaryString)

progress.Text = "Open COMM port"
MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Ready..."

Dim Fred() As Byte
Dim Thing As String

Print LenB(BinaryString)

Dim lines As Integer

lines = 1
  Do
    Thing = TestArray(lines)
    ReDim Fred(LenB(Thing))
    Fred = Thing
    MSComm1.Output = Fred

    Do
      DoEvents
      progress.Text = "Sending Program... Bytes to go: " &
MSComm1.OutBufferCount
      Loop Until MSComm1.OutBufferCount = 0
      lines = lines + 1
      Loop Until lines = Count

MSComm1.PortOpen = False

'走査終了処理

  If Number = Filee Then
    Assemble.Enabled = False
    Timer1.Enabled = False
    progress.Text = "終了"
    Exit Sub
  End If

TI = Val(TTA) * 1000
Timer1.Interval = TI

'次のファイルを読み込み

File = Val(File) + 1
Number = Number + 1
filename = File + ".txt"

Fname = filename

'時間の計算と表示

TT = 0
TTA = 0

```

```

strFileName = filename

' 空いているファイル番号を取得
intFileNo = FreeFile

' ファイルを開く
Open strFileName For Input As intFileNo

' ファイルの最後に達するまでループ
Do Until EOF(intFileNo)
    Line Input #intFileNo, strValue
    strArray = Split(strValue, " ")
    strRecBuff = strArray(3)

    TT = TT + Val(strRecBuff)
Loop

TTA = TT * 23 * (10 ^ -6)
Text1.Text = TTA

Let fMainForm.src.Text = "CreatePGM 1 'a" & vbCrLf & "Setsync 1" & vbCrLf &
ReadAllFileBytes(Fname) & "end"

'走査中に次のファイルをアセンブリして送信

Ass.InitializeAssembler

On Error GoTo ErrorHandler
Let BinaryString = Ass.AssembleString(fMainForm.src.Text)

'''
Text2.Text = ConvertBSToAsciiHex(BinaryString)
'''
Count = LenB(BinaryString) ¥ 512

    If (LenB(BinaryString) Mod 512) <> 0 Then
        Count = Count + 1
    End If

Count = Count + 1

TempCount = 1

Do
    TestArray(TempCount) = MidB(BinaryString, (TempCount - 1) * 512 + 1, 512)
    TempCount = TempCount + 1
Loop Until TempCount = Count

TestItem = Fred

MSComm1.CommPort = 2
MSComm1.Settings = "2400,N,8,1"
MSComm1.Handshaking = comRTS
MSComm1.OutBufferSize = LenB(BinaryString)

MSComm1.PortOpen = True

ChangeBaudRate

progress.Text = "Send Ready..."

Print LenB(BinaryString)

```

```

lines = 1
  Do
    Thing = TestArray(lines)
    ReDim Fred(LenB(Thing))
    Fred = Thing
    MSComm1.Output = Fred

    Do
      DoEvents
      progress.Text = "Sending Program... Bytes to go: " &
MSComm1.OutBufferCount
      Loop Until MSComm1.OutBufferCount = 0

    lines = lines + 1
  Loop Until lines = Count

MSComm1.PortOpen = False

Err:
  progress.Text = ErrText

ErrorHandler:

End Sub

```

以上，1050 行である。

参考文献

- (1) 浮田宏治, 日暮栄治, 応用物理 **63** (1994) 483
- (2) 日暮栄治, 浮田宏治, O plus E, 178 (1994) 90
- (3) 佐藤俊一, O plus E, 178 (1994) 85
- (4) 日暮栄治, レーザー研究, 24 巻, 第 11 号 (1996) 1169.
- (5) Brian . H *et al.*, Nature, **398** (1999) 51-52
- (6) S.Kawata *et al.*, Nature, **412** (2001) 697
- (7) Hong-Bo Sun, Tomokazu.T, Keiji.T, and S.Kawata, Appl.Phys.Lett., **79**
(2001) 1411-1413
- (8) 特集二光子科学 (2) , O plus E, 20 (1998)
- (9) レーザ加工学会誌 Vol.8, No3 (2001) 11-12
- (10) 河田聡, O plus E, 23 (2001) 65-67
- (11) 光学, 30 巻, 4 号, (2001) 258-259
- (12) 物理学辞典編集委員会 : 物理学辞典 (倍風館, 2001)
- (13) 特集二光子科学 (1) , O plus E, 20 (1998)
- (14) 丸谷洋二, 大川和夫, 早野誠治, 光造形法 (日刊工業新聞社 1990)
- (15) Max Born , Emil Wolf , 光学の原理 (東海大学出版会 1998) 256
- (16) 宮本健郎, 光学入門 (岩波書店 1995) 44, 96
- (17) 石黒浩三, 光学 (裳華房 1995) 128

謝辞

本研究を行うにあたり，終始丁寧なご指示，ご指導を賜りました指導教員の木村正廣教授（現高知工科大学知能機械システム工学科教授）に心から感謝致します．また，戸名正英氏（現高知工科大学知能機械システム工学科実験講師）に貴重な助言とご指導を頂きましたことを深く感謝致します．

SEM 観察において，知能機械精密工学研究室の河田耕一教授（現高知工科大学知能機械システム工学科教授），山岡正和氏（現高知工科大学大学院基盤工学専攻知能機械システムコース2回生），さらに，レーザー顕微鏡観察において，材料強度研究室の楠川量啓助教授（現高知工科大学知能機械システム工学科助教授）ご協力を頂きました．深く感謝の意を表し，お礼の言葉とさせていただきます．

また，ディーメック社からは，本研究で使用した光硬化性樹脂を提供して頂きました．ありがとうございました．

研究のサポートして頂いた光物性工学研究室の皆さんにもお礼申し上げます．